

Table of Contents

1 Order Shorthand	1
2 /order	2
2.1 Contents	2
2.2 Request Headers	2
2.3 Order Shorthand	2
2.4 Available Methods and Child Resources	2
2.5 Implicit Order Creation	2
2.6 Response Headers	2
2.7 Additional Status Code	2
2.8 OwnerId cookie	2
2.9 Examples	3
2.10 See also	4
3 Order-cloud-backup	5
4 Order a new cloud backup	6
4.1 Request	6
4.2 Create a new account for the order	6
4.3 Add "Linux Web Start" product to the order	6
4.4 Add the cloud backup service to the new product	6
4.5 Submit the order	7
5 Order-cloud-drive	8
6 Order a new cloud drive	9
6.1 Request	9
6.2 Create a new account for the order	9
6.3 Add "Linux Web Start" product to the order	9
6.4 Add the cloud drive service to the new product	9
6.5 Submit the order	10
7 Order-Domain	11
8 Order a new domain	12
8.1 Create a new order	12
8.2 Add product to the order	12
8.3 Add a service to the product	12
8.4 Add attributes to the service	12
8.5 Create a new account for the order	13
8.6 Add user info	14
8.7 Add billing contact	14
8.8 Add regular contact	14
8.9 Add payment info to the order	14
8.10 Validate the order	15
8.11 Get a quote	15
8.12 Submit the order	17
9 Order-Domain-Alias	18
10 Order a new email domain alias	19
10.1 Create a new order	19
10.2 Create a new account for the order	19
10.3 Add an email domain product	19
10.4 Add an email service to the new product	19
10.5 Add service attributes	20
10.6 Add an email domain alias service to the new product	20
10.7 Add the domain alias service attributes	20
10.8 Submit the order	20
11 Order-Domain-Shortcut	22
12 Order a new domain	23
12.1 Create a new domain registration order	23
13 Order-Email-Domain	24
14 Order a new email domain	25
14.1 Create a new order	25
14.2 Create a new account for the order	25
14.3 Add the email domain product	25
14.4 Add the email service to the new product	25
14.5 Add service attributes	26
14.6 Submit the order	26
15 Order-Exchange	27
15.1 Order a new Exchange domain	27
15.2 Create a new order	27
15.3 Create a new account for the order	27
15.4 Add an Exchange product to the order	27
15.5 Add an Exchange service to the new product	27
15.6 Obtain a quote	28
15.7 Submit the order	28
15.8 Check for status	28

Table of Contents

16 Order-Exchange-mailbox-free	30
16.1 Order a new, free Exchange mailbox	30
16.2 Create the mailbox	30
16.3 Check for status	30
17 Order-Exchange-mailbox-paid	31
17.1 Order a new, paid Exchange mailbox	31
17.2 Create the mailbox	31
17.3 Submit the order	31
17.4 Check for status	31
18 Order-Exchange-otherproduct	33
18.1 Ordering Other Exchange Products	33
18.2 Create Encryption for a domain	33
18.3 Obtain a quote	33
18.4 Submit the order	33
18.5 Check for status	33
19 Order-Products-Account	35
20 Set account on a later step when ordering products and services	36
20.1 Create a new order	36
20.2 Add a product	36
20.3 Add a service to the product with no attributes	36
20.4 Create a new account for the order	36
20.5 Set an existing account for the order	37
20.6 Submit the order	37
21 Order-Products-Account-Modify	38
22 Set and modify an account on a later step when ordering products and services	39
22.1 Create a new order	39
22.2 Add a product	39
22.3 Add a service to the product with no attributes	39
22.4 Set an existing account for the order	39
22.5 Set some account attributes	40
22.6 Set modified regular contact	40
22.7 Set account payment info	40
22.8 Submit the order	41
23 Order-vps-virtuozzo	42
23.1 Order VPS Virtuozzo products and services	42
23.2 Create a new order	42
23.3 Create a new account or add an existing one to the order	42
23.4 Add a VPS Virtuozzo product to the order	42
23.5 Add a Virtuozzo Server service to the new product	42
23.6 Add VPS Virtuozzo services to the new product	43
23.7 Submit the order	44
24 Order-webs-planet	46
24.1 Order a new WebsPlanet	46
24.2 Create a new order	46
24.3 Create a new account for the order	46
24.4 Add a Websplanet product to the order	46
24.5 Add a Webhosting service to the new product	46
24.6 Add the Webhosting service attributes	47
24.7 Add a Websplanet service to the new product	47
24.8 Add the WebsPlanet service attributes	47
25 Orders	48
26 Ordering API	49
26.1 Contents	49
26.2 Orders	49
26.3 Order Shorthand	50
26.4 Order Attributes	50
26.5 Account	50
26.6 Account Attributes	50
26.7 Account Contact Info	50
26.8 Account Payment Information	50
26.9 Account User	50
26.10 Products	50
26.11 Services	50
26.12 Service Attributes	50
26.13 Service Notes	50
26.14 Promotion Codes	51
26.15 Custom Discounts	51
26.16 Order Quote	51
26.17 See also	51
27 Order Flow Examples	52
27.1 Order Account Associations	52
27.2 Email Products	52
27.3 Microsoft Exchange Products	52
27.4 Cloud Products	52

Table of Contents

27 Order Flow Examples	
27.5 WebsPlanet Products.....	52
27.6 VPS Virtuozzo Products.....	52
27.7 Domain Registration Products.....	52
28 Orders-orderId-account-attributes:DELETE	53
29 DELETE /orders/{orderId}/account/attributes	54
29.1 Contents.....	54
29.2 Request.....	54
29.3 Response.....	54
29.4 Example.....	54
29.5 See also.....	55
30 Orders-orderId-account-attributes:GET	56
31 GET /orders/{orderId}/account/attributes	57
31.1 Contents.....	57
31.2 Request.....	57
31.3 Response.....	57
31.4 Example.....	58
31.5 See also.....	59
32 Orders-orderId-account-attributes:PUT	60
33 PUT /orders/{orderId}/account/attributes	61
33.1 Contents.....	61
33.2 Request.....	61
33.3 Response.....	62
33.4 Example.....	62
33.5 See also.....	64
34 Orders-orderId-account-contactInfo-contactType:DELETE	65
35 DELETE /orders/{orderId}/account/contactInfo/{contactType}	66
35.1 Contents.....	66
35.2 Request.....	66
35.3 Response.....	66
35.4 Example.....	66
35.5 See also.....	67
36 Orders-orderId-account-contactInfo-contactType:GET	69
37 GET /orders/{orderId}/account/contactInfo/{contactType}	70
37.1 Contents.....	70
37.2 Request.....	70
37.3 Response.....	70
37.4 Example.....	71
37.5 See also.....	72
38 Orders-orderId-account-contactInfo-contactType:PUT	73
39 PUT /orders/{orderId}/account/contactInfo/{contactType}	74
39.1 Contents.....	74
39.2 Request.....	74
39.3 Response.....	75
39.4 Example.....	76
39.5 See also.....	78
40 Orders-orderId-account-contactInfo:GET	79
41 GET /orders/{orderId}/account/contactInfo	80
41.1 Contents.....	80
41.2 Request.....	80
41.3 Response.....	80
41.4 Example.....	81
41.5 See also.....	82
42 Orders-orderId-account-paymentInfo:GET	83
43 GET /orders/{orderId}/account/paymentInfo	84
43.1 Contents.....	84
43.2 Request.....	84
43.3 Response.....	84
43.4 Example.....	85
43.5 See also.....	86
44 Orders-orderId-account-paymentInfo:PUT	87
45 PUT /orders/{orderId}/account/paymentInfo	88
45.1 Contents.....	88
45.2 Request.....	88
45.3 Response.....	89
45.4 Example.....	89
45.5 See also.....	93

Table of Contents

46 Orders-orderId-account-products-productId-discounts:DELETE	94
47 DELETE /orders/{orderId}/account/products/{productId}/discounts	95
47.1 Contents.....	95
47.2 Request.....	95
47.3 Response.....	95
47.4 Example.....	95
47.5 See also.....	96
48 Orders-orderId-account-products-productId-discounts:GET	97
49 GET /orders/{orderId}/account/products/{productId}/discounts	98
49.1 Contents.....	98
49.2 Request.....	98
49.3 Response.....	98
49.4 Example.....	99
49.5 See also.....	100
50 Orders-orderId-account-products-productId-discounts:PUT	101
51 PUT /orders/{orderId}/account/products/{productId}/discounts	102
51.1 Contents.....	102
51.2 Request.....	102
51.3 Response.....	103
51.4 Example.....	103
51.5 See also.....	106
52 Orders-orderId-account-products-productId-promotion:DELETE	107
53 DELETE /orders/{orderId}/account/products/{productId}/promotion	108
53.1 Contents.....	108
53.2 Request.....	108
53.3 Response.....	108
53.4 Example.....	108
53.5 See also.....	109
54 Orders-orderId-account-products-productId-promotion:PUT	110
55 PUT /orders/{orderId}/account/products/{productId}/promotion	111
55.1 Contents.....	111
55.2 Request.....	111
55.3 Response.....	111
55.4 Example.....	111
55.5 See also.....	113
56 Orders-orderId-account-products-productId-services-serviceld-attributes:GET	114
57 GET /orders/{orderId}/account/products/{productId}/services/{serviceld}/attributes	115
57.1 Contents.....	115
57.2 Request.....	115
57.3 Response.....	115
57.4 See also.....	116
58 Orders-orderId-account-products-productId-services-serviceld-attributes:OPTIONS	117
59 OPTIONS /orders/{orderId}/account/products/{productId}/services/{serviceld}/attributes	118
59.1 Contents.....	118
59.2 Request.....	118
59.3 Response.....	118
59.4 See also.....	118
60 Orders-orderId-account-products-productId-services-serviceld-attributes:PUT	119
61 PUT /orders/{orderId}/account/products/{productId}/services/{serviceld}/attributes	120
61.1 Contents.....	120
61.2 Request.....	120
61.3 Response.....	120
61.4 See also.....	121
62 Orders-orderId-account-products-productId-services-serviceld-discounts:DELETE	123
63 DELETE /orders/{orderId}/account/products/{productId}/services/{serviceld}/discounts	124
63.1 Contents.....	124
63.2 Request.....	124
63.3 Response.....	124
63.4 Example.....	124
63.5 See also.....	126
64 Orders-orderId-account-products-productId-services-serviceld-discounts:GET	127
65 GET /orders/{orderId}/account/products/{productId}/services/{serviceld}/discounts	128
65.1 Contents.....	128
65.2 Request.....	128
65.3 Response.....	128
65.4 Example.....	129

Table of Contents

65 GET /orders/{orderId}/account/products/{productId}/services/{serviceld}/discounts	130
65.5 See also.....	130
66 Orders-orderId-account-products-productId-services-serviceld-discounts:PUT.....	131
67 PUT /orders/{orderId}/account/products/{productId}/services/{serviceld}/discounts.....	132
67.1 Contents.....	132
67.2 Request.....	132
67.3 Response.....	133
67.4 Example.....	133
67.5 See also.....	137
68 Orders-orderId-account-products-productId-services-serviceld-notes:DELETE.....	138
69 DELETE /orders/{orderId}/account/products/{productId}/services/{serviceld}/notes.....	139
69.1 Contents.....	139
69.2 Request.....	139
69.3 Response.....	139
69.4 See also.....	140
70 Orders-orderId-account-products-productId-services-serviceld-notes:GET.....	141
71 GET /orders/{orderId}/account/products/{productId}/services/{serviceld}/notes.....	142
71.1 Contents.....	142
71.2 Request.....	142
71.3 Response.....	142
71.4 See also.....	143
72 Orders-orderId-account-products-productId-services-serviceld-notes:OPTIONS.....	144
73 OPTIONS /orders/{orderId}/account/products/{productId}/services/{serviceld}/notes.....	145
73.1 Contents.....	145
73.2 Request.....	145
73.3 Response.....	145
73.4 See also.....	145
74 Orders-orderId-account-products-productId-services-serviceld-notes:POST.....	146
75 POST /orders/{orderId}/account/products/{productId}/services/{serviceld}/notes.....	147
75.1 Contents.....	147
75.2 Request.....	147
75.3 Response.....	147
75.4 See also.....	148
76 Orders-orderId-account-products-productId-services-serviceld-notes:PUT.....	149
77 PUT /orders/{orderId}/account/products/{productId}/services/{serviceld}/notes.....	150
77.1 Contents.....	150
77.2 Request.....	150
77.3 Response.....	150
77.4 See also.....	151
78 Orders-orderId-account-products-productId-services-serviceld-promotion:DELETE.....	152
79 DELETE /orders/{orderId}/account/products/{productId}/services/{serviceld}/promotion.....	153
79.1 Contents.....	153
79.2 Request.....	153
79.3 Response.....	153
79.4 Example.....	153
79.5 See also.....	154
80 Orders-orderId-account-products-productId-services-serviceld-promotion:PUT.....	155
81 PUT /orders/{orderId}/account/products/{productId}/services/{serviceld}/promotion.....	156
81.1 Contents.....	156
81.2 Request.....	156
81.3 Response.....	156
81.4 Example.....	156
81.5 See also.....	157
82 Orders-orderId-account-products-productId-services-serviceld:DELETE.....	159
83 DELETE /orders/{orderId}/account/products/{productId}/services/{serviceld}.....	160
83.1 Contents.....	160
83.2 Request.....	160
83.3 Response.....	160
83.4 See also.....	161
84 Orders-orderId-account-products-productId-services-serviceld:GET.....	162
85 GET /orders/{orderId}/account/products/{productId}/services/{serviceld}.....	163
85.1 Contents.....	163
85.2 Request.....	163
85.3 Response.....	163
85.4 See also.....	164

Table of Contents

86 Orders-orderId-account-products-productId-services-serviceld:PUT.....	165
87 PUT /orders/{orderId}/account/products/{productId}/services/{serviceld}.....	166
87.1 Contents.....	166
87.2 Request.....	166
87.3 Response.....	166
87.4 See also.....	167
88 Orders-orderId-account-products-productId-services:GET.....	168
89 GET /orders/{orderId}/account/products/{productId}/services.....	169
89.1 Contents.....	169
89.2 Request.....	169
89.3 Response.....	169
89.4 See also.....	170
90 Orders-orderId-account-products-productId-services:POST.....	171
91 POST /orders/{orderId}/account/products/{productId}/services.....	172
91.1 Contents.....	172
91.2 Request.....	172
91.3 Response.....	173
91.4 Examples.....	176
91.5 See also.....	177
92 Orders-orderId-account-products-productId:DELETE.....	178
93 DELETE /orders/{orderId}/account/products/{productId}.....	179
93.1 Contents.....	179
93.2 Request.....	179
93.3 Response.....	179
93.4 See also.....	180
94 Orders-orderId-account-products-productId:GET.....	181
95 GET /orders/{orderId}/account/products/{productId}.....	182
95.1 Contents.....	182
95.2 Request.....	182
95.3 Response.....	182
95.4 See also.....	183
96 Orders-orderId-account-products-productId:PUT.....	184
97 PUT /orders/{orderId}/account/products/{productId}.....	185
97.1 Contents.....	185
97.2 Request.....	185
97.3 Response.....	185
97.4 See also.....	187
98 Orders-orderId-account-products:GET.....	188
99 GET /orders/{orderId}/account/products.....	189
99.1 Contents.....	189
99.2 Request.....	189
99.3 Response.....	189
99.4 See also.....	190
100 Orders-orderId-account-products:POST.....	191
101 POST /orders/{orderId}/account/products.....	192
101.1 Contents.....	192
101.2 Request.....	192
101.3 Response.....	193
101.4 See also.....	195
102 Orders-orderId-account-user:GET.....	196
103 GET /orders/{orderId}/account/user.....	197
103.1 Contents.....	197
103.2 Request.....	197
103.3 Response.....	197
103.4 Example.....	198
103.5 See also.....	199
104 Orders-orderId-account-user:PUT.....	200
105 PUT /orders/{orderId}/account/user.....	201
105.1 Contents.....	201
105.2 Request.....	201
105.3 Response.....	202
105.4 Example.....	202
105.5 See also.....	204

Table of Contents

106 Orders-orderId-account:GET	205
107 GET /orders/{orderId}/account	206
107.1 Contents	206
107.2 Request	206
107.3 Response	206
107.4 Example	207
107.5 See also	208
108 Orders-orderId-account:PUT	209
109 PUT /orders/{orderId}/account	210
109.1 Contents	210
109.2 Request	210
109.3 Response	211
109.4 Example	211
109.5 See also	213
110 Orders-orderId-action:POST	214
111 POST /orders/{orderId}/action	215
111.1 Contents	215
111.2 Request	215
111.3 Response	216
111.4 Examples	216
111.5 OPTIONS /orders/{orderId}/action	218
111.6 See also	219
112 Orders-orderId-attributes:GET	220
113 GET /orders/{orderId}/attributes	221
113.1 Contents	221
113.2 Request	221
113.3 Response	221
113.4 Example	222
113.5 See also	223
114 Orders-orderId-attributes:PUT	224
115 PUT /orders/{orderId}/attributes	225
115.1 Contents	225
115.2 Request	225
115.3 Response	226
115.4 Example	226
115.5 See also	227
116 Orders-orderId-discounts:DELETE	228
117 DELETE /orders/{orderId}/discounts	229
117.1 Contents	229
117.2 Request	229
117.3 Response	229
117.4 Example	229
117.5 See also	230
118 Orders-orderId-discounts:GET	231
119 GET /orders/{orderId}/discounts	232
119.1 Contents	232
119.2 Request	232
119.3 Response	232
119.4 Example	233
119.5 See also	234
120 Orders-orderId-discounts:PUT	235
121 PUT /orders/{orderId}/discounts	236
121.1 Contents	236
121.2 Request	236
121.3 Response	237
121.4 Example	237
121.5 See also	240
122 Orders-orderId-quote:GET	241
123 GET /orders/{orderId}/quote	242
123.1 Contents	242
123.2 Request	242
123.3 Response	242
123.4 Example	244
123.5 See also	246
124 Orders-orderId:DELETE	247

Table of Contents

125 DELETE /orders/{orderId}.....	248
125.1 Contents.....	248
125.2 Request.....	248
125.3 Response.....	248
125.4 Example.....	248
125.5 See also.....	249
126 Orders-orderId:GET.....	250
127 GET /orders/{orderId}.....	251
127.1 Contents.....	251
127.2 Request.....	251
127.3 Response.....	251
127.4 Example.....	252
127.5 OPTIONS /orders/{order_id}.....	253
127.6 See also.....	253
128 Orders:POST.....	254
129 POST /orders.....	255
129.1 Contents.....	255
129.2 Request.....	255
129.3 Response.....	255
129.4 Example.....	256
129.5 See also.....	257
130 ServiceAttributes-core.cloud.hsc.....	258
131 Cloud Compute Service Attributes.....	259
131.1 Contents.....	259
131.2 Attributes Body.....	259
131.3 See also.....	259
132 ServiceAttributes-core.dedserver.....	260
133 Dedicated Server Service Attributes.....	261
133.1 Contents.....	261
133.2 Attributes Body.....	261
133.3 See also.....	261
134 ServiceAttributes-core.dnr.....	262
135 Domain Name Registration Service Attributes.....	263
135.1 Contents.....	263
135.2 Attributes Body.....	263
135.3 See also.....	264
136 ServiceAttributes-core.fax.number.....	265
137 Fax Over Email Number Attributes.....	266
137.1 Contents.....	266
137.2 Attributes Body.....	266
137.3 See also.....	266
138 ServiceAttributes-core.fax.user.....	267
139 Fax Over Email User Attributes.....	268
139.1 Contents.....	268
139.2 Attributes Body.....	268
139.3 See also.....	268
140 ServiceAttributes-core.online.backup.account.....	269
141 Online Backup Account Attributes.....	270
141.1 Contents.....	270
141.2 Attributes Body.....	270
141.3 See also.....	270
142 ServiceAttributes-core.sharepoint.....	271
143 Sharepoint Service Attributes.....	272
143.1 Contents.....	272
143.2 Attributes Body.....	272
143.3 See also.....	272
144 ServiceAttributes-core.sharepoint2007.....	273
145 SharePoint 2007 Attributes.....	274
145.1 Contents.....	274
145.2 Attributes Body.....	274
145.3 See also.....	274
146 ServiceAttributes-core.spamexperts.....	275

Table of Contents

147 AVAS Service Attributes.....	276
147.1 Contents.....	276
147.2 Attributes Body.....	276
147.3 See also.....	276
148 ServiceAttributes-core.subdomain.....	277
149 Sub-Domain Registration Service Attributes.....	278
149.1 Contents.....	278
149.2 Attributes Body.....	278
149.3 See also.....	278
150 ServiceAttributes-core.vps.virtuozzo.server.....	279
151 VPS Virtuozzo Server Attributes.....	280
151.1 Contents.....	280
151.2 Attributes Body.....	280
151.3 See also.....	280
152 ServiceAttributes-core.webhosting.....	281
153 Web Hosting Service Attributes.....	282
153.1 Contents.....	282
153.2 Attributes Body.....	282
153.3 See also.....	282
154 ServiceAttributes-core.websplanet.....	283
155 Websplanet Attributes.....	284
155.1 Contents.....	284
155.2 Attributes Body.....	284
155.3 See also.....	284
156 ServiceAttributes-dedicated.server.firewall.....	285
157 Dedicated Server Firewall Service Attributes.....	286
157.1 Contents.....	286
157.2 Attributes Body.....	286
157.3 See also.....	286
158 ServiceAttributes-dnr.feature.domain guard.....	287
159 DomainGuard Service Attributes.....	288
159.1 Contents.....	288
159.2 Attributes Body.....	288
159.3 See also.....	288
160 ServiceAttributes-dnr.feature.private registration.....	289
161 WhoisProtector(Private Registration) Service Attributes.....	290
161.1 Contents.....	290
161.2 Attributes Body.....	290
161.3 See also.....	290
162 ServiceAttributes-mcafee.....	291
163 Excel Micro Service Attributes.....	292
163.1 See also.....	292
164 ServiceAttributesStructure.....	293
164.1 See also.....	293

1 Order Shorthand

2 /order

A shorthand that allows a user to access directly the current open order with the appropriate OwnerId.

2.1 Contents

- 1 /order
 - ◆ 1.1 Request Headers
 - ◆ 1.2 Order Shorthand
 - ◆ 1.3 Available Methods and Child Resources
 - ◆ 1.4 Implicit Order Creation
 - ◆ 1.5 Response Headers
 - ◆ 1.6 Additional Status Code
 - ◇ 1.6.1 Status Code
 - ◆ 1.7 OwnerId cookie
 - ◇ 1.7.1 Definition and purpose
 - ◇ 1.7.2 Order states and OwnerId cookie life cycle
 - 1.7.2.1 Fresh client session
 - 1.7.2.2 Subsequent calls to order shorthand resources/subresources
 - 1.7.2.3 Order instance deletion or submission
 - 1.7.2.4 Refreshing the OwnerId cookie after the order gets deleted or submitted
 - ◇ 1.7.3 CORS (Cross-origin resource sharing) and OwnerId cookie
 - ◆ 1.8 Examples
 - ◇ 1.8.1 Implicit Creation
 - ◇ 1.8.2 Alternate Hypermedia
 - ◇ 1.8.3 Accessing a Child Resource
 - ◆ 1.9 See also

2.2 Request Headers

Cookie
Required. Any call trying to use the shorthand notation has to have a valid **OwnerId** cookie set, with the exception of calls that will create a new order implicitly. A valid **Set-Cookie** header is returned with the response of either implicit or explicit order creation calls.

2.3 Order Shorthand

Using `/order` will try to find an order with status `new` that is associated with the **OwnerId** specified in the cookie header sent with the request. The value of the cookie may be set by either explicit or implicit order creation.

2.4 Available Methods and Child Resources

The shorthand supports all the methods that are available when using the `/orders/{orderId}` notation and all of the order's child resources are also accessible using the shorthand. A GET call to the order shorthand itself will include an *alternate* link to the explicit URL of the order in the hypermedia section. If an order is submitted using the shorthand URL, a **Location** header containing the explicit location of the order will be returned with a successful response for future reference.

2.5 Implicit Order Creation

If there is no current order associated with the active user, that is not submitted, a PUT or POST call to a child resource will implicitly create a new order for the user and will populate the data of the actual call to the new order. The final response of the implicitly created order will contain additional **Set-Cookie** header as described below. A GET or OPTIONS call to a non-existing order will not create it implicitly, but will return response with **404** status code.

2.6 Response Headers

Set-Cookie
If an order is created implicitly while accessing a child resource, the final response will include a **Set-Cookie** header containing an **OwnerId** identifier, in a similar way to explicit creation of an order using `POST /orders`. The cookie will be used to identify the end user, if no existing account is associated with the order, and to find which is the currently open order associated with the end client.

2.7 Additional Status Code

In addition to the response status codes that can be returned by various `/orders/{orderId}` calls, the order shorthand may also return:

2.7.1 Status Code

Status Code	Error Message	Description
405 Method Not Allowed	-	Trying to create a shortcut <code>/order</code> directly with PUT or POST
409 Conflict	-	There is already a new order associated with the same OwnerId and a second order cannot be created until the existing one is either submitted or deleted..

2.8 OwnerId cookie

2.8.1 Definition and purpose

The Order API defines the so called order shorthand. The Order shorthand is exposing all Order API resources and subresources, without supplying the order id. This means that at given time only one order instance is available to a client session. The Order API identifies the order instance by using

"Ownerld" cookie.

2.8.2 Order states and Ownerld cookie life cycle

2.8.2.1 Fresh client session

When client with fresh session starts using the Order API through the shorthand the order API is designed to create order instance implicitly. The created order instance is in "new" state. When the order instance is created, the Order API will return a Set-Cookie HTTP header for setting the "Ownerld" cookie into the client session.

2.8.2.2 Subsequent calls to order shorthand resources/subresources

Once the Ownerld has been set to the client session, the client should start providing the Ownerld cookie in all subsequent calls to Order API. Receiving request with Ownerld cookie will make Order API search for order instance in "new" state, identified by the Ownerld

2.8.2.3 Order instance deletion or submission

From "new" state the order instance can be deleted or submitted for provisioning. If the order is submitted its state will be changed to "placed".

2.8.2.4 Refreshing the Ownerld cookie after the order gets deleted or submitted

Subsequent calls, which identifies deleted or "placed" order instance will trigger Order API to implicitly create a new order instance and respond with Set-Cookie HTTP header with e new Ownerld value. The client session should start using the new Ownerld value when calling the Order API.

2.8.3 CORS (Cross-origin resource sharing) and Ownerld cookie

All browser based clients should abide the security policies enforced by the browsers itself. One such policy is the same origin security policy. To overcome the same origin security policy the browsers are providing the so called CORS (Cross-origin resource sharing) mechanism - http://en.wikipedia.org/wiki/Cross-origin_resource_sharing.

Basically the CORS mechanism makes the browsers to "disallow" requests which are originating from domain that does not match one of the Access-Control-Allow-Origin response header values. Ideally the Public API should allow calls, originated from all domains, by setting the Access-Control-Allow-Origin response header to * (Asterisk). Unfortunately when the requests are containing cookies the CORS mechanism dictates that the Access-Control-Allow-Origin should be very specific to the allowed domains(https://developer.mozilla.org/en-US/docs/HTTP/Access_control_CORS?redirectlocale=en-US&redirectslug=HTTP_access_control#Requests_with_cookies) meaning that your domain should be set into the Public API allowed domains. If your domain is not set to the Public API allowed domains, then please contact your sales or technical person and ask them to do that for you.

2.9 Examples

2.9.1 Implicit Creation

Request

```
PUT /order/account
Content-Type: application/json; charset=UTF-8
Authorization: {Base64-encoded username & password string}
```

```
{
  "currency": "USD",
  "language": "en-US"
}
```

Response

```
200 OK
Set-Cookie: OwnerId=4d84cd13-bbb3-46ba-a40c-8988351e3681; Path=/order
Set-Cookie: OwnerId=4d84cd13-bbb3-46ba-a40c-8988351e3681; Path=/orders
```

2.9.2 Alternate Hypermedia

Request

```
GET /order
Authorization: {Base64-encoded username & password string}
Cookie: OwnerId=4d84cd13-bbb3-46ba-a40c-8988351e3681; Path=/order
```

Response

```
200 OK
Content-Type: application/json; charset=UTF-8
{
  "order": {
    "OwnerId": "4d84cd13-bbb3-46ba-a40c-8988351e3681"
  },
  "links": {
    "self": {"href": "https://api.hostway.com/order/", "rel": "self"},
    "alternate": {"href": "https://api.hostway.com/orders/{orderId}/", "rel": "alternate"}
  }
}
```

2.9.3 Accessing a Child Resource

Request

```
GET /order/product/{productId}
Authorization: {Base64-encoded username & password string}
Cookie: OwnerId=4d84cd13-bbb3-46ba-a40c-8988351e3681; Path=/order
```

Response

200 OK
Content-Type: application/json; charset=UTF-8

```
{
  "status": "new",
  "services": "https://api.hostway.com/order/product/{productId}/services",
  "offering": "https://api.hostway.com/productSpecifications/{productSpecId}/offerings/{offering}",
  "links": [
    { "href": "https://api.hostway.com/order/product/{productId}/", "rel": "self" }
  ]
}
```

2.10 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

3 Order-cloud-backup

4 Order a new cloud backup

This example provides the steps to order cloud backup through the Hostway [Ordering API](#). == Create a new order == Use [POST /orders](#) to create a new order

4.1 Request

```
POST /orders
Content-Type: application/json; charset=UTF-8

{}
```

4.1.1 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf
Set-Cookie : OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "orderId": "0000013ad6dd95746a399f90000a71384bf242cf",
  "links": [
    { "href": "https://api.hostway.com/orders/", "rel": "self" },
    { "href": "https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf", "rel": "item" }
  ]
}
```

4.2 Create a new account for the order

Use [PUT /orders/{order_id}/account](#) to create an account or add an existing one to the order.

4.2.1 Request

```
PUT /orders/0000013ad6dd95746a399f90000a71384bf242cf/account
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "currency": "USD",
  "language": "en-US"
}
```

4.2.2 Response

```
204 No Content
```

4.3 Add "Linux Web Start" product to the order

The product specification offerings can be determined using the Hostway [Product Catalog API](#). Then add the new product to the order by using [POST /orders/{order_id}/account/products](#).

4.3.1 Request

```
POST /orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "offering": "https://api.hostway.com/productSpecifications/core.plan.linux.webstart/offerings/USD-12"
}
```

4.3.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products/0000013fd6dd95726c399e90000a71384bf242cf
```

4.4 Add the cloud backup service to the new product

The service specification offerings can be determined using the Hostway [Product Catalog API](#) for the core.plan.linux.webstart product. Then add the new cloud backup service to the order by using [POST /orders/{order_id}/account/products/{product_id}/services](#).

4.4.1 Request

```
POST /orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products/0000013fd6dd95726c399e90000a71384bf242cf/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "name": "test-domain.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.cloudbackup.hs2.2gb",
  "billingCycle": 12,
  "attributes": { "displayName": "test-domain.com" }
}
```

4.4.2 Response

201 Created

Content-Type: application/json; charset=UTF-8

Location : <https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products/0000013fd6dd95726c399e90000a71384bf242cf/>

4.5 Submit the order

Use [POST /orders/{order_id}/action](#) to submit the order and invoke the cloud backup provisioning.

4.5.1 Request

POST /orders/0000013ad6dd95746a399f90000a71384bf242cf/action

Content-Type: application/json; charset=UTF-8

```
{  
  "action": "submit"  
}
```

4.5.2 Response

202 Accepted

See also

- [Product Catalog API](#)
- [Ordering API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

5 Order-cloud-drive

6 Order a new cloud drive

This example provides the steps to order cloud drive through the Hostway [Ordering API](#). == Create a new order == Use [POST /orders](#) to create a new order

6.1 Request

```
POST /orders
Content-Type: application/json; charset=UTF-8

{}
```

6.1.1 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf
Set-Cookie : OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "orderId": "0000013ad6dd95746a399f90000a71384bf242cf",
  "links": [
    { "href": "https://api.hostway.com/orders/", "rel": "self" },
    { "href": "https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf", "rel": "item" }
  ]
}
```

6.2 Create a new account for the order

Use [PUT /orders/{order_id}/account](#) to create an account or add an existing one to the order.

6.2.1 Request

```
PUT /orders/0000013ad6dd95746a399f90000a71384bf242cf/account
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "currency": "USD",
  "language": "en-US"
}
```

6.2.2 Response

```
204 No Content
```

6.3 Add "Linux Web Start" product to the order

The product specification offerings can be determined using the Hostway [Product Catalog API](#). Then add the new product to the order by using [POST /orders/{order_id}/account/products](#).

6.3.1 Request

```
POST /orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "offering": "https://api.hostway.com/productSpecifications/core.plan.linux.webstart/offerings/USD-12"
}
```

6.3.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products/0000013fd6dd95726c399e90000a71384bf242cf
```

6.4 Add the cloud drive service to the new product

The service specification offerings can be determined using the Hostway [Product Catalog API](#) for the core.plan.linux.webstart product. Then add the new cloud drive service to the order by using [POST /orders/{order_id}/account/products/{product_id}/services](#).

6.4.1 Request

```
POST /orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products/0000013fd6dd95726c399e90000a71384bf242cf/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "name": "test-domain.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.clouddrive.hs2.1gb",
  "billingCycle": 12,
  "attributes": { "displayName": "test-domain.com" }
}
```

6.4.2 Response

201 Created

Content-Type: application/json; charset=UTF-8

Location : <https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products/0000013fd6dd95726c399e90000a71384bf242cf/>

6.5 Submit the order

Use [POST /orders/{order_id}/action](#) to submit the order and invoke the cloud drive provisioning.

6.5.1 Request

POST /orders/0000013ad6dd95746a399f90000a71384bf242cf/action

Content-Type: application/json; charset=UTF-8

```
{
  "action": "submit"
}
```

6.5.2 Response

202 Accepted

See also

- [Product Catalog API](#)
- [Ordering API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

7 Order-Domain

8 Order a new domain

This is a step by step example of ordering a domain through the Hostway [Ordering API](#).

8.1 Create a new order

Use [POST /orders](#) to create a new order

8.1.1 Request

```
POST /orders
Content-Type: application/json

{}
```

8.1.2 Response

```
201 Created

{
  "orderId":"00000149334635c2a163f194007f000000010001",
  "ownerId":"e497c8e7-8cf3-44ff-88d8-0c86521060da",
  "links":[
    {
      "href":"http://coreapi01.ote.chicago.hostway:8100/orders/",
      "rel":"self"
    },
    {
      "href":"http://coreapi01.ote.chicago.hostway:8100/orders/00000149334635c2a163f194007f000000010001",
      "rel":"item"
    }
  ]
}
```

8.2 Add product to the order

The domain registration product specification offerings can be determined using the Hostway [Product Catalog API](#). Add a new domain product to the order by using [POST /orders/{order_id}/account/products](#).

8.2.1 Request

```
POST /orders/00000149334635c2a163f194007f000000010001/account/products
Content-Type: application/json

{ "offering": "http://coreapi01.ote.chicago.hostway:8100/productSpecifications/core.plan.dnr.only/offerings/USD-12" }
```

8.2.2 Response

```
201 Created
Content-Type : application/json; charset=UTF-8
Connection : close
Location : http://coreapi01.ote.chicago.hostway:8100/orders/00000149334635c2a163f194007f000000010001/account/products/0000014933463b7219be46f0007f000000010001

{
  "url":"http://coreapi01.ote.chicago.hostway:8100/orders/00000149334635c2a163f194007f000000010001/account/products/0000014933463b7219be46f0007f000000010001",
  "productId":"0000014933463b7219be46f0007f000000010001"
}
```

8.3 Add a service to the product

The domain registration service specification offerings can be determined using the Hostway [Product Catalog API](#) for the corresponding core.dnr.{tld} product. Then add a new email service to the order by using [POST /orders/{order_id}/account/products/{product_id}/services](#).

8.3.1 Request

```
POST /orders/00000149334635c2a163f194007f000000010001/account/products/0000014933463b7219be46f0007f000000010001/services
Content-Type: application/json

{
  "name":"ng-test-fitness-20141021-1014.us",
  "serviceSpecification":"http://coreapi01.ote.chicago.hostway:8100/serviceSpecifications/core.dnr.us",
  "billingCycle":12
}
```

8.3.2 Response

```
201 Created
Content-Type : application/json; charset=UTF-8
Location : http://coreapi01.ote.chicago.hostway:8100/orders/00000149334635c2a163f194007f000000010001/account/products/0000014933463b7219be46f0007f000000010001

{
  "url":"http://coreapi01.ote.chicago.hostway:8100/orders/00000149334635c2a163f194007f000000010001/account/products/0000014933463b7219be46f0007f000000010001",
  "serviceId":"0000014933463eead40ac7ba007f000000010001"
}
```

8.4 Add attributes to the service

Use the [PUT /orders/{order_id}/account/products/{product_id}/services/{service_id}/attributes](#) call to add the attributes for the domain service. Note that the example below is using "extraParams" attribute which is only required for some TLDs (.us in this case).

8.4.1 Request

```
PUT /orders/0000014933612cf9fb1c626a007f000000010001/account/products/0000014933613211db39d7c4007f0000000010001/services/0000014933613528f83d7
Content-Type: application/json
```

```
{
  "domainName": "ng-test-fitness-20141021-1044.us",
  "operation": "register",
  "term": 1,
  "authInfo": "mysecret",
  "nameservers": [
    "ns1.test.com",
    "ns2.test.com"
  ],
  "extraParams": [
    {
      "name": "us_nexus",
      "value": "C11"
    },
    {
      "name": "us_purpose",
      "value": "P1"
    }
  ],
  "contactInfo": {
    "registrant": {
      "type": "registrant",
      "firstName": "John",
      "lastName": "Doe",
      "company": "ACME Inc.",
      "street1": "123 First Str.",
      "city": "Chicago",
      "zipOrPostal": "34567",
      "stateOrProvince": "IL",
      "countryCode": "US",
      "email1": "john.doe@test.com",
      "phone1": "+1.312.456789098"
    },
    "administrative": {
      "type": "administrative",
      "firstName": "Jane",
      "lastName": "Smith",
      "street1": "123 First Str.",
      "city": "Chicago",
      "zipOrPostal": "34567",
      "stateOrProvince": "IL",
      "countryCode": "US",
      "email1": "jane.smith@test.com",
      "phone1": "+1.312.456789111",
      "fax1": "+1.312.456789112"
    },
    "billing": {
      "type": "billing",
      "firstName": "James",
      "lastName": "Johnson",
      "street1": "123 First Str.",
      "city": "Chicago",
      "zipOrPostal": "34567",
      "stateOrProvince": "IL",
      "countryCode": "US",
      "email1": "james.johnson@test.com",
      "phone1": "+1.312.456789111"
    },
    "technical": {
      "type": "technical",
      "firstName": "Paul",
      "lastName": "Stevenson",
      "street1": "123 First Str.",
      "city": "Chicago",
      "zipOrPostal": "34567",
      "stateOrProvince": "IL",
      "countryCode": "US",
      "email1": "paul.stevenson@test.com",
      "phone1": "+1.312.456789111"
    }
  }
}
```

8.4.2 Response

```
204 No content
```

8.5 Create a new account for the order

Use [PUT /orders/{order_id}/account](#) to create an account or add an existing one to the order.

