Table of Contents

1 Accounts:GET **1**

2 GET /accounts **2**
- 2.1 Contents 2
- 2.2 Request 2
- 2.3 Response 2
- 2.4 Examples 3
- 2.5 See also 4

3 Accounts-accountId:GET **5**

4 GET /accounts/{accountId} **6**
- 4.1 Contents 6
- 4.2 Request 6
- 4.3 Response 6
- 4.4 Conditional Hypermedia 7
- 4.5 Alternative with partner Id 7
- 4.6 See also 8

5 Accounts-accountId:GET **9**

6 GET /accounts/{accountId} **10**
- 6.1 Contents 10
- 6.2 Request 10
- 6.3 Response 10
- 6.4 Conditional Hypermedia 11
- 6.5 Alternative with partner Id 11
- 6.6 See also 12

7 Accounts-accountId:PUT **13**

8 PUT /accounts/{accountId} **14**
- 8.1 Contents 14
- 8.2 Request 14
- 8.3 Response 14
- 8.4 Alternative with partner Id 14
- 8.5 See also 15

9 Accounts-accountId:PUT **16**

10 PUT /accounts/{accountId} **17**
- 10.1 Contents 17
- 10.2 Request 17
- 10.3 Response 17
- 10.4 Alternative with partner Id 17
- 10.5 See also 18

11 Accounts-accountId:DELETE **19**

12 DELETE /accounts/{accountId} **20**
- 12.1 Contents 20
- 12.2 Request 20
- 12.3 Response 20
- 12.4 Alternative with partner Id 20
- 12.5 See also 20

13 Accounts-accountId:DELETE **21**

14 DELETE /accounts/{accountId} **22**
- 14.1 Contents 22
- 14.2 Request 22
- 14.3 Response 22
- 14.4 Alternative with partner Id 22
- 14.5 See also 22

15 Accounts-accountId-action-interFranchiseMigration:GET **23**

16 GET /accounts/{accountId}/action/interFranchiseMigration **24**
- 16.1 Contents 24
- 16.2 Request 24
- 16.3 Response 24
- 16.4 Conditional Hypermedia 25
- 16.5 Examples 25
- 16.6 See also 26

17 Accounts-accountId-action:POST **27**

18 POST /accounts/{accountId}/action **28**
- 18.1 Contents 28
- 18.2 Request 28
- 18.3 Response 28
- 18.4 Examples 28
- 18.5 See also 29

Table of Contents

19 Accounts-accountId-attributes:GET 30

20 GET /accounts/{accountId}/attributes 31
20.1 Contents 31
20.2 Request 31
20.3 Response 31
20.4 Examples 32
20.5 See also 32

21 PartnerAccounts-partnerAccountId-attributes:GET 33

22 GET /partnerAccounts/{partnerAccountId}/attributes 34
22.1 Contents 34
22.2 Request 34
22.3 Response 34
22.4 Examples 35
22.5 See also 35

23 Accounts-accountId-attributes:POST 36

24 POST /accounts/{accountId}/attributes 37
24.1 Contents 37
24.2 Request 37
24.3 Response 37
24.4 Examples 37
24.5 See also 38

25 PartnerAccounts-partnerAccountId-attributes:POST 39

26 POST /partnerAccounts/{partnerAccountId}/attributes 40
26.1 Contents 40
26.2 Request 40
26.3 Response 40
26.4 Examples 41
26.5 See also 41

27 Accounts-accountId-attributes-attribute:GET 42

28 GET /accounts/{accountId}/attributes/{attribute} 43
28.1 Contents 43
28.2 Request 43
28.3 Response 43
28.4 Examples 44
28.5 See also 44

29 PartnerAccounts-partnerAccountId-attributes-attribute:GET 45

30 GET /partnerAccounts/{partnerAccountId}/attributes/{attribute} 46
30.1 Contents 46
30.2 Request 46
30.3 Response 46
30.4 Examples 47
30.5 See also 47

31 Accounts-accountId-attributes-attribute:PUT 48

32 PUT /accounts/{accountId}/attributes/{attribute} 49
32.1 Contents 49
32.2 Request 49
32.3 Response 49
32.4 Examples 49
32.5 See also 50

33 PartnerAccounts-partnerAccountId-attributes-attribute:PUT 51

34 PUT /partnerAccounts/{partnerAccountId}/attributes/{attribute} 52
34.1 Contents 52
34.2 Request 52
34.3 Response 52
34.4 Examples 52
34.5 See also 53

35 Accounts-accountId-attributes-attribute:DELETE 54

36 DELETE /accounts/{accountId}/attributes/{attribute} 55
36.1 Contents 55
36.2 Request 55
36.3 Response 55
36.4 Examples 55
36.5 See also 56

37 PartnerAccounts-partnerAccountId-attributes-attribute:DELETE 57

Table of Contents

38 DELETE /partnerAccounts/{partnerAccountId}/attributes/{attribute} 58
- 38.1 Contents 58
- 38.2 Request 58
- 38.3 Response 58
- 38.4 Examples 58
- 38.5 See also 59

39 Accounts-accountId-Contacts:GET 60

40 GET /accounts/{accountId}/contacts 61
- 40.1 Contents 61
- 40.2 Request 61
- 40.3 Response 61
- 40.4 Examples 63
- 40.5 See also 64

41 Accounts-accountId-Contacts-contactType:GET 65

42 GET /accounts/{accountId}/contacts/{contactType} 66
- 42.1 Contents 66
- 42.2 Request 66
- 42.3 Response 66
- 42.4 Examples 67
- 42.5 See also 68

43 Accounts-accountId-Contacts:PUT 69

44 PUT /accounts/{accountId}/contacts 70
- 44.1 Contents 70
- 44.2 Request 70
- 44.3 Response 71
- 44.4 Examples 72
- 44.5 See also 73