8.5.1 Request

```
PUT /orders/0000014933612cf9fb1c626a007f000000010001/account
Content-Type: application/json
```

```
{ "currency": "USD", "language": "en-US" }
```

8.5.2 Response

```
204 No content
```

8.6 Add user info

8.6.1 Request

```
PUT /orders/0000014933612cf9fb1c626a007f000000010001/account/user
Content-Type: application/json

{ "userName": "quah@141021-104405.com", "password": "t0pSekr1t" }
```

8.6.2 Response

204 No content

8.7 Add billing contact

8.7.1 Request

```
PUT /orders/0000014933612cf9fb1c626a007f000000010001/account/contactInfo/billing
Content-Type: application/json
```

```
{
  "name":{
    "firstName":"Billing",
    "middleName":"X",
    "lastName":"Customer",
    "company":"Hostway"
  },
  "address":{
    "street1":"100 n riverside",
    "street2":"ste 800",
    "city":"Chicago",
    "stateOrProvince":"IL",
    "postalCode":"60606",
    "countryCode":"US"
  },
  "contactMedia":{
    "phone1":"312-555-2222",
    "phone2":"312-555-3333",
    "fax":"312-555-1111",
    "email1":"james.johnson@test.com",
    "email1Format":"html",
    "email2":"gah@wev.com",
    "email2Format":"text"
  }
}
```

8.7.2 Response

204 No Content

8.8 Add regular contact

8.8.1 Request

```
PUT /orders/0000014933612cf9fb1c626a007f000000010001/account/contactInfo/regular
Content-Type: application/json
```

```
{
  "name":{
    "firstName":"Billing",
    "middleName":"X",
    "lastName":"Customer",
    "company":"Hostway"
  },
  "address":{
    "street1":"100 n riverside",
    "street2":"ste 800",
    "city":"Chicago",
    "stateOrProvince":"IL",
    "postalCode":"60606",
    "countryCode":"US"
  },
  "contactMedia":{
    "phone1":"312-555-2222",
    "phone2":"312-555-3333",
    "fax":"312-555-1111",
    "email1":"james.johnson@test.com",
    "email1Format":"html",
    "email2":"gah@wev.com",
    "email2Format":"text"
  }
}
```

8.8.2 Response

204 No Content

8.9 Add payment info to the order

8.9.1 Request

PUT /orders/0000014933612cf9fb1c626a007f000000010001/account/paymentInfo
Content-Type: application/json

```
{
  "paymentInfoType": "credit_card",
  "creditCardType": "visa",
  "accountNumber": "4111111111111111",
  "cardHolderName": "John Jello",
  "expirationDate": "2015-06",
  "verificationNumber": "521",
  "billingAddress": {
    "street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  }
}
```

8.9.2 Response

204 No Content

8.10 Validate the order

8.10.1 Request

POST /orders/0000014933612cf9fb1c626a007f000000010001/action
Content-Type: application/json

```
{ "action": "validate" }
```

8.10.2 Response

204 No Content

8.11 Get a quote

8.11.1 Request

GET /orders/0000014933612cf9fb1c626a007f000000010001/quote

8.11.2 Response

200 OK
Content-Type : application/json; charset=UTF-8

```
{
  "quote": {
    "PurchasedPlan": [
      {
        "Product": [
          {
            "ChildProduct": [
            ],
            "Name": "ng-test-fitness-20141021-1044.us",
            "MonthlyCycle": 12,
            "ConfigType": "core.dnr.us",
            "Fees": [
              {
                "ProrationAllowed": "0",
                "StartDate": "2014-10-21",
                "NumCharged": "0",
                "EndDate": "2015-10-20",
                "Description": "Domain registration fee ng-test-fitness-20141021-1044.us",
                "Discounts": [
                ],
                "BaseAmount": "7.95",
                "UnitAmount": "7.95",
                "Prorated": "0",
                "MonthlyCycle": "12",
                "Currency": "USD",
                "Amount": "7.95",
                "NumCharges": "1",
                "ProratedUnitAmount": "7.95",
                "Type": "domain_registration_fee",
                "Quantity": "1",
                "Deferred": "0"
              }
            ],
            "Data": {
              "Register": {
                "ContactList": {
                  "Contact": [
                    {
                      "@city": "Chicago",
                      "@province": "IL",
                      "@first_name": "Paul",
                      "@fax": "",
                      "@address1": "123 First Str.",
                      "@address2": ""
                    }
                  ]
                }
              }
            }
          }
        ]
      }
    ]
  }
}
```



```

        "@phone":"+1.312.456789111",
        "@country_code":"US",
        "@last_name":"Stevenson",
        "@postal":"34567",
        "@type":"technical",
        "@email":"paul.stevenson@test.com",
        "@company":""
    },
    {
        "@city":"Chicago",
        "@province":"IL",
        "@first_name":"Jane",
        "@fax":"",
        "@address1":"123 First Str.",
        "@address2":"",
        "@phone":"+1.312.456789111",
        "@country_code":"US",
        "@last_name":"Smith",
        "@postal":"34567",
        "@type":"admin",
        "@email":"jane.smith@test.com",
        "@company":""
    },
    {
        "@city":"Chicago",
        "@province":"IL",
        "@first_name":"John",
        "@fax":"",
        "@address1":"123 First Str.",
        "@address2":"",
        "@phone":"+1.312.456789098",
        "@country_code":"US",
        "@last_name":"Doe",
        "@postal":"34567",
        "@type":"company",
        "@email":"john.doe@test.com",
        "@company":"ACME Inc."
    },
    {
        "@city":"Chicago",
        "@province":"IL",
        "@first_name":"James",
        "@fax":"",
        "@address1":"123 First Str.",
        "@address2":"",
        "@phone":"+1.312.456789111",
        "@country_code":"US",
        "@last_name":"Johnson",
        "@postal":"34567",
        "@type":"billing",
        "@email":"james.johnson@test.com",
        "@company":""
    }
]
},
"@term":"1",
"@lock":"",
"@domain_name":"ng-test-fitness-20141021-1044.us",
"@agreed":"",
"@nameservers":"ns1.test.com,ns2.test.com",
"@authInfo":"mysecret",
"@reason":"",
"@legal_type":"",
"@agreement_version":"",
"@operation":"register",
"@ExtraParams":[
    {
        "Name":"us_nexus",
        "Value":"C11"
    },
    {
        "Name":"us_purpose",
        "Value":"P1"
    }
]
},
"@PromotionCode":null,
"@ProductGUID":"0000014933613528f83d7290007f000000010001"
}
],
"@displayName":"",
"@Name":"DNR Only Plan",
"@ReferralKey":null,
"@MonthlyCycle":12,
"@PartnerId":"",
"@Plan":"core.plan.dnr.only",
"@Fees":[
    {
        "@ProrationAllowed":"0",
        "@StartDate":"2014-10-21",
        "@NumCharged":"0",
        "@EndDate":"2015-10-20",
        "@Description":"Recurring fee DNR Only Plan, ng-test-fitness-20141021-1044.us 10/21/2014 through 10/20/2015",
        "@Discounts":[]
    },
    {
        "@BaseAmount":"0.00",
        "@UnitAmount":"0.00",
        "@Prorated":"0",
        "@MonthlyCycle":"12",
        "@Currency":"USD",
        "@Amount":"0.00",
        "@NumCharges":"-1",
        "@ProratedUnitAmount":"0.00",
        "@Type":"recurring",
        "@Quantity":"1",
    }
]

```

```

        "Deferred": "0"
    }
  ],
  "PlanNumber": "0000014933613211db39d7c4007f000000010001",
  "PromotionCode": null,
  "PurchasedPlanNumber": "acct258192-0000014933615c88d6149e83007f00000001000"
}
},
"Hash": "b2f4dba3062c44084d3339561781d3bbc7bb9f70",
"Currency": "USD",
"Total": "7.95",
"SubTotal": "7.95",
"OneTimeFeeAmount": "7.95",
"TaxTotal": "0",
"DeferredSubTotals": [],
"DeferredTaxes": [],
"DeferredTotals": []
},
"links": {
  "self": {
    "href": "http://coreapi01.ote.chicago.hostway:8100/orders/0000014933612cf9fb1c626a007f000000010001/quote/",
    "rel": "self"
  }
}
}
}

```

8.12 Submit the order

Use [POST /orders/{order_id}/action](#) to submit the order.

8.12.1 Request

```

POST /orders/0000014933612cf9fb1c626a007f000000010001/action
Content-Type: application/json

```

```

{
  "action": "submit",
  "quoteHash": "b2f4dba3062c44084d3339561781d3bbc7bb9f70"
}

```

8.12.2 Response

```

202 Accepted

```

See also

- [Product Catalog API](#)
- [Ordering API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

9 Order-Domain-Alias

10 Order a new email domain alias

This example provides the steps to order a new email domain through the Hostway [Ordering API](#).

10.1 Create a new order

Use `POST /orders` to create a new order

10.1.1 Request

```
POST /orders
Content-Type: application/json; charset=UTF-8
{}

```

10.1.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001
Set-Cookie : OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "orderId": "00000143d407b34742ee26ae007f000000010001",
  "links": [
    {"href": "https://api.hostway.com/orders/", "rel": "self"},
    {"href": "https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001", "rel": "item"}
  ]
}

```

10.2 Create a new account for the order

Use `PUT /orders/{order_id}/account` to create an account or add an existing one to the order.

10.2.1 Request

```
PUT /orders/00000143d407b34742ee26ae007f000000010001/account
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "currency": "USD",
  "language": "en-US"
}

```

10.2.2 Response

```
204 No Content

```

10.3 Add an email domain product

The email domain product specification offerings can be determined using the Hostway [Product Catalog API](#). Then add a new email domain product to the order by using `POST /orders/{order_id}/account/products`.

10.3.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "offering": "https://api.hostway.com/productSpecifications/core.plan.email.business/offerings/USD-12"
}

```

10.3.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001

```

10.4 Add an email service to the new product

The email service specification offerings can be determined using the Hostway [Product Catalog API](#) for the core.plan.email.business product. Then add a new email service to the order by using `POST /orders/{order_id}/account/products/{product_id}/services`.

10.4.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "name": "test-email-domain.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.email.mfg2",
  "billingCycle": 1
}

```

10.4.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/
```

10.5 Add service attributes

Use the [PUT /orders/{order_id}/account/products/{product_id}/services/{service_id}/attributes](#) call to add the domain name for the email service.

10.5.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services/00000143d407bea7e2b8
```

```
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
```

```
{
  "domain": "test-email-domain.com"
}
```

10.5.2 Response

```
204 No content
```

10.6 Add an email domain alias service to the new product

The email service specification offerings can be determined using the Hostway [Product Catalog API](#) for the core.plan.email.business product. Then add a new email service to the order by using [POST /orders/{order_id}/account/products/{product_id}/services](#). The "relatedTo" attribute should contain the email domain product URL.

10.6.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
```

```
{
  "name": "test-email-domain-alias.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.email.domainalias.mfg2",
  "relatedTo": ["https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/"],
  "billingCycle": 1
}
```

10.6.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/
```

10.7 Add the domain alias service attributes

Use the [PUT /orders/{order_id}/account/products/{product_id}/services/{service_id}/attributes](#) call to add the domain alias name for the email service.

10.7.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services/00000143d407bea7e2b8
```

```
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
```

```
{
  "domain": "test-email-domain.com",
  "domain_alias": "test-email-domain-alias.com"
}
```

10.7.2 Response

```
204 No content
```

10.8 Submit the order

Use [POST /orders/{order_id}/action](#) to submit the order and invoke the email domain provisioning.

10.8.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/action
Content-Type: application/json; charset=UTF-8
```

```
{
  "action": "submit"
}
```

10.8.2 Response

202 Accepted

See also

- [Product Catalog API](#)
- [Ordering API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

11 Order-Domain-Shortcut

12 Order a new domain

This is an example of ordering a domain with the [Domain Order Shortcut](#). It is the same as the [Order Domain](#) process, using an existing account in the URL.

12.1 Create a new domain registration order

Use `PUT /accounts/{accountId}/domains/{domainName}` to create a new domain registration order for an existing account.

12.1.1 Request

```
PUT /accounts/acct839326/domains/ng-test-Fitness-shortcut-20141022-1033.com
Content-Type: application/json
```

```
{
  "domainName": "ng-test-Fitness-shortcut-20141022-1033.com",
  "operation": "register",
  "term": 1,
  "authInfo": "mysecret",
  "nameservers": [
    "nsl.test.com",
    "ns2.test.com"
  ],
  "contactInfo": {
    "registrant": {
      "type": "registrant",
      "firstName": "John",
      "lastName": "Doe",
      "company": "ACME Inc.",
      "street1": "123 First Str.",
      "city": "Chicago",
      "zipOrPostal": "34567",
      "stateOrProvince": "IL",
      "countryCode": "US",
      "email1": "john.doe@test.com",
      "phone1": "+1.9494567890"
    },
    "administrative": {
      "type": "administrative",
      "firstName": "Jane",
      "lastName": "Smith",
      "street1": "123 First Str.",
      "city": "Chicago",
      "zipOrPostal": "34567",
      "stateOrProvince": "IL",
      "countryCode": "US",
      "email1": "jane.smith@test.com",
      "phone1": "+1.9494567891",
      "fax1": "+1.9494567891"
    },
    "billing": {
      "type": "billing",
      "firstName": "James",
      "lastName": "Johnson",
      "street1": "123 First Str.",
      "city": "Chicago",
      "zipOrPostal": "34567",
      "stateOrProvince": "IL",
      "countryCode": "US",
      "email1": "james.johnson@test.com",
      "phone1": "+1.9494567891"
    },
    "technical": {
      "type": "technical",
      "firstName": "Paul",
      "lastName": "Stevenson",
      "street1": "123 First Str.",
      "city": "Chicago",
      "zipOrPostal": "34567",
      "stateOrProvince": "IL",
      "countryCode": "US",
      "email1": "paul.stevenson@test.com",
      "phone1": "+1.9494567891"
    }
  }
}
```

12.1.2 Response

```
202 Accepted
Content-Type : application/json
```

```
{
  "orderId": "00000149387d9e73153111a9007f000000010001",
  "links": [
    {
      "href": "http://coreapi01.ote.chicago.hostway:8100/orders/00000149387d9e73153111a9007f000000010001",
      "rel": "self"
    }
  ]
}
```

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

13 Order-Email-Domain

14 Order a new email domain

This example provides the steps to order a new email domain through the Hostway [Ordering API](#).

14.1 Create a new order

Use `POST /orders` to create a new order

14.1.1 Request

```
POST /orders
Content-Type: application/json; charset=UTF-8

{}
```

14.1.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf
Set-Cookie : OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "orderId": "0000013ad6dd95746a399f90000a71384bf242cf",
  "links": [
    { "href": "https://api.hostway.com/orders/", "rel": "self" },
    { "href": "https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf", "rel": "item" }
  ]
}
```

14.2 Create a new account for the order

Use `PUT /orders/{order_id}/account` to create an account or add an existing one to the order.

14.2.1 Request

```
PUT /orders/0000013ad6dd95746a399f90000a71384bf242cf/account
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "currency": "USD",
  "language": "en-US"
}
```

14.2.2 Response

```
204 No Content
```

14.3 Add the email domain product

The email domain product specification offerings can be determined using the Hostway [Product Catalog API](#). Then add a new email domain product to the order by using `POST /orders/{order_id}/account/products`.

14.3.1 Request

```
POST /orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "offering": "https://api.hostway.com/productSpecifications/core.plan.email.business/offerings/USD-12"
}
```

14.3.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products/0000013fd6dd95726c399e90000a71384bf242cf
```

14.4 Add the email service to the new product

The email service specification offerings can be determined using the Hostway [Product Catalog API](#) for the core.plan.email.business product. Then add a new email service to the order by using `POST /orders/{order_id}/account/products/{product_id}/services`.

14.4.1 Request

```
POST /orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products/0000013fd6dd95726c399e90000a71384bf242cf/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/

{
  "name": "test-email-domain.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.email.mfg2",
  "billingCycle": 1
}
```

14.4.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products/0000013fd6dd95726c399e90000a71384bf242cf/
```

14.5 Add service attributes

Use the [PUT /orders/{order_id}/account/products/{product_id}/services/{service_id}/attributes](#) call to add the domain name for the email service.

14.5.1 Request

```
POST /orders/0000013ad6dd95746a399f90000a71384bf242cf/account/products/0000013fd6dd95726c399e90000a71384bf242cf/services/0000013fe6dd95726c399e90000a71384bf242cf/
```

```
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/
```

```
{
  "domain": "test-email-domain.com"
}
```

14.5.2 Response

```
204 No content
```

14.6 Submit the order

Use [POST /orders/{order_id}/action](#) to submit the order and invoke the email domain provisioning.

14.6.1 Request

```
POST /orders/0000013ad6dd95746a399f90000a71384bf242cf/action
Content-Type: application/json; charset=UTF-8
```

```
{
  "action": "submit"
}
```

14.6.2 Response

```
202 Accepted
```

See also

- [Product Catalog API](#)
- [Ordering API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

15 Order-Exchange

15.1 Order a new Exchange domain

This example provides the steps for ordering a new Exchange domain

15.2 Create a new order

Use [POST /orders](#) to create a new order

15.2.1 Request

```
POST /orders
Content-Type: application/json; charset=UTF-8
{}

```

15.2.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001
Set-Cookie : OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "orderId": "00000143d407b34742ee26ae007f000000010001",
  "links": [
    {"href": "https://api.hostway.com/orders/", "rel": "self"},
    {"href": "https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001", "rel": "item"}
  ]
}

```

15.3 Create a new account for the order

Use [PUT /orders/{order_id}/account](#) to create an account or add an existing one to the order.

15.3.1 Request

```
PUT /orders/00000143d407b34742ee26ae007f000000010001/account
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "currency": "USD",
  "language": "en-US"
}

```

15.3.2 Response

```
204 No Content

```

15.4 Add an Exchange product to the order

The Exchange product specification offerings can be determined using the Hostway [Product Catalog API](#). Then add a new websplanet product to the order by using [POST /orders/{order_id}/account/products](#).

15.4.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "offering": "https://api.hostway.com/productSpecifications/core.plan.email.exchange2010/offerings/USD-1"
}

```

15.4.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001

```

15.5 Add an Exchange service to the new product

15.5.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{"name": "myexchangedomain.com",
 "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.email.exchange2010",
 "billingCycle": 1,
 "attributes": {

```

```
    "domain": "myexchangedomain.com"
  }
}
```

15.5.2 Response

201 Created

Content-Type: application/json; charset=UTF-8

Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/

15.6 Obtain a quote

15.6.1 Request

```
GET /orders/00000143d407b34742ee26ae007f000000010001/quote
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
```

15.6.2 Response

200 OK

Content-Type: application/json; charset=UTF-8

```
{
  "quote" : {
    "Hash" : "b6a714a2ba76b6f15df9e46964df21ca726594a0",
    "Currency" : "USD",
    "Total" : "0.00",
    "SubTotal" : "0.00",
    "OneTimeFeeAmount" : "0",
    "TaxTotal" : "0"
  },
  "links" : {
    "self" : {
      "href" : "http://coreapi01.ote.chicago.hostway:8092/orders/00000143d407b34742ee26ae007f000000010001/quote/",
      "rel" : "self"
    }
  }
}
```

15.7 Submit the order

15.7.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/action
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{"action": "submit"}
```

15.7.2 Response

202 Accepted

no-body

15.8 Check for status

15.8.1 Request

```
GET /orders/00000143d407b34742ee26ae007f000000010001
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
Authorization: Basic cXd1QHhcnRuzXJfYXBpX2RldmVsb3Blcl9lc19tYWluOml1MzR0OGhm <- credentials that have access to the account
```

15.8.2 Response

200 OK

```
{
  "order" : {
    "Status" : "Processing",
    "Account" : "http://coreapi01.ote.chicago.hostway:8092/accounts/acct999999" <- "Completed" means that everything has p
  },
  "links" : [ {
    "href" : "http://coreapi01.ote.chicago.hostway:8092/orders/00000143d407b34742ee26ae007f000000010001/",
    "rel" : "self"
  } ]
}
```

See also

- [Product Catalog API](#)
- [Ordering API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

16 Order-Exchange-mailbox-free

16.1 Order a new, free Exchange mailbox

This example provides the steps for ordering a new Exchange mailbox in the case where it's offered without charge

16.2 Create the mailbox

16.2.1 Request

```
POST /accounts/acct999999/exchange/myexchangedomain.com/mailboxes
Content-Type: application/json; charset=UTF-8
Authorization: Basic cXdlQHBhcnRuZXJfYXBpX2RldmVsb3B1c191c19tYWluOml1MzR0OGhm
```

```
{
  "owner": {
    "username": "a.a",
    "password": "tester1234"
  },
  "admin": 0,
  "notes": "some text",
  "recipientLimit": 10,
  "deliverRedirect": "zsds",
  "forwardTo": "asdasd",
  "name": {
    "name": "John Smith",
    "company": "My Company",
    "department": "Sales",
    "title": "President"
  },
  "address": {
    "address": "100 Main St.",
    "city": "Chicago",
    "state": "IL",
    "zip": "60001",
    "country": "US"
  },
  "phones": {
    "businessPhone": "555-555-5501",
    "homePhone": "555-555-5502",
    "fax": "555-555-5503",
    "mobile": "555-555-5504"
  }
}
```

16.2.2 Response

```
202 Accepted
Location      http://coreapi01.ote.chicago.hostway:8092/orders/00000145ff9ba7e58e7c2c51007f000000010001
Set-Cookie   OwnerId=f9cafb10-2e85-4b1a-bd69-1b1a91437d50; Path=/
no-body
```

Note: order has already been submitted at this point!

16.3 Check for status

16.3.1 Request

```
GET /orders/00000145ff9ba7e58e7c2c51007f000000010001
Cookie: OwnerId=f9cafb10-2e85-4b1a-bd69-1b1a91437d50; Path=/
Authorization: Basic cXdlQHBhcnRuZXJfYXBpX2RldmVsb3B1c191c19tYWluOml1MzR0OGhm
```

16.3.2 Response

```
200 OK
{
  "order": {
    "Status": "Processing",
    "Account": "http://coreapi01.ote.chicago.hostway:8092/accounts/acct999999"
  },
  "links": [ {
    "href": "http://coreapi01.ote.chicago.hostway:8092/orders/00000145ff9ba7e58e7c2c51007f000000010001/",
    "rel": "self"
  } ]
}
```

<- "Completed" means that everything provisioned

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

17 Order-Exchange-mailbox-paid

17.1 Order a new, paid Exchange mailbox

This example provides the steps for ordering a new Exchange mailbox in the case where a charge applies

17.2 Create the mailbox

17.2.1 Request

```
POST /accounts/acct999999/exchange/myexchangedomain.com/mailboxes
Content-Type: application/json; charset=UTF-8
Authorization: Basic cXdlQHBhcnRuZXJfYXBpX2RldmVsb3B1c191c19tYWluOml1MzR0OGhm
```

```
{
  "owner": {
    "username": "a.a",
    "password": "tester1234"
  },
  "admin": 0,
  "notes": "some text",
  "recipientLimit": 10,
  "deliverRedirect": "zsds",
  "forwardTo": "asdasd",
  "name": {
    "name": "John Smith",
    "company": "My Company",
    "department": "Sales",
    "title": "President"
  },
  "address": {
    "address": "100 Main St.",
    "city": "Chicago",
    "state": "IL",
    "zip": "60001",
    "country": "US"
  },
  "phones": {
    "businessPhone": "555-555-5501",
    "homePhone": "555-555-5502",
    "fax": "555-555-5503",
    "mobile": "555-555-5504"
  }
}
```

17.2.2 Response

```
202 Accepted
Set-Cookie: OwnerId=bd048fa5-8714-462f-86f6-1feffd04023b; Path=/
{
  "total": [ "5.18", "USD" ],
  "subtotal": [ "5.18", "USD" ],
  "deferredTotal": [],
  "deferredSubtotal": [],
  "quotehash": "0dd6b20d1ce3ae0e44882f3db11124d03f60ba53",
  "links": [ {
    "href": "http://coreapi01.ote.chicago.hostway:8092/accounts/acct999999/exchange/myexchangedomain.com/mailboxes/",
    "rel": "self"
  }, {
    "href": "http://coreapi01.ote.chicago.hostway:8092/orders/00000145ff82ec6bf27d6421007f000000010001/action",
    "rel": "related"
  } ]
}
```

17.3 Submit the order

17.3.1 Request

```
POST /orders/00000145ff82ec6bf27d6421007f000000010001/action
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=bd048fa5-8714-462f-86f6-1feffd04023b; Path=/
{"action": "submit"}
</nowiki>
```

```
=== Response ===
<nowiki>
202 Accepted
no-body
```

17.4 Check for status

17.4.1 Request

```
GET /orders/00000145ff82ec6bf27d6421007f000000010001
Cookie: OwnerId=bd048fa5-8714-462f-86f6-1feffd04023b; Path=/
Authorization: Basic cXdlQHBhcnRuZXJfYXBpX2RldmVsb3B1c191c19tYWluOml1MzR0OGhm
```


17.4.2 Response

200 OK

```
{
  "order" : {
    "Status" : "Processing",
    "Account" : "http://coreapi01.ote.chicago.hostway:8092/accounts/acct999999"
  },
  "links" : [ {
    "href" : "http://coreapi01.ote.chicago.hostway:8092/orders/00000145ff82ec6bf27d6421007f000000010001/",
    "rel" : "self"
  } ]
}
```

<- "Completed" means that everything provisioned

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

18 Order-Exchange-otherproduct

18.1 Ordering Other Exchange Products

This order flow applies to encryption, archiving, mailing list and blackberry products. The example illustrates ordering encryption.

18.2 Create Encryption for a domain

18.2.1 Request

```
PUT /accounts/acct999999/exchange/myexchangedomain.com/emailEncryption
Authorization: Basic cXdlQHBhcnRuZlJfYXBpX2RldmVsb3Blcl91c19tYWluOml1MzR0OGhm
{
  "adminEmail": "someone@somewhere.com"
}
```

18.2.2 Response

```
202 Accepted
Set-Cookie: OwnerId=73adb746-2443-4cc8-879e-886348385b9a; Path=/
Location: http://coreapi01.ote.chicago.hostway:8092/orders/00000145ffa79d31de17529f007f000000010001
no-body
```

Note: specified order has not yet been submitted.

18.3 Obtain a quote

18.3.1 Request

```
GET /orders/00000145ffa79d31de17529f007f000000010001/quote
Cookie: OwnerId=73adb746-2443-4cc8-879e-886348385b9a; Path=/
Authorization: Basic cXdlQHBhcnRuZlJfYXBpX2RldmVsb3Blcl91c19tYWluOml1MzR0OGhm
```

18.3.2 Response

```
200 OK
Content-Type: application/json; charset=UTF-8
{
  "quote" : {
    "Hash" : "b6a714a2ba76b6f15df9e46964df21ca726594a0",
    "Currency" : "USD",
    "Total" : "0.00",
    "SubTotal" : "0.00",
    "OneTimeFeeAmount" : "0",
    "TaxTotal" : "0",
    "DeferredSubTotals" : [],
    "DeferredTaxes" : [],
    "DeferredTotals" : []
  },
  "links" : {
    "self" : {
      "href" : "http://coreapi01.ote.chicago.hostway:8092/orders/00000145ffa79d31de17529f007f000000010001/quote/",
      "rel" : "self"
    }
  }
}
```

18.4 Submit the order

18.4.1 Request

```
POST /orders/00000145ffa79d31de17529f007f000000010001/action
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=73adb746-2443-4cc8-879e-886348385b9a; Path=/
Authorization: Basic cXdlQHBhcnRuZlJfYXBpX2RldmVsb3Blcl91c19tYWluOml1MzR0OGhm
{"action": "submit"}
</nowiki>
```

```
=== Response ===
<nowiki>
202 Accepted
no-body
```

18.5 Check for status

18.5.1 Request

```
GET /orders/00000145ffa79d31de17529f007f000000010001
Cookie: OwnerId=73adb746-2443-4cc8-879e-886348385b9a; Path=/
Authorization: Basic cXdlQHBhcnRuZlJfYXBpX2RldmVsb3Blcl91c19tYWluOml1MzR0OGhm
```

18.5.2 Response

200 OK

```
{
  "order" : {
    "Status" : "Processing",
    "Account" : "http://coreapi01.ote.chicago.hostway:8092/accounts/acct999999"
  },
  "links" : [ {
    "href" : "http://coreapi01.ote.chicago.hostway:8092/orders/00000145ffa79d31de17529f007f000000010001/",
    "rel" : "self"
  } ]
}
```

<- "Completed" means that everything has p

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

19 Order-Products-Account

20 Set account on a later step when ordering products and services

This example provides the steps to set order's account on a later step.

20.1 Create a new order

Use `POST /orders` to create a new order

20.1.1 Request

```
POST /orders
Content-Type: application/json; charset=UTF-8
{}

```

20.1.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/000001452259ee0103d4e652007f000000010001
Set-Cookie : OwnerId=d0977ed2-86e7-4101-adfe-2675ba0e26ac; Path=/
{
  "orderId": "000001452259ee0103d4e652007f000000010001",
  "links": [
    { "href": "https://api.hostway.com/orders/", "rel": "self" },
    { "href": "https://api.hostway.com/orders/000001452259ee0103d4e652007f000000010001", "rel": "item" }
  ]
}

```

20.2 Add a product

Use `POST /orders/{order_id}/account/products` to add a product

20.2.1 Request

```
POST /orders/000001452259ee0103d4e652007f000000010001/account/products
Content-Type: application/json; charset=UTF-8

```

```
{
  "offering": "https://api.hostway.com/productSpecifications/core.plangroup.dedserver.plan4/offerings/USD-1"
}

```

20.2.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/000001452259ee0103d4e652007f000000010001/account/products/00000145225a87f17bd8d66d007f000000010001/
{
  "url": "https://api.hostway.com/orders/000001452259ee0103d4e652007f000000010001/account/products/00000145225a87f17bd8d66d007f000000010001/"
}

```

20.3 Add a service to the product with no attributes

Use `POST /orders/{order_id}/account/products/{productId}/services` to add a product's service

20.3.1 Request

```
POST /orders/000001452259ee0103d4e652007f000000010001/account/products/00000145225a87f17bd8d66d007f000000010001/services
Content-Type: application/json; charset=UTF-8

```

```
{
  "name": "test123.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.dedserver",
  "billingCycle": 1
}

```

20.3.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/000001452292557b24c2acc0007f000000010001/account/products/0000014522925d3222f0ce5b007f000000010001/
{
  "url": "https://api.hostway.com/orders/000001452292557b24c2acc0007f000000010001/account/products/0000014522925d3222f0ce5b007f000000010001/se"
}

```

20.4 Create a new account for the order

Use `PUT /orders/{order_id}/account` to add a new account to the order

20.4.1 Request

```
PUT /orders/000001452292557b24c2acc0007f000000010001/account
Content-Type: application/json; charset=UTF-8

```

```
{
  "currency": "USD",
  "language": "en-US"
}
```

20.4.2 Response

204 No Content

20.5 Set an existing account for the order

Use [PUT /orders/{order_id}/account](#) to assign an existing account to the order

20.5.1 Request

```
PUT /orders/000001452292557b24c2acc0007f000000010001/account
Content-Type: application/json; charset=UTF-8
```

```
{
  "accountNumber": "test793347"
}
```

20.5.2 Response

204 No Content

20.6 Submit the order

Use [POST /orders/{order_id}/action](#) to submit the order and to invoke the ordered product provisioning.

20.6.1 Request

```
POST /orders/000001452292557b24c2acc0007f000000010001/action
Content-Type: application/json; charset=UTF-8
```

```
{
  "action": "submit"
}
```

20.6.2 Response

202 Accepted

See also

- [Product Catalog API](#)
- [Ordering API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

21 Order-Products-Account-Modify

22 Set and modify an account on a later step when ordering products and services

This example provides the steps to set order's account on a later step and modify it afterward.

22.1 Create a new order

Use `POST /orders` to create a new order

22.1.1 Request

```
POST /orders
Content-Type: application/json; charset=UTF-8

{}
```

22.1.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/000001452259ee0103d4e652007f000000010001
Set-Cookie : OwnerId=d0977ed2-86e7-4101-adfe-2675ba0e26ac; Path=/

{
  "orderId": "000001452259ee0103d4e652007f000000010001",
  "links": [
    {"href": "https://api.hostway.com/orders/", "rel": "self"},
    {"href": "https://api.hostway.com/orders/000001452259ee0103d4e652007f000000010001", "rel": "item"}
  ]
}
```

22.2 Add a product

Use `POST /orders/{order_id}/account/products` to add a product

22.2.1 Request

```
POST /orders/000001452259ee0103d4e652007f000000010001/account/products
Content-Type: application/json; charset=UTF-8

{
  "offering": "https://api.hostway.com/productSpecifications/core.plangroup.dedserver.plan4/offerings/USD-1"
}
```

22.2.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/000001452259ee0103d4e652007f000000010001/account/products/00000145225a87f17bd8d66d007f000000010001/

{
  "url": "https://api.hostway.com/orders/000001452259ee0103d4e652007f000000010001/account/products/00000145225a87f17bd8d66d007f000000010001/"
}
```

22.3 Add a service to the product with no attributes

Use `POST /orders/{order_id}/account/products/{productId}/services` to add a product's service

22.3.1 Request

```
POST /orders/000001452259ee0103d4e652007f000000010001/account/products/00000145225a87f17bd8d66d007f000000010001/services
Content-Type: application/json; charset=UTF-8

{
  "name": "test123.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.dedserver",
  "billingCycle": 1
}
```

22.3.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/000001452292557b24c2acc0007f000000010001/account/products/0000014522925d3222f0ce5b007f000000010001/

{
  "url": "https://api.hostway.com/orders/000001452292557b24c2acc0007f000000010001/account/products/0000014522925d3222f0ce5b007f000000010001/"
}
```

22.4 Set an existing account for the order

Use `PUT /orders/{order_id}/account` to assign an existing account to the order

22.4.1 Request

```
PUT /orders/000001452292557b24c2acc0007f000000010001/account
Content-Type: application/json; charset=UTF-8
```

```
{
  "accountNumber": "test793347"
}
```

22.4.2 Response

204 No Content

22.5 Set some account attributes

Use [PUT /orders/{orderId}/account/attributes](#) to set the account attributes in an order.

22.5.1 Request

```
PUT /orders/0000014522fbfd383c2b1a33007f000000010001/account/attributes
Content-Type: application/json; charset=UTF-8
```

```
[
  {
    "name": "partner_account_id",
    "value": "paccid_030414051600"
  }
]
```

22.5.2 Response

204 No Content

22.6 Set modified regular contact

Use [PUT /orders/{orderId}/account/contactInfo/{contactType}](#) to set the account contact details for a given contact type in an order.

22.6.1 Request

```
PUT /orders/0000014522fbfd383c2b1a33007f000000010001/account/contactInfo/regular
Content-Type: application/json; charset=UTF-8
```

```
{
  "name":
  {
    "firstName": "Test-Regular",
    "lastName": "Contact",
    "company": "MyCompany"
  },
  "address":
  {
    "street1": "111 My Street",
    "street2": "street2",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  },
  "contactMedia":
  {
    "phone1": "312-555-1111",
    "phone2": "312-555-222",
    "fax": "312-555-1111",
    "email1": "myemail@something.com",
    "emailFormat": "html"
  }
}
```

22.7 Set account payment info

Use [PUT /orders/{orderId}/account/paymentInfo](#) to set the account's payment info.

22.7.1 Request

```
PUT /orders/0000014522fbfd383c2b1a33007f000000010001/account/paymentInfo
Content-Type: application/json; charset=UTF-8
```

```
{
  "paymentInfoType": "credit_card",
  "creditCardType": "visa",
  "accountNumber": "1234567890123456",
  "cardHolderName": "John Jello",
  "expirationDate": "2015-06",
  "verificationNumber": "503",
  "billingAddress":
  {
    "street1": "My street",
    "street2": "",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  }
}
```

22.7.2 Response

204 No Content

22.7.3 Response

204 No Content

22.8 Submit the order

Use [POST /orders/{order_id}/action](#) to submit the order and to invoke the ordered product provisioning.

22.8.1 Request

```
POST /orders/000001452292557b24c2acc0007f000000010001/action
Content-Type: application/json; charset=UTF-8
```

```
{
  "action": "submit"
}
```

22.8.2 Response

202 Accepted

See also

- [Product Catalog API](#)
- [Ordering API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

23 Order-vps-virtuozzo

23.1 Order VPS Virtuozzo products and services

This example provides the steps to order a new VPS Virtuozzo product and services through the Hostway OrderAPI.