45 Accounts-accountId-Contacts-contactType:PUT 74

46 PUT /accounts/{accountId}/contacts/{contactType} 75
- 46.1 Contents 75
- 46.2 Request 75
- 46.3 Response 76
- 46.4 Examples 76
- 46.5 See also 78

47 Initiate-Email-Verification-Action:POST 79

48 POST /accounts/{account_number}/contacts/action 80
- 48.1 Contents 80
- 48.2 Request 80
- 48.3 Response 80
- 48.4 Examples 80
- 48.5 See also 81

49 ZenDesk-SSO-Link-Per-Contact-Type:GET 82

50 GET /accounts/{accountId}/contacts/{contactType}/supportSSOLink 83
- 50.1 Contents 83
- 50.2 Request 83
- 50.3 Response 83
- 50.4 Examples 83
- 50.5 See also 84

51 Accounts-accountId-users:GET 85

52 GET /accounts/{accountId}/users 86
- 52.1 Contents 86
- 52.2 Request 86
- 52.3 Response 86
- 52.4 See Also 87

53 Accounts-accountId-users:POST 88

54 POST /accounts/{accountId}/users 89
- 54.1 Contents 89
- 54.2 Request 89
- 54.3 Response 89
- 54.4 Allowed Security Questions 90
- 54.5 See also 90

55 Accounts-accountId-users-userName:GET 91

56 GET /accounts/{accountId}/users/{userName} 92
- 56.1 Contents 92
- 56.2 Request 92
- 56.3 Response 92
- 56.4 See also 92

Table of Contents

57 Accounts-accountId-users-userName:PUT 94

58 PUT /accounts/{accountId}/users/{userName} 95
- 58.1 Contents 95
- 58.2 Request 95
- 58.3 Response 95
- 58.4 See also 96

59 Accounts-accountId-user:PUT 97

60 PUT /accounts/{accountId}/user 98
- 60.1 Contents 98
- 60.2 Request 98
- 60.3 Response 98
- 60.4 Examples 99
- 60.5 See also 100

61 Accounts-accountId-users-userName:DELETE 101

62 DELETE /accounts/{accountId}/users/{userName} 102
- 62.1 Contents 102
- 62.2 Request 102
- 62.3 Response 102
- 62.4 See also 102

63 Accounts-accountId-users-userName-roles:GET 103

64 GET /accounts/{accountId}/users/{userName}/roles 104
- 64.1 Contents 104
- 64.2 Request 104
- 64.3 Response 104
- 64.4 See also 104

65 Accounts-accountId-users-userName-roles:PUT 105

66 PUT /accounts/{accountId}/users/{userName}/roles 106
- 66.1 Contents 106
- 66.2 Request 106
- 66.3 Response 106
- 66.4 See also 106

67 Accounts-accountId-products:GET 107

68 GET /accounts/{accountId}/products 108
- 68.1 Contents 108
- 68.2 Request 108
- 68.3 Response 108
- 68.4 Alternative with partner Id 109
- 68.5 See also 109

69 Accounts-accountId-products-productId:GET 110

70 GET /accounts/{accountId}/products/{productId} 111
- 70.1 Contents 111
- 70.2 Request 111
- 70.3 Response 111
- 70.4 See also 112

71 Accounts-accountId-products-productId:PUT 113

72 PUT /accounts/{accountId}/products/{productId}' 114
- 72.1 Contents 114
- 72.2 Request 114
- 72.3 Response 114
- 72.4 See also 114

73 Accounts-accountId-products-productId-action:POST 115

74 POST /accounts/{accountId}/products/{productId}/action 116
- 74.1 Contents 116
- 74.2 Request 116
- 74.3 Response 116
- 74.4 See also 116

75 Accounts-accountId-products-productId-services:GET 118

76 GET /accounts/{accountId}/products/{productId}/services 119
- 76.1 Contents 119
- 76.2 Request 119
- 76.3 Response 119
- 76.4 See also 120

77 Accounts-accountId-products-productId-services-serviceId:GET 121

Table of Contents

78 GET /accounts/{accountId}/products/{productId}/services/{serviceId} 122
78.1 Contents 122
78.2 Request 122
78.3 Response 122
78.4 See also 123

79 Accounts-accountId-products-productId-services-serviceId:PUT 124

80 PUT /accounts/{accountId}/products/{productId}/services/{serviceId}' 125
80.1 Contents 125
80.2 Request 125
80.3 Response 125
80.4 See also 125

81 Accounts-accountId-products-productId-services-serviceId-getTargetServiceSpecifications:GET 126

82 GET /accounts/{accountId}/products/{productId}/services/{serviceId}/?getTargetServiceSpecifications=1 127
82.1 Contents 127
82.2 Request 127
82.3 Response 127
82.4 See also 128

83 Accounts-accountId-products-productId-services-serviceId-action:POST 129

84 POST /accounts/{accountId}/products/{productId}/services/{serviceId}/action 130
84.1 Contents 130
84.2 Request 130
84.3 Response 130
84.4 See also 131

85 Accounts-accountId-products-productId-bandwidth-daily:GET 132

86 GET /accounts/{accountId}/products/{productId}/bandwidth/daily?filterType={filterType}&from={filterStartDate}&to={filterEndDate} 133
86.1 Contents 133
86.2 Request 133
86.3 Response 133
86.4 Examples 134
86.5 See also 134

87 Accounts-accountId-products-productId-bandwidth-weekly:GET 135

88 GET /accounts/{accountId}/products/{productId}/bandwidth/weekly?filterType={filterType}&from={filterStartDate}&to={filterEndDate} 136
88.1 Contents 136
88.2 Request 136
88.3 Response 136
88.4 Examples 137
88.5 See also 138

89 Accounts-accountId-products-productId-bandwidth-monthly:GET 139

90 GET /accounts/{accountId}/products/{productId}/bandwidth/monthly?filterType={filterType}&from={filterStartDate}&to={filterEndDate} 140
90.1 Contents 140
90.2 Request 140
90.3 Response 140
90.4 Examples 141
90.5 See also 142

91 Accounts-accountId-products-productId-bandwidth-yearly:GET 143

92 GET /accounts/{accountId}/products/{productId}/bandwidth/yearly?filterType={filterType}&from={filterStartDate}&to={filterEndDate} 144
92.1 Contents 144
92.2 Request 144
92.3 Response 144
92.4 Examples 145
92.5 See also 145

93 Accounts-accountId-products-productId-usage:GET 146

94 GET /accounts/{accountId}/products/{productId}/usage?from={fromDate}&to={toDate}&rate=true 147
94.1 Contents 147
94.2 Request 147
94.3 Response 147
94.4 Examples 149
94.5 See also 150

95 Accounts-accountId-paymentInfo:GET 151

96 GET /accounts/{accountId}/paymentInfo 152
96.1 Contents 152
96.2 Request 152
96.3 Response 152
96.4 See also 153

Table of Contents

97 Accounts-accountId-paymentInfo:POST 154

98 POST /accounts/{accountId}/paymentInfo 155
98.1 Contents 155
98.2 Request 155
98.3 Response 156
98.4 Examples 157
98.5 See also 157

99 Accounts-accountId-paymentInfo-default:PUT 158

100 PUT /accounts/{accountId}/paymentInfo/default 159
100.1 Contents 159
100.2 Request 159
100.3 Response 159
100.4 See also 159

101 Accounts-accountId-paymentInfo-default:DELETE 160

102 DELETE /accounts/{accountId}/paymentInfo/default 161
102.1 Contents 161
102.2 Request 161
102.3 Response 161
102.4 See also 161

103 Accounts-accountId-paymentInfo-paymentInfoId:GET 162

104 GET /accounts/{accountId}/paymentInfo/{paymentInfoId} 163
104.1 Contents 163
104.2 Request 163
104.3 Response 163
104.4 See also 164

105 Accounts-accountId-paymentInfo-paymentInfoId:DELETE 165

106 DELETE /accounts/{accountId}/paymentInfo/{paymentInfoId} 166
106.1 Contents 166
106.2 Request 166
106.3 Response 166
106.4 See also 166

107 Accounts-accountId-paymentInfo-paymentInfoId-payment:POST 167

108 POST /accounts/{accountId}/paymentInfo/{paymentInfoId}/payment 168
108.1 Contents 168
108.2 Request 168
108.3 Response 168
108.4 Examples 169
108.5 See also 169

109 Accounts-accountId-transactionsInfo:GET 170

110 GET /accounts/{accountId}/transactions 171
110.1 Contents 171
110.2 Request 171
110.3 Response 171
110.4 Examples 172
110.5 See also 172

111 Accounts-accountId-transactionsInfo-filtered-type:GET 173

112 GET /accounts/{accountId}/transactions?filterType={filterType} 174
112.1 Contents 174
112.2 Request 174
112.3 Response 174
112.4 Examples 175
112.5 See also 175

113 Accounts-accountId-transactionsInfo-sorted:GET 176

114 GET /accounts/{accountId}/transactions?sortField={sortField}&sortOrder={sortOrder} 177
114.1 Contents 177
114.2 Request 177
114.3 Response 177
114.4 Examples 178
114.5 See also 178

115 Accounts-accountId-invoicesInfo:GET 179

116 GET /accounts/{accountId}/invoices 180
116.1 Contents 180
116.2 Request 180
116.3 Response 180
116.4 Examples 181
116.5 See also 181

Table of Contents

117 Accounts-accountId-invoicesInfo-filtered-status:GET 182

118 GET /accounts/{accountId}/invoice?filterStatus={filterStatus} 183
- 118.1 Contents 183
- 118.2 Request 183
- 118.3 Response 183
- 118.4 Examples 184
- 118.5 See also 184

119 Accounts-accountId-invoicesInfo-sorted:GET 185

120 GET /accounts/{accountId}/invoice?sortField={sortField} 186
- 120.1 Contents 186
- 120.2 Request 186
- 120.3 Response 186
- 120.4 Examples 187
- 120.5 See also 187

121 Accounts-accountId-invoicesInfo-invoiceId:GET 188

122 GET /accounts/{accountId}/invoces/{invoiceId} 189
- 122.1 Contents 189
- 122.2 Request 189
- 122.3 Response 189
- 122.4 Examples 190
- 122.5 See also 190

123 Accounts-accountId-invoices-invoiceId-transactions:GET 191

124 GET /accounts/{accountId}/invoices/{invoiceId}/transactions 192
- 124.1 Contents 192
- 124.2 Request 192
- 124.3 Response 192
- 124.4 Examples 193
- 124.5 See also 193

125 Accounts-accountId-discount:GET 194

126 GET /accounts/{accountId}/discount 195
- 126.1 Contents 195
- 126.2 Request 195
- 126.3 Response 195
- 126.4 Examples 195
- 126.5 See also 196

127 Accounts-accountId-os-security-groups:GET 197

128 GET /accounts/{accountId}/os-security-groups 198
- 128.1 Request 198
- 128.2 Response 198
- 128.3 Response Codes 198
- 128.4 Examples 198
- 128.5 Alternative URI /accounts/os-security-groups 200

129 Accounts-accountId-os-security-groups:POST 201

130 POST /accounts/{accountId}/os-security-groups 202
- 130.1 Request 202
- 130.2 Response 202
- 130.3 Response Codes 202
- 130.4 Examples 202
- 130.5 Alternative URI /accounts/os-security-groups 204