23.2 Create a new order

Use [POST /orders](#) to create a new order

23.2.1 Request

```
POST /orders
Content-Type: application/json; charset=UTF-8
{}

```

23.2.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001
Set-Cookie : OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "orderId": "00000143d407b34742ee26ae007f000000010001",
  "links": [
    {"href": "https://api.hostway.com/orders/", "rel": "self"},
    {"href": "https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001", "rel": "item"}
  ]
}

```

23.3 Create a new account or add an existing one to the order

Use [PUT /orders/{order_id}/account](#) to create an account or add an existing one to the order.

23.3.1 Request for creating a new account

```
PUT /orders/00000143d407b34742ee26ae007f000000010001/account
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "currency": "USD",
  "language": "en-US"
}

```

23.3.2 Request for adding an existing account

```
PUT /orders/00000143d407b34742ee26ae007f000000010001/account
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "accountNumber": "acct123456"
}

```

23.3.3 Response

```
204 No Content

```

23.4 Add a VPS Virtuozzo product to the order

The VPS Virtuozzo product specification offerings can be determined using the Hostway [Product Catalog API](#). Then add a new VPS Virtuozzo product to the order by using [POST /orders/{order_id}/account/products](#).

23.4.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "offering": "https://api.hostway.com/productSpecifications/core.plan.vps.virtuozzo.ul01/offerings/USD-1"
}

```

23.4.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001

```

23.5 Add a Virtuozzo Server service to the new product

Virtuozzo Server Windows/Linux service needs to be created first before any other VPS Virtuozzo services(IP, Memory, ControlPanel, Disk, Bandwidth, PowerPack) are added.

23.5.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "name": "VPS Virtuozzo Test",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.vps.virtuozzo.server.u100",
  "billingCycle": 1
}
```

23.5.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/
```

23.6 Add VPS Virtuozzo services to the new product

Please note that VPS Virtuozzo services(IP, Memory, ControlPanel, Disk, Bandwidth) can be added only as a child service to the Virtuozzo Server Windows/Linux service. You have to specify Virtuozzo Server Windows/Linux service you want your VPS Virtuozzo services(IP, Memory, ControlPanel, Disk, Bandwidth) to be added to.

23.6.1 Add VPS Virtuozzo Controlpanel service

Assuming you have already a Virtuozzo Server Windows/Linux service in your order, you can add Virtuozzo Controlpanel service using [POST /orders/{order_id}/account/products/{product_id}/services](#). The "relatedTo" attribute should contain the Virtuozzo Server Windows/Linux service's URL.

23.6.2 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "name": "VPS Virtuozzo Test",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.vps.virtuozzo.controlpanel.plesk9-10domain",
  "billingCycle": 1,
  "relatedTo": ["https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/"]
}
```

23.6.3 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/
```

23.6.4 Add VPS Virtuozzo Additional Memory service

Assuming you have already a Virtuozzo Server Windows/Linux service in your order, you can add VPS Virtuozzo Additional Memory service using [POST /orders/{order_id}/account/products/{product_id}/services](#). The "relatedTo" attribute should contain the Virtuozzo Server Windows/Linux service's URL.

23.6.5 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "name": "VPS Virtuozzo Test",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.vps.virtuozzo.additional_memory.256mb",
  "billingCycle": 1,
  "relatedTo": ["https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/"]
}
```

23.6.6 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/
```

23.6.7 Add VPS Virtuozzo Additional Disk Space service

Assuming you have already a Virtuozzo Server Windows/Linux service in your order, you can add VPS Virtuozzo Additional Disk Space service using [POST /orders/{order_id}/account/products/{product_id}/services](#). The "relatedTo" attribute should contain the Virtuozzo Server Windows/Linux service's URL.

23.6.8 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "name": "VPS Virtuozzo Test",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.vps.virtuozzo.additional_disk.10gb",
  "billingCycle": 1,
  "relatedTo": ["https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/"]
}
```

23.6.9 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/
```

23.6.10 Add VPS Virtuozzo Additional Bandwidth service

Assuming you have already a Virtuozzo Server Windows/Linux service in your order, you can add Virtuozzo Additional Bandwidth service using [POST /orders/{order_id}/account/products/{product_id}/services](#). The "relatedTo" attribute should contain the Virtuozzo Server Windows/Linux service's URL.

23.6.11 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "name": "VPS Virtuozzo Test",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.vps.virtuozzo.additional_bandwidth.1000gb",
  "billingCycle": 1,
  "relatedTo": ["https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/"]
}
```

23.6.12 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/
```

23.6.13 Add VPS Virtuozzo IP service

Assuming you have already a Virtuozzo Server Windows/Linux service in your order, you can add Virtuozzo IP service using [POST /orders/{order_id}/account/products/{product_id}/services](#). The "relatedTo" attribute should contain the Virtuozzo Server Windows/Linux service's URL.

23.6.14 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "name": "VPS Virtuozzo Test",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.vps.virtuozzo.ip",
  "billingCycle": 1,
  "relatedTo": ["https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/"]
}
```

23.6.15 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/
```

23.6.16 Add VPS Virtuozzo PowerPack service

Assuming you have already a Virtuozzo Server Windows/Linux service in your order, you can add VPS Virtuozzo PowerPack service using [POST /orders/{order_id}/account/products/{product_id}/services](#). The "relatedTo" attribute should contain the Virtuozzo Server Windows/Linux service's URL.

23.6.17 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{
  "name": "VPS Virtuozzo Test",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.vps.virtuozzo.powerpack",
  "billingCycle": 1,
  "relatedTo": ["https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/"]
}
```

23.6.18 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/
```

23.7 Submit the order

Use [POST /orders/{order_id}/action](#) to submit the order and invoke the cloud drive provisioning.

23.7.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/action
Content-Type: application/json; charset=UTF-8
{
  "action": "submit"
}
```

23.7.2 Response

202 Accepted

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

24 Order-webs-planet

24.1 Order a new WebsPlanet

This example provides the steps to order a new webs planet product through the Hostway Ordering API.

24.2 Create a new order

Use `POST /orders` to create a new order

24.2.1 Request

```
POST /orders
Content-Type: application/json; charset=UTF-8

{}
```

24.2.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001
Set-Cookie : OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/

{
  "orderId": "00000143d407b34742ee26ae007f000000010001",
  "links": [
    { "href": "https://api.hostway.com/orders/", "rel": "self" },
    { "href": "https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001", "rel": "item" }
  ]
}
```

24.3 Create a new acout for the order

Use `PUT /orders/{order_id}/account` to create an account or add an existing one to the order.

24.3.1 Request

```
PUT /orders/00000143d407b34742ee26ae007f000000010001/account
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/

{
  "currency": "USD",
  "language": "en-US"
}
```

24.3.2 Response

```
204 No Content
```

24.4 Add a Websplanet product to the order

The websplanet product specification offerings can be determined using the Hostway [Product Catalog API](#). Then add a new websplanet product to the order by using `POST /orders/{order_id}/account/products`.

24.4.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/

{
  "offering": "https://api.hostway.com/productSpecifications/core.plan.linux.webstart/offerings/USD-1"
}
```

24.4.2 Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001
```

24.5 Add a Webhosting service to the new product

A Websplanet service needs to be related to either a Windows or a Linux webhosting service. To do that, the webhosting service needs to be created first.

24.5.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
{"name": "Websplanet",
```

```
"serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.webhosting.linux.5GB",
"billingCycle": 1}
```

24.5.2 Response

```
201 Created
```

```
Content-Type: application/json; charset=UTF-8
```

```
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/
```

24.6 Add the Webhosting service attributes

```
PUT /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services/00000143d407b34742ee26ae007f000000010001/
```

```
Content-Type: application/json; charset=UTF-8
```

```
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
```

```
{"domain": "ng-test-fitness-websplanet-20140505-0505.us"}
```

24.7 Add a Websplanet service to the new product

Please note that WebsPlanet can be added only as a child service to the webhosting service. You have to specify webhosting service you want websplanet to be added to.

Assuming you have a webhosting service in your order, you can add websplanet using [POST /orders/{order_id}/account/products/{product_id}/services](#). The "relatedTo" attribute should contain the webhosting service's URL.

24.7.1 Request

```
POST /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services
```

```
Content-Type: application/json; charset=UTF-8
```

```
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
```

```
{"name": "Websplanet",
```

```
"serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.websplanet.1",
```

```
"billingCycle": 1,
```

```
"relatedTo": ["https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/"]}
```

24.7.2 Response

```
201 Created
```

```
Content-Type: application/json; charset=UTF-8
```

```
Location : https://api.hostway.com/orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/
```

24.8 Add the WebsPlanet service attributes

```
PUT /orders/00000143d407b34742ee26ae007f000000010001/account/products/00000143d407b725d9eb9a41007f000000010001/services/00000143d407bea7e2b84
```

```
Content-Type: application/json; charset=UTF-8
```

```
Cookie: OwnerId=09185e9b-2ea0-4bef-950e-fc9f8b89861f; Path=/
```

```
{"domain": "ng-test-fitness-websplanet-20140505-0505.us",
```

```
"email": "test@example.com"}
```

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

25 Orders

26 Ordering API

The ordering API supports the operations of creating, building and placing orders. The orders are the main method of creating new accounts and adding and managing the products for an account.

26.1 Contents

- 1 Ordering API
 - ◆ 1.1 Orders
 - ◆ 1.2 Order Shorthand
 - ◆ 1.3 Order Attributes
 - ◆ 1.4 Account
 - ◆ 1.5 Account Attributes
 - ◆ 1.6 Account Contact Info
 - ◆ 1.7 Account Payment Information
 - ◆ 1.8 Account User
 - ◆ 1.9 Products
 - ◆ 1.10 Services
 - ◆ 1.11 Service Attributes
 - ◆ 1.12 Service Notes
 - ◆ 1.13 Promotion Codes
 - ◇ 1.13.1 Product Promotion Codes
 - ◇ 1.13.2 Service Promotion Codes
 - ◆ 1.14 Custom Discounts
 - ◇ 1.14.1 Order Discounts
 - ◇ 1.14.2 Product Discounts
 - ◇ 1.14.3 Service Discounts
 - ◆ 1.15 Order Quote
 - ◆ 1.16 See also
- 2 Order Flow Examples
 - ◆ 2.1 Order Account Associations
 - ◆ 2.2 Email Products
 - ◆ 2.3 Microsoft Exchange Products
 - ◆ 2.4 Cloud Products
 - ◆ 2.5 WebsPlanet Products
 - ◆ 2.6 VPS Virtuozzo Products
 - ◆ 2.7 Domain Registration Products

[download this selection of articles as a PDF book](#)

26.2 Orders

Order management resources.

- [POST /orders](#) - Creates a new order
- [GET /orders/{orderId}](#) - Returns order details
- [DELETE /orders/{orderId}](#) - Deletes an order
- [POST /orders/{orderId}/action](#) - Performs an action on an order

26.3 Order Shorthand

An URL shortcut allowing the user to access the currently open order without specifying the order ID.

- [Order Shorthand](#)

26.4 Order Attributes

Custom attributes for an Order.

- [PUT /orders/{orderId}/attributes](#) - Sets order attributes
- [GET /orders/{orderId}/attributes](#) - Returns order attributes

26.5 Account

Customer information in a given order. Only one account is allowed per order.

- [PUT /orders/{orderId}/account](#) - Add account to the order
- [GET /orders/{orderId}/account](#) - Returns the account details

26.6 Account Attributes

Custom attributes for an Account in a given Order.

- [PUT /orders/{orderId}/account/attributes](#) - Sets account attributes
- [GET /orders/{orderId}/account/attributes](#) - Returns account attributes
- [DELETE /orders/{orderId}/account/attributes](#) - Deletes all account attributes

26.7 Account Contact Info

Various types of Contact Info set for an Account in a given Order.

- [PUT /orders/{orderId}/account/contactInfo/{contactType}](#) - Set contact details
- [GET /orders/{orderId}/account/contactInfo/{contactType}](#) - Returns contact details
- [GET /orders/{orderId}/account/contactInfo](#) - Returns all account contacts
- [DELETE /orders/{orderId}/account/contactInfo/{contactType}](#) - Deletes specific contact

26.8 Account Payment Information

Payment method information for an Account in a given Order. An account cannot have multiple payment methods.

- [PUT /orders/{orderId}/account/paymentInfo](#) - Sets the payment method for an account
- [GET /orders/{orderId}/account/paymentInfo](#) - Returns the payment method for an account

26.9 Account User

User information for an Account in a given Order.

- [PUT /orders/{orderId}/account/user](#) - Sets the account user credentials
- [GET /orders/{orderId}/account/user](#) - Returns information about account user credentials

26.10 Products

Products included in the order.

- [POST /orders/{orderId}/account/products](#) - Adds product to the account
- [GET /orders/{orderId}/account/products](#) - Gets all products
- [GET /orders/{orderId}/account/products/{productId}](#) - Gets the product details
- [PUT /orders/{orderId}/account/products/{productId}](#) - Updates product details
- [DELETE /orders/{orderId}/account/products/{productId}](#) - Removes product from the order

26.11 Services

Services included in the products.

- [POST /orders/{orderId}/account/products/{productId}/services](#) - Adds service to a product
- [GET /orders/{orderId}/account/products/{productId}/services](#) - Gets all services for a product
- [GET /orders/{orderId}/account/products/{productId}/services/{serviceId}](#) - Gets service details
- [PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}](#) - Updates service details
- [DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}](#) - Removes service from a product

26.12 Service Attributes

Attributed for the included services.

- [PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes](#) - Adds service attributes
- [GET /orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes](#) - Gets service attributes
- [OPTIONS /orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes](#) - Returns information about supported attributes
- [Service Attributes Structure](#) - Structure of attributes set supported by Hostway API

26.13 Service Notes

Notes for services.

- [POST /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes](#) - Adds a service note

- PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes - Updates the service notes
- GET /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes - Gets all the service notes
- DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes - Deletes the service notes
- OPTIONS /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes - Returns information about supported attributes

26.14 Promotion Codes

Describes handling of promotion codes.

26.14.1 Product Promotion Codes

- PUT /orders/{orderId}/account/products/{productId}/promotion - Add or update promotion code for a product.
- DELETE /orders/{orderId}/account/products/{productId}/promotion - Removes the promotion code for a product.

26.14.2 Service Promotion Codes

- PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/promotion - Add or update promotion code for a service.
- DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}/promotion - Removes the promotion code for a service.

26.15 Custom Discounts

Custom discounts at the order, product and service levels. Managing discounts requires elevated permissions.

26.15.1 Order Discounts

- PUT /orders/{orderId}/discounts - Adds order discounts
- GET /orders/{orderId}/discounts - Returns all order discounts
- DELETE /orders/{orderId}/discounts - Removes all order discounts

26.15.2 Product Discounts

- PUT /orders/{orderId}/account/products/{productId}/discounts - Adds product discounts
- GET /orders/{orderId}/account/products/{productId}/discounts - Returns all product discounts
- DELETE /orders/{orderId}/account/products/{productId}/discounts - Removes all product discounts

26.15.3 Service Discounts

- PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts - Adds service discounts
- GET /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts - Returns all service discounts
- DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts - Removes all service discounts

26.16 Order Quote

Quote information for a given order.

- GET /orders/{orderId}/quote - Returns quote for the products and services in the order

26.17 See also

- Common Features

27 Order Flow Examples

27.1 Order Account Associations

- [Order Products Account](#) - Example order flow with adding initially products and than account
- [Order Products Account Modify](#) - Example order flow with adding initially products than account and modify it later

27.2 Email Products

- [Order Email Domain](#) - Example order flow for purchasing an email domain
- [Order Domain Alias](#) - Example order flow for purchasing an email domain alias

27.3 Microsoft Exchange Products

- [Order Exchange domain](#) - Example order flow for purchasing exchange domain
- [Order Free Exchange Mailbox](#)
- [Order Paid Exchange Mailbox](#)
- [Ordering Other Exchange Products](#) - Example order flow that applies to Exchange email encryption, archiving, mailing list and blackberry products

27.4 Cloud Products

- [Order Cloud Drive](#) - Example order flow for purchasing a Cloud Drive
- [Order Cloud Backup](#) - Example order flow for purchasing a Cloud Backup

27.5 WebsPlanet Products

- [WebsPlanet](#) - Example order flow for purchasing a WebsPlanet

27.6 VPS Virtuozzo Products

- [VpsVirtuozzo](#) - Example order flow for purchasing VPS Virtuozzo products and services

27.7 Domain Registration Products

- [Order Domain](#) - Example order flow for purchasing a domain
- [Order Domain Shortcut](#) - Example order flow for purchasing a domain by using the domain shortcut

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

28 Orders-orderId-account-attributes:DELETE

29 DELETE /orders/{orderId}/account/attributes

Remove all attributes for an account in a given order.

29.1 Contents

- 1 DELETE /orders/{orderId}/account/attributes
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Successfully delete all attributes
 - ◇ 1.3.2 Delete attributes with wrong OwnerId
 - ◇ 1.3.3 Unassociated user without cookie
 - ◆ 1.4 See also

29.2 Request

DELETE /orders/{orderId}/account/attributes

29.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

29.2.2 URI Parameters

None.

29.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

29.3 Response

29.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The account attributes were successfully deleted.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to delete attributes for this account.
403 Forbidden	Unauthorized: Attributes failed permission check	Access order account attributes as an authenticated user, not associated with the order.
404 Not Found	Resource not found	The specified order does not exist.

29.4 Example

29.4.1 Successfully delete all attributes

Request

```
DELETE /orders/{orderId}/account/attributes
Cookie: OwnerId={owner_id}; Path=
```

Response

```
204 No Content
```

29.4.2 Delete attributes with wrong OwnerId

Request

```
DELETE /orders/{orderId}/account/attributes
Cookie: OwnerId={wrong_order_id}; Path=
```

Response

```
401 Unauthorized
Content-Type: text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

29.4.3 Unassociated user without cookie

Request

```
DELETE /orders/{orderId}/account/attributes
Authorization: {unassociated_user_authorization}
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8
```

```
{
  "forbidden" : {
    "guid" : "66f63a81-52e9-4ec6-a50a-42acebc021e7",
    "message" : "Unauthorized: Attributes failed permission check",
    "code" : 403,
    "details" : ""
  }
}
```

29.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

30 Orders-orderId-account-attributes:GET

31 GET /orders/{orderId}/account/attributes

Get the list of attributes for an account in a given order.

31.1 Contents

- 1 GET /orders/{orderId}/account/attributes
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Get order information
 - ◇ 1.3.2 Get attributes with wrong OwnerId
 - ◇ 1.3.3 Unassociated user without cookie
 - ◇ 1.3.4 Get attributes of unset account
 - ◆ 1.4 See also

31.2 Request

GET /orders/{orderId}/account/attributes

31.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

31.2.2 URI Parameters

None.

31.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

31.3 Response

31.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The operation was successfully completed.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to get attributes for this account.
403 Forbidden	Unauthorized: Attributes failed permission check	Access order account attributes as an authenticated user, not associated with the order.
404 Not Found	Resource not found	The specified order does not exist.
	MetaData is missing. Make sure the account is set before querying attributes.	An account should be set before getting info for attributes.

31.3.2 Response Body

```
{
  "list": [
    {
      "name": "{attributeName}",
      "value": "{attributeValue}"
    }
  ]
}
```

```

    },
    ...
  ],
  "links": [
    { "href": "https://api.hostway.com/orders/{orderId}/account/attributes/", "rel": "self" }
  ]
}

```

31.3.2.1 Parameters

attributeName - *string*
The unique name of the attribute.

attributeValue - *string*
The value of the attribute.

links - *list*
[Hypermedia](#) for this resource.

31.4 Example

31.4.1 Get order information

Request

```

GET /orders/{orderId}/account/attributes
Cookie: OwnerId={ownerId}; Path=/

```

Response

```

200 OK
Content-Type: application/json; charset=UTF-8

```

```

{
  "list" : [ {
    "name" : "partner_account_id",
    "value" : "paccid_050914054136"
  } ],
  "links" : [ {
    "href" : "https://api.hostway.com/orders/{ownerId}/account/attributes/",
    "rel" : "self"
  } ]
}

```

31.4.2 Get attributes with wrong OwnerId

Request

```

GET /orders/{orderId}/account/attributes
Cookie: OwnerId={wrong_order_id}; Path=/

```

Response

```

401 Unauthorized
Content-Type: text/plain; charset=UTF-8

```

```

401 Unauthorized

```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

31.4.3 Unassociated user without cookie

Request

```

GET /orders/{orderId}/account/attributes
Authorization: {unassociated_user_authorization}

```

Response

```

403 Forbidden
Content-Type: application/json; charset=UTF-8

```

```

{
  "forbidden" : {
    "guid" : "66f63a81-52e9-4ec6-a50a-42acebc021e7",
    "message" : "Unauthorized: Attributes failed permission check",
    "code" : 403,
    "details" : ""
  }
}

```

31.4.4 Get attributes of unset account

Request

```

GET /orders/{orderId}/account/attributes
Cookie: OwnerId={owner_id}; Path=/

```

Response

```

404 Not Found
Content-Type : application/json; charset=UTF-8

```

```
{
  "itemNotFound": {
    "guid": "0068a205-0c4c-4105-9a14-dafe313bfc3a",
    "message": "MetaData is missing. Make sure the account is set before querying attributes.",
    "code": 404,
    "details": ""
  }
}
```

31.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

32 Orders-orderId-account-attributes:PUT

33 PUT /orders/{orderId}/account/attributes

Set the list of attributes for an account in a given order.

33.1 Contents

- 1 PUT /orders/{orderId}/account/attributes
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - ◆ 1.3 Example
 - ◇ 1.3.1 Set valid account attribute
 - ◇ 1.3.2 Set bad account attributes
 - ◇ 1.3.3 Set an attribute with too long value
 - ◇ 1.3.4 Set too long Partner Account ID value
 - ◇ 1.3.5 Set attributes with wrong OwnerId
 - ◇ 1.3.6 Unassociated user without cookie
 - ◇ 1.3.7 Set existing Partner Account ID
 - ◇ 1.3.8 Account has not been set
 - ◆ 1.4 See also

33.2 Request

PUT /orders/{orderId}/account/attributes

33.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

33.2.2 URI Parameters

None.

33.2.3 Request Headers

Content-Type

Required. Set this header to `application/json; charset=UTF-8`

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

33.2.4 Request Body

```
[
  {
    "name": "{attributeName}",
    "value": "{attributeValue}"
  },
  {
    "name": "{attributeName}",
```

```

    "value": "{attributeValue}"
  }
]

```

33.2.4.1 Parameters

attributeName - *string*

The unique name of the attribute. The list of supported attribute names is specific to the franchise. The common attributes include:
partner_account_id - unique account identifier in the partner system. The value should be unique within the franchise.

attributeValue - *string*

The value of the attribute max 200 characters. Should adhere to the rules for the attribute specified by the name.
 The value of the "partner_account_id" attribute is max 100 characters. Allowed characters are alphanumeric characters, underscore, dot and dash.

33.3 Response

33.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The account attributes were successfully set.
	POST data error	The request body format is invalid and all attributes must have less than 200 characters.
400 Bad Request	Not allowed or not existing MetaData Type Partner Account ID cannot exceed 100 characters	Setting some bad account attributes
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to add attributes for this account.
403 Forbidden	Unauthorized: Attributes failed permission check	Access order account attributes as an authenticated user, not associated with the order.
404 Not Found	Resource not found	The specified order does not exist.
409 Conflict	Account ID already in use	Attribute value for Partner Account ID, supposed to be unique, is already in use.
500 Internal Server Error	-	Account has not been set for this order.
502 Bad Gateway	-	A critical (unknown) error occurred.

33.3.2 Response Body

None.

33.4 Example

33.4.1 Set valid account attribute

Request

```

PUT /orders/{orderId}/account/attributes
Cookie: OwnerId={owner_id}; Path=/

```

```

[
  {
    "name": "partner_account_id",
    "value": "paccid_050914054136"
  }
]

```

Response

204 No Content

33.4.2 Set bad account attributes

Request

```

PUT /orders/{orderId}/account/attributes
Cookie: OwnerId={owner_id}; Path=/

```

```

[
  {
    "name": "attribute_a",
    "value": "value_a"
  },
  {
    "name": "attribute_b",
    "value": "value_b"
  }
]

```

Response

```

400 Bad Request
Content-Type : application/json; charset=UTF-8

```

```

{
  "computeFault": {
    "guid": "5268a14c-ced6-4f32-87ee-aa6709414396",
    "message": "Not allowed or not existing MetaData Type",
    "code": 400,
    "details": ""
  }
}

```



```
403 Forbidden
Content-Type: application/json; charset=UTF-8
```

```
{
  "forbidden" : {
    "guid" : "66f63a81-52e9-4ec6-a50a-42acebc021e7",
    "message" : "Unauthorized: Attributes failed permission check",
    "code" : 403,
    "details" : ""
  }
}
```

33.4.7 Set existing Partner Account ID

Request

```
PUT /orders/{orderId}/account/attributes
Cookie: OwnerId={owner_id}; Path=/
```

```
[
  {
    "name": "partner_account_id",
    "value": "ext_140213131348"
  }
]
```

Response

```
409 Conflict
Content-Type : application/json; charset=UTF-8
```

```
{
  "conflict": {
    "guid": "a8d38abe-1f68-49c5-bafe-29894671ed06",
    "message": "Account ID already in use",
    "code": 409,
    "details": ""
  }
}
```

33.4.8 Account has not been set

Request

```
PUT /orders/{orderId}/account/attributes
Cookie: OwnerId={owner_id}; Path=/
```

```
[
  {
    "name": "partner_account_id",
    "value": "paccid_050914054136"
  }
]
```

Response

```
500 Internal Server Error
Content-Type : application/json; charset=UTF-8
```

```
{
  "computeFault" : {
    "guid" : "a4eae57d-b172-45ff-ae6e-474a8b6061a0",
    "timestamp" : "2014-09-10 10:12:41",
    "code" : 500,
    "errorRefId" : "a4eae57d-b172-45ff-ae6e-474a8b6061a0"
  }
}
```

33.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

34 Orders-orderId-account-contactInfo-contactType:DELETE

35 DELETE /orders/{orderId}/account/contactInfo/{contactType}

Deletes the account contact details for a given contact type in an order.

35.1 Contents

- 1 DELETE /orders/{orderId}/account/contactInfo/{contactType}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◆ 1.3 Example
 - ◇ 1.3.1 Successfully delete contact type of an order
 - ◇ 1.3.2 Get contact info with wrong OwnerId
 - ◇ 1.3.3 Unassociated user without cookie
 - ◇ 1.3.4 Delete contact info of unset contact type
 - ◇ 1.3.5 Account has not been set
 - ◇ 1.3.6 Order does not exist
 - ◆ 1.4 See also

35.2 Request

DELETE /orders/{orderId}/account/contactInfo/{contactType}

35.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

contact_type - *string*

Contact Type (regular, billing, administrator or technical)

35.2.2 URI Parameters

None.

35.2.3 Request Headers

Content-Type

Required. Set this header to `application/json; charset=UTF-8`

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

35.3 Response

Status Code	Error Message	Description
204 No Content	-	The contact info was successfully deleted. The specified order does not exist.
404 Not Found	Resource not found	An account has not been added to the given order. The specified contact type is not set
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to delete contact info for this order.
403 Forbidden	Unauthorized: ContactInfo failed permission check	Delete account contact info as an authenticated user, not associated with the order.
502 Bad Gateway	-	Unexpected backend response. Cannot delete contact info.

35.4 Example

35.4.1 Successfully delete contact type of an order

Request

```
DELETE /orders/{orderId}/account/contactInfo/billing
Cookie: OwnerId={owner_id}; Path=/
```

Response

204 No Content

35.4.2 Get contact info with wrong OwnerId

Request

```
DELETE /orders/{orderId}/account/contactInfo/billing
Cookie: OwnerId={wrong_order_id}; Path=/
```

Response

```
401 Unauthorized
Content-Type: text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

35.4.3 Unassociated user without cookie

Request

```
DELETE /orders/{orderId}/account/contactInfo/billing
Authorization: {unassociated_user_authorization}
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8
```

```
{
  "forbidden" : {
    "guid" : "53fd5c41-a4b1-48d1-8196-43fc5ba32cf2",
    "message" : "Unauthorized: ContactInfo failed permission check",
    "code" : 403,
    "details" : ""
  }
}
```

35.4.4 Delete contact info of unset contact type

Request

```
DELETE /orders/{orderId}/account/contactInfo/billing
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "3f8d5d31-3986-4075-8a13-5fe3e20a2be4",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

35.4.5 Account has not been set

Request

```
DELETE /orders/{orderId}/account/contactInfo/billing
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "3f8d5d31-3986-4075-8a13-5fe3e20a2be4",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

35.4.6 Order does not exist

Request

```
DELETE /orders/{non-existent_orderId}/account/contactInfo/billing
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "80d5e49e-0795-4209-a929-1ee051e0f252",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

35.5 See also

- [Orders API](#)

- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

36 Orders-orderId-account-contactInfo-contactType:GET

37 GET /orders/{orderId}/account/contactInfo/{contactType}

Get specific contact type details for an account in a given order.

37.1 Contents

- 1 GET /orders/{orderId}/account/contactInfo/{contactType}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Get contact info with wrong OwnerId
 - ◇ 1.3.2 Unassociated user without cookie
 - ◇ 1.3.3 Get contact info of unset contact type
 - ◇ 1.3.4 Account has not been set
 - ◇ 1.3.5 Order does not exist
 - ◆ 1.4 See also

37.2 Request

GET /orders/{orderId}/account/contactInfo/{contactType}

37.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

contact_type - *string*

Contact Type (regular, billing, administrative or technical)

37.2.2 URI Parameters

None.

37.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

37.3 Response

37.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The operation was successfully completed.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to get contact type info for this order.
403 Forbidden	Unauthorized: ContactInfo failed permission check	Access account contact info as an authenticated user, not associated with the order.
404 Not Found	Resource not found	The specified order does not exist. An account was not set for the order.
502 Bad Gateway	-	The order does not have contact info of this type. Unexpected backend response. Cannot retrieve contact info.

37.3.2 Response Body

```
{
  "contactInfo": {
    "name": {
      "salutation": "{salutation}",
      "firstName": "{firstname}",
      "middleName": "{middleName}",
      "lastName": "{lastName}",
      "company": "{company}"
    },
    "address": {
      "street1": "{street1}",
      "street2": "{street2}",
      "city": "{city}",
      "stateOrProvince": "{stateOrProvince}",
      "postalCode": "{postalCode}",
      "countryCode": "{countryCode}"
    },
    "contactMedia": {
```

```

    "phone1": "{phone1}",
    "phone2": "{phone2}",
    "fax": "{fax}",
    "email1": "{email1}",
    "email1Format": "{html|text}",
    "email2": "{email2}",
    "email2Format": "{html|text}"
  }
},
"links": [
  {"href": "https://api.hostway.com/orders/{orderId}/account/contactInfo/{contactType}/", "rel": "self"}}
]
}

```

37.3.2.1 Parameters

salutation
Contact's salutation

firstName
Contact's first name.

middleName
Contact's middle name (optional).

lastName
Contact's last name.

company
Contact's company name (optional).

street1
Contact's street address.

street2
Contact's street address second line (optional).

city
Contact's city.

stateOrProvince
Contact's state or province.

postalCode
Contact's postal or ZIP code.

country
Contact's country code per ISO 3166-1 alpha-2[1].

phone1
Contact's primary phone number in ITU-T E.164[2] format.

phone2
Contact's secondary phone number in ITU-T E.164[3] format (optional).

fax
Contact's fax number in ITU-T E.164[4] format (optional).

email1
Contact's primary e-mail address.

email1Format
Primary e-mail delivery format: *html* or *text*.

email2
Contact's secondary e-mail address (optional).

email2Format
Secondary e-mail delivery format: *html* or *text* (optional).

links - list
[Hypermedia](#) for this resource.

37.4 Example

37.4.1 Get contact info with wrong OwnerId

Request

```

GET /orders/{orderId}/account/contactInfo/billing
Cookie: OwnerId={wrong_order_id}; Path=/

```

Response

```

401 Unauthorized
Content-Type: text/plain; charset=UTF-8

```

```

401 Unauthorized

```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

37.4.2 Unassociated user without cookie

Request

```
GET /orders/{orderId}/account/contactInfo/billing
Authorization: {unassociated_user_authorization}
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8

{
  "forbidden": {
    "guid": "53fd5c41-a4b1-48d1-8196-43fc5ba32cf2",
    "message": "Unauthorized: ContactInfo failed permission check",
    "code": 403,
    "details": ""
  }
}
```

37.4.3 Get contact info of unset contact type

Request

```
GET /orders/{orderId}/account/contactInfo/billing
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8

{
  "itemNotFound": {
    "guid": "3f8d5d31-3986-4075-8a13-5fe3e20a2be4",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

37.4.4 Account has not been set

Request

```
GET /orders/{orderId}/account/contactInfo/billing
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8

{
  "itemNotFound": {
    "guid": "3f8d5d31-3986-4075-8a13-5fe3e20a2be4",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

37.4.5 Order does not exist

Request

```
GET /orders/{non-existent_orderId}/account/contactInfo/billing
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8

{
  "itemNotFound": {
    "guid": "80d5e49e-0795-4209-a929-1ee051e0f252",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

37.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

38 Orders-orderId-account-contactInfo-contactType:PUT

39 PUT /orders/{orderId}/account/contactInfo/{contactType}

Set the account contact details for a given contact type in an order.

39.1 Contents

- 1 PUT
/orders/{orderId}/account/contactInfo/{contactType}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - ◆ 1.3 Example
 - ◇ 1.3.1 Set bad contact info values
 - ◇ 1.3.2 Set bad contact info values 2
 - ◇ 1.3.3 Set contact info with wrong OwnerId
 - ◇ 1.3.4 Unassociated user without cookie
 - ◇ 1.3.5 Set contact info in non-existent order
 - ◇ 1.3.6 Set contact info on an unset account
 - ◆ 1.4 See also

39.2 Request

PUT /orders/{orderId}/account/contactInfo/{contactType}

39.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

contact_type - *string*

Contact Type (regular, billing, administrator or technical)

39.2.2 URI Parameters

None.

39.2.3 Request Headers

Content-Type

Required. Set this header to `application/json; charset=UTF-8`

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

39.2.4 Request Body

```
{
  "name": {
    "salutation": "{salutation}",
    "firstName": "{firstname}",
    "middleName": "{middleName}",
    "lastName": "{lastName}",
    "company": "{company}"
  },
  "address": {
    "street1": "{street1}",
    "street2": "{street2}",
    "city": "{city}",
    "stateOrProvince": "{stateOrProvince}",
    "postalCode": "{postalCode}",
    "countryCode": "{countryCode}"
  },
  "contactMedia": {
    "phone1": "{phone1}",
    "phone2": "{phone2}",
    "fax": "{fax}",
    "email1": "{email1}",
    "email1Format": "{html|text}",
    "email2": "{email2}",
    "email2Format": "{html|text}"
  }
}
```

39.2.4.1 Parameters

- salutation - *string*
(*OPTIONAL*)
May contain one of the salutations provided by [GET /salutations](#)
- firstName - *string*
Required string max 50 characters. May contain only alphanumerics, white space and the following characters: -.'
- middleName - *string*
Optional string max 50 characters. May contain only alphanumerics, white space and the following characters: -.'
- lastName - *string*
Required string max 50 characters. May contain only alphanumerics, white space and the following characters: -.'
- company - *string*
Optional string max 200 characters. May contain only alphanumerics, white space and the following characters: -.'
- street1 - *string*
Required string max 100 characters. May contain only alphanumerics, white space and the following characters: -.'
- street2 - *string*
Optional string max 100 characters. May contain only alphanumerics, white space and the following characters: -.'
- city - *string*
Required string max 50 characters. May contain only alphanumerics, white space and the following characters: -.'
- stateOrProvince - *string*
Required string max 20 characters. May contain only alphanumerics, white space and the following characters: -.' If the selected country is United States, then this must be a valid two-letter US state abbreviation. If the selected country is Canada, then this value must be a valid two-letter Canadian province abbreviation.
- postalCode - *string*
Required string max 30 characters. May contain only alphanumerics, white space and the following characters: -.' If the selected country is United States, then the postalCode must be in format "#####" or "#####-####" (where "#" represents any numeric digit). If the selected country is Canada, then the postalCode must be in format "A#A#A#" or "A#A-#A#" or "A#A #A#" (where "A" represents any latin alphabet character and "#" represents any numeric digit).
- country - *string*
Required string ISO 3166-1 alpha-2[1] country code.
- phone1 - *string*
Required string ITU-T E.164[2] formatted phone number max 20 digits.
- phone2 - *string*
Optional string ITU-T E.164[3] formatted phone number max 20 digits.
- fax - *string*
Optional string ITU-T E.164[4] formatted phone number max 20 digits.
- email1 - *string*
Required string valid e-mail address max 100 characters.
- email1Format - *string*
Optional string ? one of *html* or *text*. The default value of *html* will be applied if omitted from the request.
- email2 - *string*
Optional string valid e-mail address max 100 characters.
- email2Format - *string*
Optional string ? one of *html* or *text*. The default value of *html* will be used if *email2* is supplied and *email2Format* is omitted.

39.3 Response

39.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The operation was successfully completed.
400 Bad Request	POST data error Field cannot have any alphabet letters; only numbers and symbols allowed.	The value of one or more contact info fields is not valid or missing.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to set contact info for this order.
403 Forbidden	Unauthorized: ContactInfo failed permission check	Set account contact info as an authenticated user, not associated with the order.
404 Not Found	Resource not found	The specified order does not exist.
502 Bad Gateway	-	Account has not been set. Cannot add contact info.

39.3.2 Response Body

None.