131 Accounts-accountId-os-security-groups-groupId:GET 205

132 GET /accounts/{accountId}/os-security-groups/{groupId} 206
- 132.1 Parameters 206
- 132.2 Response 206
- 132.3 Response Codes 206
- 132.4 Examples 206
- 132.5 Alternative URI /accounts/os-security-groups/{groupId} 208

133 Accounts-accountId-os-security-groups-groupId:PUT 209

134 PUT /accounts/{accountId}/os-security-groups/{groupId} 210
- 134.1 Parameters 210
- 134.2 Request 210
- 134.3 Response 210
- 134.4 Response Codes 210
- 134.5 Examples 210

135 Accounts-accountId-os-security-groups-groupId:DELETE 212

Table of Contents

136 DELETE /accounts/{accountId}/os-security-groups/{groupId} 213
136.1 Parameters 213
136.2 Response Codes 213
136.3 Examples 213
136.4 Alternative URI /accounts/os-security-groups/{groupId} 214

137 Accounts-accountId-os-security-group-rules:POST 215

138 POST /accounts/{accountId}/os-security-group-rules 216
138.1 Request 216
138.2 Response 216
138.3 Response Codes 217
138.4 Examples 217
138.5 Alternative URI /accounts/os-security-group-rules 219

139 Accounts-accountId-os-security-group-rules-ruleId:DELETE 221

140 DELETE /accounts/{accountId}/os-security-group-rules/{ruleId} 222
140.1 Parameters 222
140.2 Response Codes 222
140.3 Examples 222
140.4 Alternative URI /accounts/os-security-group-rules/{ruleId} 223

141 Common Features 224

1 Accounts:GET

2 GET /accounts

Retrieves a list of all accounts authorized for access with the provided credentials.

2.1 Contents

- 1 GET /accounts
  - 1.1 Request
    - 1.1.1 URI Parameters
    - 1.1.2 Request Headers
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 Listing current account only when provided credentials don't give elevated access
    - 1.3.2 Paginated request when more then one account is returned
  - 1.4 See also

2.2 Request

```
GET /accounts
```

2.2.1 URI Parameters

filterStatus - *string*
: Optional. Only accounts with status matching the value of the parameter will be returned. Allowed values are **pending**, **open**, **suspended**, **closed**.

page - *integer*
: Optional. Specifies which page should be displayed. Requires a pageSize parameter to also be provided.

pageSize - *integer*
: Optional. Specifies the number of entries to be displayed on a page. Default size is 100. Requires a page parameter to also be provided.

2.2.2 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

2.3 Response

2.3.1 Status Code

200 OK
: Success

400 Bad Request

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: Forbidden access to a user not authorized for such access.

415 Unsupported Media Type
: The Content-Type header was not specified or the specified type is not supported. Use application/json.

2.3.2 Response Body

```
{
  "list" : [{
    "status" : "{status}",
    "links" : [{
      "href" : "https://api.hostway.com/accounts/{account-number}/",
      "rel" : "self"
    }],
    "accountNumber" : "{account-number}",
    "createdDate" : "{created-date}",
    "currency": "{currency}"
  }],
  "links" : [{
      "href" : "https://api.hostway.com/accounts?pageSize={pageSize}&page={page}",
      "rel" : "self"
    },
    {
      "href" : "https://api.hostway.com/accounts?pageSize={pageSize}&page={page}",
      "rel" : "next"
    },
    {
      "href" : "https://api.hostway.com/accounts?pageSize={pageSize}&page={page}",
      "rel" : "last"
  }]
}
```

2.3.2.1 Parameters

list - *list*
A collection of dictionaries containing information about specific account

status - *string*
Describes the status of the account. Valid values are **pending**, **open**, **suspended**, **closed**

accountNumber - *string*
The Id of the account

createdDate - *date*
The date account was created in format Y-m-d

currency - *string*
An ISO-4217[2] currency code assigned to the account.

links - *list*
Hypermedia for this account resource. Links to the account.

links - *list*
Hypermedia Link to self and other resources relevant to the account lists

2.4 Examples

2.4.1 Listing current account only when provided credentials don't give elevated access

Request

```
GET /accounts
```

Response

```
{
   "list": [{
      "status": "open",
      "links": [{
         "href": "https://api.hostway.com/accounts/test-account/",
         "rel": "self"
      }],
      "accountNumber": "test-account",
      "createdDate": "2009-12-07",
      "currency": "USD"
   }],
   "links": [{
      "href": "https://api.hostway.com/accounts",
      "rel": "self"
   }]
}
```

2.4.2 Paginated request when more then one account is returned

Request

```
GET /accounts?pageSize=2&page=1
```

Response

```
{
   "list" : [{
         "status" : "open",
         "links" : [{
            "href" : "https://api.hostway.com/accounts/test-account-1/",
            "rel" : "self"
         }],
         "accountNumber" : "test-account-1",
         "createdDate": "2009-12-07",
         "currency": "USD"
      },
         "status" : "closed",
         "links" : [{
            "href" : "https://api.hostway.com/accounts/test-account-2/",
            "rel" : "self"
         }],
         "accountNumber" : "test-account-2",
         "createdDate": "20012-11-01",
         "currency": "CAD"
      }],
   "links" : [{
         "href" : "https://api.hostway.com/accounts?pageSize=2&page=1",
         "rel" : "self"
      },
      {
         "href" : "https://api.hostway.com?pageSize=2&page=2",
         "rel" : "next"
      },
      {
         "href" : "https://api.hostway.com?pageSize=2&page=11637",
         "rel" : "last"
      }]
}
```

2.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

3 Accounts-accountId:GET

4 GET /accounts/*{accountId}*

Get details about an account.

4.1 Contents

- 1 GET /accounts/{accountId}
  - 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 Conditional Hypermedia
  - 1.4 Alternative with partner Id
    - 1.4.1 Parameters
  - 1.5 See also

4.2 Request

`GET /accounts/{accountId}`

4.2.1 Parameters

accountId - *string*
: The Id of the account.

4.2.2 URI Parameters

None.

4.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

4.3 Response

4.3.1 Status Code

200 OK
: Success.

403 Forbidden
: Forbidden access to **closed** and **suspended** accounts by a user not authorized for such access.

404 Not Found
: The account Id was not found.

4.3.2 Response Body

```
{
  "accountNumber": "{accountId}",
  "createdDate": "{createdDate}",
  "currency": "{currency}",
  "status": "open",
  "paymentInfo": [
    {"href": "https://api.hostway.com/accounts/{account-number}/paymentInfo", "rel": "related"}
  ],
  "products": [
    {"href": "https://api.hostway.com/accounts/{account-number}/products", "rel": "related"}
  ],
  "exchange": [
    {"href": "https://api.hostway.com/accounts/{account-number}/exchange", "rel": "related"}
  ],
  "dedicatedServers": [
    {"href": "https://api.hostway.com/accounts/{account-number}/dedicatedServers", "rel": "related"}
  ],
```

```
  "contacts": [
    {"href": "https://api.hostway.com/accounts/{account-number}/contacts", "rel": "related"}
  ],
  "email": [
    {"href": "https://api.hostway.com/accounts/{account-number}/email", "rel": "related"}
  ],
  "domains": [
    {"href": "https://api.hostway.com/accounts/{account-number}/domains", "rel": "related"}
  ],
  "attributes": [
    {"href": "https://api.hostway.com/accounts/{account-number}/attributes", "rel": "related"}
  ],
  "flexcloud": [
     {"href": "https://api.hostway.com/accounts/{account-number}/flexcloud", "rel": "related"}
  ],
  "users": [
     {"href": "https://api.hostway.com/accounts/{account-number}/users", "rel": "related"}
  ],
  "links": [
    {"href": "https://api.hostway.com/accounts/{account-number}/", "rel": "self"}
  ]
}
```

4.3.2.1 Parameters

accountNumber - *string*
: The Id of the account

createdDate - *date*
: The date account was created in format Y-m-d

currency - *string*
: An ISO-4217[2] currency code assigned to the account.

status - *string*
: status of the account. Valid values are **open**, **closed** or **suspended**. Accounts with certain status may not be accessible by regular users.

paymentInfo - *list*
: *(OPTIONAL)*
Hypermedia to a resource of the payment instruments associated with the account.

users - *list*
: *(OPTIONAL)*
Hypermedia to a resource of the users associated with the account.

products - *list*
: *(OPTIONAL)*
Hypermedia to a resource containing all the products of the account. This value will appear depending on the user's access rights.

dedicatedServers - *list*
: *(OPTIONAL)*
Hypermedia to a resource containing all the dedicated server products of the account.

exchange - *list*
: *(OPTIONAL)*
Hypermedia to a resource containing all the Exchange domains of the account.

contacts - *list*
: *(OPTIONAL)*
Hypermedia to a resource handling the contact details of the account.

email - *list*
: *(OPTIONAL)*
Hypermedia to a resource containing all the email domains of the account.

domains - *list*
: *(OPTIONAL)*
Hypermedia linking to the domain name management of the account.

attributes - *list*
: *(OPTIONAL)*
Hypermedia to a partner attributes of the account.

flexcloud - *list*
: *(OPTIONAL)*
Hypermedia to the Flexcloud management of the account.

links - *list*
: Hypermedia for this resource. Links to self.

4.4 Conditional Hypermedia

Hypermedia fields marked as optional will appear on the basis of the accessing user's permissions.

4.5 Alternative with partner Id

As an alternative to using the account Id the client can use a partner-specific Id set via the account attribute partner_account_id. The response to this request is the same as the response to a request using the account Id.

```
GET /partnerAccounts/{partnerAccountId}
```

4.5.1 Parameters

partnerAccountId - *string*
Partner account Id set via the partner_account_id attribute.

4.6 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

5 Accounts-accountId:GET

6 GET /accounts/*{accountId}*

Get details about an account.

6.1 Contents

- 1 GET /accounts/{accountId}
  - 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 Conditional Hypermedia
  - 1.4 Alternative with partner Id
    - 1.4.1 Parameters
  - 1.5 See also

6.2 Request

`GET /accounts/{accountId}`

6.2.1 Parameters

accountId - *string*
: The Id of the account.

6.2.2 URI Parameters

None.

6.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

6.3 Response

6.3.1 Status Code

200 OK
: Success.

403 Forbidden
: Forbidden access to **closed** and **suspended** accounts by a user not authorized for such access.

404 Not Found
: The account Id was not found.

6.3.2 Response Body

```
{
  "accountNumber": "{accountId}",
  "createdDate": "{createdDate}",
  "currency": "{currency}",
  "status": "open",
  "paymentInfo": [
    {"href": "https://api.hostway.com/accounts/{account-number}/paymentInfo", "rel": "related"}
  ],
  "products": [
    {"href": "https://api.hostway.com/accounts/{account-number}/products", "rel": "related"}
  ],
  "exchange": [
    {"href": "https://api.hostway.com/accounts/{account-number}/exchange", "rel": "related"}
  ],
  "dedicatedServers": [
    {"href": "https://api.hostway.com/accounts/{account-number}/dedicatedServers", "rel": "related"}
  ],
```

```
  "contacts": [
    {"href": "https://api.hostway.com/accounts/{account-number}/contacts", "rel": "related"}
  ],
  "email": [
    {"href": "https://api.hostway.com/accounts/{account-number}/email", "rel": "related"}
  ],
  "domains": [
    {"href": "https://api.hostway.com/accounts/{account-number}/domains", "rel": "related"}
  ],
  "attributes": [
    {"href": "https://api.hostway.com/accounts/{account-number}/attributes", "rel": "related"}
  ],
  "flexcloud": [
     {"href": "https://api.hostway.com/accounts/{account-number}/flexcloud", "rel": "related"}
  ],
  "users": [
     {"href": "https://api.hostway.com/accounts/{account-number}/users", "rel": "related"}
  ],
  "links": [
    {"href": "https://api.hostway.com/accounts/{account-number}/", "rel": "self"}
  ]
}
```