39.4 Example

39.4.1 Set bad contact info values

Request

```
PUT /orders/{orderId}/account/contactInfo/billing
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "name": {
    "firstName": "&",
    "middleName": "&",
    "lastName": "&",
    "company": "Hostway",
    "salutation": "invalid"
  },
  "address": {
    "street1": "(",
    "street2": "(",
    "city": "&",
    "stateOrProvince": "&",
    "postalCode": "asfagag",
    "countryCode": "^"
  },
  "contactMedia": {
    "phone1": "a",
    "phone2": "a",
    "fax": "a",
    "email1": "em",
    "email1Format": "test"
  }
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "badRequest": {
    "guid": "1c9b2dc5-2feb-4599-9753-aae2c39853dd",
    "message": "POST data error",
    "code": 400,
    "details": {
      "name.lastName": "Only alphanumerics, spaces, and the following characters are allowed in lastName: -'.",
      "address.city": "Only alphanumerics, spaces, and the following characters are allowed in city: -'.",
      "address.street2": "Only alphanumerics, spaces, and the following characters are allowed in street2: -'.",
      "name.firstName": "Only alphanumerics, spaces, and the following characters are allowed in firstName: -'.",
      "address.street1": "Only alphanumerics, spaces, and the following characters are allowed in street1: -'.",
      "name.salutation": "\"invalid\" is not one of Mr., Ms., Mrs., Dr.",
      "address.stateOrProvince": "Only alphanumerics, spaces, and the following characters are allowed in stateOrProvince: -'.",
      "name.middleName": "Only alphanumerics, spaces, and the following characters are allowed in middleName: -'.",
      "contactMedia.email1": "Invalid email address in email1: em",
      "address.countryCode": "Invalid Country Code",
      "contactMedia.email1Format": "\"test\" is not one of html, text"
    }
  }
}
```

39.4.2 Set bad contact info values 2

Request

```
PUT /orders/{orderId}/account/contactInfo/billing
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "name": {
    "firstName": "Billing",
    "middleName": "X",
    "lastName": "Customer",
    "company": "Hostway"
  },
  "address": {
    "street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  },
  "contactMedia": {
    "phone1": "test",
    "phone2": "test",
    "fax": "test",
    "email1": "petre.popescu@hostway.com",
    "email1Format": "html",
    "email2": "gah@wev.com",
    "email2Format": "text"
  }
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "computeFault": {
    "guid": "3fb07aa5-157e-474d-b7cc-ecce0cf87f36e",
    "message": "Field cannot have any alphabet letters; only numbers and symbols allowed.",
    "code": 400,
  }
}
```

```
    "details": ""
  }
}
```

39.4.3 Set contact info with wrong OwnerId

Request

```
PUT /orders/{orderId}/account/contactInfo/billing
Cookie: OwnerId={wrong_order_id}; Path=
```

```
{
  "name": {
    "firstName": "Billing",
    "middleName": "X",
    "lastName": "Customer",
    "company": "Hostway",
    "salutation": "Mr."
  },
  "address": {
    "street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  },
  "contactMedia": {
    "phone1": "312-555-2222",
    "phone2": "312-555-3333",
    "fax": "312-555-1111",
    "email1": "petre.popescu@hostway.com",
    "email1Format": "html",
    "email2": "gah@wev.com",
    "email2Format": "text"
  }
}
```

Response

```
401 Unauthorized
Content-Type: text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

39.4.4 Unassociated user without cookie

Request

```
PUT /orders/{orderId}/account/contactInfo/billing
Authorization: {unassociated_user_authorization}
```

```
{
  "name": {
    "firstName": "Billing",
    "middleName": "X",
    "lastName": "Customer",
    "company": "Hostway",
    "salutation": "Mr."
  },
  "address": {
    "street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  },
  "contactMedia": {
    "phone1": "312-555-2222",
    "phone2": "312-555-3333",
    "fax": "312-555-1111",
    "email1": "petre.popescu@hostway.com",
    "email1Format": "html",
    "email2": "gah@wev.com",
    "email2Format": "text"
  }
}
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8
```

```
{
  "forbidden" : {
    "guid" : "53fd5c41-a4b1-48d1-8196-43fc5ba32cf2",
    "message" : "Unauthorized: ContactInfo failed permission check",
    "code" : 403,
    "details" : ""
  }
}
```

39.4.5 Set contact info in non-existent order

Request

```
PUT /orders/{non-existent_orderId}/account/contactInfo/billing
Cookie: OwnerId={owner_id}; Path=
```

```

{
  "name": {
    "firstName": "Billing",
    "middleName": "X",
    "lastName": "Customer",
    "company": "Hostway",
    "salutation": "Mr."
  },
  "address": {
    "street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  },
  "contactMedia": {
    "phone1": "312-555-2222",
    "phone2": "312-555-3333",
    "fax": "312-555-1111",
    "email1": "petre.popescu@hostway.com",
    "email1Format": "html",
    "email2": "gah@wev.com",
    "email2Format": "text"
  }
}

```

Response

404 Not Found
Content-Type : application/json; charset=UTF-8

```

{
  "itemNotFound" : {
    "guid" : "ed5f236d-6983-4576-b1d3-9cd69c4255a0",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}

```

39.4.6 Set contact info on an unset account

Request

PUT /orders/{orderId}/account/contactInfo/billing
Cookie: OwnerId={owner_id}; Path=/

```

{
  "name": {
    "firstName": "Billing",
    "middleName": "X",
    "lastName": "Customer",
    "company": "Hostway",
    "salutation": "Mr."
  },
  "address": {
    "street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  },
  "contactMedia": {
    "phone1": "312-555-2222",
    "phone2": "312-555-3333",
    "fax": "312-555-1111",
    "email1": "petre.popescu@hostway.com",
    "email1Format": "html",
    "email2": "gah@wev.com",
    "email2Format": "text"
  }
}

```

Response

502 Bad Gateway
Content-Type : application/json; charset=UTF-8

```

{
  "computeFault" : {
    "guid" : "5db3e3c9-37ac-4a1d-8085-c931577e1c16",
    "timestamp" : "2014-09-10 12:10:12",
    "code" : 502,
    "errorRefId" : "5db3e3c9-37ac-4a1d-8085-c931577e1c16"
  }
}

```

39.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

40 Orders-orderId-account-contactInfo:GET

41 GET /orders/{orderId}/account/contactInfo

Get all contact infos for an account in a given order.

41.1 Contents

- 1 GET /orders/{orderId}/account/contactInfo
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Get contact info with wrong OwnerId
 - ◇ 1.3.2 Unassociated user without cookie
 - ◇ 1.3.3 Non-existent contact info for the order
 - ◇ 1.3.4 Order does not exist
 - ◇ 1.3.5 Unset account for the order
 - ◆ 1.4 See also

41.2 Request

GET /orders/{orderId}/account/contactInfo

41.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

41.2.2 URI Parameters

None.

41.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

41.3 Response

41.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The operation was successfully completed.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to get contact info for this order.
403 Forbidden	Unauthorized: ContactInfo failed permission check	Access account contact info as an authenticated user, not associated with the order.
404 Not Found	Resource not found	The order does not have any contact info.
	AccountNumber is missing. Make sure the account is set before querying.	The specified order does not exist.
502 Bad Gateway	-	An account was not set for the order.
		Unexpected backend response. Cannot retrieve contact infos.

41.3.2 Response Body

```
{
  "contactInfo": {
    "regular": {...},
    "billing": {...},
    "administrator": {...},
    "technical": {...}
  },
  "links": [
    { "href": "https://api.hostway.com/orders/{orderId}/account/contactInfo/", "rel": "self" }
  ]
}
```

41.3.2.1 Parameters

contactInfo - *dictionary*
Details for all contact types specified for the account.

links - *list*
[Hypermedia](#) for this resource.

41.4 Example

41.4.1 Get contact info with wrong OwnerId

Request

```
GET /orders/{orderId}/account/contactInfo
Cookie: OwnerId={wrong_order_id}; Path=
```

Response

```
401 Unauthorized
Content-Type: text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

41.4.2 Unassociated user without cookie

Request

```
GET /orders/{orderId}/account/contactInfo
Authorization: {unassociated_user_authorization}
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8
```

```
{
  "forbidden" : {
    "guid" : "53fd5c41-a4b1-48d1-8196-43fc5ba32cf2",
    "message" : "Unauthorized: ContactInfo failed permission check",
    "code" : 403,
    "details" : ""
  }
}
```

41.4.3 Non-existent contact info for the order

Request

```
GET /orders/{orderId}/account/contactInfo
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "80d5e49e-0795-4209-a929-1ee051e0f252",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

41.4.4 Order does not exist

Request

```
GET /orders/{non-existent_orderId}/account/contactInfo
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "80d5e49e-0795-4209-a929-1ee051e0f252",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

41.4.5 Unset account for the order

Request

```
GET /orders/{orderId}/account/contactInfo
Cookie: OwnerId={owner_id}; Path=/
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "80d5e49e-0795-4209-a929-1ee051e0f252",
    "message": "AccountNumber is missing. Make sure the account is set before querying.",
    "code": 404,
    "details": ""
  }
}
```

41.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

42 Orders-orderId-account-paymentInfo:GET

43 GET /orders/{orderId}/account/paymentInfo

Get the payment method information for a given order. Only a minimum amount of information will be returned to recognize the payment method.

43.1 Contents

- 1 GET /orders/{orderId}/account/paymentInfo
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 No authorization
 - ◇ 1.3.3 Unauthorized user
 - ◇ 1.3.4 Order does not exist
 - ◇ 1.3.5 Account has not been set
 - ◇ 1.3.6 Payment info has not been set
 - ◆ 1.4 See also

43.2 Request

GET /orders/{orderId}/account/paymentInfo

43.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

43.2.2 URI Parameters

None.

43.2.3 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

43.3 Response

43.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The operation was successfully completed.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to get payment info for this order.
403 Forbidden	Unauthorized: PaymentInfo failed permission check	An unauthorized user cannot set or get payment info.
404 Not Found	Resource not found	The specified order does not exist.
	-	An account was not set for the order.
	-	No payment info for the order was found.
502 Bad Gateway	-	Unexpected backend error

43.3.2 Response Body

The following is the response body for credit cards.

```
{
  "paymentInfoType": "credit_card",
  "accountNumber": "*****1111",
  "creditCardType": "visa",
  "expirationDate": "2012-08",
  "links": [
```

```
    {"href": "https://api.hostway.com/orders/{order_id}/account/paymentInfo/", "rel": "self"}
  ]
}
```

The following is the response for Verizon payment information.

```
{
  "paymentInfoType": "vz_payment",
  "paymentReference": "123432429034",
  "sessionId": "23432434244",
  "links": [
    {"href": "https://api.hostway.com/orders/{order_id}/account/paymentInfo/", "rel": "self"}
  ]
}
```

The following is the response for offline payment information.

```
{
  "paymentInfoType": "offline",
  "links": [
    {"href": "https://api.hostway.com/orders/{order_id}/account/paymentInfo/", "rel": "self"}
  ]
}
```

43.3.2.1 Parameters

paymentInfoType - *string*

The type of payment method. Possible value(s) are:

credit_card
vz_payment
offline

accountNumber - *string*

The credit card number. Only the last four digits will be revealed. Only applies for credit cards.

creditCardType - *string*

The type of credit card such as visa, mastercard, discover, etc. Only applies for credit cards.

expirationDate - *string*

The credit card expiration date in YYYY-MM format. Only applies for credit cards.

paymentReference - *string*

Verizon payment reference id. Only applies for Verizon payment type.

sessionId - *string*

Verizon payment session id. Only applies for Verizon payment type.

links - *list*

[Hypermedia](#) for this resource. Links to self.

43.4 Example

43.4.1 Success

Request

```
GET /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=/
```

Response

```
200 OK
Content-Type : application/json; charset=UTF-8
```

```
{
  "links": [
    {
      "href": "https://api.hostway.com/orders/00000148546ce1ac1d12585e007f000000010001/account/paymentinfo/",
      "rel": "self"
    }
  ],
  "creditCardType": "visa",
  "expirationDate": "2015-06",
  "paymentInfoType": "credit_card",
  "accountNumber": "*****1111"
}
```

43.4.2 No authorization

Request

```
GET /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={wrong_owner_id}; Path=/
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

43.4.3 Unauthorized user

Request

```
GET /orders/{orderId}/account/paymentInfo
Authorization: Basic {wrong_user_authorization}
```

Response

```
403 Forbidden
Content-Type : application/json; charset=UTF-8
```

```
{
  "forbidden": {
    "guid": "5a199170-843a-44fc-bbb0-87155eb15fc6",
    "message": "Unauthorized: PaymentInfo failed permission check",
    "code": 403,
    "details": ""
  }
}
```

43.4.4 Order does not exist

Request

```
GET /orders/{non-existent_orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "cca219c5-3e2b-481d-8e7f-7083064872ff",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

43.4.5 Account has not been set

Request

```
GET /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "9ae11205-0650-471a-9ed7-7642107011dc",
    "message": "",
    "code": 404,
    "details": ""
  }
}
```

43.4.6 Payment info has not been set

Request

```
GET /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "9ae11205-0650-471a-9ed7-7642107011dc",
    "message": "",
    "code": 404,
    "details": ""
  }
}
```

43.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

44 Orders-orderId-account-paymentInfo:PUT

45 PUT /orders/{orderId}/account/paymentInfo

Set the account's payment information for a given order.

45.1 Contents

- 1 PUT /orders/{orderId}/account/paymentInfo
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 Offline payment for supported franchises with agent access
 - ◇ 1.3.3 Invalid payment info type
 - ◇ 1.3.4 Missing attribute for paymentInfoType credit_card
 - ◇ 1.3.5 Credit card number length exceeds 16 characters
 - ◇ 1.3.6 Credit card number cannot contain hyphens
 - ◇ 1.3.7 Credit card number with wrong length
 - ◇ 1.3.8 Invalid verification number
 - ◇ 1.3.9 Invalid date format
 - ◇ 1.3.10 No authorization
 - ◇ 1.3.11 Unauthorized user
 - ◇ 1.3.12 Unset account for the order
 - ◆ 1.4 See also

45.2 Request

PUT /orders/{orderId}/account/paymentInfo

45.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

45.2.2 URI Parameters

None.

45.2.3 Request Headers

Content-Type

Required. Set this header to `application/json; charset=UTF-8`

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

45.2.4 Request Body

The request format depends on the type of payment method that is added. The following format is required for credit cards.

```
{
  "paymentInfoType": "credit_card",
  "accountNumber": "4111111111111111",
  "cardHolderName": "John Smith",
  "creditCardType": "visa",
  "expirationDate": "2012-08",
  "verificationNumber": "123",
  "billingAddress": {
    "city": "Chicago",
    "countryCode": "US",
    "postalCode": "60606",
    "stateOrProvince": "IL",
    "street1": "100 N Riverside",
    "street2": ""
  }
}
```

45.2.4.1 Parameters

paymentInfoType - *string*

The type of payment method that will be used. Allowed value(s) are:

credit_card
offline (Available only for agents on supported franchises)

accountNumber - *string*

Credit card number. Must consist of numbers only. Required if paymentInfoType is set to **credit_card**.

cardHolderName - *string*

The owner name as it appears on the card. Required if paymentInfoType is set to **credit_card**.

creditCardType - *string*

The type of credit card - **visa, mastercard, discover, amex**. Required if paymentInfoType is set to **credit_card**.

expirationDate - *string*

The credit card expiration date in YYYY-MM format. Should exceed the current date. Required if paymentInfoType is set to **credit_card**.

verificationNumber - *string*

The credit card security code. This is only used to validate the credit card and will not be stored. Required if paymentInfoType is set to **credit_card**.

billingAddress - *dictionary*

Contains the credit card's billing address. Required if paymentInfoType is set to **credit_card**.

45.3 Response

45.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The operation was successfully completed.
	POST data error	A required attribute for the payment info type is missing.
	declined	paymentInfoType can only be credit_card.
	Invalid verificationNumber	Invalid verificationNumber
400 Bad Request	Invalid date format	Wrong expirationDate format
	invalid number or type	creditCardType can be only one of visa, mastercard, amex, discover.
		Credit card number with wrong length
		Credit card number does not comply with visa rules
		expirationDate set in past
	Incorrect padding	Hyphens or spaces in credit card number are not allowed
	Input strings must be a multiple of 16 in length	Credit card number length should be 16.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to add payment info.
403 Forbidden	Unauthorized: PaymentInfo failed permission check	An unauthorized user cannot set or get payment info.
404 Not Found	Resource not found	The specified order does not exist.
502 Bad Gateway	-	An account has not been added to the given order. Unexpected backend error.

45.3.2 Response Body

None.

45.4 Example

45.4.1 Success

Request

```
PUT /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "paymentInfoType": "credit_card",
  "creditCardType": "visa",
  "accountNumber": "4111111111111111",
  "cardHolderName":
  "John Jello",
  "expirationDate":
  "2015-06",
  "verificationNumber": "505",
  "billingAddress": {
    "street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  }
}
```

Response

```
204 No Content
```

45.4.2 Offline payment for supported franchises with agent access

Request

```
PUT /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=/
Authorization: Basic Auth_String
```

```
{
```

```
"paymentInfoType": "offline"
}
```

Response

204 No Content

45.4.3 Invalid payment info type

Request

PUT /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=

```
{
  "paymentInfoType": "direct_debit",
  "creditCardType": "visa",
  "accountNumber": "4111111111111111",
  "cardHolderName":
  "John Jello",
  "expirationDate":
  "2015-06",
  "verificationNumber": "505",
  "billingAddress": {
    "street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  }
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "badRequest": {
    "guid": "7ca8b2cc-88d5-4e5b-aab8-e4015a5b3ad8",
    "message": "POST data error",
    "code": 400,
    "details": {
      "paymentInfoType": "\"direct_debit\" is not one of credit_card, vz_payment"
    }
  }
}
```

45.4.4 Missing attribute for paymentInfoType credit_card

Request

PUT /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=

```
{
  "paymentInfoType": "credit_card",
  "accountNumber": "4111111111111111",
  "cardHolderName": "John Smith",
  "creditCardType": "visa",
  "expirationDate": "2012-08",
  "billingAddress": {
    "city": "Chicago",
    "countryCode": "US",
    "postalCode": "60606",
    "stateOrProvince": "IL",
    "street1": "100 N Riverside",
    "street2": ""
  }
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "badRequest": {
    "guid": "2829e9e2-04e1-4fe2-8e05-be738b4c467c",
    "message": "POST data error",
    "code": 400,
    "details": {
      "verificationNumber": "Required"
    }
  }
}
```

45.4.5 Credit card number length exceeds 16 characters

Request

PUT /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=

```
{
  "paymentInfoType": "credit_card",
  "accountNumber": "4111-1111-1111-1111",
  "cardHolderName": "John Smith",
  "creditCardType": "visa",
  "expirationDate": "2012-08",
  "verificationNumber": "505",
}
```

```
"billingAddress": {
  "city": "Chicago",
  "countryCode": "US",
  "postalCode": "60606",
  "stateOrProvince": "IL",
  "street1": "100 N Riverside",
  "street2": ""
}
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "badRequest": {
    "guid": "f5ed5738-086d-4751-8ad5-0dfead39ea3a",
    "message": "Input strings must be a multiple of 16 in length",
    "code": 400,
    "details": ""
  }
}
```

45.4.6 Credit card number cannot contain hyphens

Request

PUT /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=

```
{
  "paymentInfoType": "credit_card",
  "accountNumber": "4111-111-111-111",
  "cardHolderName": "John Smith",
  "creditCardType": "visa",
  "expirationDate": "2012-08",
  "verificationNumber": "505",
  "billingAddress": {
    "city": "Chicago",
    "countryCode": "US",
    "postalCode": "60606",
    "stateOrProvince": "IL",
    "street1": "100 N Riverside",
    "street2": ""
  }
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "computeFault": {
    "guid": "5a757a5a-870e-4d4f-acdf-aed95e066b3d",
    "message": "Incorrect padding",
    "code": 400,
    "details": ""
  }
}
```

45.4.7 Credit card number with wrong length

Request

PUT /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=

```
{
  "paymentInfoType": "credit_card",
  "accountNumber": "411111111111111122",
  "cardHolderName": "John Smith",
  "creditCardType": "visa",
  "expirationDate": "2012-08",
  "verificationNumber": "505",
  "billingAddress": {
    "city": "Chicago",
    "countryCode": "US",
    "postalCode": "60606",
    "stateOrProvince": "IL",
    "street1": "100 N Riverside",
    "street2": ""
  }
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "badRequest": {
    "guid": "ced1a8a8-21dc-4a10-9d36-5a201214a5ce",
    "message": "invalid number or type",
    "code": 400,
    "details": ""
  }
}
```

45.4.8 Invalid verification number

Request

```
PUT /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "paymentInfoType": "credit_card",
  "creditCardType": "visa",
  "accountNumber": "4111111111111111",
  "cardHolderName": "John Jello",
  "expirationDate": "2015-06",
  "verificationNumber": "0000",
  "billingAddress": {"street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  }
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "computeFault" : {
    "guid" : "031b8bc2-8da3-41c7-860c-ca3743cef135",
    "message" : "declined",
    "code" : 400,
    "details" : ""
  }
}
```

45.4.9 Invalid date format

Request

```
PUT /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "paymentInfoType": "credit_card",
  "accountNumber": "4111111111111111",
  "cardHolderName": "John Smith",
  "creditCardType": "visa",
  "expirationDate": "15-06",
  "verificationNumber": "505",
  "billingAddress": {
    "city": "Chicago",
    "countryCode": "US",
    "postalCode": "60606",
    "stateOrProvince": "IL",
    "street1": "100 N Riverside",
    "street2": ""
  }
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "badRequest": {
    "guid": "30e66cfe-649c-4b1c-bfaf-c65caf47b99d",
    "message": "Invalid date format",
    "code": 400,
    "details": ""
  }
}
```

45.4.10 No authorization

Request

```
PUT /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={wrong_owner_id}; Path=
```

```
{
  "paymentInfoType": "credit_card",
  "creditCardType": "visa",
  "accountNumber": "4111111111111111",
  "cardHolderName":
    "John Jello",
  "expirationDate":
    "2015-06",
  "verificationNumber": "505",
  "billingAddress": {
    "street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  }
}
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

45.4.11 Unauthorized user

Request

```
PUT /orders/{orderId}/account/paymentInfo
Authorization: Basic {wrong_user_authorization}
```

```
{
  "paymentInfoType": "credit_card",
  "creditCardType": "visa",
  "accountNumber": "4111111111111111",
  "cardHolderName":
  "John Jello",
  "expirationDate":
  "2015-06",
  "verificationNumber": "505",
  "billingAddress": {
    "street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  }
}
```

Response

```
403 Forbidden
Content-Type : application/json; charset=UTF-8
```

```
{
  "forbidden": {
    "guid": "5a199170-843a-44fc-bbb0-87155eb15fc6",
    "message": "Unauthorized: PaymentInfo failed permission check",
    "code": 403,
    "details": ""
  }
}
```

45.4.12 Unset account for the order

Request

```
PUT /orders/{orderId}/account/paymentInfo
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "paymentInfoType": "credit_card",
  "creditCardType": "visa",
  "accountNumber": "4111111111111111",
  "cardHolderName":
  "John Jello",
  "expirationDate":
  "2015-06",
  "verificationNumber": "505",
  "billingAddress": {
    "street1": "100 n riverside",
    "street2": "ste 800",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60606",
    "countryCode": "US"
  }
}
```

Response

```
502 Bad Gateway
Content-Type : application/json; charset=UTF-8
```

```
{
  "computeFault": {
    "guid": "a4f6b310-11d9-4263-8482-81bccf261df7",
    "timestamp": "2014-09-08 08:34:20",
    "code": 502,
    "errorRefId":
    "a4f6b310-11d9-4263-8482-81bccf261df7"
  }
}
```

45.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

46 Orders-orderId-account-products-productId-discounts:DELETE

47 DELETE /orders/{orderId}/account/products/{productId}/discounts

Remove all discounts that were previously added at the Product level. This will not remove discounts that were added at the order or service level.

47.1 Contents

- 1 DELETE
/orders/{orderId}/account/products/{productId}/discounts
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 Missing authorization
 - ◇ 1.3.3 Insufficient privileges
 - ◇ 1.3.4 Order does not exist
 - ◇ 1.3.5 Product could not be found
 - ◆ 1.4 See also

47.2 Request

DELETE /orders/{orderId}/account/products/{productId}/discounts

47.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*

UUID specifying the product for which actions are to be performed

47.2.2 URI Parameters

None.

47.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

47.3 Response

47.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The request was executed successfully.
401 Unauthorized	-	The request was not properly authorized.
403 Forbidden	Unauthorized: DiscountsView failed permission check	The client does not have sufficient privileges.
404 Not Found	Resource not found	Order does not exist. Product could not be found.
502 Bad Gateway	-	Unexpected backend response. Cannot delete custom product discounts.

47.4 Example

47.4.1 Success

Request

```
DELETE /orders/{orderId}/account/products/{productId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/
```

Response

```
204 No Content
```

47.4.2 Missing authorization

Request

```
DELETE /orders/{orderId}/account/products/{productId}/discounts
Cookie: OwnerId={owner_id}; Path=/
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
401 Unauthorized
```


This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

47.4.3 Insufficient privileges

Request

```
DELETE /orders/{orderId}/account/products/{productId}/discounts
Authorization: Basic ZWdjUlBtZWZiWnJVQHNPdGVjb250cm9sX2hvc3R3YXlfdXNfbWVpYjBpTdnVROGlWOUFTVTc0
Cookie: OwnerId={owner_id}; Path=
```

Response

```
403 Forbidden
Content-Type : application/json; charset=UTF-8

{
  "forbidden" : {
    "guid" : "a0c6a265-eea9-4187-b4eb-0b0b7b112810",
    "message" : "Unauthorized: DiscountsView failed permission check",
    "code" : 403,
    "details" : ""
  }
}
```

47.4.4 Order does not exist

Request

```
DELETE /orders/{non-existent_orderId}/account/products/{productId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8

{
  "itemNotFound" : {
    "guid" : "8a56063d-b099-4688-b020-d31035529d45",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}
```

47.4.5 Product could not be found

Request

```
DELETE /orders/{orderId}/account/products/{non-existent_productId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8

{
  "itemNotFound" : {
    "guid" : "42f454f4-a534-46ea-888e-c05af56c1130",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}
```

47.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

48 Orders-orderId-account-products-productId-discounts:GET

49 GET /orders/{orderId}/account/products/{productId}/discounts

Get discounts that were added at the Product level.

49.1 Contents

- 1 GET /orders/{orderId}/account/products/{productId}/discounts
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 Missing authorization
 - ◇ 1.3.3 Insufficient privileges
 - ◇ 1.3.4 Order does not exist
 - ◇ 1.3.5 Product could not be found
 - ◆ 1.4 See also

49.2 Request

GET /orders/{orderId}/account/products/{productId}/discounts

49.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product for which actions are to be performed

49.2.2 URI Parameters

None.

49.2.3 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

49.3 Response

49.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The request was executed successfully.
401 Unauthorized	-	The request was not properly authorized.
403 Forbidden	Unauthorized: DiscountsView failed permission check	The client does not have sufficient privileges.
404 Not Found	Resource not found	Order does not exist. Product could not be found.
502 Bad Gateway	-	Unexpected backend response. Cannot retrieve product discounts.

49.3.2 Response Body

```
{
  "discounts": [
    {
      "priceName": "recurring",
      "recurrences": -1,
      "type": "amount",
      "value": "1.00"
    }
  ],
  "links": [
    {
      "href": "https://api.hostway.com/orders/{orderId}/account/product/{productId}/discounts",
      "rel": "self"
    }
  ]
}
```

49.3.2.1 Parameters

discounts - *list*
A list of discounts. Each discount contains the following:

priceName - *string*
The type of price to which the discount will be applied.

recurrences - *integer*

The number of times that the discount will be applied. A value of "-1" means the discount will be applied every time the fee is charged.

type - *string*

Specifies the discount type which determines the method used to calculate the amount that is discounted. Possible values are:
amount: Discount will be treated as an exact amount.

value - *string*

The discount amount. The value will be interpreted based on the specified discount type.

links - *list*

[Hypermedia](#) for this resource. Links to self.

49.4 Example

49.4.1 Success

Request

```
GET /orders/{orderId}/account/products/{productId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/
```

Response

```
200 OK
Content-Type : application/json; charset=UTF-8
```

```
{
  "discounts" : [ {
    "priceName" : "recurring",
    "type" : "amount",
    "value" : "5.00",
    "recurrences" : -1
  } ],
  "links" : [ {
    "href" : "http://coreapi01.ote.chicago.hostway:8092/orders/00000148598765a3088232de007f00000010001/account/products/000001485988d9197a60",
    "rel" : "self"
  } ]
}
```

49.4.2 Missing authorization

Request

```
GET /orders/{orderId}/account/products/{productId}/discounts
Cookie: OwnerId={owner_id}; Path=/
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

49.4.3 Insufficient privileges

Request

```
GET /orders/{orderId}/account/products/{productId}/discounts
Authorization: Basic ZWdjUlBtZWZiWnJVQHNPdGVjb250cm9sX2hvc3R3YXlfdXNfbWVpbjTdnVROGlWOUFTVtC0
Cookie: OwnerId={owner_id}; Path=/
```

Response

```
403 Forbidden
Content-Type : application/json; charset=UTF-8
```

```
{
  "forbidden": {
    "guid": "b5b513b8-8436-4eee-b49e-ac1d369a9d1b",
    "message": "Unauthorized: DiscountsView failed permission check",
    "code": 403,
    "details": ""
  }
}
```

49.4.4 Order does not exist

Request

```
GET /orders/{non-existent_orderId}/account/products/{productId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound" : {
    "guid" : "4383366c-845c-4311-97ee-9dd7d7d0045b",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}
```

49.4.5 Product could not be found

Request

```
GET /orders/{orderId}/account/products/{non-existent_productId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound" : {
    "guid" : "40caab58-6c0b-4056-97eb-2219afe92ec1",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}
```

49.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

50 Orders-orderId-account-products-productId-discounts:PUT

51 PUT /orders/{orderId}/account/products/{productId}/discounts

Set discounts for a Product in an Order. Discounts will be applied only to the specified product. **Note: This will replace any existing discounts that were configured for the specified product.**

51.1 Contents

- 1 PUT
/orders/{orderId}/account/products/{productId}/discounts
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 Invalid discount amount
 - ◇ 1.3.3 Invalid discount type
 - ◇ 1.3.4 Invalid price type
 - ◇ 1.3.5 Invalid discount number of recurrences
 - ◇ 1.3.6 Missing price type
 - ◇ 1.3.7 Missing authorization
 - ◇ 1.3.8 Insufficient privileges
 - ◇ 1.3.9 Order does not exist
 - ◆ 1.4 See also

51.2 Request

PUT /orders/{orderId}/account/products/{productId}/discounts

51.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product for which actions are to be performed

51.2.2 URI Parameters

None.

51.2.3 Request Headers

Content-Type
Required. Set this header to `application/json; charset=UTF-8`

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

51.2.4 Request Body

```
{
  "discounts": [
    {
      "priceName": "recurring",
      "recurrences": -1,
      "type": "amount",
      "value": "1.00"
    }
  ]
}
```

51.2.4.1 Parameters

discounts - *list*
A list of discounts. Each discount contains the following:

priceName - *string*
The type of price to which the discount will be applied.
Validation: Should match the fee types configured in the back-end (eg. 'recurring').

recurrences - *integer*
The number of times that the discount will be applied. A value of "-1" means the discount will be applied every time the fee is charged.
Validation: Should be either a positive integer or -1.
Note: Value must be -1 if the priceName is "recurring".

type - *string*
Specifies the discount type which determines the method used to calculate the amount that is discounted. Possible values are:
amount: Discount will be treated as an exact amount
percent: Discount will be treated as a percentage

Validation: Should be set to either **amount** or **percent**.

value - *string*

The discount amount. The value will be interpreted based on the specified discount type.
Validation: Should be a positive number.

51.3 Response

51.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The discount was successfully set.
	POST data error	Required fields are missing.
		Validation Error:
400 Bad Request	{priceName} discount fee type is invalid {value} discount amount is invalid {type} discount type is unsupported {recurrences} discount period is invalid	Type of price is not recognized by the system. Discount amount should be a positive number. Type should be set to either amount or percent . Number of recurrences should be either a positive integer or -1 . If priceName is recurring , it should be -1 .
401 Unauthorized	-	The request was not properly authorized.
403 Forbidden	Unauthorized: DiscountsView failed permission check	The client does not have sufficient privileges.
404 Not Found	Resource not found	Order does not exist. Product could not be found.
502 Bad Gateway	-	Unexpected backend response. Cannot set custom discounts.

51.4 Example

51.4.1 Success

Request

```
PUT /orders/{orderId}/account/products/{productId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

204 No Content

51.4.2 Invalid discount amount

Request

```
PUT /orders/{orderId}/account/products/{productId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "-5.00"
  }]
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "computeFault": {
    "guid": "c3563e61-a297-4dbe-b072-24aace7b5e44",
    "message": "-5.00 discount amount is invalid",
    "code": 400,
    "details": ""
  }
}
```

51.4.3 Invalid discount type

Request

```
PUT /orders/{orderId}/account/products/{productId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```



```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "numbers",
    "value": "5.00"
  }]
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "computeFault": {
    "guid": "1a94f744-5b33-4d54-ab28-77b875934829",
    "message": "numbers discount type is unsupported",
    "code": 400,
    "details": ""
  }
}
```

51.4.4 Invalid price type

Request

PUT /orders/{orderId}/account/products/{productId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=

```
{
  "discounts": [{
    "priceName": "nonrecurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "computeFault" : {
    "guid" : "62967417-b2d6-46fa-8a8d-90d9231ce844",
    "message" : "nonrecurring discount fee type is invalid",
    "code" : 400,
    "details" : ""
  }
}
```

51.4.5 Invalid discount number of recurrences

Request

PUT /orders/{orderId}/account/products/{productId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-10",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "computeFault" : {
    "guid" : "bc360508-3392-4185-a07f-8f8cfff6884e",
    "message" : "-10 discount period is invalid",
    "code" : 400,
    "details" : ""
  }
}
```

51.4.6 Missing price type

Request

PUT /orders/{orderId}/account/products/{productId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=

```
{
  "discounts": [{,
    "recurrences": "-10",
    "type": "amount",
    "value": "5.00"
  }]
}
```

```
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "badRequest" : {
    "guid" : "5f2c19c5-ae18-4fee-b509-476b7ebe4b9c",
    "message" : "POST data error",
    "code" : 400,
    "details" : {
      "discounts.0.priceName" : "Required"
    }
  }
}
```

51.4.7 Missing authorization

Request

```
PUT /orders/{orderId}/account/products/{productId}/discounts
Content-Type : application/json; charset=UTF-8
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

51.4.8 Insufficient privileges

Request

```
PUT /orders/{orderId}/account/products/{productId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic ZWdjUlBtZWZiWnJVQHNPdGVjb250cm9sX2hvc3R3YXlfdXNlbnFpYjBpTdnVROGlWOUFTVTc0
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

```
403 Forbidden
Content-Type : application/json; charset=UTF-8
```

```
{
  "forbidden": {
    "guid": "1912d9fa-5f8e-4e16-b0ba-00d50c9c1ec4",
    "message": "Unauthorized: DiscountsView failed permission check",
    "code": 403,
    "details": ""
  }
}
```

51.4.9 Order does not exist

Request

```
PUT /orders/{non-existent_orderId}/account/products/{productId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "cca219c5-3e2b-481d-8e7f-7083064872ff",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

51.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

52 Orders-orderId-account-products-productId-promotion:DELETE

53 DELETE /orders/{orderId}/account/products/{productId}/promotion

Set promotion code for a product in an order.

53.1 Contents

- 1 DELETE
/orders/{orderId}/account/products/{productId}/promotion
 - ◆ 1.1 Request
 - ◇ 1.1.1 URI Parameters
 - ◇ 1.1.2 Query String Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Successfully delete a promotion code for product
 - ◇ 1.3.2 Product does not exist
 - ◆ 1.4 See also

53.2 Request

DELETE /orders/{orderId}/account/products/{productId}/promotion

53.2.1 URI Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*

UUID specifying the product for which actions are to be performed

53.2.2 Query String Parameters

None.

53.2.3 Request Headers

Content-Type

Required. Set this header to `application/json; charset=UTF-8`

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

53.3 Response

53.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The operation was successfully completed.
404 Not Found	Resource not found	Order does not exist. Product does not exist.
502 Bad Gateway	-	Unexpected backend response. Cannot delete promotion code.

53.4 Example

53.4.1 Successfully delete a promotion code for product

Request

```
DELETE /orders/{orderId}/account/products/{product_id}/promotion
Cookie: OwnerId={owner_id}; Path=/
```

Response

```
204 No Content
```

53.4.2 Product does not exist

Request

```
DELETE /orders/{order_id}/account/products/{non-existent_product_id}/promotion
Cookie: OwnerId={owner_id}; Path=/
```

Response

```
404 Not found
Content-Type : application/json; charset=UTF-8
{
  "itemNotFound": {
    "guid": "ee20516d-011e-44ea-99d2-ae7a2b6fe99a",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

```
} }
```

53.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

54 Orders-orderId-account-products-productId-promotion:PUT

55 PUT /orders/{orderId}/account/products/{productId}/promotion

Set promotion code for a product in an order.

55.1 Contents

- 1 PUT /orders/{orderId}/account/products/{productId}/promotion
 - ◆ 1.1 Request
 - ◇ 1.1.1 URI Parameters
 - ◇ 1.1.2 Query String Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Request Body Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Successfully add a promotion code for product
 - ◇ 1.3.2 Missing promoCode field
 - ◇ 1.3.3 No authorization
 - ◇ 1.3.4 Unassociated user without cookie
 - ◇ 1.3.5 Order does not exist
 - ◆ 1.4 See also

55.2 Request

PUT /orders/{orderId}/account/products/{productId}/promotion

55.2.1 URI Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product for which actions are to be performed

55.2.2 Query String Parameters

None.

55.2.3 Request Headers

Content-Type
Required. Set this header to `application/json; charset=UTF-8`

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

55.2.4 Request Body

```
{
  "promoCode": "ThanksgivingPriceDrop"
}
```

55.2.4.1 Request Body Parameters

promoCode - *string*
A promotion code to be applied to the product fees when retrieving the quote.

55.3 Response

55.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The operation was successfully completed.
400 Bad Request	POST data error	promoCode was not provided.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to add the promotion code.
403 Forbidden	Unauthorized: PromotionView failed permission check	Set product promotion as an authenticated user, not associated with the order.
404 Not Found	Resource not found	Order does not exist. Product does not exist.
502 Bad Gateway	-	Unexpected backend response. Cannot apply promotion code.

55.4 Example

55.4.1 Successfully add a promotion code for product

Request

```
PUT /orders/{orderId}/account/products/{product_id}/promotion
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "promoCode": "ThanksgivingPriceDrop"
}
```

Response

```
204 No Content
```

55.4.2 Missing promoCode field

Request

```
PUT /orders/{orderId}/account/products/{product_id}/promotion
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "code": "ThanksgivingPriceDrop"
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "badRequest": {
    "guid": "ee20516d-011e-44ea-99d2-ae7a2b6fe99a",
    "message": "Resource not found",
    "code": 400,
    "details": {
      "promoCode": "Required"
    }
  }
}
```

55.4.3 No authorization

Request

```
PUT /orders/{orderId}/account/user
Cookie: OwnerId={wrong_owner_id}; Path=
```

```
{
  "promoCode": "ThanksgivingPriceDrop"
}
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

55.4.4 Unassociated user without cookie

Request

```
PUT /orders/{orderId}/account/products/{productId}/promotion
Authorization: {unassociated_user_authorization}
```

```
{
  "promoCode": "truflex600_mitko"
}
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8
```

```
{
  "forbidden" : {
    "guid" : "3e6e1847-402f-4a91-b083-190ca0713b23",
    "message" : "Unauthorized: PromotionView failed permission check",
    "code" : 403,
    "details" : ""
  }
}
```

55.4.5 Order does not exist

Request

```
PUT /orders/{non-existent_orderId}/account/products/{product_id}/promotion
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "promoCode": "ThanksgivingPriceDrop"
}
```

Response

404 Not found

Content-Type : application/json; charset=UTF-8

```
{
  "itemNotFound": {
    "guid": "ee20516d-011e-44ea-99d2-ae7a2b6fe99a",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

55.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

56 Orders-orderId-account-products-productId-services-serviceId-attributes:GET

57 GET

`/orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes`

Get the Attributes of a given Service. Not all services will have attributes; depending on the service, this call *may* return a response code of 405 (Method not supported). If attributes are supported on the service, but have not yet been set, this call will return an empty dictionary in the response body.

57.1 Contents

- 1 GET
`/orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes`
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful responses
 - 1.2.2.2 Error responses
 - ◆ 1.3 See also

57.2 Request

GET `/orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes`

57.2.1 Request Parameters

`orderId` - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

`productId` - *string*
UUID specifying the product

`serviceId` - *string*
UUID specifying the service

57.2.2 URI Parameters

None.

57.2.3 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

57.3 Response

57.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The operation was successfully completed.
400 Bad Request	No attributes defined for the provided service specification	Trying to get attributes for service with no attributes.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The <i>OwnerId</i> cookie was not specified or is incorrect.
403 Forbidden	Unauthorized: Attributes failed permission check	The client does not have permissions for this operation.
404 Not Found	Resource not found	The client is trying to add attribute for non existing order, product or service.
405 Method Not Allowed	-	The service does not support attributes.
502 Bad Gateway	-	Unexpected backend response.

57.3.2 Response Body

57.3.2.1 Successful responses

Will be a dictionary whose keys/values are specific to the service category of the service. See [Service Attributes Structure](#).

57.3.2.2 Error responses

```
400 Bad Request - trying to get the attributes for a service that does not have attributes.
{
  "computeFault": {
    "guid": "756b755a-84cf-4692-b2c6-70808dc9d00d",
    "message": "No attributes defined for the provided service specification",
    "code": 400, "details": ""
  }
}
```

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

404 Not Found - the order, product or service was not found.

```
{
  "itemNotFound": {
    "guid": "e1792d8a-5b0a-43ef-adf5-34a0710c8fc5",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

502 Bad Gateway - unexpected error from the backend.

```
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}
```

57.4 See also

- [Orders API](#)
- [Common Features](#)
- [Service Attributes Structure](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

59 OPTIONS

/orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes

Get list of supported methods for this resource, plus some hints on the expected resource structure.

59.1 Contents

- 1 OPTIONS
/orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Response Status Code
 - ◇ 1.2.2 Response Headers
 - ◇ 1.2.3 Response Body
 - ◆ 1.3 See also

59.2 Request

OPTIONS /orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes

59.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*

UUID specifying the product

serviceId - *string*

UUID specifying the service

59.3 Response

59.3.1 Response Status Code

Status Code	Error Message	Description
200 OK	-	The operation was successfully completed.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The <i>OwnerId</i> cookie was not specified or is incorrect.
403 Forbidden	Unauthorized: Attributes failed permission check	The client does not have permissions for this operation.
404 Not Found	Resource not found	The client is trying to add attribute for non existing order, product or service.
502 Bad Gateway	-	Unexpected backend response.

59.3.2 Response Headers

Allow

Contains a comma-separated list of methods that are allowed on this resource. For services that do not support attributes, this will contain only the OPTIONS method.

59.3.3 Response Body

A dictionary whose keys/values are specific to the service category of the service. Rather than returning actual service data, the body will contain a dictionary that provides hints for how to build the request body for a PUT request, for example:

```
{
  "name": "String",
  "numWidgets": "Integer",
  "widgets": [
    {
      "id": "String",
      "widgetType": "String",
      "otherAttribute": "String"
    }
  ]
}
```

59.4 See also

- [Orders API](#)
- [Common Features](#)
- [Service Attributes Structure](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

60 Orders-orderId-account-products-productId-services-serviceId-attributes:PUT

61 PUT

`/orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes`

Adds Attributes to the Service. Not all services support this call. Depending on the service, this call *may* return a response code of 405 (method not supported).

61.1 Contents

- 1 PUT
`/orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes`
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - ◇ 1.1.5 Validation Rules
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful response
 - 1.2.2.2 Error responses
 - ◆ 1.3 See also

61.2 Request

PUT `/orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes`

61.2.1 Request Parameters

`orderId` - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

`productId` - *string*
UUID specifying the product

`serviceId` - *string*
UUID specifying the service

61.2.2 URI Parameters

None.

61.2.3 Request Headers

`Content-Type`
Required. Set this header to `application/json; charset=UTF-8`

`Cookie`
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

61.2.4 Request Body

A dictionary whose keys/values will depend on the service category of the service. See [Service Attributes Structure](#).

```
{
  "key_one": "Value One",
  "key_two": "Value Two",
  ...
}
```

61.2.5 Validation Rules

1. The service URL (`https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}`) should point to a valid service already added to the product.
2. The attribute key/value pair should be allowed for the service specification.

61.3 Response

61.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The operation was successfully completed.
	POST data error	Trying to create an attribute without body or with invalid body.
400 Bad Request	Invalid term supplied -1	Trying to add attributes for transfer domain request with invalid value for term (should be 1).
	No attributes defined for the provided service specification	Trying to get attributes for service with no attributes.
401 Unauthorized		

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

The *OwnerId* cookie was not specified or is incorrect.

403 Forbidden Unauthorized: Attributes failed permission check

The client does not have permissions to add attributes.

404 Not Found Resource not found

The client is trying to add attribute for non existing order, product or service.

405 Method Not Allowed
502 Bad Gateway -

The service does not support attributes.

Unexpected backend response.

61.3.2 Response Body

61.3.2.1 Successful response

None.

61.3.2.2 Error responses

400 Bad Request - no request body is passed

```
{
  "badRequest": {
    "guid": "bd80dfdc-5752-4472-96f2-90f590ddb3f9",
    "message": "POST data error",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - invalid body is passed (required field is missing)

```
{
  "badRequest": {
    "guid": "aae865ae-5102-49d9-blef-8a5419cc0ce7",
    "message": "POST data error",
    "code": 400,
    "details": {"serverName": "Required"}
  }
}
```

400 Bad Request - invalid attributes are passed

```
{
  "badRequest": {
    "guid": "fdd0b796-c690-47af-9666-0478e2bcae27",
    "message": "POST data error",
    "code": 400,
    "details": {"serverLocation": "String does not match expected pattern"}
  }
}
```

400 Bad Request - invalid term for transfer request is passed.

```
{
  "computeFault": {
    "guid": "1e98f8ea-d298-4867-a9aa-0a6486dbefb0",
    "message": "Invalid term supplied -1",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - trying to get the attributes for a service that does not have attributes.

```
{
  "computeFault": {
    "guid": "756b755a-84cf-4692-b2c6-70808dc9d00d",
    "message": "No attributes defined for the provided service specification",
    "code": 400, "details": ""
  }
}
```

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

404 Not Found - the order or product was not found.

```
{
  "itemNotFound": {
    "guid": "e1792d8a-5b0a-43ef-adf5-34a0710c8fc5",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

502 Bad Gateway - unexpected error from the backend.

```
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}
```

61.4 See also

- [Orders API](#)
- [Common Features](#)
- [Service Attributes Structure](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

62 Orders-orderId-account-products-productId-services-serviceId-discounts:DELETE

63 DELETE

/orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts

Remove all discounts that were previously added at the Service level. This will not remove discounts that were added at the order or product level.

63.1 Contents

- 1 DELETE
 - /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 Missing authorization
 - ◇ 1.3.3 Insufficient privileges
 - ◇ 1.3.4 Order does not exist
 - ◇ 1.3.5 Product could not be found
 - ◇ 1.3.6 Service could not be found
 - ◆ 1.4 See also

63.2 Request

DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts

63.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product for which actions are to be performed

serviceId - *string*
UUID specifying the service for which actions are to be performed

63.2.2 URI Parameters

None.

63.2.3 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

63.3 Response

63.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The request was executed successfully.
401 Unauthorized	-	The request was not properly authorized.
403 Forbidden	Unauthorized: DiscountsView failed permission check	The client does not have sufficient privileges. Order does not exist.
404 Not Found	Resource not found	Product could not be found in this order. Service could not be found for this product.
502 Bad Gateway	-	Custom discount could not be deleted.

63.4 Example

63.4.1 Success

Request

```
DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

Response

```
204 No Content
```

63.4.2 Missing authorization

Request

```
DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
```

Cookie: OwnerId={owner_id}; Path=/

Response

401 Unauthorized
Content-Type : text/plain; charset=UTF-8

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

63.4.3 Insufficient privileges

Request

DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Authorization: Basic ZWdjUlBtZWZiWnJVQHNPdGVjb250cm9sX2hvc3R3YXlfdXNfbWFnPbJpTdnVROGlWOUFTVTc0
Cookie: OwnerId={owner_id}; Path=/

Response

403 Forbidden
Content-Type : application/json; charset=UTF-8

```
{
  "forbidden" : {
    "guid" : "a0c6a265-aaa9-4187-b4eb-0b0b7b112810",
    "message" : "Unauthorized: DiscountsView failed permission check",
    "code" : 403,
    "details" : ""
  }
}
```

63.4.4 Order does not exist

Request

DELETE /orders/{non-existent_orderId}/account/products/{productId}/services/{serviceId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/

Response

404 Not Found
Content-Type : application/json; charset=UTF-8

```
{
  "itemNotFound" : {
    "guid" : "8a56063d-b099-4688-b020-d31035529d45",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}
```

63.4.5 Product could not be found

Request

DELETE /orders/{orderId}/account/products/{non-existent_productId}/services/{serviceId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/

Response

404 Not Found
Content-Type : application/json; charset=UTF-8

```
{
  "itemNotFound" : {
    "guid" : "42f454f4-a534-46ea-888e-c05af56c1130",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}
```

63.4.6 Service could not be found

Request

DELETE /orders/{orderId}/account/products/{productId}/services/{non-existent_serviceId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/

Response

404 Not Found
Content-Type : application/json; charset=UTF-8

```
{
  "itemNotFound" : {
    "guid" : "42f454f4-a534-46ea-888e-c05af56c1130",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}
```

63.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

64 Orders-orderId-account-products-productId-services-serviceId-discounts:GET

65 GET

/orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts

Get discounts that were added at the Service level.

65.1 Contents

- 1 GET
/orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 Missing authorization
 - ◇ 1.3.3 Insufficient privileges
 - ◇ 1.3.4 Order does not exist
 - ◇ 1.3.5 Product could not be found
 - ◇ 1.3.6 Service could not be found
 - ◆ 1.4 See also

65.2 Request

GET /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts

65.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product for which actions are to be performed

serviceId - *string*
UUID specifying the service for which actions are to be performed

65.2.2 URI Parameters

None.

65.2.3 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

65.3 Response

65.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The request was executed successfully.
401 Unauthorized	-	The request was not properly authorized.
403 Forbidden	Unauthorized: DiscountsView failed permission check	The client does not have sufficient privileges.
404 Not Found	Resource not found	Order does not exist. Product could not be found in this order. Service could not be found for this product.
502 Bad Gateway	-	Failed to retrieve service discount info.

65.3.2 Response Body

```
{
  "discounts": [
    {
      "priceName": "recurring",
      "recurrences": -1,
      "type": "amount",
      "value": "1.00"
    }
  ],
  "links": [
    {
      "href": "https://api.hostway.com/orders/{orderId}/account/product/{productId}/services/{serviceId}/discounts",
      "rel": "self"
    }
  ]
}
```

65.3.2.1 Parameters

discounts - *list*

A list of discounts. Each discount contains the following:

priceName - *string*

The type of price to which the discount will be applied.

recurrences - *integer*

The number of times that the discount will be applied. A value of "-1" means the discount will be applied every time the fee is charged.

type - *string*

Specifies the discount type which determines the method used to calculate the amount that is discounted. Possible values are:
amount: Discount will be treated as an exact amount.

value - *string*

The discount amount. The value will be interpreted based on the specified discount type.

links - *list*

[Hypermedia](#) for this resource. Links to self.

65.4 Example

65.4.1 Success

Request

```
GET /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

Response

```
200 OK
Content-Type : application/json; charset=UTF-8
```

```
{
  "discounts" : [ {
    "priceName" : "recurring",
    "type" : "amount",
    "value" : "5.00",
    "recurrences" : -1
  } ],
  "links" : [ {
    "href" : "https://api.hostway.com/orders/0000014859b641999db63a3d007f000000010001/account/products/0000014859b8411671c04a92007f000000010001/services/0000014859b88f1f066e3a67007f000000010001/discounts/",
    "rel" : "self"
  } ]
}
```

65.4.2 Missing authorization

Request

```
GET /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Cookie: OwnerId={owner_id}; Path=
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

65.4.3 Insufficient privileges

Request

```
GET /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Authorization: Basic ZWdjUlBtZWZiWnJVQHNPdGVjb250cm9sX2hvc3R3YXlfdXNfbWFPbjpTdnVROGlWOUFTVTc0
Cookie: OwnerId={owner_id}; Path=
```

Response

```
403 Forbidden
Content-Type : application/json; charset=UTF-8
```

```
{
  "forbidden": {
    "guid": "b5b513b8-8436-4eee-b49e-ac1d369a9d1b",
    "message": "Unauthorized: DiscountsView failed permission check",
    "code": 403,
    "details": ""
  }
}
```

65.4.4 Order does not exist

Request

```
GET /orders/{non-existent_orderId}/account/products/{productId}/services/{serviceId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8

{
  "itemNotFound" : {
    "guid" : "4383366c-845c-4311-97ee-9dd7d7d0045b",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}
```

65.4.5 Product could not be found

Request

```
GET /orders/{orderId}/account/products/{non-existent_productId}/services/{serviceId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8

{
  "itemNotFound" : {
    "guid" : "40caab58-6c0b-4056-97eb-2219afe92ec1",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}
```

65.4.6 Service could not be found

Request

```
GET /orders/{orderId}/account/products/{productId}/services/{non-existent_serviceId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8

{
  "itemNotFound" : {
    "guid" : "fdbff03e-5992-4c64-a769-94cd61bbb927",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}
```

65.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

66 Orders-orderId-account-products-productId-services-serviceId-discounts:PUT

67 PUT

/orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts

Set discounts for the specified service. **Note: This will replace any existing discounts that were configured for the specified service.**

67.1 Contents

- 1 PUT
 - /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 Invalid discount amount
 - ◇ 1.3.3 Invalid discount type
 - ◇ 1.3.4 Unsupported price type
 - ◇ 1.3.5 Invalid discount number of recurrences
 - ◇ 1.3.6 Missing price type
 - ◇ 1.3.7 Missing authorization
 - ◇ 1.3.8 Insufficient privileges
 - ◇ 1.3.9 Order does not exist
 - ◇ 1.3.10 Product not found
 - ◇ 1.3.11 Service not found
 - ◆ 1.4 See also

67.2 Request

PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts

67.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product for which actions are to be performed

serviceId - *string*
UUID specifying the service for which actions are to be performed

67.2.2 URI Parameters

None.

67.2.3 Request Headers

Content-Type
Required. Set this header to `application/json; charset=UTF-8`

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

67.2.4 Request Body

Body for adding an indefinite discount of 1 dollar for the recurring fee for the service:

```
{
  "discounts": [
    {
      "priceName": "recurring",
      "recurrences": -1,
      "type": "amount",
      "value": 1.00
    }
  ]
}
```

Body for adding a 1 dollar amount discount for the first cycle for a domain registration fee:

```
{
  "discounts": [
    {
      "priceName": "domain_registration_fee",
      "recurrences": 1,
      "type": "amount",
      "value": 1.00
    }
  ]
}
```

Body for adding a 10% discount for the first three monthly cycles for a service's recurring fee:

```
{
  "discounts": [
    {
      "priceName": "recurring",
      "recurrences": 3,
      "type": "percent",
      "value": 0.1
    }
  ]
}
```

67.2.4.1 Parameters

discounts - *list*

A list of discounts. Each discount contains the following:

priceName - *string*

The type of price to which the discount will be applied.

Validation: Should match the fee types configured in the back-end (eg. 'recurring', 'domain_registration_fee', 'setup_fee' etc.) and should be configured for the corresponding franchise.

recurrences - *integer*

The number of times that the discount will be applied. A value of "-1" means the discount will be applied every time the fee is charged.

Validation: Should be either a positive integer or -1.

Note: Value must be -1 if the priceName is "recurring".

type - *string*

Specifies the discount type which determines the method used to calculate the amount that is discounted. Possible values are:

amount: Discount will be treated as an exact amount

percent: Discount will be treated as a percentage

Validation: Should be set to either **amount** or **percent**.

value - *string*

The discount amount. The value will be interpreted based on the specified discount type.

Validation: Should be a positive number.

67.3 Response

67.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The discount was successfully set.
	POST data error	Required fields are missing. Validation Error:
400 Bad Request	{priceName} discount fee type is invalid {value} discount amount is invalid {type} discount type is unsupported {recurrences} discount period is invalid	Type of price is not recognized by the system. Discount amount should be a positive number. Type should be set to either amount or percent . Number of recurrences should be either a positive integer or -1. If price type is recurring , it should be -1.
401 Unauthorized	-	The request was not properly authorized.
403 Forbidden	Unauthorized: DiscountsView failed permission check	The client does not have sufficient privileges.
404 Not Found	Resource not found	Order does not exist. Product could not be found in this order. Service could not be found for this product.
502 Bad Gateway	-	Unexpected backend response. Cannot set custom discounts.

67.4 Example

67.4.1 Success

Request

```
PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

```
204 No Content
```

67.4.2 Invalid discount amount

Request

```
PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Content-Type : application/json; charset=UTF-8
```

Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "-5.00"
  }]
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "computeFault": {
    "guid": "c3563e61-a297-4dbe-b072-24aace7b5e44",
    "message": "-5.00 discount amount is invalid",
    "code": 400,
    "details": ""
  }
}
```

67.4.3 Invalid discount type

Request

PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "numbers",
    "value": "5.00"
  }]
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "computeFault": {
    "guid": "1a94f744-5b33-4d54-ab28-77b875934829",
    "message": "numbers discount type is unsupported",
    "code": 400,
    "details": ""
  }
}
```

67.4.4 Unsupported price type

Request

PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/

```
{
  "discounts": [{
    "priceName": "recurring_yearly",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "computeFault" : {
    "guid" : "43357b7a-0bba-4e40-b565-21422937b7df",
    "message" : "recurring_yearly discount fee type is invalid",
    "code" : 400,
    "details" : ""
  }
}
```

67.4.5 Invalid discount number of recurrences

Request

PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/

```
{
  "discounts": [{
    "priceName": "recurring",
```

```
    "recurrences": "-10",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "computeFault" : {
    "guid" : "bc360508-3392-4185-a07f-8f8cfff6884e",
    "message" : "-10 discount period is invalid",
    "code" : 400,
    "details" : ""
  }
}
```

67.4.6 Missing price type

Request

PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/

```
{
  "discounts": [{
    "recurrences": "-10",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

400 Bad Request
Content-Type : application/json; charset=UTF-8

```
{
  "badRequest" : {
    "guid" : "5f2c19c5-ae18-4fee-b509-476b7ebe4b9c",
    "message" : "POST data error",
    "code" : 400,
    "details" : {
      "discounts.0.priceName" : "Required"
    }
  }
}
```

67.4.7 Missing authorization

Request

PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Content-Type : application/json; charset=UTF-8
Cookie: OwnerId={owner_id}; Path=/

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

401 Unauthorized
Content-Type : text/plain; charset=UTF-8

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

67.4.8 Insufficient privileges

Request

PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic ZWdjUlBtZWZiWnJVQHNPdGVjb250cm9sX2hvc3R3YXlfdXNfbWpjb2VudmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response


```
403 Forbidden
Content-Type : application/json; charset=UTF-8
```

```
{
  "forbidden": {
    "guid": "1912d9fa-5fbc-4e16-b0ba-00d50c9c1ec4",
    "message": "Unauthorized: DiscountsView failed permission check",
    "code": 403,
    "details": ""
  }
}
```

67.4.9 Order does not exist

Request

```
PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "cca219c5-3e2b-481d-8e7f-7083064872ff",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

67.4.10 Product not found

Request

```
PUT /orders/{orderId}/account/products/{non-existent_productId}/services/{serviceId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "cca219c5-3e2b-481d-8e7f-7083064872ff",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

67.4.11 Service not found

Request

```
PUT /orders/{orderId}/account/products/{productId}/services/{non-existent_serviceId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "cca219c5-3e2b-481d-8e7f-7083064872ff",

```

```
"message": "Resource not found",  
"code": 404,  
"details": ""  
}  
}
```

67.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

68 Orders-orderId-account-products-productId-services-serviceId-notes:DELETE

69 DELETE

/orders/{orderId}/account/products/{productId}/services/{serviceId}/notes

Removes notes from a service.

69.1 Contents

- 1 DELETE
/orders/{orderId}/account/products/{productId}/services/{serviceId}/notes
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful response
 - 1.2.2.2 Error responses
 - ◆ 1.3 See also

69.2 Request

DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes

69.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
Product's unique Id.

serviceId - *string*
UUID specifying the service to be deleted

69.2.2 URI Parameters

None.

69.2.3 Request Headers

69.3 Response

69.3.1 Status Code

Status Code	Error Message	Description
204	-	The operation was successfully completed.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The request was not properly authorized.
403 Forbidden	Unauthorized: NotesView failed permission check	The client does not have permissions to delete the notes.
404 Not Found	Resource not found	The order, product or service is not found.
502 Bad Gateway	-	Unexpected backend response.

69.3.2 Response Body

69.3.2.1 Successful response

None.

69.3.2.2 Error responses

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden - the client does not have permissions to delete the notes.

```
{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: NotesView failed permission check",
    "code": 403,
    "details": ""
  }
}
```

404 Not Found

```
{
  "itemNotFound": {
    "guid": "579951da-5f2c-4db7-9169-25658bfaafdb",
    "message": "Resource not found",
  }
}
```

```
    "code": 404,  
    "details": ""  
  }  
}  
502 Bad Gateway - unexpected error from the backend.  
{  
  "computeFault": {  
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",  
    "timestamp": "2014-09-10 20:59:50",  
    "code": 502,  
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"  
  }  
}
```

69.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

70 Orders-orderId-account-products-productId-services-serviceId-notes:GET

71 GET /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes

Get all notes for a given service.

71.1 Contents

- 1 GET /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful response
 - 1.2.2.1.1 Parameters
 - 1.2.2.2 Error responses
 - ◆ 1.3 See also

71.2 Request

GET /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes

71.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product

serviceId - *string*
UUID specifying the product

71.2.2 URI Parameters

None.

71.2.3 Request Headers

71.3 Response

71.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The notes for the service (if any) are successfully returned.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The request was not properly authorized.
403 Forbidden	Unauthorized: NotesView failed permission check	The client does not have permissions to get information about the notes.
404 Not Found	Resource not found	The client is trying to get the notes for non existing order, product or service.
502 Bad Gateway	-	Unexpected backend response.

71.3.2 Response Body

71.3.2.1 Successful response

```
200 OK
{
  "notes": ["xx", "zz"],
  "links": [
    { "href": "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}/notes", "rel": "self" }
  ]
}
```

71.3.2.1.1 Parameters

notes - *list*
List of the notes for the given service.

links - *list*
[Hypermedia](#) for the list of notes resources

71.3.2.2 Error responses

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden - the client does not have access to get the information about the notes for a service.

```
{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: NotesView failed permission check",
    "code": 403,
    "details": ""
  }
}

404 Not Found - the order, the product or the service doesn't exist.
{
  "itemNotFound": {
    "guid": "3a422918-24c3-4c20-9bee-f1c61695e987",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}

502 Bad Gateway - unexpected error from the backend.
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}
```

71.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

72 Orders-orderId-account-products-productId-services-serviceId-notes:OPTIONS

73 OPTIONS

/orders/{orderId}/account/products/{productId}/services/{serviceId}/notes

Get list of supported methods for this resource, plus some hints on the expected resource structure.

73.1 Contents

- 1 OPTIONS
/orders/{orderId}/account/products/{productId}/services/{serviceId}/notes
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Response Status Code
 - ◇ 1.2.2 Response Headers
 - ◇ 1.2.3 Response Body
 - ◆ 1.3 See also

73.2 Request

OPTIONS /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes

73.2.1 Request Parameters

- orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.
- productId - *string*
UUID specifying the product
- serviceId - *string*
UUID specifying the service

73.3 Response

73.3.1 Response Status Code

Status Code	Error Message	Description
200 OK	-	The operation was successfully completed.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The request was not properly authorized.
403 Forbidden	Unauthorized: NotesView failed permission check	The client does not have permissions for this operation.
404 Not Found	Resource not found	The order, product or service doesn't exist.

73.3.2 Response Headers

- Allow
Contains a comma-separated list of methods that are allowed on this resource. For services that do not support attributes, this will contain only the OPTIONS method.

73.3.3 Response Body

A dictionary whose keys/values are specific to the service category of the service. Rather than returning actual service data, the body will contain a dictionary that provides hints for how to build the request body for a PUT request, for example:

73.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

74 Orders-orderId-account-products-productId-services-serviceId-notes:POST

75 POST

`/orders/{orderId}/account/products/{productId}/services/{serviceId}/notes`

Adds a new note to a service in the order.

75.1 Contents

- 1 POST
`/orders/{orderId}/account/products/{productId}/services/{serviceId}/notes`
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Headers
 - ◇ 1.2.3 Response Body
 - 1.2.3.1 Successful response
 - 1.2.3.2 Error responses
 - ◆ 1.3 See also

75.2 Request

POST `/orders/{orderId}/account/products/{productId}/services/{serviceId}/notes`

75.2.1 Request Parameters

`orderId` - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

`productId` - *string*
UUID specifying the product

`serviceId` - *string*
UUID specifying the service

75.2.2 URI Parameters

None.

75.2.3 Request Headers

`Content-Type`
Required. Set this header to `application/json; charset=UTF-8`

75.2.4 Request Body

```
{  
  "note": "xx"  
}
```

75.2.4.1 Parameters

`note` - *string*
The text of the note

75.3 Response

75.3.1 Status Code

Status Code	Error Message	Description
201 Created	-	The note was successfully added to the service.
400 Bad Request	POST data error	Trying to create a note without passing all the required attributes.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The request was not properly authorized.
403 Forbidden	Unauthorized: NotesView failed permission check	The client does not have permissions to add notes.
404 Not Found	Resource not found	The order or product or service doesn't exist.
405 Method not allowed	Method not allowed	Trying to add notes for placed order.
502 Bad Gateway	-	Unexpected backend response.

75.3.2 Response Headers

None.

75.3.3 Response Body

75.3.3.1 Successful response

201 Created

75.3.3.2 Error responses

400 Bad Request - request body is missing some of the required attributes

```
{
  "badRequest": {
    "guid": "23bf964e-fb93-4bd6-816e-3dcb31b82755",
    "message": "POST data error",
    "code": 400,
    "details": {
      "note": "Required"
    }
  }
}
```

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden - the client does not have access to add notes.

```
{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: NotesView failed permission check",
    "code": 403,
    "details": ""
  }
}
```

404 Not Found - the order, product or the service was not found.

```
{
  "itemNotFound": {
    "guid": "e1792d8a-5b0a-43ef-adf5-34a0710c8fc5",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}

{
  "badMethod": {
    "guid": "3e9418e0-5a5d-4652-ae50-0e665d39571a",
    "message": "Method not allowed",
    "code": 405,
    "details": ""
  }
}
```

502 Bad Gateway - unexpected error from the backend.

```
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}
```

75.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

76 Orders-orderId-account-products-productId-services-serviceId-notes:PUT

77 PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes

Updates the notes for a service in the order.

77.1 Contents

- 1 PUT
/orders/{orderId}/account/products/{productId}/services/{serviceId}/notes
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Headers
 - ◇ 1.2.3 Response Body
 - 1.2.3.1 Successful response
 - 1.2.3.2 Error responses
 - ◆ 1.3 See also

77.2 Request

PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/notes

77.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product

serviceId - *string*
UUID specifying the service

77.2.2 URI Parameters

None.

77.2.3 Request Headers

Content-Type
Required. Set this header to `application/json; charset=UTF-8`

77.2.4 Request Body

```
{  
  "notes": ["xx", "zz"]  
}
```

77.2.4.1 Parameters

notes - *list*
List of notes to update the existing notes for a the service

77.3 Response

77.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The notes were successfully added to the service.
400 Bad Request	POST data error	Trying to update the notes without passing all the required attributes.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The request was not properly authorized.
403 Forbidden	Unauthorized: NotesView failed permission check	The client does not have permissions to update the notes.
404 Not Found	Resource not found	The order, product or service doesn't exist.
405 Method not allowed	Method not allowed	Trying to update the notes for placed order.
502 Bad Gateway	-	Unexpected backend response.

77.3.2 Response Headers

None.

77.3.3 Response Body

77.3.3.1 Successful response

201 Created

77.3.3.2 Error responses

400 Bad Request - request body is missing some of the required attributes

```
{
  "badRequest": {
    "guid": "23bf964e-fb93-4bd6-816e-3dcb31b82755",
    "message": "POST data error",
    "code": 400,
    "details": {
      "notes": "Required"
    }
  }
}
```

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden - the client does not have access to update the notes.

```
{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: NotesView failed permission check",
    "code": 403,
    "details": ""
  }
}
```

404 Not Found - the order, product or the service was not found.

```
{
  "itemNotFound": {
    "guid": "e1792d8a-5b0a-43ef-adf5-34a0710c8fc5",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}

{
  "badMethod": {
    "guid": "3e9418e0-5a5d-4652-ae50-0e665d39571a",
    "message": "Method not allowed",
    "code": 405,
    "details": ""
  }
}
```

502 Bad Gateway - unexpected error from the backend.

```
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}
```

77.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

78 Orders-orderId-account-products-productId-services-serviceId-promotion:DELETE

79 DELETE

/orders/{orderId}/account/products/{productId}/services/{serviceId}/promotion

Remove promotion code for a service in an order.

79.1 Contents

- 1 DELETE
 - /orders/{orderId}/account/products/{productId}/services/{serviceId}/promotion
 - ◆ 1.1 Request
 - ◇ 1.1.1 URI Parameters
 - ◇ 1.1.2 Query String Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Successfully remove a promotion code for service
 - ◇ 1.3.2 No authorization
 - ◇ 1.3.3 Service does not exist
 - ◆ 1.4 See also

79.2 Request

DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}/promotion

79.2.1 URI Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product

serviceId - *string*
UUID specifying the service for which actions are to be performed

79.2.2 Query String Parameters

None.

79.2.3 Request Headers

Content-Type
Required. Set this header to `application/json; charset=UTF-8`

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

79.3 Response

79.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The operation was successfully completed.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to delete the promotion code. Only authenticated users may access the order, and no authentication header provided for the current request.
404 Not Found	Resource not found	Order does not exist. Product or service does not exist.
502 Bad Gateway	-	Unexpected backend response. Cannot delete promotion code.

79.4 Example

79.4.1 Successfully remove a promotion code for service

Request

```
DELETE /orders/{order_id}/account/products/{product_id}/services/{service_id}/promotion
Cookie: OwnerId={owner_id}; Path=
```

Response

```
204 No Content
```

79.4.2 No authorization

Request

```
DELETE /orders/{orderId}/account/products/{product_id}/services/{serviceId}/promotion
Cookie: OwnerId={wrong_owner_id}; Path=
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

79.4.3 Service does not exist

Request

```
DELETE /orders/{order_id}/account/products/{product_id}/services/{non_existing_service_id}/promotion
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "{guid}",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

79.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

80 Orders-orderId-account-products-productId-services-serviceId-promotion:PUT

81 PUT

/orders/{orderId}/account/products/{productId}/services/{serviceId}/promotion

Set promotion code for a service in an order.

81.1 Contents

- 1 PUT
/orders/{orderId}/account/products/{productId}/services/{serviceId}/promotion
 - ◆ 1.1 Request
 - ◇ 1.1.1 URI Parameters
 - ◇ 1.1.2 Query String Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Request Body Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Successfully add a promotion code for service
 - ◇ 1.3.2 Missing promoCode field
 - ◇ 1.3.3 No authorization
 - ◇ 1.3.4 Order does not exist
 - ◆ 1.4 See also

81.2 Request

PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}/promotion

81.2.1 URI Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product

serviceId - *string*
UUID specifying the service for which actions are to be performed

81.2.2 Query String Parameters

None.

81.2.3 Request Headers

Content-Type
Required. Set this header to `application/json; charset=UTF-8`

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

81.2.4 Request Body

```
{  
  "promoCode": "{promoCode}"  
}
```

81.2.4.1 Request Body Parameters

promoCode - *string* (Required)
A promotion code to be applied to the service fees when retrieving the quote.

81.3 Response

81.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The operation was successfully completed.
400 Bad Request	POST data error	promoCode was not provided. No owner Id specified or the specified owner is not authorized to add the promotion code.
401 Unauthorized	-	Only authenticated users may access the order, and no authentication header provided for the current request.
404 Not Found	Resource not found	Order does not exist. Product or service does not exist.
502 Bad Gateway	-	Unexpected backend response. Cannot apply promotion code.

81.4 Example

81.4.1 Successfully add a promotion code for service

Request

```
PUT /orders/{orderId}/account/products/{product_id}/services/{serviceId}/promotion
```

```
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "promoCode": "ThanksgivingPriceDrop"
}
```

Response

```
204 No Content
```

81.4.2 Missing promoCode field

Request

```
PUT /orders/{orderId}/account/products/{product_id}/services/{serviceId}/promotion
```

```
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "code": "ThanksgivingPriceDrop"
}
```

Response

```
400 Bad Request
```

```
Content-Type : application/json; charset=UTF-8
```

```
{
  "badRequest": {
    "guid": "{guid}",
    "message": "Resource not found",
    "code": 400,
    "details": {
      "promoCode": "Required"
    }
  }
}
```

81.4.3 No authorization

Request

```
PUT /orders/{orderId}/account/products/{product_id}/services/{serviceId}/promotion
```

```
Cookie: OwnerId={wrong_owner_id}; Path=
```

```
{
  "promoCode": "ThanksgivingPriceDrop"
}
```

Response

```
401 Unauthorized
```

```
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

81.4.4 Order does not exist

Request

```
PUT /orders/{non-existing_orderId}/account/products/{product_id}/services/{serviceId}/promotion
```

```
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "promoCode": "ThanksgivingPriceDrop"
}
```

Response

```
404 Not found
```

```
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "{guid}",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

81.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

82 Orders-orderId-account-products-productId-services-serviceId:DELETE

83 DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}

Removes service from a product.

83.1 Contents

- 1 DELETE
/orders/{orderId}/account/products/{productId}/services/{serviceId}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful response
 - 1.2.2.2 Error responses
 - ◆ 1.3 See also

83.2 Request

DELETE /orders/{orderId}/account/products/{productId}/services/{serviceId}

83.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
Product's unique Id.

serviceId - *string*
UUID specifying the service to be deleted

83.2.2 URI Parameters

None.

83.2.3 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

83.3 Response

83.3.1 Status Code

Status Code	Error Message	Description
204	-	The operation was successfully completed.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The OwnerId cookie or the Authorization header are not specified or do not permit access to the specified order.
403 Forbidden	Unauthorized: Services failed permission check	The client does not have permissions to delete a service.
404 Not Found	Resource not found	The order, product or service is not found.
502 Bad Gateway	-	Unexpected backend response.

83.3.2 Response Body

83.3.2.1 Successful response

None.

83.3.2.2 Error responses

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden - the client does not have permissions to delete a service.

```
{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: Services failed permission check",
    "code": 403,
    "details": ""
  }
}
```

404 Not Found

```
{
  "itemNotFound": {
    "guid": "579951da-5f2c-4db7-9169-25658bfaafdb",
    "message": "Resource not found",
    "code": 404,
  }
}
```

```
    "details": ""
  }
}

502 Bad Gateway - unexpected error from the backend.
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}
```

83.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

84 Orders-orderId-account-products-productId-services-serviceId:GET

85 GET /orders/{orderId}/account/products/{productId}/services/{serviceId}

Get details about a specific service in a given product.

85.1 Contents

- 1 GET /orders/{orderId}/account/products/{productId}/services/{serviceId}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful response
 - 1.2.2.1.1 Parameters
 - 1.2.2.2 Error responses
 - ◆ 1.3 See also

85.2 Request

GET /orders/{orderId}/account/products/{productId}/services/{serviceId}

85.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product

serviceId - *string*
UUID specifying the service

85.2.2 URI Parameters

None.

85.2.3 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

85.3 Response

85.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The services for the order and product (if any) are successfully returned.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The OwnerId cookie or the Authorization header are not specified or do not permit access to the specified order.
403 Forbidden	Unauthorized: Services failed permission check	The client does not have access to get the information about the service.
404 Not Found	Resource not found	The client is trying to get the information for a service for non existing order, product or service.
502 Bad Gateway	-	Unexpected backend response.