6.3.2.1 Parameters

accountNumber - *string*
: The Id of the account

createdDate - *date*
: The date account was created in format Y-m-d

currency - *string*
: An ISO-4217[2] currency code assigned to the account.

status - *string*
: status of the account. Valid values are **open**, **closed** or **suspended**. Accounts with certain status may not be accessible by regular users.

paymentInfo - *list*
: *(OPTIONAL)*
Hypermedia to a resource of the payment instruments associated with the account.

users - *list*
: *(OPTIONAL)*
Hypermedia to a resource of the users associated with the account.

products - *list*
: *(OPTIONAL)*
Hypermedia to a resource containing all the products of the account. This value will appear depending on the user's access rights.

dedicatedServers - *list*
: *(OPTIONAL)*
Hypermedia to a resource containing all the dedicated server products of the account.

exchange - *list*
: *(OPTIONAL)*
Hypermedia to a resource containing all the Exchange domains of the account.

contacts - *list*
: *(OPTIONAL)*
Hypermedia to a resource handling the contact details of the account.

email - *list*
: *(OPTIONAL)*
Hypermedia to a resource containing all the email domains of the account.

domains - *list*
: *(OPTIONAL)*
Hypermedia linking to the domain name management of the account.

attributes - *list*
: *(OPTIONAL)*
Hypermedia to a partner attributes of the account.

flexcloud - *list*
: *(OPTIONAL)*
Hypermedia to the Flexcloud management of the account.

links - *list*
: Hypermedia for this resource. Links to self.

6.4 Conditional Hypermedia

Hypermedia fields marked as optional will appear on the basis of the accessing user's permissions.

6.5 Alternative with partner Id

As an alternative to using the account Id the client can use a partner-specific Id set via the account attribute partner_account_id. The response to this request is the same as the response to a request using the account Id.

```
GET /partnerAccounts/{partnerAccountId}
```

6.5.1 Parameters

partnerAccountId - *string*
: Partner account Id set via the partner_account_id attribute.

6.6 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

7 Accounts-accountId:PUT

8 PUT /accounts/*{accountId}*

Changes the status of an account. This operation requires elevated permissions.

8.1 Contents

- 1 PUT /accounts/{accountId}
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 Alternative with partner Id
    - 1.3.1 Parameters
  - 1.4 See also

8.2 Request

`PUT /accounts/{accountId}`

8.2.1 Parameters

accountId - *string*
: The Id of the account.

8.2.2 URI Parameters

None.

8.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

8.2.4 Request Body

```
{
  "status": "suspended"
}
```

8.2.4.1 Parameters

status - *string*
: The new status of the account. Accepted values are **open**, **closed** or **suspended**.

8.3 Response

8.3.1 Status Code

204 No Content
: Successful account status change

404 Not Found
: The account Id was not found.

8.3.2 Response Body

None.

8.4 Alternative with partner Id

As an alternative to using the account Id the client can use a partner-specific Id set via the account attribute partner_account_id. The response to this request is the same as the response to a request using the account Id.

```
PUT /partnerAccounts/{partnerAccountId}
```

8.4.1 Parameters

partnerAccountId - *string*
: Partner account Id set via the partner_account_id attribute.

8.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

9 Accounts-accountId:PUT

10 PUT /accounts/*{accountId}*

Changes the status of an account. This operation requires elevated permissions.

10.1 Contents

- 1 PUT /accounts/{accountId}
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 Alternative with partner Id
    - 1.3.1 Parameters
  - 1.4 See also

10.2 Request

`PUT /accounts/{accountId}`

10.2.1 Parameters

accountId - *string*
: The Id of the account.

10.2.2 URI Parameters

None.

10.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

10.2.4 Request Body

```
{
  "status": "suspended"
}
```

10.2.4.1 Parameters

status - *string*
: The new status of the account. Accepted values are **open**, **closed** or **suspended**.

10.3 Response

10.3.1 Status Code

204 No Content
: Successful account status change

404 Not Found
: The account Id was not found.

10.3.2 Response Body

None.

10.4 Alternative with partner Id

As an alternative to using the account Id the client can use a partner-specific Id set via the account attribute partner_account_id. The response to this request is the same as the response to a request using the account Id.

`PUT /partnerAccounts/{partnerAccountId}`

10.4.1 Parameters

partnerAccountId - *string*
: Partner account Id set via the partner_account_id attribute.

10.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

11 Accounts-accountId:DELETE

12 DELETE /accounts/*{accountId}*

Purges a user account. This call requires elevated permissions.

12.1 Contents

- 1 DELETE /accounts/{accountId}
  - 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
  - 1.3 Alternative with partner Id
    - 1.3.1 Parameters
  - 1.4 See also

12.2 Request

`DELETE /accounts/{accountId}`

12.2.1 Parameters

accountId - *string*
: The Id of the account.

12.2.2 URI Parameters

None.

12.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

12.3 Response

12.3.1 Status Code

204 No Content
: Success.

404 Not Found
: The account Id was not found.

12.3.2 Response Body

None.

12.4 Alternative with partner Id

As an alternative to using the account Id the client can use a partner-specific Id set via the account attribute partner_account_id. The response to this request is the same as the response to a request using the account Id.

`DELETE /partnerAccounts/{partnerAccountId}`

12.4.1 Parameters

partnerAccountId - *string*
: Partner account Id set via the partner_account_id attribute.

12.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

13 Accounts-accountId:DELETE

14 DELETE /accounts/*{accountId}*

Purges a user account. This call requires elevated permissions.

14.1 Contents

- 1 DELETE /accounts/{accountId}
  - 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
  - 1.3 Alternative with partner Id
    - 1.3.1 Parameters
  - 1.4 See also

14.2 Request

`DELETE /accounts/{accountId}`

14.2.1 Parameters

accountId - *string*
: The Id of the account.

14.2.2 URI Parameters

None.

14.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

14.3 Response

14.3.1 Status Code

204 No Content
: Success.

404 Not Found
: The account Id was not found.

14.3.2 Response Body

None.

14.4 Alternative with partner Id

As an alternative to using the account Id the client can use a partner-specific Id set via the account attribute partner_account_id. The response to this request is the same as the response to a request using the account Id.

`DELETE /partnerAccounts/{partnerAccountId}`

14.4.1 Parameters

partnerAccountId - *string*
: Partner account Id set via the partner_account_id attribute.

14.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

15 Accounts-accountId-action-interFranchiseMigration:GET

16 GET /accounts/*{accountId}*/action/interFranchiseMigration

Get details about an account's migration to another franchise

16.1 Contents

- 1 GET /accounts/{accountId}/action/interFranchiseMigration
  - 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 Conditional Hypermedia
  - 1.4 Examples
    - 1.4.1 Requesting the status of a migration that has been submitted but not started
    - 1.4.2 Requesting the status of a migration that is being processed (10% done)
    - 1.4.3 Requesting the status of a migration that is being processed (70% done)
    - 1.4.4 Requesting the status of a migration that is finished
    - 1.4.5 Requesting the status of a migration that was never started
    - 1.4.6 Requesting the status of a migration that does not exist
  - 1.5 See also

16.2 Request

`GET /accounts/{accountId}/action/interFranchiseMigration`

16.2.1 Parameters

accountId - *string*
: The Id of the account.

16.2.2 URI Parameters

None.

16.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

16.3 Response

16.3.1 Status Code

200 OK
: Success.

403 Forbidden
: Forbidden access to accounts by a user not authorized for such access.

404 Not Found
: The account Id was not found or a migration request has not been created for this account.

16.3.2 Response Body

```
{
  "status": "{status}",
  "progress": {progress},
  "links": [
    {"href": "https://api.hostway.com/accounts/{account-number}/action/interFranchiseMigration", "rel": "self"}
  ]
}
```

16.3.2.1 Parameters

status - *string*
: The status of the migration. Valid values are **pending**, **complete**, **inprogress** or **unknown** - in case there is no InterFranchise migration started on the account, for this case the progress is None.

progress - *float*
: A float from 0 to 1 representing the migration request's progress. The progress would indicated which is the current flow of the InterFranchise migration process - 0.1 would indicate that it is on the first flow, 0.9 on the last, 1 would indicated that the migration was completed. Can be *null* if the status is **unknown**.

newAccountId - *string*
: The newly-generated accountId for the target account in the target franchise. Available only when the status is **complete**.

newFranchise - *string*
: The target franchise this account was migrated to. Available only when the status is **complete**.

links - *list*
Hypermedia for this resource. Links to self.

16.4 Conditional Hypermedia

Hypermedia fields marked as optional will appear on the basis of the accessing user's permissions.

16.5 Examples

16.5.1 Requesting the status of a migration that has been submitted but not started

Request

```
GET {{APIBaseURL}}/accounts/account123/action/interFranchiseMigration
```

Response

```
HTTP/1.1 200 OK
```

```
{
  "status": "pending",
  "progress": 0,
  "links": [
    {"href": "{{APIBaseURL}}/accounts/account123/action/interFranchiseMigration", "rel": "self"}
  ]
}
```

16.5.2 Requesting the status of a migration that is being processed (10% done)

Request

```
GET {{APIBaseURL}}/accounts/account123/action/interFranchiseMigration
```

Response

```
HTTP/1.1 200 OK
```

```
{
  "status": "inprogress",
  "progress": 0.1,
  "links": [
    {"href": "{{APIBaseURL}}/accounts/account123/action/interFranchiseMigration", "rel": "self"}
  ]
}
```

16.5.3 Requesting the status of a migration that is being processed (70% done)

Request

```
GET {{APIBaseURL}}/accounts/account123/action/interFranchiseMigration
```

Response

```
HTTP/1.1 200 OK
```

```
{
  "status": "inprogress",
  "progress": 0.7,
  "links": [
    {"href": "{{APIBaseURL}}/accounts/account123/action/interFranchiseMigration", "rel": "self"}
  ]
}
```

16.5.4 Requesting the status of a migration that is finished

Request

```
GET {{APIBaseURL}}/accounts/account123/action/interFranchiseMigration
```

Response

```
HTTP/1.1 200 OK
```

```
{
  "status": "complete",
  "progress": 1,
  "newAccountId": "acc853234524",
  "newFranchise": "domainpeople_us_main",
  "links": [
    {"href": "{{APIBaseURL}}/accounts/account123/action/interFranchiseMigration", "rel": "self"}
  ]
}
```

16.5.5 Requesting the status of a migration that was never started

Request

```
GET {{APIBaseURL}}/accounts/account123/action/interFranchiseMigration
```

Response

```
HTTP/1.1 200 OK
```

```
{
  "status": "unknown",
  "progress": null,
  "links": [
    {"href": "{{APIBaseURL}}/accounts/account123/action/interFranchiseMigration", "rel": "self"}
  ]
}
```

16.5.6 Requesting the status of a migration that does not exist

Request

```
GET {{APIBaseURL}}/accounts/account123/action/interFranchiseMigration
```

Response

```
HTTP/1.1 404 Not Found
```

```
{
   "itemNotFound" : {
      "message": "Resource not found",
      "code": 404,
      "details": "",
      "guid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
}
```

16.6 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

17 Accounts-accountId-action:POST

18 POST /accounts/*{accountId}*/action

Initiate an action on an account

18.1 Contents

- 1 POST /accounts/{accountId}/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
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 Examples
    - 1.3.1 Migrate an account to franchise "franchise-2" and set the new partner account Id to "partner1"
  - 1.4 See also

18.2 Request

`POST /accounts/{accountId}/action`

18.2.1 Parameters

accountId - *string*
: The Id of the account.