85.3.2 Response Body

85.3.2.1 Successful response

```
{
  "status": "new",
  "name": "john.smith1",
  "links": [
    {
      "href": "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}/",
      "rel": "self"
    }
  ],
  "relatedTo": [ "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{relatedServiceId}" ],
  "associatedTo": [ "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}" ],
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/{serviceSpecificationId}",
  "billingCycle": 1,
  "quantity": 1,
  "attributes": "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes"
}
```

85.3.2.1.1 Parameters

status - *string*

Current status of the service. Allowed values are **new**, **placed**, **pending**, **open**, **suspended**, **close_pending**, **closed**, **purge_pending** and **purged**

name - *string*

The name of the specific service

links - *list*

[Hypermedia](#) for the product resource

relatedTo - *list*

A list containing other services related in some way to the current service

associatedTo - *list* (optional)

A list containing other services that are associated in some way to the service. The list may contain links to [other new services in the order](#) and/or a link to an [existing service](#) that belongs to the same account and product. In most cases the list may be empty unless otherwise stated.

serviceSpecification - *string*

Link to the product catalogue specification describing the given service

billingCycle - *integer*

Billing cycle of the service. Allowed values are from 1 to 120

quantity - *integer*

Quantity (if multiple services is allowed) of the service.

attributes - *string*

A link to the attributes of the service

85.3.2.2 Error responses

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden - the client does not have access to get the information about the service.

```
{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: Services failed permission check",
    "code": 403,
    "details": ""
  }
}
```

404 Not Found - the order, product or service doesn't exist.

```
{
  "itemNotFound": {
    "guid": "3a422918-24c3-4c20-9bee-f1c61695e987",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

502 Bad Gateway - unexpected error from the backend.

```
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}
```

85.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

86 Orders-orderId-account-products-productId-services-serviceId:PUT

87 PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}

Updates a service in order builder. Currently supports only updating the name and the billing cycle of the service.

87.1 Contents

- 1 PUT
/orders/{orderId}/account/products/{productId}/services/{serviceId}
- ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - ◇ 1.1.5 Validation Rules
- ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful response
 - 1.2.2.2 Error responses
- ◆ 1.3 See also

87.2 Request

PUT /orders/{orderId}/account/products/{productId}/services/{serviceId}

87.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product

serviceId - *string*
UUID specifying the service

87.2.2 URI Parameters

None.

87.2.3 Request Headers

Content-Type
Required. Set this header to `application/json; charset=UTF-8`

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

87.2.4 Request Body

A dictionary containing the keys "billingCycle" and "name" and their values.

```
{
  "billingCycle": 12,
  "name": "Service Name",
  "associatedTo": [ "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}", "https://api.hostway.com/ac"
]
```

87.2.5 Validation Rules

1. The service URL (`https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}`) should point to a valid service already added to the product.
2. The dictionary key/value pairs should be valid billing cycle and service name.
3. The billingCycle and name parameters are required.
4. Each service URL from the associated services list should point to a valid service already added to the product.
5. Any associated service that already exists on the account must be open.

87.3 Response

87.3.1 Status Code

Status Code	Error Message	Description
204 No Content -		The operation was successfully completed.
400 Bad Request	POST data error Unable to change service: invalid cycle	Trying to update service with empty body or with invalid body. Trying to change the billing cycle to value that is not supported by the current service.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The <i>OwnerId</i> cookie was not specified or is incorrect.

403 Forbidden Unauthorized: Service failed permission check

404 Not Found Resource not found

502 Bad Gateway

87.3.2 Response Body

87.3.2.1 Successful response

None.

87.3.2.2 Error responses

400 Bad Request - invalid or empty body is passed (required field is missing)

```
{
  "badRequest": {
    "guid": "aae865ae-5102-49d9-b1ef-8a5419cc0ce7",
    "message": "POST data error",
    "code": 400,
    "details": {"billingCycle": "Required", "name": "Required"}
  }
}
```

400 Bad Request - missing billingCycle

```
{
  "badRequest": {
    "guid": "aae865ae-5102-49d9-b1ef-8a5419cc0ce7",
    "message": "POST data error",
    "code": 400,
    "details": {"billingCycle": "Required"}
  }
}
```

400 Bad Request - missing name

```
{
  "badRequest": {
    "guid": "aae865ae-5102-49d9-b1ef-8a5419cc0ce7",
    "message": "POST data error",
    "code": 400,
    "details": {"name": "Required"}
  }
}
```

400 Bad Request - invalid billing cycle

```
{
  "computeFault": {
    "guid": "7c72847f-5138-44e8-ad93-cc80bb5ba7bc",
    "message": "Unable to change service: invalid cycle",
    "code": 400,
    "details": ""
  }
}
```

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

404 Not Found - the order or product was not found.

```
{
  "itemNotFound": {
    "guid": "e1792d8a-5b0a-43ef-adf5-34a0710c8fc5",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

502 Bad Gateway - unexpected error from the backend.

```
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fal5e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fal5e6ce"
  }
}
```

87.4 See also

- [Orders API](#)
- [Common Features](#)

The client does not have permissions to update the service.

The client is trying to update a non existing service.

Unexpected backend response.

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

88 Orders-orderId-account-products-productId-services:GET

89 GET /orders/{orderId}/account/products/{productId}/services

Get all Services in a given product.

89.1 Contents

- 1 GET /orders/{orderId}/account/products/{productId}/services
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful response
 - 1.2.2.1.1 Parameters
 - 1.2.2.2 Error responses
 - ◆ 1.3 See also

89.2 Request

GET /orders/{orderId}/account/products/{productId}/services

89.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product

89.2.2 URI Parameters

None.

89.2.3 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

89.3 Response

89.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The services for the order and product (if any) are successfully returned.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The OwnerId cookie or the Authorization header are not specified or do not permit access to the specified order.
403 Forbidden	Unauthorized: Services failed permission check	The client does not have permissions to get information about the services.
404 Not Found	Resource not found	The client is trying to get the services for non existing order or non existing product.
502 Bad Gateway	-	Unexpected backend response.

89.3.2 Response Body

89.3.2.1 Successful response

```
200 OK
{
  "list": [
    {
      "status": "new",
      "name": "john.smith1",
      "links": [
        { "href": "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}/", "rel": "self" }
      ],
      "relatedTo": [ "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}" ],
      "associatedTo": [ "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}" ],
      "serviceSpecification": "https://api.hostway.com/serviceSpecifications/{serviceSpecificationId}",
      "billingCycle": 1,
      "quantity": 1,
      "attributes": "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}/attributes"
    }
  ],
  "links": [
    { "href": "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/", "rel": "self" }
  ]
}
```

89.3.2.1.1 Parameters

- list - list*
List of the services for the given product. Each element of the list has the following attributes:
- status - string*
Current status of the service. Allowed values are **new**, **pending**, **open**, **suspended**, **close_pending**, **closed**, **purge_pending** and **purged**
 - name - string*
The name of the specific service
 - links - list*
[Hypermedia](#) for the product resource
 - relatedTo - list*
A list containing other services in the current order related in some way to the current service
 - associatedTo - list (optional)*
A list containing other services that are associated in some way to the service. The list may contain links to [other new services in the order](#) and/or a link to an [existing service](#) that belongs to the same account and product. In most cases the list may be empty unless otherwise stated.
 - serviceSpecification - string*
Link to the product catalogue specification describing the given service
 - billingCycle - integer*
Billing cycle (length of billing period in months) of the service. Allowed values are from **1** to **120**
 - quantity - integer*
Quantity (if multiple services is allowed) of the service.
 - attributes - string*
A link to the attributes of the service

links - list
[Hypermedia](#) for the list of product resources

89.3.2.2 Error responses

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden - the client does not have access to get the information about the services.

```
{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: Services failed permission check",
    "code": 403,
    "details": ""
  }
}
```

404 Not Found - the order or the product doesn't exist.

```
{
  "itemNotFound": {
    "guid": "3a422918-24c3-4c20-9bee-f1c61695e987",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

502 Bad Gateway - unexpected error from the backend.

```
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}
```

89.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

90 Orders-orderId-account-products-productId-services:POST

91 POST /orders/{orderId}/account/products/{productId}/services

Adds a new service to a product in the order.

91.1 Contents

- 1 POST /orders/{orderId}/account/products/{productId}/services
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◇ 1.1.5 Validation Rules
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Headers
 - ◇ 1.2.3 Response Body
 - 1.2.3.1 Successful response
 - 1.2.3.2 Error responses
 - ◆ 1.3 Examples
 - ◇ 1.3.1 Create a service with attributes
 - ◇ 1.3.2 Create a service with attributes and make it related to an existing service
 - ◇ 1.3.3 Create a service with attributes and make it related to another new service in the same order
 - ◇ 1.3.4 Create a service with quantity but without attributes and make it related to another new service in the same order
 - ◇ 1.3.5 Create a service and make it associated to an existing service
 - ◇ 1.3.6 Create a service and make it associated to another new service in the same order
 - ◇ 1.3.7 Create a service and make it associated to two other new services in the same order
 - ◇ 1.3.8 Create a service and make it related to another new service and associate it with a second new service in the same order
 - ◆ 1.4 See also

91.2 Request

POST /orders/{orderId}/account/products/{productId}/services

91.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*

UUID specifying the product

91.2.2 URI Parameters

None.

91.2.3 Request Headers

Content-Type

Required. Set this header to `application/json; charset=UTF-8`

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

91.2.4 Request Body

```
{
  "name": "john.smith1",
  "relatedTo": [ "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}" ],
  "associatedTo": [ "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}" ],
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/{serviceSpecificationId}",
  "promoCode": "{promoCode}"
  "billingCycle": 1,
  "quantity": 1,
  "attributes": {
    "{attribute-name}": "{attribute-value}",
    ....
  }
}
```

91.2.4.1 Parameters

name - *string*

The name of the new service

relatedTo - *list* (optional)

A list containing other services related in some way to the newly created service. The relatedTo may be omitted, but relationships between some types of services have to be established for the order to be validated or submitted. The list may contain links to [other new services in the order](#) and/or a link to an [existing service](#) that belongs to the same account. If a link to an existing service is specified, a parent-child relationship will be created between the existing service and the newly-added service.

serviceSpecification - *string*

Link to the [product catalog specification](#) describing the new service

billingCycle - *integer*

Billing cycle (length of billing period in months) of the service. Allowed values are from **1** to **120**

quantity - *integer* (optional)

Quantity of the service. Must be greater than **0**. It is **1** by default. Quantity greater than 1 only allowed for services configured to support multiple quantities when ordering.

attributes - *dictionary* (optional)

a dictionary of the new service's attributes values. Each (*attribute-name*, *attribute-value*) pair present the service attribute name and its relevant value to be set. The allowed attribute names depends on the *serviceSpecification* field value. See [Service Attributes Structure](#).

promoCode - *string* (optional)

A promotion code for the new service [promotion codes](#).

associatedTo - *list* (optional)

A list containing other services that are associated in some way to the newly created service. The list may contain links to [other new services in the order](#) and/or a link to an [existing service](#) that belongs to the same account and product. Adding a service to this list will not alter existing or create new parent-child relationships between the existing service and the newly-added service. In most cases the list may be empty unless otherwise stated.

91.2.5 Validation Rules

1. The service specification URL (<https://api.hostway.com/serviceSpecifications/{serviceSpecificationId}>) should point to an existing service.
2. Each service URL (<https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceld}>) from the related services list should point to a valid service already added to the product.
3. The related services list may contain at most one URL to an existing service (ex. <https://api.hostway.com/accounts/{accountId}/products/{productId}/services/{serviceld}>) that is open on the account.
4. The service billing cycle should be allowed for the product.
5. The product should support adding the service.
6. Each service URL from the associated services list should point to a valid service already added to the product.
7. Any associated service that already exists on the account must be open.

91.3 Response

91.3.1 Status Code

Status Code	Error Message	Description
201 Created	-	The service was successfully added to the order.
	POST data error	Trying to create a service without passing all the required attributes.
	Invalid service specification	Malformed serviceSpecification was passed or the service is not found in the backend.
	Invalid cycle	Trying to add service with wrong billingCycle.
	Invalid quantity	Trying to add service with invalid quantity input. It should be a number greater than 0.
400 Bad Request	Product does not support quantity	Trying to set quantity on service that is not configured to support multiple units.
	Invalid relatedTo service ID provided: fake.service	Trying to add a service with invalid relatedTo.
	Invalid associatedTo service ID: fake.service	Trying to add a service with invalid associatedTo.
	Product does not support this service	Trying to add service to a product, which doesn't support the service.
	Current service requires being a child service.	The service that can be added only as a child is not added as a child of another service.
	Cannot create domain alias to the same domain name.	Trying to create domain alias to the same domain name.
	Parent service doesn't exist.	Tried to add the service as a child service of an existing service that doesn't exist.
Parent service is not open.	Tried to add the service as a child service of an existing service that is not open.	
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The <i>OwnerId</i> cookie was not specified or is incorrect.
403 Forbidden	Unauthorized: Services failed permission check	The client does not have permissions to add services.
404 Not Found	Resource not found	The order or product doesn't exist.
	Domain-dependent services:	
	Cannot create domain alias to the same domain name.	Duplicate domain.
	Email domain alias service: A domain alias would produce impossible to delivery loops	
409 Conflict	Domain {domain_name} has circular forwards to alias {alias_name}	There is an email forwarder from the target domain name to a mailbox of the domain alias with the same username
		There is an email forwarder from the domain alias to a mailbox of the target domain name with the same username
502 Bad Gateway	-	Unexpected backend response.

91.3.2 Response Headers

Location

Returned with successful response. Contains the URI of the newly added service and the id of the service.

91.3.3 Response Body

91.3.3.1 Successful response

201 Created

```
{
  "serviceId": "00000140735d43640f62c50f007f000000010001",
  "url": "https://api.hostway.com/order/account/products/00000140728b013c52e249a4007f000000010001/services/00000140735d43640f62c50f007f000000010001"
}
```

Note: The service Id will change once the order is provisioned.

91.3.3.2 Error responses

400 Bad Request - request body is missing some of the required attributes

```
{
  "badRequest": {
    "guid": "23bf964e-fb93-4bd6-816e-3dcb31b82755",
    "message": "POST data error",
    "code": 400,
    "details": {
      "serviceSpecification": "Required",
      "billingCycle": "Required"
    }
  }
}
```

400 Bad Request - malformed serviceSpecification was passed or the service is not found in the backend

```
{
  "computeFault": {
    "guid": "e4443fe7-1139-4078-b407-39d868c02d26",
    "message": "Invalid service specification",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - wrong billingCycle

```
{
  "computeFault": {
    "guid": "e4443fe7-1139-4078-b407-39d868c02d26",
    "message": "Invalid cycle",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - invalid quantity

```
{
  "computeFault": {
    "guid": "e4443fe7-1139-4078-b407-39d868c02d26",
    "message": "Invalid quantity",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - unsupported quantity

```
{
  "computeFault": {
    "guid": "e4443fe7-1139-4078-b407-39d868c02d26",
    "message": "Product does not support quantity",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - invalid service was passed as relatedTo

```
{
  "computeFault": {
    "guid": "e4443fe7-1139-4078-b407-39d868c02d26",
    "message": "Invalid relatedTo service ID provided: fake.service",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - the service is not supported by the product

```
{
  "computeFault": {
    "guid": "e4443fe7-1139-4078-b407-39d868c02d26",
    "message": "Product does not support this service",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - the service that can be added only as a child was not added as a child of another service

```
{
  "computeFault": {
    "guid": "e4443fe7-1139-4078-b407-39d868c02d26",
    "message": "Current service requires being a child service.",
    "code": 400,
    "details": ""
  }
}
```

```

400 Bad Request - request body provided incorrect attributes
{
  "badRequest": {
    "guid": "5386544b-6dbf-4d21-b22a-879979070c01",
    "message": "POST data error",
    "code": 400,
    "details": {"attributes.serverName": "Required"}
  }
}

400 Bad Request - trying to create domain alias to the same domain name
{
  "computeFault": {
    "guid": "c270f3a1-331f-4736-b420-c006cbdae689",
    "message": "Cannot create domain alias to the same domain name.",
    "code": 400,
    "details": ""
  }
}

400 Bad Request - trying to add domain alias with invalid domain name
{
  "badRequest": {
    "guid": "6321f804-285a-43fb-890a-bbdbclbd8db3",
    "message": "POST data error",
    "code": 400,
    "details": {"attributes.domain": "Invalid domain name in domain: invalid", "attributes.domain_alias": "Invalid domain name in domain_"}
  }
}

400 Bad Request - trying to specify a relationship to an existing service that doesn't exist
{
  "badRequest": {
    "guid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "message": "Parent service doesn't exist.",
    "code": 400,
    "details": ""
  }
}

400 Bad Request - trying to specify a relationship to an existing service that isn't open
{
  "badRequest": {
    "guid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "message": "Parent service is not open.",
    "code": 400,
    "details": ""
  }
}

400 Bad Request - trying to specify an association to an existing service that doesn't exist
{
  "badRequest": {
    "guid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "message": "Invalid associatedTo service ID: xxxxx",
    "code": 400,
    "details": ""
  }
}

400 Bad Request - trying to specify an association to an existing service that isn't open
{
  "badRequest": {
    "guid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "message": "Associated service is not open",
    "code": 400,
    "details": ""
  }
}

401 Unauthorized
This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden - the client does not have assess to get the information about the services.
{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: Services failed permission check",
    "code": 403,
    "details": ""
  }
}

404 Not Found - the order or product was not found.
{
  "itemNotFound": {
    "guid": "e1792d8a-5b0a-43ef-adf5-34a0710c8fc5",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}

409 Conflict
{
  "conflict": {
    "guid": "685b8586-c854-4dd8-ae9d-6233fe007a86",
    "message": "Domain ng-test-hwus-drago-140723b.com has circular forwards to alias ng-test-hwus-drago-140723b-alias140814.com",
    "code": 409,
    "details": ""
  }
}

502 Bad Gateway - unexpected error from the backend.
{

```



```

"computeFault": {
  "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
  "timestamp": "2014-09-10 20:59:50",
  "code": 502,
  "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
}
}

```

91.4 Examples

91.4.1 Create a service with attributes

Request

```

{
  "name": "ng-test-hwus-test.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.dedserver",
  "billingCycle": 1,
  "attributes": {
    "serverName": "sampleServerName",
    "serverLocation": "Tampa"
  }
}

```

91.4.2 Create a service with attributes and make it related to an existing service

Request

```

{
  "name": "ng-test-hwus-test.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.dedserver",
  "billingCycle": 1,
  "attributes": {
    "serverName": "sampleServerName",
    "serverLocation": "Tampa"
  },
  "relatedTo": ["https://api.hostway.com/accounts/{accountId}/products/{productId}/services/{serviceId}"]
}

```

91.4.3 Create a service with attributes and make it related to another new service in the same order

Request

```

{
  "name": "ng-test-hwus-test.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.dedserver",
  "billingCycle": 1,
  "attributes": {
    "serverName": "sampleServerName",
    "serverLocation": "Tampa"
  },
  "relatedTo": ["https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}"]
}

```

91.4.4 Create a service with quantity but without attributes and make it related to another new service in the same order

Request

```

{
  "name": "ng-test-hwus-test.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.dedserver.hardware.memory",
  "billingCycle": 1,
  "quantity": 4,
  "relatedTo": ["https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}"]
}

```

91.4.5 Create a service and make it associated to an existing service

Request

```

{
  "name": "ng-test-hwus-test.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.dedicatedserver.hardware.disk.500gb",
  "billingCycle": 1,
  "quantity": 4,
  "associatedTo": ["https://api.hostway.com/accounts/{accountId}/products/{productId}/services/{serviceId}"]
}

```

91.4.6 Create a service and make it associated to another new service in the same order

Request

```

{
  "name": "ng-test-hwus-test.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.dedicatedserver.hardware.disk.500gb",
  "billingCycle": 1,
  "quantity": 4,
  "associatedTo": ["https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{serviceId}"]
}

```

91.4.7 Create a service and make it associated to two other new services in the same order

Request

```
{
  "name": "ng-test-hwus-test.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.dedicatedserver.hardware.diskcontroller.sataraid",
  "billingCycle": 1,
  "quantity": 4,
  "associatedTo": [
    "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{service1Id}",
    "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{service2Id}"
  ]
}
```

91.4.8 Create a service and make it related to another new service and associate it with a second new service in the same order

Request

```
{
  "name": "ng-test-hwus-test.com",
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/core.dedicatedserver.hardware.disk.500gb",
  "billingCycle": 1,
  "quantity": 4,
  "relatedTo": ["https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{service1Id}"],
  "associatedTo": ["https://api.hostway.com/orders/{orderId}/account/products/{productId}/services/{service2Id}"]
}
```

91.5 See also

- [Orders API](#)
- [Common Features](#)
- [Service Attributes Structure](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

92 Orders-orderId-account-products-productId:DELETE

93 DELETE /orders/{orderId}/account/products/{productId}

Removes a product from the account's order.

93.1 Contents

- 1 DELETE /orders/{orderId}/account/products/{productId>
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful response
 - 1.2.2.2 Error responses
 - ◆ 1.3 See also

93.2 Request

DELETE /orders/{orderId}/account/products/{productId}

93.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*

Product's unique Id.

93.2.2 URI Parameters

None.

93.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

93.3 Response

93.3.1 Status Code

Status Code	Error Message	Description
204	-	The product was successfully deleted.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The OwnerId cookie or the Authorization header are not specified or invalid.
403 Forbidden	Unauthorized: Products failed permission check	The client does not have permissions to delete products.
404 Not Found	Resource not found	The client is trying to delete a non existing product or the order does not exist.
502 Bad Gateway	-	Unexpected backend response.

93.3.2 Response Body

93.3.2.1 Successful response

None.

93.3.2.2 Error responses

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden - the client does not have permission to delete products.

```
{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: Products failed permission check",
    "code": 403,
    "details": ""
  }
}
```

404 Not Found

```
{
  "itemNotFound": {
    "guid": "579951da-5f2c-4db7-9169-25658bfaafdb",

```

```
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}

502 Bad Gateway - unexpected error from the backend.
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}
```

93.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

94 Orders-orderId-account-products-productId:GET

95 GET /orders/{orderId}/account/products/{productId}

Get details about a specific Product in an account.

95.1 Contents

- 1 GET /orders/{orderId}/account/products/{productId}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful response
 - 1.2.2.1.1 Parameters
 - 1.2.2.2 Error responses
 - ◆ 1.3 See also

95.2 Request

GET /orders/{orderId}/account/products/{productId}

95.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
Product's unique Id.

95.2.2 URI Parameters

None.

95.2.3 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

95.3 Response

95.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The operation was successfully completed.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The OwnerId cookie or the Authorization header are not specified or are invalid
403 Forbidden	Unauthorized: Products failed permission check	The client does not have permission to get the product.
404 Not Found	Resource not found	The client is trying to get a product for non existing order or is trying to get non existing product.
502 Bad Gateway	-	Unexpected backend response.

95.3.2 Response Body

95.3.2.1 Successful response

```
200 OK
{
  "status" : "open",
  "services" : "https://api.hostway.com/orders/{orderId}/account/products/{productId}/services",
  "offering" : "https://api.hostway.com/productSpecifications/{productSpecificationId}/offerings/{productOfferingId}",
  "displayName": "My custom product name",
  "partnerId": "Partner Id",
  "links" : [
    {"href": "https://api.hostway.com/orders/{orderId}/account/products/{productId}", "rel": "self" }
  ]
}
```

95.3.2.1.1 Parameters

status - *string*
Current status of the product. Allowed values are **new**, **pending**, **open**, **suspended**, **closed** and **purged**

services - *string*
A link to the services associated with the product

offering - *string*
Link to the [product catalog offering](#) specifying the given product

displayName - *string*
A custom display name for the product.

partnerId - *string*
Partner id.

links - *list*
[Hypermedia](#) for the product resource

95.3.2.2 Error responses

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden - the client does not have permission to get the product.

```
{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: Products failed permission check",
    "code": 403,
    "details": ""
  }
}
```

404 Not Found

```
{
  "itemNotFound": {
    "guid": "579951da-5f2c-4db7-9169-25658bfaafdb",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

502 Bad Gateway - unexpected error from the backend.

```
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}
```

95.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

96 Orders-orderId-account-products-productId:PUT

97 PUT /orders/{orderId}/account/products/{productId}

Updates an existing product in order builder. Currently supports only updating the billing cycle.

97.1 Contents

- 1 PUT /orders/{orderId}/account/products/{productId}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - ◇ 1.1.5 Validation Rules
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful response
 - 1.2.2.2 Error responses
 - ◆ 1.3 See also

97.2 Request

PUT /orders/{orderId}/account/products/{productId}

97.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

productId - *string*
UUID specifying the product

97.2.2 URI Parameters

None.

97.2.3 Request Headers

Content-Type
Required. Set this header to `application/json; charset=UTF-8`

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

97.2.4 Request Body

A dictionary containing the key "billingCycle" and a value that is integer.

```
{
  "billingCycle": 12
}
```

97.2.5 Validation Rules

1. The product URL (<https://api.hostway.com/orders/{orderId}/account/products/{productId}>) should point to a valid product already added to the order.
2. The billing cycle should be a valid integer between 1 and 120 and also a valid billing cycle as defined in ServicePlex.

97.3 Response

97.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The operation was successfully completed.
	POST data error	Missing billing cycle value.
	POST data error	Trying to add a billing cycle less than 1.
400 Bad Request	POST data error	Trying to update the billing cycle to value > 120.
	Unable to update the product 0000014ee39eee9c95fa4b5b007f000000010001	Trying to update the billing cycle to value that is not available for the product.

401 Unauthorized This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

The *OwnerId* cookie was not specified or is incorrect.

403 Forbidden Unauthorized: Products failed permission check

The client does not have permissions to update the product.

404 Not Found Resource not found

The client is trying to update the billing cycle for a non existing product or order.

502 Bad Gateway -

Unexpected backend response.

97.3.2 Response Body

97.3.2.1 Successful response

None.

97.3.2.2 Error responses

400 Bad Request - empty body is passed

```
{
  "badRequest": {
    "guid": "bd80dfdc-5752-4472-96f2-90f590ddb3f9",
    "message": "POST data error",
    "code": 400,
    "details": {"billingCycle": "Required"}
  }
}
```

400 Bad Request - invalid body is passed (required field is missing)

```
{
  "badRequest": {
    "guid": "aae865ae-5102-49d9-blef-8a5419cc0ce7",
    "message": "POST data error",
    "code": 400,
    "details": {"billingCycle": "Required"}
  }
}
```

400 Bad Request - invalid billing cycle is passed

```
{
  "badRequest": {
    "guid": "fdd0b796-c690-47af-9666-0478e2bcae27",
    "message": "POST data error",
    "code": 400,
    "details": {"billingCycle": "0 is less than minimum value 1"}
  }
}
```

400 Bad Request - invalid billing cycle is passed

```
{
  "computeFault": {
    "guid": "1e98f8ea-d298-4867-a9aa-0a6486dbefb0",
    "message": "POST data error",
    "code": 400,
    "details": {"billingCycle": "122 is greater than maximum value 120"}
  }
}
```

400 Bad Request - billing cycle is not available for the product

```
{
  "computeFault": {
    "guid": "1e98f8ea-d298-4867-a9aa-0a6486dbefb0",
    "message": "Unable to update the product 000014edfe57d8b9b2c78bf007f00000010001",
    "code": 400,
    "details": ""
  }
}
```

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden

```
{
  "forbidden": {
    "guid": "6af8e3e6-c3dc-4083-8389-1c1a038c62c0",
    "message": "Unauthorized: Products failed permission check",
    "code": 403,
    "details": ""
  }
}
```

404 Not Found - the order or product was not found.

```
{
  "itemNotFound": {
    "guid": "e1792d8a-5b0a-43ef-adf5-34a0710c8fc5",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

502 Bad Gateway - unexpected error from the backend.

```
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}
```

```
} }
```

97.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

98 Orders-orderId-account-products:GET

99 GET /orders/{orderId}/account/products

Get all Product in an account.

99.1 Contents

- 1 GET /orders/{orderId}/account/products
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Successful response
 - 1.2.2.1.1 Parameters
 - 1.2.2.2 Error responses
 - ◆ 1.3 See also

99.2 Request

GET /orders/{orderId}/account/products

99.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

99.2.2 URI Parameters

None.

99.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

99.3 Response

99.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The products for the order (if any) are successfully returned.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The OwnerId cookie or the Authorization header are not specified or do not permit access to the specified order.
403 Forbidden	Unauthorized: Products failed permission check	The client does not have permission to get the products.
404 Not Found	Resource not found	The client is trying to get the products for non existing order.
502 Bad Gateway	-	Unexpected backend response.

99.3.2 Response Body

99.3.2.1 Successful response

```
200 OK
{
  "list": [
    {
      "status": "open",
      "services": "https://api.hostway.com/orders/{orderId}/account/products/{orderProductId}/services",
      "offering": "https://api.hostway.com/productSpecifications/{productSpecificationId}/offerings/{productOfferingId}",
      "displayName": "My custom product name",
      "partnerId": "PartnerId",
      "links": [
        { "href": "https://api.hostway.com/orders/{orderId}/account/products/{orderProductId}", "rel": "self" }
      ]
    }
  ]
}
```

```

    }
  ],
  "links": [
    {
      "href": "https://api.hostway.com/orders/{orderId}/account/products/",
      "rel": "self"
    }
  ]
}

```

99.3.2.1.1 Parameters

`list` - *list*

List of the products for the given account order. Each element of the list has the following attributes:

`status` - *string*

Current status of the product. Allowed values are **new**, **pending**, **open**, **suspended**, **closed** and **purged**

`services` - *string*

A link to the services associated with the product

`displayName` - *string*

A custom display name for the product.

`partnerId` - *string*

Partner id.

`links` - *list*

[Hypermedia](#) for the product resource

`offering` - *string*

Link to the product catalogue offering specifying the given product

`links` - *list*

[Hypermedia](#) for the list of product resources

99.3.2.2 Error responses

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

403 Forbidden - the client does not have permission to get the products.

```

{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: Products failed permission check",
    "code": 403,
    "details": ""
  }
}

```

404 Not Found - there is no such order.

```

{
  "itemNotFound": {
    "guid": "579951da-5f2c-4db7-9169-25658bfaafdb",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}

```

502 Bad Gateway - unexpected error from the backend.

```

{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}

```

99.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

100 Orders-orderId-account-products:POST

101 POST /orders/{orderId}/account/products

Adds a new or existing Product to the Account's Order.

If, at the time the product is created, there is not yet an Account in the order, then this call will implicitly add a very rudimentary Account (containing only a default language & the currency of this product).

If there is already an Account in the order at the time this call is made, the currency of the specified product offering *must* match the currency of the Account.

101.1 Contents

- 1 POST /orders/{orderId}/account/products
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◇ 1.1.5 Validation Rules
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Headers
 - ◇ 1.2.3 Response Body
 - 1.2.3.1 Successful response
 - 1.2.3.2 Error responses
 - ◆ 1.3 See also

101.2 Request

POST /orders/{orderId}/account/products

101.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

101.2.2 URI Parameters

None.

101.2.3 Request Headers

Content-Type

Required. Set this header to `application/json; charset=UTF-8`

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

101.2.4 Request Body

There are two possible formats. To add a new product that does not already exist:

```
{
  "offering": "https://api.hostway.com/productSpecifications/{productSpecificationId}/offerings/{productOfferingId}",
  "promoCode": "{promoCode}",
  "displayName": "My custom product name",
  "partnerId": "PartnerId",
}
```

OR

To specify an existing product, use the following:

```
{
  "existingProductUrl": "https://api.hostway.com/accounts/{accountId}/products/{productId}"
}
```

}

101.2.4.1 Parameters

offering - *string*

Link to the [product catalog offering](#) specifying the new product.

promoCode - *string* (optional)

A promotion code for the new product [promotion codes](#).

displayName - *string* (optional)

A custom display name for the product.

partnerId - *string* (optional)

Partner id.

existingProductUrl - *string*

Link to an [existing product](#). Only required when adding an existing product to the order.

101.2.5 Validation Rules

1. The product specification URL (<https://api.hostway.com/productSpecifications/{productSpecificationId}>) should point to an existing product.
2. The product offering currency should match the account currency or if no account is specified should be allowed for the franchise.
3. The product offering URL (<https://api.hostway.com/productSpecifications/{productSpecificationId}/offerings/{productOfferingId}>) should point to an existing product offering.
4. The product use type (public, internal, internal view only, etc.) should be allowed for the current user.
5. The product offering should be valid according to the back-end.
6. The account product URL (<{APIBaseURL}/accounts/{accountId}/products/{productId}>) should point to an existing product that is open and belongs to the account.

101.3 Response

101.3.1 Status Code

Status Code	Error Message	Description
201 Created	-	The operation was successfully completed.
	POST data error	Trying to create a product with invalid body.
	Cannot add non-existent product "fake"	Trying to create a product by passing fake offering.
	Invalid offering URI	Trying to create a product by passing invalid offering (missing currency, monthly cycle or productSpecifications)
	The requested offering is invalid	Trying to create a product with invalid monthly cycle (eg: 100)
400 Bad Request	Trying to add inconsistent currency: product {product} with FAKE while account uses USD	Trying to create a product with currency, which does not match the account's currency
	Currency FAKE not allowed for this franchise	Trying to add a product with invalid for the franchise currency to an order for which no account has been set.
	Trying to add Internal product {product} without authorization	Trying to add product, which is internal, without the proper credentials.
	Trying to add Internal View Only product {product}	Trying to add product, which is internal view only, without the proper credentials.
	Invalid product URI	Tried to add an existing product with wrong existingProductUrl
	The account of the order and the account of the product doesn't match	Tried to add an existing product with account, which is different from the account of the order.
	Product doesn't exist.	Tried to add an existing product which does not exist
	Cannot add a product that is not open.	Tried to add an existing product that is not open.
401 Unauthorized	This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.	The <i>OwnerId</i> cookie was not specified or is incorrect.
403 Forbidden	Unauthorized: Products failed permission check	The client does not have permissions to add products.
404 Not Found	Resource not found	The client is trying to add a product to non existing order.
502 Bad Gateway	-	Unexpected backend response.

101.3.2 Response Headers

Location

Returned with successful response. Contains the URI of the newly added product and the id of the product.

101.3.3 Response Body

101.3.3.1 Successful response

```

201 Created
{
  "productId": "00000148842a5f6448a61ce6007f000000010001",
  "url" : "https://api.hostway.com/orders/0000014884293798c68eafa5007f000000010001/account/products/00000148842a5f6448a61ce6007f000000010001/"
}

```

Note: The product Id will change once the order is provisioned.

101.3.3.2 Error responses

400 Bad Request - trying to order product, which is internal, without the proper authorization.

```
{
  "computeFault": {
    "guid": "c940c886-5343-4a49-b728-0b94cbb1a3e9",
    "message": "Trying to add Internal product {product} without authorization",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - trying to add a product, which is internal view only, such products cannot be added via the Public API.

```
{
  "computeFault": {
    "guid": "3ee2bd37-81ee-4a57-bd63-dc84705b3985",
    "message": "Trying to add Internal View Only product {product}",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - trying to add a product without supplying the required attributes in the body.

```
{
  "badRequest": {
    "guid": "0f3e0435-21ac-43e5-acfd-ac33886fbe70",
    "message": "POST data error",
    "code": 400,
    "details": {
      "offering": "Required"
    }
  }
}
```

400 Bad Request - trying to add a product that does not exist.

```
{
  "computeFault": {
    "guid": "12cf7a72-3261-4c57-a82e-1f2f1c124d50",
    "message": "Cannot add non-existent product \"fake\"",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - trying to add a product by supplying invalid offering URI (eg: "offering": "http://coreapi01.ote.chicago.hostway:8092/prod

```
{
  "computeFault": {
    "guid": "0ced627e-c9fc-4835-92da-1a4256165607",
    "message": "Invalid offering URI",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - trying to add a product with invalid currency (case when account is set and the currency provided does not match the account

```
{
  "computeFault": {
    "guid": "742e193a-4667-47fd-a5ff-ec1f7788d459",
    "message": "Trying to add inconsistent currency: product {product} with FAKE while account uses USD",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - trying to add a product with invalid for the franchise currency (case when an account is not set so the currency it is determined

```
{
  "computeFault": {
    "guid": "3ee2bd37-81ee-4a57-bd63-dc84705b3985",
    "message": "Currency FAKE not allowed for this franchise",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - trying to add a product with invalid month cycle (eg: 100)

```
{
  "computeFault": {
    "guid": "8165a0b2-80f5-436f-ab47-075d824144d9",
    "message": "The requested offering is invalid",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - trying to add an existing product with invalid existingProductUri

```
{
  "computeFault": {
    "guid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "message": "Invalid product URI",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - trying to add an existing product that belongs to account different from the account of the order.

```
{
  "computeFault": {
    "guid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "message": "The account of the order and the account of the product doesn't match",
    "code": 400,
    "details": ""
  }
}
```

400 Bad Request - trying to add an existing product that does not exist

```
{
  "computeFault": {
```

```

    "guid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "message": "Product doesn't exist.",
    "code": 400,
    "details": ""
  }
}
400 Bad Request - trying to add an existing product that is not open
{
  "computeFault": {
    "guid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "message": "Cannot add a product that is not open.",
    "code": 400,
    "details": ""
  }
}
401 Unauthorized
This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b
403 Forbidden - the client does not have permission to add products.
{
  "forbidden": {
    "guid": "d336dd8e-bc52-44f3-b672-9aa2fac7fbf4",
    "message": "Unauthorized: Products failed permission check",
    "code": 403,
    "details": ""
  }
}
404 Not Found - the order does not exist
{
  "itemNotFound": {
    "guid": "e1792d8a-5b0a-43ef-adf5-34a0710c8fc5",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
502 Bad Gateway - unexpected error from the backend.
{
  "computeFault": {
    "guid": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce",
    "timestamp": "2014-09-10 20:59:50",
    "code": 502,
    "errorRefId": "8ced5fe7-f6a9-4e8f-8700-cfc9fa15e6ce"
  }
}

```

101.4 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

102 Orders-orderId-account-user:GET

103 GET /orders/{orderId}/account/user

Get details of the account user for the given order. A user with auto-generated information is created by default when the account is created. The user password is excluded from the response.

103.1 Contents

- 1 GET /orders/{orderId}/account/user
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 No authorization
 - ◇ 1.3.3 Unauthorized user
 - ◇ 1.3.4 Order does not exist
 - ◇ 1.3.5 Account has not been set
 - ◆ 1.4 See also

103.2 Request

GET /orders/{orderId}/account/user

103.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

103.2.2 URI Parameters

None.

103.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

103.3 Response

103.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The operation was successfully completed.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to set user credentials.
403 Forbidden	Unauthorized: User failed permission check	Retrieve user as an authenticated user, not associated with the order.
404 Not Found	Resource not found	The specified order does not exist.
		An account was not set for the order.
		No user data found for the account.

103.3.2 Response Body

```
{
  "userName": "{userName}",
  "links": [
    { "href": "https://api.hostway.com/orders/{order_id}/account/user/", "rel": "self" }
  ]
}
```

103.3.2.1 Parameters

userName - *string*

The username of the account login credentials.

links - *list*

[Hypermedia](#) for this resource. Links to self.

103.4 Example

103.4.1 Success

Request

```
GET /orders/{orderId}/account/user
Cookie: OwnerId={owner_id}; Path=
```

Response

```
200 OK
Content-Type : application/json; charset=UTF-8
```

```
{
  "userName": "accountUser1",
  "links": [{
    "href": "https://api.hostway.com/orders/0000014854c737a28b6dffbe007f000000010001/account/user/",
    "rel": "self"
  }]
}
```

103.4.2 No authorization

Request

```
GET /orders/{orderId}/account/user
Cookie: OwnerId={wrong_owner_id}; Path=
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

103.4.3 Unauthorized user

Request

```
GET /orders/{orderId}/account/user
Authorization: {unassociated_user_authorization}
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8
```

```
{
  "forbidden" : {
    "guid" : "66f63a81-52e9-4ec6-a50a-42acebc021e7",
    "message" : "Unauthorized: User failed permission check",
    "code" : 403,
    "details" : ""
  }
}
```

103.4.4 Order does not exist

Request

```
GET /orders/{non-existent_orderId}/account/user
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "b0b38b3e-b693-4021-8116-b59f7ec3fea7",
    "message": "Resource not found",
  }
}
```

```
"code": 404,  
"details": ""  
}  
}
```

103.4.5 Account has not been set

Request

```
GET /orders/{orderId}/account/user  
Cookie: OwnerId={owner_id}; Path=/  

```

Response

```
404 Not found  
Content-Type : application/json; charset=UTF-8
```

```
{  
  "itemNotFound": {  
    "guid": "0c0c1f50-8d80-4423-b501-0a3cc5f070af",  
    "message": "",  
    "code": 404,  
    "details": ""  
  }  
}
```

103.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

104 Orders-orderId-account-user:PUT

105 PUT /orders/{orderId}/account/user

Sets or updates the account user credentials in a given order.

105.1 Contents

- 1 PUT /orders/{orderId}/account/user
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - ◆ 1.3 Example
 - ◇ 1.3.1 Successfully add/update user to order
 - ◇ 1.3.2 Missing username
 - ◇ 1.3.3 Invalid username
 - ◇ 1.3.4 Invalid password
 - ◇ 1.3.5 No authorization
 - ◇ 1.3.6 Unauthorized user
 - ◇ 1.3.7 Order does not exist
 - ◇ 1.3.8 Existing account in the system
 - ◇ 1.3.9 Account has not been set
 - ◆ 1.4 See also

105.2 Request

PUT /orders/{orderId}/account/user

105.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

105.2.2 URI Parameters

None.

105.2.3 Request Headers

Content-Type
Required. Set this header to `application/json; charset=UTF-8`

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

105.2.4 Request Body

```
{
  "userName": "mytestuser1",
  "password": "Mytestpass1"
}
```

105.2.4.1 Parameters

userName - *string*
Required string max 50 characters. Only letters, digits and dot (.), dash (-), underscore (_) and at sign (@) are allowed.

password - *string*

Required string min 8, max 50 characters. Must include at least one letter and one digit. The following characters are not allowed: &`"'\<>\$.

105.3 Response

105.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The operation was successfully completed.
400 Bad Request	POST data error	JSON is invalid Username or password is missing. Username or password breaks its format rules.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to set user credentials.
403 Forbidden	Unauthorized: User failed permission check	Set user credentials as an authenticated user, not associated with the order.
404 Not Found	Resource not found	Order does not exist.
409 Conflict	Data conflict	The same username already exists in the system.
502 Bad Gateway	-	Account hasn't been added

105.3.2 Response Body

None.

105.4 Example

105.4.1 Successfully add/update user to order

Request

```
PUT /orders/{orderId}/account/user
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "userName": "accountUser1",
  "password": "userPass1"
}
```

Response

204 No Content

105.4.2 Missing username

Request

```
PUT /orders/{orderId}/account/user
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "password": "userPass1"
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "badRequest": {
    "guid": "50f051a3-8325-4064-b174-d82d78da83d9",
    "message": "POST data error",
    "code": 400,
    "details": {
      "userName": "Required"
    }
  }
}
```

105.4.3 Invalid username

Request

```
PUT /orders/{orderId}/account/user
Cookie: OwnerId={owner_id}; Path=
```

```
{
  "userName": "user:name"
  "password": "userPass1"
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "badRequest": {
    "guid": "50f051a3-8325-4064-b174-d82d78da83d9",
    "message": "POST data error",
    "code": 400,
    "details": {
      "userName": "Only letters, digits and dot (.), dash (-), underscore (_) and at sign (@) are allowed"
    }
  }
}
```

```
}  
}  
}
```

105.4.4 Invalid password

Request

```
PUT /orders/{orderId}/account/user  
Cookie: OwnerId={owner_id}; Path=/  

```

```
{  
  "userName": "accountUser1"  
  "password": "userPass$$$"  
}
```

Response

```
400 Bad Request  
Content-Type : application/json; charset=UTF-8
```

```
{  
  "badRequest": {  
    "guid": "9cdfae0e-ad2a-4759-a5ef-ceddd2ac14be",  
    "message": "POST data error",  
    "code": 400,  
    "details": {  
      "password": "The following characters are not allowed: &`'\\"/>  
    }  
  }  
}
```

105.4.5 No authorization

Request

```
PUT /orders/{orderId}/account/user  
Cookie: OwnerId={wrong_owner_id}; Path=/  

```

```
{  
  "userName": "accountUser1",  
  "password": "userPass1"  
}
```

Response

```
401 Unauthorized  
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

105.4.6 Unauthorized user

Request

```
PUT /orders/{orderId}/account/user  
Authorization: {unassociated_user_authorization}
```

```
{  
  "userName": "accountUser1",  
  "password": "userPass1"  
}
```

Response

```
403 Forbidden  
Content-Type: application/json; charset=UTF-8
```

```
{  
  "forbidden" : {  
    "guid" : "66f63a81-52e9-4ec6-a50a-42acebc021e7",  
    "message" : "Unauthorized: User failed permission check",  
    "code" : 403,  
    "details" : ""  
  }  
}
```

105.4.7 Order does not exist

Request

```
PUT /orders/{non-existent_orderId}/account/user  
Cookie: OwnerId={owner_id}; Path=/  

```

Response

```
404 Not found  
Content-Type : application/json; charset=UTF-8
```

```
{  
  "itemNotFound": {  
    "guid": "cca219c5-3e2b-481d-8e7f-7083064872ff",  
    "message": "Resource not found",  
    "code": 404,  
    "details": ""  
  }  
}
```

```
}
```

105.4.8 Existing account in the system

Request

```
PUT /orders/{orderId}/account/user  
Cookie: OwnerId={owner_id}; Path=/  
  
{  
  "userName": "zxc01"  
  "password": "Mytestpass1"  
}
```

Response

```
409 Conflict  
Content-Type : application/json; charset=UTF-8  
  
{  
  "conflict": {  
    "guid": "5c817267-5302-4ea6-abd7-bc9b77f01082",  
    "message": "Data conflict",  
    "code": 409,  
    "details": ""  
  }  
}
```

105.4.9 Account has not been set

Request

```
PUT /orders/{orderId}/account/user  
Cookie: OwnerId={owner_id}; Path=/  
  
{  
  "userName": "accountUser1"  
  "password": "userPass1"  
}
```

Response

```
502 Bad Gateway  
Content-Type : application/json; charset=UTF-8  
  
{  
  "computeFault": {  
    "guid": "a4f6b310-11d9-4263-8482-81bccf261df7",  
    "timestamp": "2014-09-08 08:34:20",  
    "code": 502,  
    "errorRefId":  
    "a4f6b310-11d9-4263-8482-81bccf261df7"  
  }  
}
```

105.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

106 Orders-orderId-account:GET

107 GET /orders/{orderId}/account

Get details of an Account in a given Order. It must be **added** to an order before details can be retrieved.

107.1 Contents

- 1 GET /orders/{orderId}/account
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Wrong authorization
 - ◇ 1.3.2 Unassociated user without cookie
 - ◇ 1.3.3 Account has not been set
 - ◇ 1.3.4 Order does not exist
 - ◆ 1.4 See also

107.2 Request

GET /orders/{orderId}/account

107.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the **POST /orders** call.

107.2.2 URI Parameters

None.

107.2.3 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

107.3 Response

107.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The operation was successfully completed.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to perform add account for this order.
403 Forbidden	Unauthorized: OrderAccount failed permission check	Access order account as an authenticated user, not associated with the order.
404 Not Found	Resource not found	An account has not been added to the given order. The order does not exist.

107.3.2 Response Body

```
{
  "status": "{status}",
  "paymentInfo": "https://api.hostway.com/orders/{orderId}/account/paymentInfo/",
  "links": [
    {"href": "https://api.hostway.com/orders/{orderId}/account/", "rel": "self"}
  ]
}
```

```

"language": "{language}",
"number": "{accountId}",
"currency": "{currency}",
"products": "https://api.hostway.com/orders/{orderId}/account/products/",
"user": "https://api.hostway.com/orders/{orderId}/account/user/",
"attributes": "https://api.hostway.com/orders/{orderId}/account/attributes/",
"contactInfo": "https://api.hostway.com/orders/{orderId}/account/contactInfo/",
}

```

107.3.2.1 Parameters

- currency** - *string*
An ISO-4217[1] currency code assigned to the account.
- language** - *string*
An IETF[2] language tag (ex. en, en-CA, en-US, etc.) specifying account's language.
- number** - *string*
The account's Id.
- status** - *string*
The account's current status. Accounts that have not been provisioned will have a status of "new".
- attributes** - *string*
A link to the account's [attributes](#).
- contactInfo** - *string*
A link to the account's [contact info](#).
- paymentInfo** - *string*
A link to the account's [payment information](#).
- user** - *string*
A link to the account's [user](#).
- products** - *string*
A link to the account's [products](#).
- links** - *list*
[Hypermedia](#) for this resource.

107.4 Example

107.4.1 Wrong authorization

Request

```

GET /orders/{orderId}/account
Cookie: OwnerId={wrong_owner_id}; Path=/

```

Response

```

401 Unauthorized
Content-Type : text/plain; charset=UTF-8

401 Unauthorized

```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

107.4.2 Unassociated user without cookie

Request

```

GET /orders/{orderId}/account
Authorization: {unassociated_user_authorization}

```

Response

```

403 Forbidden
Content-Type: application/json; charset=UTF-8

{
  "forbidden" : {
    "guid" : "95646dcf-55cd-46cb-9bef-e1ef5d06f6c8",
    "message" : "Unauthorized: OrderAccount failed permission check",
    "code" : 403,
    "details" : ""
  }
}

```

107.4.3 Account has not been set

Request

```

GET /orders/{orderId}/account
Cookie: OwnerId={owner_id}; Path=/

```

Response


```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "9c2bf07d-4e74-4434-9d3e-77d57408d839",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

107.4.4 Order does not exist

Request

```
GET /orders/{non-existent_orderId}/account
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "402282af-9f53-4973-8437-469087465843",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

107.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

108 Orders-orderId-account:PUT

109 PUT /orders/{orderId}/account

Assign an Account to an Order.

109.1 Contents

- 1 PUT /orders/{orderId}/account
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Request Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - ◆ 1.3 Example
 - ◇ 1.3.1 The currency is not supported by the franchise
 - ◇ 1.3.2 Missing currency
 - ◇ 1.3.3 Missing currency and language
 - ◇ 1.3.4 No authorization
 - ◇ 1.3.5 Unassociated user without cookie
 - ◇ 1.3.6 A test partner agent assigns a non-test account
 - ◇ 1.3.7 Inconsistent currency
 - ◆ 1.4 See also

109.2 Request

PUT /orders/{orderId}/account

109.2.1 Request Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

109.2.2 URI Parameters

None.

109.2.3 Request Headers

Content-Type
Required. Set this header to `application/json; charset=UTF-8`

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

109.2.4 Request Body

Request Body for adding a new Account to the Order

```
{
  "currency": "{currency}",
  "language": "{language}"
}
```

Request Body for adding an existing Account to the Order

```
{
  "accountNumber": "{accountNumber}"
}
```

109.2.4.1 Request Parameters

currency - *string*
An ISO-4217[1] currency code assigned to the account.

language - *string*
An IETF[2] language tag (ex. en, en-CA, en-US, etc.) specifying account's language.

accountNumber - *string*
The unique account number for an existing account.

109.3 Response

109.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The account was successfully set.
400 Bad Request	POST data error Currency {unsupported_currency} not allowed for this franchise Language {unsupported_language} not allowed for this franchise	The request body format is invalid. Missing language or currency. The currency is not supported by the franchise. The language is not supported by the franchise.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to perform add account for this order.
403 Forbidden	Trying to add a non-test account with invalid credentials: test partner agents cannot add non-test accounts Unauthorized: OrderAccount failed permission check	-
409 Conflict	Trying to add an account with inconsistent currency: the order has already products with {product_currency} while the account you are adding uses {account_currency}	Attempt to set currency that is different from currencies of the products or that of the existing account.
502 Bad Gateway	-	Unexpected backend response.

109.3.2 Response Body

None.

109.4 Example

109.4.1 The currency is not supported by the franchise

Request

```
PUT /orders/{orderId}/account
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "currency": "BGN",
  "language": "en-US"
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "computeFault" : {
    "guid" : "ee39580b-8c74-4c0a-aa11-4de5864d8e51",
    "message" : "Currency BGN not allowed for this franchise",
    "code" : 400,
    "details" : ""
  }
}
```

109.4.2 Missing currency

Request

```
PUT /orders/{orderId}/account
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "language": "en-US"
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "badRequest": {
    "guid": "6f96c728-e6b6-4ca3-bd97-f931d2cd326c",
    "message": "POST data error",
    "code": 400, "details": {
      "currency": "Required"
    }
  }
}
```

109.4.3 Missing currency and language

Request

```
PUT /orders/{orderId}/account
Cookie: OwnerId={owner_id}; Path=/
```

```
{ }
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "badRequest" : {
    "guid" : "f4b6f4b6-fa15-4649-9c31-ec9b7b8150db",
    "message" : "POST data error",
    "code" : 400,
    "details" : {
      "accountNumber" : "Either accountNumber or both currency AND language should be provided"
    }
  }
}
```

109.4.4 No authorization

Request

```
PUT /orders/{orderId}/account
```

```
{
  "currency": "USD",
  "language": "en-US"
}
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

109.4.5 Unassociated user without cookie

Request

```
PUT /orders/{orderId}/account
Authorization: {unassociated_user_authorization}
```

```
{
  "accountNumber": "smith909090"
}
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8
```

```
{
  "forbidden" : {
    "guid" : "f9583d13-1ad3-4ea9-bf26-a5315af5f234",
    "message" : "Unauthorized: OrderAccount failed permission check",
    "code" : 403,
    "details" : ""
  }
}
```

109.4.6 A test partner agent assigns a non-test account

Request

```
PUT /orders/{orderId}/account
Authorization: {test_partner_agent_authorization}
Content-Type : application/json; charset=UTF-8
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "accountNumber": "smith909090"
}
```

Response

```
403 Forbidden
Content-Type : application/json; charset=UTF-8
```

```
{
  "forbidden" : {
    "guid" : "7a7d4ce6-241c-44a2-a2e3-cefb26f33a69",
    "message" : "Trying to add a non-test account with invalid credentials: test partner agents cannot add non-test accounts",
    "code" : 403,
    "details" : ""
  }
}
```

109.4.7 Inconsistent currency

Request

```
PUT /orders/{orderId}/account
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "currency": "CAD",
  "language": "en-US"
}
```

Response

409 Conflict

Content-Type : application/json; charset=UTF-8

```
{
  "conflict" : {
    "guid" : "8ec26107-a368-4809-8590-3523d9178893",
    "message" : "Trying to add an account with inconsistent currency: the order has already products with USD while the account you are adding has products with EUR",
    "code" : 409,
    "details" : ""
  }
}
```

109.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

110 Orders-orderId-action:POST

111 POST /orders/{orderId}/action

Performs an action like validating or submitting an order.

111.1 Contents

- 1 POST /orders/{orderId}/action
 - ◆ 1.1 Request
 - ◇ 1.1.1 Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◇ 1.1.5 Notes
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Error Response
 - ◆ 1.3 Examples
 - ◇ 1.3.1 Submit an order
 - ◇ 1.3.2 Validate an order
 - ◇ 1.3.3 Submit an order with invalid quote
 - ◇ 1.3.4 Order without contact info
 - ◇ 1.3.5 Submit an order before a service is added
 - ◇ 1.3.6 Order without domain attribute
 - ◇ 1.3.7 Unassociated user without cookie
 - ◇ 1.3.8 Order containing a domain with blacklisted word
 - ◇ 1.3.9 Domain associated with another account
 - ◆ 1.4 OPTIONS /orders/{orderId}/action
 - ◇ 1.4.1 Parameters
 - ◇ 1.4.2 Request Headers
 - ◇ 1.4.3 Response Status Code
 - ◇ 1.4.4 Response Headers
 - ◆ 1.5 See also

111.2 Request

POST /orders/{orderId}/action

111.2.1 Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

111.2.2 URI Parameters

None.

111.2.3 Request Headers

Content-Type

Required. Set this header to `application/json; charset=UTF-8`

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

111.2.4 Request Body

```
{
  "action": "{action}",
  "actionData": {actionData}
  "quoteHash": {quoteId}
}
```

111.2.4.1 Parameters

action - *string*

The action that should be performed. Allowed values are:

validate - checks the integrity of the order.

submit - submits the order for processing.

actionData - *dictionary*

Optional. Reserved for data specific for the requested action.

quoteHash - *string*

Optional. Specifies the Id of the previously obtained quote to be used with the action.

111.2.5 Notes

If a sales agent authorization is provided, the submitted order will be assigned an agent.

111.3 Response

111.3.1 Status Code

Status Code	Error Message	Description
202 Accepted	-	The requested action was initiated successfully (in case of "submit" action). Check the status of the order for future updates.
204 No Content	-	The requested action was successfully completed (in case of "validate" action)
	Account has not been set	Cannot check contact info because no account (new or existing) has been set in the order.
		Cannot add test attribute because account has not been set.
400 Bad Request	No {attribute_name} attribute set.	A required account attribute has not been set.
	Contact info type {contact_type} is not set	A required contact info type has not been set.
	All plans in the order must contain at least one product.	A product does not contain any service.
	Missing plugin data for service type(s): set([u'core.www'])	Service is missing attribute domain .
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to perform actions on this order.
403 Forbidden	Unauthorized: Action failed permission check	An unassociated user cannot validate or submit order.
404 Not Found	Resource not found	The specified order does not exist.
		Domain-dependent services:
	Cannot create domain alias to the same domain name.	Duplicate domain.
	Domain {domain_name} is associated with another account	One of the domains to be ordered is already owned by another account
409 Conflict	A domain in the order contains a restricted word	One of the domains contains blacklisted word.
	Contains a blacklisted domain	One of the domains to be ordered is blacklisted.
	Transfer is already in progress for this domain	A transfer is in progress for a domain.
		Email domain alias service: A domain alias would produce impossible to delivery loops
	Domain {domain_name} has circular forwards to alias {alias_name}	There is an email forwarder from the target domain name to a mailbox of the domain alias with the same username
		There is an email forwarder from the domain alias to a mailbox of the target domain name with the same username
502 Bad Gateway	-	Unexpected backend response. Cannot submit or validate order.

111.3.2 Response Body

Depends on the result.

111.3.2.1 Error Response

```
{
  "{errorType}": {
    "message": "{errorMessage}",
    "code": {errorCode},
    "details": "{errorDetails}"
  }
}
```

111.4 Examples

111.4.1 Submit an order

Request

```
POST /orders/0000013ca61880fbfc068b33000a02c9d8ef861c/action
Content-Type:application/json; Charset=UTF-8
Cookie: OwnerId=948ce0d4-7cf0-43b6-88a5-2b3f1331b689; Path=/
```

```
{
  "action": "submit"
}
```

Response

```
202 Accepted
```

111.4.2 Validate an order

Request

```
POST /orders/0000013ca61880fbfc068b33000a02c9d8ef861c/action
Content-Type:application/json; Charset=UTF-8
Cookie: OwnerId=948ce0d4-7cf0-43b6-88a5-2b3f1331b689; Path=/
```

```
{
  "action": "validate"
}
```

Response

204 No Content

111.4.3 Submit an order with invalid quote

Request

```
POST /orders/0000013ca61880fbfc068b33000a02c9d8ef861c/action
Content-Type:application/json; Charset=UTF-8
Cookie: OwnerId=948ce0d4-7cf0-43b6-88a5-2b3f1331b689; Path=/
```

```
{
  "action" : "submit",
  "quoteHash" : "invalid_quote_id"
}
```

Response

400 Bad Request

```
{
  "computeFault": {
    "message": "Account has not been set",
    "code": 400,
    "details": ""
  }
}
```

111.4.4 Order without contact info

Request

```
POST /orders/{orderId}/action
Content-Type:application/json; Charset=UTF-8
Cookie: OwnerId={ownerId}; Path=/
```

```
{
  "action" : "submit",
  "quoteHash" : "{quoteId}"
}
```

Response

400 Bad Request

```
{
  "computeFault": {
    "guid": "32adf6a1-e710-4362-83c7-7743c7097db9",
    "message": "Contact info type \"regular\" is not set",
    "code": 400,
    "details": ""
  }
}
```

111.4.5 Submit an order before a service is added

Request

```
POST /orders/{orderId}/action
Content-Type:application/json; Charset=UTF-8
Cookie: OwnerId={ownerId}; Path=/
```

```
{
  "action" : "submit",
  "quoteHash" : "{quoteId}"
}
```

Response

400 Bad Request

```
{
  "computeFault": {
    "guid": "7ef492af-3ebe-4334-9e37-5b0365977ef2",
    "message": "All plans in the order must contain at least one product.",
    "code": 400,
    "details": ""
  }
}
```

111.4.6 Order without domain attribute

Request

```
POST /orders/{orderId}/action
Content-Type:application/json; Charset=UTF-8
Cookie: OwnerId={ownerId}; Path=/
```

```
{
  "action" : "submit",
  "quoteHash" : "{quoteId}"
}
```

```
}
```

Response

400 Bad Request

```
{
  "computeFault": {
    "guid": "bc2e3c0c-b4da-43e2-b576-7801180bfb6e",
    "message": "Missing plugin data for service type(s): set ([u'core.www'])",
    "code": 400,
    "details": ""
  }
}
```

111.4.7 Unassociated user without cookie

Request

```
POST /orders/000001483b9336c4be0a0c89000a19010d347ea2/action
Authorization: {unassociated_user_authorization}
Content-Type: application/json; charset=UTF-8
```

```
{
  "action": "validate"
}
```

Response

403 Forbidden
Content-Type: application/json; charset=UTF-8

```
{
  "forbidden": {
    "guid": "4d1e97e2-1497-48d5-b7a3-b3726656c4b5",
    "message": "Unauthorized: Order failed permission check",
    "code": 403,
    "details": ""
  }
}
```

111.4.8 Order containing a domain with blacklisted word

Request

```
POST /orders/{orderId}/action
Content-Type:application/json; Charset=UTF-8
Cookie: OwnerId={ownerId}; Path=/
```

```
{
  "action" : "submit",
  "quoteHash" : "{quoteId}"
}
```

Response

409 Conflict

```
{
  "conflict": {
    "guid": "bfb2988f-d745-4b2a-9eff-f2deed3451db",
    "message": "A domain in the order contains a restricted word",
    "code": 409,
    "details": ""
  }
}
```

111.4.9 Domain associated with another account

Request

```
POST /orders/{orderId}/action
Content-Type:application/json; Charset=UTF-8
Cookie: OwnerId={ownerId}; Path=/
```

```
{
  "action" : "submit",
  "quoteHash" : "{quoteId}"
}
```

Response

409 Conflict

```
{
  "conflict": {
    "guid": "3fcd11ec-ffb9-47be-b14e-53a74ecb7de2",
    "message": "Domain ng-test-hwus-petre-140718a.com is associated with another account",
    "code": 409,
    "details": ""
  }
}
```

111.5 OPTIONS /orders/{orderId}/action

In addition to the POST method the API supports the OPTIONS method for direct browser access to the API. The format of the request is:

```
OPTIONS /orders/{orderId}/action
```

111.5.1 Parameters

orderId - *string*

The unique order Id. The full URI of each order is returned by the [POST /orders](#) call.

111.5.2 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

111.5.3 Response Status Code

200 OK

Success

404 Not Found

The specified order does not exist.

111.5.4 Response Headers

Access-Control-Allow-Methods

A comma-separated list of request methods supported on this URI.

Access-Control-Allow-Origin

All origins are allowed. This header has value `*`.

111.6 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

112 Orders-orderId-attributes:GET

113 GET /orders/{orderId}/attributes

Get the list of attributes for an order.

113.1 Contents

- 1 GET /orders/{orderId}/attributes
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Successfully retrieve order attributes
 - ◇ 1.3.2 Wrong authorization
 - ◇ 1.3.3 Unassociated user without cookie
 - ◇ 1.3.4 Order does not exist
 - ◇ 1.3.5 No attributes set
 - ◆ 1.4 See also

113.2 Request

GET /orders/{orderId}/attributes

113.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

113.2.2 URI Parameters

None.

113.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

113.3 Response

113.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	Successfully retrieved order attributes.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to retrieve order attributes.
403 Forbidden	Unauthorized: Attributes failed permission check	Access order attributes as an authenticated user, not associated with the order.
404 Not Found	Resource not found MetaData is missing.	Order does not exist. Attributes have not been set.
502 Bad Gateway	Unable to get attributes	Unexpected (backend) error. Unable to retrieve order attributes.

113.3.2 Response Body

```
{
  "list": [
    {
      "name": "{attributeName}",
      "value": "{attributeValue}"
    },
    ...
  ]
}
```

```
],
"links": [
  {"href": "https://api.hostway.com/orders/{orderId}/attributes/", "rel": "self"}
]
}
```

113.3.2.1 Parameters

attributeName - *string*
The unique name of the attribute.

attributeValue - *string*
The value of the attribute.

links - *list*
[Hypermedia](#) for this resource.

113.4 Example

113.4.1 Successfully retrieve order attributes

Request

```
GET /orders/000001485a3c75c8569447b3007f000000010001/attributes
Cookie: OwnerId={owner_id}; Path=
```

Response

```
200 OK
Content-Type : application/json; charset=UTF-8
```

```
{
  "list" : [ {
    "name" : "_utmb",
    "value" : "utmb_val"
  }, {
    "name" : "_utma",
    "value" : "utma_val"
  } ],
  "links" : [ {
    "href" : "https://api.hostway.com/orders/000001485a3c75c8569447b3007f000000010001/attributes/",
    "rel" : "self"
  } ]
}
```

113.4.2 Wrong authorization

Request

```
GET /orders/{orderId}/attributes
Cookie: OwnerId={wrong_owner_id}; Path=
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

113.4.3 Unassociated user without cookie

Request

```
GET /orders/{orderId}/attributes
Authorization: {unassociated_user_authorization}
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8
```

```
{
  "forbidden": {
    "guid": "4d1e97e2-1497-48d5-b7a3-b3726656c4b5",
    "message": "Unauthorized: Attributes failed permission check",
    "code": 403,
    "details": ""
  }
}
```

113.4.4 Order does not exist

Request

```
GET /orders/{wrong_orderId}/attributes
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not found
```

Content-Type : application/json; charset=UTF-8

```
{
  "itemNotFound": {
    "guid": "ee20516d-011e-44ea-99d2-ae7a2b6fe99a",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

113.4.5 No attributes set

Request

GET /orders/{orderId}/attributes
Cookie: OwnerId={owner_id}; Path=/

Response

404 Not found
Content-Type : application/json; charset=UTF-8

```
{
  "itemNotFound": {
    "guid": "ee20516d-011e-44ea-99d2-ae7a2b6fe99a",
    "message": "MetaData is missing.",
    "code": 404,
    "details": ""
  }
}
```

113.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

114 Orders-orderId-attributes:PUT

115 PUT /orders/{orderId}/attributes

Set the list of attributes for an order.

115.1 Contents

- 1 PUT /orders/{orderId}/attributes
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Set order attributes correctly
 - ◇ 1.3.2 Set order attributes with invalid key
 - ◇ 1.3.3 Set order attributes with malformed body
 - ◇ 1.3.4 Set order attributes without authenticating
 - ◇ 1.3.5 Unassociated user without cookie
 - ◇ 1.3.6 Set order attributes of unexisting order
 - ◆ 1.4 See also

115.2 Request

PUT /orders/{orderId}/attributes

115.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

115.2.2 URI Parameters

None.

115.2.3 Request Headers

Content-Type

Required. Set this header to `application/json; charset=UTF-8`

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

115.2.4 Request Body

```
[
  {
    "name": "{attributeName}",
    "value": "{attributeValue}"
  },
  {
    "name": "{attributeName}",
    "value": "{attributeValue}"
  }
]
```

115.2.4.1 Parameters

attributeName - *string*

The unique name of the attribute. The list of supported attribute names is specific to the franchise. The common attributes include:
AffiliateId - identifies an affiliate.

attributeValue - *string*

The value of the attribute max 200 characters. Should adhere to the rules for the attribute specified by the name.

115.3 Response

115.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The order attributes were successfully set.
400 Bad Request	POST data error Could not find MetaData Type {attribute_key} for franchise {franchise}.	The request body format is invalid. The attribute name is not supported by the franchise, the attribute value is invalid (e.g: too long, contains invalid characters, etc).
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to set order attributes.
403 Forbidden	Unauthorized: Attributes failed permission check	Set order attributes as an authenticated user, not associated with the order.
404 Not Found	Resource not found	The specified order does not exist.
502 Bad Gateway	-	A critical (unknown) error occurred.

115.4 Example

115.4.1 Set order attributes correctly

Request

```
PUT /orders/{orderId}/attributes
Cookie: OwnerId={owner_id}; Path=/
[ { "name": "_utma", "value": "utma_val" }, { "name": "_utmb", "value": "utmb_val" } ]
```

Response

204 No Content

115.4.2 Set order attributes with invalid key

Request

```
PUT /orders/{orderId}/attributes
Cookie: OwnerId={owner_id}; Path=/
[ { "name": "nonexistent_key", "value": "some_value" } ]
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8

{
  "computeFault": {
    "guid": "5b01960d-4110-43f0-b58f-d3178db61c6d",
    "message": "Could not find MetaData Type nonexistent_key for franchise hostway_us_main",
    "code": 400, "details": ""
  }
}
```

115.4.3 Set order attributes with malformed body

Request

```
PUT /orders/{orderId}/attributes
Cookie: OwnerId={owner_id}; Path=/
[ { "_utma": "utma_val" } ]
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8

{
  "badRequest": {
    "guid": "9a6b18b4-66cf-4b5e-bee6-f4f44c0b9450",
    "message": "POST data error",
    "code": 400,
    "details": {
      "0.name": "Required",
      "0.value": "Required"
    }
  }
}
```

115.4.4 Set order attributes without authenticating

Request

```
PUT /orders/{orderId}/attributes
Cookie: OwnerId={owner_id}; Path=/
[ { "name": "nonexistent_key", "value": "some_value" } ]
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

401 Unauthorized
This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

115.4.5 Unassociated user without cookie

Request

```
PUT /orders/{orderId}/attributes
Authorization: {unassociated_user_authorization}

[ {"name": "_utma", "value": "utma_val"}, {"name": "_utmb", "value": "utmb_val"} ]
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8

{
  "forbidden": {
    "guid": "4d1e97e2-1497-48d5-b7a3-b3726656c4b5",
    "message": "Unauthorized: Attributes failed permission check",
    "code": 403,
    "details": ""
  }
}
```

115.4.6 Set order attributes of unexisting order

Request

```
PUT /orders/{orderId}/attributes
Cookie: OwnerId={owner_id}; Path=/

[ {"_utma": "utma_val"} ]
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8

{
  "itemNotFound": {
    "guid": "6273e2fb-6efd-4294-9281-7a1ea664cf11",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

115.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the comments powered by Disqus. [blog comments powered by Disqus](#)

116 Orders-orderId-discounts:DELETE

117 DELETE /orders/{orderId}/discounts

Remove all discounts that were previously added at the Order level. This will not remove discounts that were added to specific products or services.

117.1 Contents

- 1 DELETE /orders/{orderId}/discounts
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 Missing authorization
 - ◇ 1.3.3 Insufficient privileges
 - ◇ 1.3.4 Order does not exist
 - ◆ 1.4 See also

117.2 Request

DELETE /orders/{orderId}/discounts

117.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

117.2.2 URI Parameters

None.

117.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

117.3 Response

117.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The request was executed successfully.
401 Unauthorized	-	The request was not properly authorized.
403 Forbidden	Unauthorized: DiscountsView failed permission check	The client does not have sufficient privileges.
404 Not Found	Resource not found	Order does not exist.
502 Bad Gateway	-	Unexpected backend response. Cannot delete custom discounts.

117.4 Example

117.4.1 Success

Request

```
DELETE /orders/{orderId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/
```

Response

```
204 No Content
```

117.4.2 Missing authorization

Request

```
DELETE /orders/{orderId}/discounts
Cookie: OwnerId={owner_id}; Path=
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

117.4.3 Insufficient privileges

Request

```
DELETE /orders/{orderId}/discounts
Authorization: Basic ZWdjUlBtZWZiWnJVQHNpdGVjb250cm9sX2hvc3R3YXlfdXNfbWVpbjpbTdnVROGlWOUFTVTc0
Cookie: OwnerId={owner_id}; Path=
```

Response

```
403 Forbidden
Content-Type : application/json; charset=UTF-8
```

```
{
  "forbidden": {
    "guid": "8e235035-f8fe-465f-9b5d-85ef49829fd3",
    "message": "Unauthorized: DiscountsView failed permission check",
    "code": 403,
    "details": ""
  }
}
```

117.4.4 Order does not exist

Request

```
DELETE /orders/{non-existent_orderId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "7f3091bd-5930-4f14-9f02-86780cd8405f",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

117.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

118 Orders-orderId-discounts:GET

119 GET /orders/{orderId}/discounts

Get discounts that were added at the Order level. Discounts that are applied to an entire order will be applied to all of the product and services within the order.

119.1 Contents

- 1 GET /orders/{orderId}/discounts
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 Missing authorization
 - ◇ 1.3.3 Insufficient privileges
 - ◇ 1.3.4 Order does not exist
 - ◆ 1.4 See also

119.2 Request

GET /orders/{orderId}/discounts

119.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

119.2.2 URI Parameters

None.

119.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

119.3 Response

119.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The request was executed successfully.
401 Unauthorized	-	The request was not properly authorized.
403 Forbidden	Unauthorized: DiscountsView failed permission check	The client does not have sufficient privileges.
404 Not Found	Resource not found	Order does not exist.
502 Bad Gateway	-	Unexpected backend response. Cannot retrieve order discounts.

119.3.2 Response Body

```
{
  "discounts": [
    {
      "priceName": "recurring",
      "recurrences": -1,
      "type": "amount",
      "value": "1.00"
    }
  ],
  "links": [
```

```

    {
      "href": "https://api.hostway.com/orders/{orderId}/account/user/",
      "rel": "self"
    }
  ]
}

```

119.3.2.1 Parameters

discounts - *list*

A list of discounts. Each discount contains the following:

priceName - *string*

The type of price to which the discount will be applied.

recurrences - *integer*

The number of times that the discount will be applied. A value of "-1" means the discount will be applied every time the fee is charged.

type - *string*

Specifies the discount type which determines the method used to calculate the amount that is discounted. Possible values are:
amount: Discount will be treated as an exact amount.

value - *string*

The discount amount. The value will be interpreted based on the specified discount type.

links - *list*

[Hypermedia](#) for this resource. Links to self.

119.4 Example

119.4.1 Success

Request

```

GET /orders/{orderId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/

```

Response

```

200 OK
Content-Type : application/json; charset=UTF-8

```

```

{
  "discounts": [
    {
      "priceName": "recurring",
      "type": "amount",
      "value": "5.00",
      "recurrences": -1
    }
  ],
  "links": [
    {
      "href": "https://api.hostway.com/orders/000001485584ddfe2c27d60a007f000000010001/discounts/",
      "rel": "self"
    }
  ]
}

```

119.4.2 Missing authorization

Request

```

GET /orders/{orderId}/discounts
Cookie: OwnerId={owner_id}; Path=/

```

Response

```

401 Unauthorized
Content-Type : text/plain; charset=UTF-8

```

```

401 Unauthorized

```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

119.4.3 Insufficient privileges

Request

```

GET /orders/{orderId}/discounts
Authorization: Basic ZWdjUlBtZWZiWnJVQHNPdGVjb250cm9sX2hvc3R3YXlfdXNfbWFPbjpTdGVROGlWOUFTVtC0
Cookie: OwnerId={owner_id}; Path=/

```

Response

```

403 Forbidden
Content-Type : application/json; charset=UTF-8

```

```

{

```

```
"forbidden": {
  "guid": "b5b513b8-8436-4eee-b49e-ac1d369a9d1b",
  "message": "Unauthorized: DiscountsView failed permission check",
  "code": 403,
  "details": ""
}
```

119.4.4 Order does not exist

Request

```
GET /orders/{non-existent_orderId}/discounts
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2Vydm1jZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "402282af-9f53-4973-8437-469087465843",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

119.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

120 Orders-orderId-discounts:PUT

121 PUT /orders/{orderId}/discounts

Set discounts for the entire Order. Discounts that are applied to an entire order will be applied to all of the product and services within the order. **Note:** This will replace any existing discounts that were configured at the order level.

121.1 Contents

- 1 PUT /orders/{orderId}/discounts
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◇ 1.1.4 Request Body
 - 1.1.4.1 Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 Invalid discount amount
 - ◇ 1.3.3 Invalid discount type
 - ◇ 1.3.4 Invalid discount number of recurrences
 - ◇ 1.3.5 Missing price type
 - ◇ 1.3.6 Missing authorization
 - ◇ 1.3.7 Insufficient privileges
 - ◇ 1.3.8 Order does not exist
 - ◆ 1.4 See also

121.2 Request

PUT /orders/{orderId}/discounts

121.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

121.2.2 URI Parameters

None.

121.2.3 Request Headers

Content-Type

Required. Set this header to `application/json; charset=UTF-8`

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

121.2.4 Request Body

```
{
  "discounts": [
    {
      "priceName": "recurring",
      "recurrences": -1,
      "type": "amount",
      "value": "1.00"
    }
  ]
}
```

121.2.4.1 Parameters

discounts - *list*

A list of discounts. Each discount contains the following:

priceName - *string*

The type of price to which the discount will be applied.
Validation: Should match the fee types configured in the back-end.

Note: Only "recurring" is currently supported

recurrences - *integer*

The number of times that the discount will be applied. A value of "-1" means the discount will be applied every time the fee is charged.

Validation: Should be either a positive integer or -1.

Note: Value must be -1 if the priceName is "recurring".

type - *string*

Specifies the discount type which determines the method used to calculate the amount that is discounted. Possible values are:

amount: Discount will be treated as an exact amount

percent: Discount will be treated as a percentage

Validation: Should be set to either **amount** or **percent**.

value - *string*

The discount amount. The value will be interpreted based on the specified discount type.

Validation: Should be a positive number.

121.3 Response

121.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	The discount was successfully set.
	POST data error	A required field is missing.
		Validation Error:
400 Bad Request	Bad request	Recurrences should be either a positive integer or -1.
	{value} discount amount is invalid	Price type should be recurring .
	{type} discount type is unsupported	Discount amount should be a positive number.
401 Unauthorized	-	Type should be set to either amount or percent .
403 Forbidden	Unauthorized: DiscountsView failed permission check	The request was not properly authorized.
404 Not Found	Resource not found	The client does not have sufficient privileges.
502 Bad Gateway	-	Order does not exist.
		Unexpected backend response. Cannot set custom discounts.

121.4 Example

121.4.1 Success

Request

```
PUT /orders/{orderId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

204 No Content

121.4.2 Invalid discount amount

Request

```
PUT /orders/{orderId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "-5.00"
  }]
}
```

```
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "computeFault": {
    "guid": "c3563e61-a297-4d54-ab28-77b875934829",
    "message": "-5.00 discount amount is invalid",
    "code": 400,
    "details": ""
  }
}
```

121.4.3 Invalid discount type

Request

```
PUT /orders/{orderId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "numbers",
    "value": "5.00"
  }]
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "computeFault": {
    "guid": "1a94f744-5b33-4d54-ab28-77b875934829",
    "message": "amounts discount type is unsupported",
    "code": 400,
    "details": ""
  }
}
```

121.4.4 Invalid discount number of recurrences

Request

```
PUT /orders/{orderId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-10",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "computeFault": {
    "guid": "e15a6425-1cad-4154-b8f9-c10a0e46bf2c",
    "message": "Bad Request",
    "code": 400,
    "details": ""
  }
}
```

121.4.5 Missing price type

Request

```
PUT /orders/{orderId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "discounts": [{
    "recurrences": "-1",
    "type": "amount",
    "value": "-5.00"
  }]
}
```

Response

```
400 Bad Request
Content-Type : application/json; charset=UTF-8
```

```
{
  "badRequest" : {
    "guid" : "bb3da718-f0e1-4787-acd4-8cf4215f0c44",
    "message" : "POST data error",
    "code" : 400,
    "details" : {
      "discounts.0.priceName" : "Required"
    }
  }
}
```

121.4.6 Missing authorization

Request

```
PUT /orders/{orderId}/discounts
Content-Type : application/json; charset=UTF-8
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

```
401 Unauthorized
Content-Type : text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

121.4.7 Insufficient privileges

Request

```
PUT /orders/{orderId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic ZWdjUlBtZWZiWnJVQHNPdGVjb250cm9sX2hvc3R3YXlfdXNfbWFPbjpTdnVROGlWOUFTVTc0
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

```
403 Forbidden
Content-Type : application/json; charset=UTF-8
```

```
{
  "forbidden": {
    "guid": "1912d9fa-5fbc-4e16-b0ba-00d50c9c1ec4",
    "message": "Unauthorized: DiscountsView failed permission check",
    "code": 403,
    "details": ""
  }
}
```

121.4.8 Order does not exist

Request

```
PUT /orders/{non-existent_orderId}/discounts
Content-Type : application/json; charset=UTF-8
Authorization: Basic dGVzdC52cGxleEBjb3JwX2xkYXBfc2VydmljZV9jaGk6VHJAcXNQODch
Cookie: OwnerId={owner_id}; Path=/
```

```
{
  "discounts": [{
    "priceName": "recurring",
    "recurrences": "-1",
    "type": "amount",
    "value": "5.00"
  }]
}
```

Response

```
404 Not Found
Content-Type : application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "cca219c5-3e2b-481d-8e7f-7083064872ff",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```


}

121.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

122 Orders-orderId-quote:GET

123 GET /orders/{orderId}/quote

Get quote for a given order.

123.1 Contents

- 1 GET /orders/{orderId}/quote
 - ◆ 1.1 Request
 - ◇ 1.1.1 Request Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Get empty quote info
 - ◇ 1.3.2 No authorization
 - ◇ 1.3.3 Unauthorized user
 - ◇ 1.3.4 Order does not exist
 - ◇ 1.3.5 Get quote with child product
 - ◆ 1.4 See also

123.2 Request

GET /orders/{orderId}/quote

123.2.1 Request Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

123.2.2 URI Parameters

None.

123.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

123.3 Response

123.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	The operation was successfully completed.
401 Unauthorized	-	No owner Id specified or the specified owner is not authorized to get quote for this order.
403 Forbidden	Unauthorized: Fees failed permission check	The user does not have sufficient privilege.
404 Not Found	Resource not found	The specified order does not exist.
502 Bad Gateway	-	Unexpected backend response. Cannot retrieve order quote.

123.3.2 Response Body

```
{
  "quote": {
    "Hash": "42fc1e100808efc62dadf8ad027d424dff30d6f1",
    "Currency": "USD",
    "Products": [
      {
        "Services": [],
        "PlanNumber": "00000139b5dd73be7f14d3f8000a88fa0fa65d70",
        "PromotionCode": "promo_1234",
        "PromotionName": "Seasonal Discount 25% off all dedicated servers for 3 months",
        "ChildProduct": [],
        "Fees": [
          {
            "ProrationAllowed": "0",
            "StartDate": "2012-09-11",
            "NumCharged": "0",
            "EndDate": "2012-09-11",
            "Description": "Setup fee ",
            "BaseAmount": "100.00",
            "UnitAmount": "100.00",
            "Prorated": "0",
            "MonthlyCycle": "1",
            "Currency": "USD",
            "Amount": "100.00",
            "NumCharges": "1",
            "ProratedUnitAmount": "100.00",
            "Type": "setup_fee",
            "Quantity": "1",
            "Deferred": "0"
          },
          {
            "ProrationAllowed": "1",
            "StartDate": "2012-09-11",
            "NumCharged": "0",
            "EndDate": "2012-10-10",
            "Description": "Recurring fee",
            "BaseAmount": "239.00",
            "UnitAmount": "239.00",
            "Prorated": "0",
            "MonthlyCycle": "1",
            "Currency": "USD",
            "Amount": "239.00",
            "NumCharges": "-1",
            "ProratedUnitAmount": "239.00",
            "Type": "recurring",
            "Quantity": "1",
            "Deferred": "1"
          }
        ]
      }
    ],
    "Total": "100.00",
    "SubTotal": "100.00",
    "OneTimeFeeAmount": "100.00",
    "TaxTotal": "0",
    "DeferredSubTotals": [
      {
        "PurchasedPlanNumber": "00000139b5dd73be7f14d3f8000a88fa0fa65d70",
        "Amount": "239.00"
      }
    ],
    "DeferredTaxes": [],
    "DeferredTotals": [
      {
        "PurchasedPlanNumber": "00000139b5dd73be7f14d3f8000a88fa0fa65d70",
        "Amount": "239.00"
      }
    ],
    "links": [
      {
        "href": "https://api.hostway.com/orders/{orderId}/quote/",
        "rel": "self"
      }
    ]
  }
}
```

123.3.2.1 Parameters

quote - *dictionary*

Quote's ordered items info

Hash - *string*

Hash string of this particular order.

Currency - *string*

The currency for this particular order. Must be a ISO-4217 currency code. ex. USD, CAD, etc.

Products - *list*

List of plan products

Services - *list*

List of product services

PlanNumber - *string*

Plan number identifier

PromotionCode - *string*

Promotion code, if any (applied at the time the product was added to the order)

PromotionName - *string*

(OPTIONAL)

Name or short description of the promotion that was applied, if any

ChildProduct - *list*
List of products of the plan product

Fees - *list*
List of product fees

ProrationAllowed - *integer*
Number indicating whether fee proration is allowed

StartDate - *date*
Start date of the fee

NumCharged - *integer*
Number how many times this fee is charged

EndDate - *date*
End date the fee is applied

Description - *string*
Description info for this fee

BaseAmount - *decimal*
Base amount for this fee

UnitAmount - *decimal*
Unit amount for this fee

Prorated - *integer*
Integer indicating whether this fee is prorated

MonthlyCycle - *integer*
Billing cycle in months for this fee

Currency - *string*
The currency for this particular fee. Must be a ISO-4217 currency code. ex. USD, CAD, etc.

Amount - *decimal*
The actual fee amount

NumCharges - *integer*
Number of charges for this fee

ProratedUnitAmount - *decimal*
The prorated unit amount

Type - *string*
The type of this fee

Quantity - *integer*
The quantity for this fee

Deferred - *integer*
Indicates whether the fee is deferred

Total - *decimal*
Total sum for this quote. Does not include deferred charges.

SubTotal - *decimal*
Subtotal sum for this quote. Does not include deferred charges.

OneTimeFeeAmount - *decimal*
One time payable fee amount

TaxTotal - *decimal*
Tax total value. Does not include taxes on deferred charges.

DeferredSubTotals - *list*
A list of dictionaries, containing the sub of deferred subtotals for each PurchasedPlan in the order. If no deferred fees are found in the plan, it might not be present.

DeferredTaxes - *list*
A list of dictionaries, containing the sub of deferred taxes for each PurchasedPlan in the order. If no deferred fees are found in the plan, it might not be present.

DeferredTotals - *list*
A list of dictionaries, containing the sub of deferred totals (subtotal + tax) for each PurchasedPlan in the order. If no deferred fees are found in the plan, it might not be present.

links - *list*
[Hypermedia](#) for this resource. Links to self.

123.4 Example

123.4.1 Get empty quote info

Request

```
GET /orders/{orderId}/quote
Cookie: OwnerId={owner_id}; Path=/
```

Response

200 OK
Content-Type : application/json; charset=UTF-8

```
{
  "quote": {
    "PurchasedPlan": [],
    "Hash": "26d42ebf46c38b3eaabca48a0f588d90e8f542fc",
    "Currency": "USD",
    "Total": "0",
    "SubTotal": "0",
    "OneTimeFeeAmount": "0",
    "TaxTotal": "0"
  },
  "links": {
    "self": {
      "href": "https://api.hostway.com/orders/0000014854ec1d454e9cfd3d007f000000010001/quote/",
      "rel": "self"
    }
  }
}
```

123.4.2 No authorization

Request

GET /orders/{orderId}/quote
Cookie: OwnerId={wrong_owner_id}; Path=

Response

401 Unauthorized
Content-Type : text/plain; charset=UTF-8

401 Unauthorized

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., bad password), or your browser does not understand how to supply the credentials required.

123.4.3 Unauthorized user

Request

GET /orders/{orderId}/quote
Authorization: Basic {wrong_user_authorization}; Path=

Response

403 Forbidden
Content-Type : application/json; charset=UTF-8

```
{
  "forbidden": {
    "guid": "ea88f675-dc71-410e-9e72-84f0944a289a",
    "message": "Unauthorized: Fees failed permission check",
    "code": 403,
    "details": ""
  }
}
```

123.4.4 Order does not exist

Request

GET /orders/{non-existent_orderId}/quote
Cookie: OwnerId={owner_id}; Path=

Response

404 Not found
Content-Type : application/json; charset=UTF-8

```
{
  "itemNotFound": {
    "guid": "92b7773f-e224-4971-a3dd-60fbd9c7129a",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

123.4.5 Get quote with child product

Request

GET /orders/{orderId}/quote
Cookie: OwnerId={owner_id}; Path=

Response

200 OK
Content-Type : application/json; charset=UTF-8

```
{
  "quote": {
```

```

"PurchasedPlan": [
  "Product": [
    "PromotionCode":null,
    "ChildProduct": [
      {
        "PromotionCode":null,
        "ChildProduct":[],
        "Name":"ChildProductName",
        "ProductNumber":null,
        "MonthlyCycle":1
        "ConfigType":"ptc",
        "Fees": [
          {
            "Currency":"USD",
            "Amount":"0.00",
            "NumCharges":"1",
            "ProratedUnitAmount":"0.00",
            "Type":"setup_fee",
            "Discounts":[],
            "Quantity":"1"
          },
          {
            "BaseAmount":"10.00",
            "UnitAmount":"10.00",
            "Prorated":"0",
            "MonthlyCycle":"1"
          }
        ]
      }
    ]
  ],
  "Name":"ParentProduct",
  "ProductNumber":null,
  "MonthlyCycle":1,
  "ConfigType":"parent.product",
  "Fees":[ ],
  "OrderNotes":[ ],
  "ProductGUID":"0000015100dcc330be07657a007f000000010001"
},
"DisplayName":"",
"Name":"plan name",
"ReferralKey":null,
"MonthlyCycle":1,
"PartnerId":"",
"Plan":"test.plan",
"Fees":[ ],
"PlanNumber":"0000015100dcbf46c7576990007f000000010001",
"PromotionCode":null
],
"Hash": "26d42ebf46c38b3eaabca48a0f588d90e8f542fc",
"Currency": "USD",
"DeferredSubTotals":[],
"DeferredTotals":[],
"Total": "0",
"SubTotal": "0",
"OneTimeFeeAmount": "0",
"DeferredTaxes":[],
"TaxTotal": "0"
},
"links": {
  "self": {
    "href": "https://api.hostway.com/orders/0000014854ec1d454e9cfd3d007f000000010001/quote/",
    "rel": "self"
  }
}
}
}

```

123.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

124 Orders-orderId:DELETE

125 DELETE /orders/{orderId}

Deletes an order. Only new orders that were not submitted can be deleted.

125.1 Contents

- 1 DELETE /orders/{orderId}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◆ 1.3 Example
 - ◇ 1.3.1 Success
 - ◇ 1.3.2 No permission
 - ◇ 1.3.3 Unassociated user without cookie
 - ◇ 1.3.4 Order does not exist
 - ◇ 1.3.5 Order has already been submitted
 - ◆ 1.4 See also

125.2 Request

DELETE /orders/{orderId}

125.2.1 Parameters

orderId - *string*

The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

125.2.2 URI Parameters

None.

125.2.3 Request Headers

Cookie

A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

125.3 Response

125.3.1 Status Code

Status Code	Error Message	Description
204 No Content	-	Order was successfully deleted.
401 Unauthorized	-	The OwnerId cookie or the Authorization header are not specified or do not permit deletion of the specified order.
403 Forbidden	Unauthorized: Order failed permission check	Access an associated order as an authenticated user, not associated with the order.
404 Not Found	Resource not found	The specified order does not exist or it was previously deleted.
405 Method Not Allowed	Method not allowed	The order was submitted and cannot be deleted.

125.4 Example

125.4.1 Success

Request

```
DELETE /orders/0000013ca61880fbfc068b33000a02c9d8ef861c
Cookie: OwnerId=948ce0d4-7cf0-43b6-88a5-2b3f1331b689; Path=/
```

Response

```
204 No Content
```

125.4.2 No permission

Request

```
DELETE /orders/000001483b9336c4be0a0c89000a19010d347ea2
Cookie: OwnerId=wrong_owner_id; Path=/
```

Response

```
401 Unauthorized
Content-Type: text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

125.4.3 Unassociated user without cookie

Request

```
DELETE /orders/000001483b9336c4be0a0c89000a19010d347ea2
Authorization: {unassociated_user_authorization}
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8
```

```
{
  "forbidden" : {
    "guid" : "bb99d8e5-cb4a-4c87-9c03-57743ef06ac7",
    "message" : "Unauthorized: Order failed permission check",
    "code" : 403,
    "details" : ""
  }
}
```

125.4.4 Order does not exist

Request

```
DELETE /orders/000001483b9336c4be0a0c89000a19010d347ea2
Cookie: OwnerId=0b3fc7bd-6905-4de7-bddb-43d2c7365e7a; Path=/
```

Response

```
404 Not Found
Content-Type: application/json; charset=UTF-8
```

```
{
  "itemNotFound": {
    "guid": "3d9b09b4-9737-473b-91a5-6a8d2d212581",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

125.4.5 Order has already been submitted

Request

```
DELETE /orders/0000013ca61880fbfc068b33000a02c9d8ef861c
Cookie: OwnerId=948ce0d4-7cf0-43b6-88a5-2b3f1331b689; Path=/
```

Response

```
405 Method Not Allowed
Content-Type: application/json; charset=UTF-8
```

```
{
  "badMethod": {
    "guid": "93343aac-1451-4ed1-bca8-ac143f263cbe",
    "message": "Method not allowed",
    "code": 405,
    "details": ""
  }
}
```

125.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

126 Orders-orderId:GET

127 GET /orders/{orderId}

Returns the content of an order specified by the URI.

127.1 Contents

- 1 GET /orders/{orderId}
 - ◆ 1.1 Request
 - ◇ 1.1.1 Parameters
 - ◇ 1.1.2 URI Parameters
 - ◇ 1.1.3 Request Headers
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Body
 - 1.2.2.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Get order information
 - ◇ 1.3.2 Get info with wrong OwnerId
 - ◇ 1.3.3 Unassociated user without cookie
 - ◇ 1.3.4 Unexisting order
 - ◆ 1.4 OPTIONS /orders/{order_id}
 - ◇ 1.4.1 Parameters
 - ◇ 1.4.2 Request Headers
 - ◇ 1.4.3 Response Status Code
 - ◇ 1.4.4 Response Headers
 - ◆ 1.5 See also

127.2 Request

GET /orders/{orderId}

127.2.1 Parameters

orderId - *string*
The unique order Id. The URI of the order is returned by the [POST /orders](#) call.

127.2.2 URI Parameters

None.

127.2.3 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

Authorization
Optional. Specify the **Authorization** header when requesting details for an order that was already submitted. Follow the instructions for [API Authentication](#).

127.3 Response

127.3.1 Status Code

Status Code	Error Message	Description
200 OK	-	Success.
401 Unauthorized	-	

The **OwnerId** cookie or the **Authorization** header are not specified or do not permit access to the specified order.

403 Forbidden Unauthorized: Order failed permission check

Access an associated order as an authenticated user, not associated with the order.

404 Not Found Resource not found

The specified order does not exist.

127.3.2 Response Body

The response body may contain additional information beyond what is specified in the documentation. Such information should be ignored or treated as volatile, since it may change in the future.

```
{
  "order": {
    "Status": "{orderStatus}",
    "Account": "{accountUrl}"
  },
  "links": {
    "self": {"href": "https://api.hostway.com/orders/{orderId}/", "rel": "self"}
  }
}
```

127.3.2.1 Parameters

orderStatus - *string*

The status of the order.

accountUrl - *string (optional)*

[Hypermedia](#) for the account of this order. Link to the account.

links - *list*

[Hypermedia](#) for this resource. Link to the order.

127.4 Example

127.4.1 Get order information

Request

```
GET /orders/0000013ca61880fbfc068b33000a02c9d8ef861c
Cookie: OwnerId=948ce0d4-7cf0-43b6-88a5-2b3f1331b689; Path=/
```

Response

```
200 OK
Content-Type: application/json; charset=UTF-8
```

```
{
  "order": {
    "Status": "new",
    "Account": "http://accounts/acc20130821"
  },
  "links": [ {
    "href": "https://api.hostway.com/orders/0000013ca61880fbfc068b33000a02c9d8ef861c/",
    "rel": "self"
  } ]
}
```

127.4.2 Get info with wrong OwnerId

Request

```
GET /orders/000001483b9336c4be0a0c89000a19010d347ea1
Cookie: OwnerId={wrong_order_id}; Path=/
```

Response

```
401 Unauthorized
Content-Type: text/plain; charset=UTF-8
```

```
401 Unauthorized
```

This server could not verify that you are authorized to access the document you requested. Either you supplied the wrong credentials (e.g., b

127.4.3 Unassociated user without cookie

Request

```
GET /orders/000001483b9336c4be0a0c89000a19010d347ea2
Authorization: {unassociated_user_authorization}
```

Response

```
403 Forbidden
Content-Type: application/json; charset=UTF-8
```

```
{
  "Forbidden": {
    "guid": "4dle97e2-1497-48d5-b7a3-b3726656c4b5",
    "message": "Unauthorized: Order failed permission check",
    "code": 403,
    "details": ""
  }
}
```

127.4.4 Unexisting order

Request

```
GET /orders/000001483b9336c4be0a0c89000a19010d347ea2
Cookie: OwnerId=0b3fc7bd-6905-4de7-bddb-43d2c7365e7a; Path=/
```

Response

```
404 Not Found
Content-Type: application/json; charset=UTF-8

{
  "itemNotFound": {
    "guid": "286aab08-7a2b-41a7-81fa-6530f14ed728",
    "message": "Resource not found",
    "code": 404,
    "details": ""
  }
}
```

127.5 OPTIONS /orders/{order_id}

In addition to the GET method the API supports the OPTIONS method for direct browser access to the API. The format of the request is:

```
OPTIONS /orders/{order_id}
```

127.5.1 Parameters

`order_id` - *string*

The unique order Id. The full URI of each order is returned by the [POST /orders](#) call.

127.5.2 Request Headers

Cookie
A cookie named **OwnerId** to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to the create order request in the **Set-Cookie** response header.

127.5.3 Response Status Code

200 OK
Success.

404 Not Found
The specified order does not exist.

127.5.4 Response Headers

Access-Control-Allow-Methods
A comma-separated list of request methods supported on this URI.

Access-Control-Allow-Origin
All origins are allowed. This header has value `*`.

127.6 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

128 Orders:POST

129 POST /orders

Creates a new order and returns the unique order URI.

129.1 Contents

- 1 POST /orders
 - ◆ 1.1 Request
 - ◇ 1.1.1 URI Parameters
 - ◇ 1.1.2 Request Headers
 - ◇ 1.1.3 Request Body
 - 1.1.3.1 Parameters
 - ◆ 1.2 Response
 - ◇ 1.2.1 Status Code
 - ◇ 1.2.2 Response Headers
 - ◇ 1.2.3 Response Body
 - 1.2.3.1 Parameters
 - ◆ 1.3 Example
 - ◇ 1.3.1 Create Order successfully
 - ◇ 1.3.2 Create without permission to pass client IP
 - ◇ 1.3.3 An open order already exists
 - ◆ 1.4 See also

129.2 Request

POST /orders

129.2.1 URI Parameters

None.

129.2.2 Request Headers

Content-Type

Required. Set this header to `application/json; charset=UTF-8`

Cookie

Optional. A cookie named **OwnerId** may be specified to identify the client (for most browser clients, the browser will do this automatically, depending on the browser configuration). The value of the cookie would come from the response to a previous request for this resource (in the **Set-Cookie** response header).

129.2.3 Request Body

```
{
  "clientId": {clientId}
}
```

129.2.3.1 Parameters

clientId - *string*

Optional. The IP address of the client that submitted the order. If not provided, the IP address of the client that called the API will be used. The IP address may be used to check the order for possible fraud. This parameter may be useful in cases where the API client accepts and relays orders from a 3rd-party customer (such as a shopping cart application). This parameter requires elevated permissions.

129.3 Response

129.3.1 Status Code

Status Code	Error Message	Description
201 Created	-	The new order was created.
400 Bad Request	User cannot specify the client IP address	The client attempted to pass the client IP address and is not allowed. Use of this parameter requires elevated permissions.
409 Conflict	A new order has already been created for this ownerId	The client already has an open order. If the OwnerId cookie was passed in the request headers, a 409 will be returned if there's already an open (unsubmitted) order for the same client.

129.3.2 Response Headers

Location

Returned with successful response. Contains the order URI. The client must use this URI for all subsequent operations with the order.

Set-Cookie

Returned with successful response. Contains an **OwnerId** value which must be submitted with all subsequent requests for this order. The **OwnerId** identifies the client and prohibits it from having more than one open (unsubmitted) orders.

129.3.3 Response Body

```
{
  "orderId": "{orderId}",
  "links": [
    {"href": "https://api.hostway.com/orders/", "rel": "self"},
    {"href": "https://api.hostway.com/orders/{orderId}", "rel": "item"}
  ]
}
```

129.3.3.1 Parameters

`orderId` - *string*
The Id of the new order

`links` - *list*
[Hypermedia](#) for this resource. Links to self and the newly created order.

129.4 Example

129.4.1 Create Order successfully

Request

```
POST /orders
Content-Type: application/json; charset=UTF-8
```

```
{}
```

Response

```
201 Created
Content-Type: application/json; charset=UTF-8
Location : https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf
Set-Cookie : OwnerId=b0d6352a-f66a-476d-8e1d-2f3f9dea5c49; Path=/
```

```
{
  "orderId": "0000013ad6dd95746a399f90000a71384bf242cf",
  "links": [
    {"href": "https://api.hostway.com/orders/", "rel": "self"},
    {"href": "https://api.hostway.com/orders/0000013ad6dd95746a399f90000a71384bf242cf", "rel": "item"}
  ]
}
```

129.4.2 Create without permission to pass client IP

Request

```
POST /orders
Content-Type: application/json; charset=UTF-8
Authorization: Basic AuthString
```

```
{
  "clientId": "1.2.3.4"
}
```

Response

```
400 Bad Request
Content-Type: application/json; charset=UTF-8
```

```
{
  "computeFault": {
    "guid": "143336cd-1763-4fe9-b3e3-c1e172b14fae",
    "message": "User cannot specify the client IP address",
    "code": 400, "details": ""
  }
}
```

129.4.3 An open order already exists

Request

```
POST /orders
Content-Type: application/json; charset=UTF-8
Cookie: OwnerId=0b3fc7bd-6905-4de7-bddb-43d2c7365e7a
```

```
{}
```

Response

```
409 Conflict
Content-Type: application/json; charset=UTF-8
```

```
{
  "conflict": {
    "guid": "e250bf7e-688d-45ba-8308-58374529be58",
    "message": "A new order has already been created for this ownerId",
    "code": 409,
    "details": ""
  }
}
```

129.5 See also

- [Orders API](#)
- [Common Features](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

130 ServiceAttributes-core.cloud.hscc

131 Cloud Compute Service Attributes

Attributes used when configuring the Cloud computing service:

131.1 Contents

- 1 Cloud Compute Service Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

131.2 Attributes Body

```
{  
  "accountNumber": "acct123456",  
}
```

131.2.1 Parameters

accountNumber
The account Number of the Hostway account.

131.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

132 ServiceAttributes-core.dedserver

133 Dedicated Server Service Attributes

Attributes used when configuring the Dedicated Server service.

133.1 Contents

- 1 Dedicated Server Service Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

133.2 Attributes Body

```
{  
  "serverName": "MyServer",  
  "serverLocation": "Tampa"  
}
```

133.2.1 Parameters

serverName

A name for the dedicated server.

serverLocation

The name of the data center where the server is to be located.

133.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

134 ServiceAttributes-core.dnr

135 Domain Name Registration Service Attributes

Attributes used when configuring the Domain Name Registration service.

135.1 Contents

- 1 Domain Name Registration Service Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

135.2 Attributes Body

```
{
  "domainName": "mydomain.com",
  "operation": "register",
  "term": 1,
  "authInfo": "mysecret",
  "nameservers": [ "ns1.myprovider.com", "ns2.myprovider.com" ],
  "extraParams": [{"name": "us_nexus", "value": "C11"}, {"name": "us_purpose", "value": "P1"}],
  "contactInfo": {
    "registrant": {
      "type": "registrant",
      "firstName": "John",
      "lastName": "Doe",
      "company": "ACME Inc.",
      "street1": "123 First Str.",
      "city": "Chicago",
      "zipOrPostal": "60000",
      "stateOrProvince": "IL",
      "countryCode": "US",
      "email1": "john.doe@test.com",
      "phone1": "+1.123.5555555" },
    "administrative": {
      "type": "administrative",
      ...
    },
    "billing": {
      "type": "billing",
      ...
    },
    "technical": {
      "type": "technical",
      ...
    }
  }
}
```

135.2.1 Parameters

- domainName** - *string*
The domain name
- operation** - *string*
"register" to register a domain
"transfer" to transfer a domain
- term** - *integer*
the registration term (period) in years
- authInfo** - *string*
authentication info, required for transfer operation
- nameservers** - *list of strings*
domain name servers (DNS) for the domain
- extraParams** - *list of extra params* (optional)
List of additional parameters required for some TLDs. Each parameter is an object with "name" and "value" fields. For example, .us registration requires the fields "us_nexus" which can be one of the following [C11, C12, C21, C31, C32] - see [What are Domain Name Application Purpose Codes?](#) and "us_purpose" which can be one of [P1, P2, P3, P4, P5] - see [What do the Nexus Codes mean?](#) You can find more information on extra parameters in the [GET /domains/tlds/{tld}/extrparams](#) documentation.
- company** - *string* (optional)
The company or organization name
- firstName** - *string*
The first name of the contact person
- lastName** - *string*
The last name of the contact person
- street1** - *string*
The street address
- street2** - *string* (optional)
The second line of the street address (if any)
- city** - *string*
The city
- countryCode** - *string*
The international 2-letter country code
- stateOrProvince** - *string*
The state or province. For US states use the 2-letter state code
- zipOrPostal** - *string*
The postal code
- email1** - *string*
The e-mail address of the contact person
- phone1** - *string*
The phone number of the contact person in valid [E.164](#) format, e.g. +1.3209270625

135.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

136 ServiceAttributes-core.fax.number

137 Fax Over Email Number Attributes

Attributes used for the FOE fax number.

137.1 Contents

- 1 Fax Over Email Number Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

137.2 Attributes Body

```
{
  "areaCode": "{area_code}",
  "name": "{name}"
}
```

137.2.1 Parameters

areaCode - *string*

(*REQUIRED*)

Standard 3 digit US area code (e.g. 312, 412).

Represents one of the supported area codes for a fax number within a particular ZIP area.

Please refer to the [FOE Supported Area Codes by ZIP code API](#) page for a detailed information about the supported area codes.

name - *string*

Customer friendly name/label for the fax number.

137.3 See also

- [Supported Area Codes by ZIP code](#)
- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

138 ServiceAttributes-core.fax.user

139 Fax Over Email User Attributes

Attributes used for the FOE user.

139.1 Contents

- 1 Fax Over Email User Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

139.2 Attributes Body

```
{  
  "zipCode": "{zip_code}",  
}
```

139.2.1 Parameters

zipCode - *string*
(*REQUIRED*)
US standard ZIP code - both ZIP code 5 and ZIP code 5 plus 4 formats are supported (e.g. 98001, 98001-9000).
Represents the zip code of the user.

139.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

140 ServiceAttributes-core.online backup.account

141 Online Backup Account Attributes

Attributes used for the online backup account.

141.1 Contents

- [1 Online Backup Account Attributes](#)
 - ◆ [1.1 Attributes Body](#)
 - ◇ [1.1.1 Parameters](#)
 - ◆ [1.2 See also](#)

141.2 Attributes Body

```
{  
  "defaultUserQuota": {default_user_quota},  
}
```

141.2.1 Parameters

defaultUserQuota - *integer*
(*REQUIRED*)
The default quota in MB for each user to be created.

141.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

142 ServiceAttributes-core.sharepoint

143 Sharepoint Service Attributes

Attributes used when configuring the Web Hosting service.

143.1 Contents

- 1 Sharepoint Service Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

143.2 Attributes Body

```
{  
  "domain": "mydomain.com", "is_shared": 0|1, "name": "portalname", "template_key": "templatename"  
}
```

143.2.1 Parameters

domain The domain name of the web hosting site.
is_shared
name
template_key

143.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

144 ServiceAttributes-core.sharepoint2007

145 SharePoint 2007 Attributes

Attributes used when configuring the SharePoint 2007.

145.1 Contents

- 1 SharePoint 2007 Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

145.2 Attributes Body

```
{  
  "domain": "{domain}",  
  "is_shared": 0|1,  
  "name": "{name}",  
  "template_key": "{template_key}"  
}
```

145.2.1 Parameters

domain - *String*
(REQUIRED)
The domain name for the SharePoint product.

is_shared - *Integer*
(REQUIRED)
Flag that specifies whether the domain attribute is shared domain or not. Possible values are **1** or **0** (a.k.a true or false).

name - *String*
(REQUIRED)
Sub domain name which will be the prefix of the actual **FQDN** for the SharePoint portal resource.

template_key - *String*
(OPTIONAL)
Specific template text key for the SharePoint resource.

145.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

146 ServiceAttributes-core.spamexperts

147 AVAS Service Attributes

Attributes used when configuring the Antivirus Antispam service.

147.1 Contents

- 1 AVAS Service Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

147.2 Attributes Body

```
{  
  "domain": "mydomain.com"  
}
```

147.2.1 Parameters

domain The email domain to be filtered.

147.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

148 ServiceAttributes-core.subdomain

149 Sub-Domain Registration Service Attributes

Attributes used when configuring the Sub-Domain Registration service.

149.1 Contents

- [1 Sub-Domain Registration Service Attributes](#)
 - ◆ [1.1 Attributes Body](#)
 - ◇ [1.1.1 Parameters](#)
 - ◆ [1.2 See also](#)

149.2 Attributes Body

```
{  
  "name": "sub-domain",  
  "auto_generate": "0"  
}
```

149.2.1 Parameters

name The name of the sub-domain.

auto_generate When set to *1* forces the sub-domain name to be generated. In this case if the *name* is specified, it will be ignored.

149.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

150 ServiceAttributes-core.vps.virtuozzo.server

151 VPS Virtuozzo Server Attributes

Attributes used when configuring the Virtuozzo server.

151.1 Contents

- 1 VPS Virtuozzo Server Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

151.2 Attributes Body

```
{  
  "serverHostName": "host",  
}
```

151.2.1 Parameters

serverHostName - *String*
(*OPTIONAL*)
The hostname of the new server.

151.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

152 ServiceAttributes-core.webhosting

153 Web Hosting Service Attributes

Attributes used when configuring the Web Hosting service.

153.1 Contents

- [1 Web Hosting Service Attributes](#)
 - ◆ [1.1 Attributes Body](#)
 - ◇ [1.1.1 Parameters](#)
 - ◆ [1.2 See also](#)

153.2 Attributes Body

```
{  
  "domain": "mydomain.com"  
}
```

153.2.1 Parameters

domain The domain name of the web hosting site.

153.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

154 ServiceAttributes-core.websplanet

155 Websplanet Attributes

Attributes used when configuring the Websplanet service.

155.1 Contents

- 1 Websplanet Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

155.2 Attributes Body

```
{  
  "domain": "ng-test-drago-140416a.com",  
  "email": "test@example.com"  
}
```

155.2.1 Parameters

domain Domain name for Websplanet.

email email address.

155.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

156 ServiceAttributes-dedicated.server.firewall

157 Dedicated Server Firewall Service Attributes

Attributes used when configuring the Dedicated Server Firewall service.

157.1 Contents

- 1 Dedicated Server Firewall Service Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

157.2 Attributes Body

```
{  
  "serverFirewallType": "{type}"  
}
```

157.2.1 Parameters

serverFirewallType
Actual type of the dedicated server firewall. Possible values - **shared** or **dedicated**.

157.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

158 ServiceAttributes-dnr.feature.domain guard

159 DomainGuard Service Attributes

Attributes used when configuring the DomainGuard service.

159.1 Contents

- 1 DomainGuard Service Attributes
 - ◆ 1.1 Attributes Body
 - ◇ 1.1.1 Parameters
 - ◆ 1.2 See also

159.2 Attributes Body

```
{  
  "domain": "mydomain.com",  
  "email": "test@email.com",  
  "password": "test1234"  
}
```

159.2.1 Parameters

domain - *String*
The domain name to be protected.

email - *String*
Email address of the DomainGuardian, who is the only person that will have access to modify the guarded domain

password - *String*
The password used for a DomainGuard authorization.

159.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

160 ServiceAttributes-dnr.feature.private registration

161 WhoisProtector(Private Registration) Service Attributes

Attributes used when configuring the WhoisProtector service.

161.1 Contents

- [1 WhoisProtector\(Private Registration\) Service Attributes](#)
 - ◆ [1.1 Attributes Body](#)
 - ◇ [1.1.1 Parameters](#)
 - ◆ [1.2 See also](#)

161.2 Attributes Body

```
{  
  "domain": "mydomain.com",  
  "forwardEmail": "test@email.com"  
}
```

161.2.1 Parameters

domain - *String*
The domain name to be protected.

forwardEmail - *String (OPTIONAL)*
Forward email address. When messages are received at a private WhoisProtector email address, after SPAM filtering, they will be forwarded to this privacy email address.

161.3 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

162 ServiceAttributes-mcafee

163 Excel Micro Service Attributes

No service attributes are available for the Excel micro service-specifications. Attempt to setting or getting attributes for a Excel Micro will result in a **404** error.

163.1 See also

- [Service Attributes Structure](#)
- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)

164 ServiceAttributesStructure

Services supported by Hostway Public Order API:

- [Web Hosting](#)
- [Domain Name Registration](#)
- [Sub-Domain Registration](#)
- [Dedicated Server](#)
- [Dedicated Server Firewall](#)
- [Exchange2010](#)
- [WhoisProtector \(Private Registration\)](#)
- [DomainGuard](#)
- [VPS Virtuozzo Server](#)
- [WebsPlanet](#)
- [Excel Micro](#)
- [Cloud Compute](#)
- [Sharepoint](#)
- [Sharepoint 2007](#)
- [Fax Over Email - User](#)
- [Fax Over Email - Number](#)
- [Online Backup - Account](#)
- [AVAS](#)

164.1 See also

- [Orders API](#)
- [Account Management API](#)

Please enable JavaScript to view the [comments powered by Disqus](#). [blog comments powered by Disqus](#)