18.2.2 URI Parameters

None.

18.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

18.2.4 Request Body

```
{
  "action": "{action}",
  "actionData": {actionData}
}
```

18.2.4.1 Parameters

action - *string*
: The action that should be performed. Allowed values are:
  interFranchiseMigration - Submits a request to migrate this account to another franchise.

actionData - *dictionary*
: Optional. Reserved for data specific for the requested action.

18.3 Response

18.3.1 Status Code

202 Accepted
: The request has been successfully initiated. Check the action status for future updates.

400 Bad Request
: The request body is not formatted correctly, is missing a required field, or contains an invalid value.

403 Forbidden
: Forbidden access to users that are not authorized for such access.

404 Not Found
: The account Id was not found.

409 Conflict
: An action has already been submitted for this account and is in progress.

18.3.2 Response Body

None

18.4 Examples

18.4.1 Migrate an account to franchise "franchise-2" and set the new partner account Id to "partner1"

Request

```
POST {{APIBaseURL}}/accounts/account123/action
Content-Type:application/json; Charset=UTF-8
```

```
{
  action: "interFranchiseMigration",
  actionData: {
    franchise: "franchise-2",
    partnerAccountId: "partner1"
  }
}
```

Response

```
HTTP/1.1 202 Accepted
```

18.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

19 Accounts-accountId-attributes:GET

20 GET /accounts/*{accountId}*/attributes

Get attributes list.

20.1 Contents

- 1 GET /accounts/{accountId}/attributes
  - 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 Examples
    - 1.3.1 Sending unauthorized request
    - 1.3.2 Sending request without setting Content-Type
    - 1.3.3 Getting list of attributes of not existing account
    - 1.3.4 Success scenario
  - 1.4 See also

20.2 Request

`GET /accounts/{accountId}/attributes`

20.2.1 Parameters

accountId - *string*
: The Id of the Hostway account.

20.2.2 URI Parameters

None.

20.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Content-Type
: application/json

20.3 Response

20.3.1 Status Code

200 OK
: Success.

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: Forbidden access to a user not authorized for such access.

404 Not Found
: The account Id or attribute were not found.

415 Unsupported Media Type
: The Content-Type header was not specified or the specified type is not supported. Use application/json.

20.3.2 Response Body

```
{
  "list": [
    "name": "{name}",
    "value": "{value}",
    "links" : [{"href": "https://api.hostway.com/accounts/{accountId}/attributes/{name}", "rel": "related"}]
  ],
  "links": [
    {"href": "https://api.hostway.com/accounts/{accountId}/attributes", "rel": "self"}
  ]
}
```

20.3.2.1 Parameters

list - *list*
: A collection of dictionaries containing information about specific attribute

  name - *string*
  : Descriptive name of the attribute

value - *string*
Value of the attribute

NOTE: The "partner_account_id" attribute will not be returned in this list.

links - *list*
Hypermedia for this product resource. Links to the attribute.

links - *list*
Hypermedia for this resource. Links to self.

20.4 Examples

20.4.1 Sending unauthorized request

Request

```
GET /accounts/account-id/attributes
```

Response

```
401 Unauthorized
```

20.4.2 Sending request without setting Content-Type

Request

```
GET /accounts/account-id/attributes
Authorization: Basic base64_encode(username:password)
```

Response

```
415 Unsupported Media Type
```

20.4.3 Getting list of attributes of not existing account

Request

```
GET /accounts/not-existing-account-id/attributes
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
404 Not Found
```

20.4.4 Success scenario

Request

```
GET /accounts/account-id/attributes
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
200 OK

{
    "list": [
        {
            "name": "test_attribute",
            "value": "123",
            "links": [{"href": "{{APIBaseURL}}/accounts/account-id/attributes/test_attribute/", "rel": "related"}]
        }
    ],
    "links": [
        {
            "href": "{{APIBaseURL}}/accounts/account-id/attributes/",
            "rel": "self"
        }
    ]
}
```

20.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

21 PartnerAccounts-partnerAccountId-attributes:GET

22 GET /partnerAccounts/*{partnerAccountId}*/attributes

Get attributes list.

22.1 Contents

- 1 GET /partnerAccounts/{partnerAccountId}/attributes
  - 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 Examples
    - 1.3.1 Sending unauthorized request
    - 1.3.2 Sending request without setting Content-Type
    - 1.3.3 Getting list of attributes of not existing partner
    - 1.3.4 Success scenario
  - 1.4 See also

22.2 Request

`GET /partnerAccounts/{partnerAccountId}/attributes`

22.2.1 Parameters

partnerAccountId - *string*
The Id of the partner account.

22.2.2 URI Parameters

None.

22.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
The Authentication credentials of the client application.

Content-Type
application/json

22.3 Response

22.3.1 Status Code

200 OK
Success.

401 Unauthorized
The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
Forbidden access to a user not authorized for such access.

404 Not Found
The partner account Id or attribute were not found.

415 Unsupported Media Type
The Content-Type header was not specified or the specified type is not supported. Use application/json.

22.3.2 Response Body

```
{
  "list": [
    "name": "{name}",
    "value": "{value}",
    "links" : [{"href": "https://api.hostway.com/partnerAccounts/{partnerAccountId}/attributes/{name}", "rel": "related"}]
  ],
  "links": [
    {"href": "https://api.hostway.com/partnerAccounts/{partnerAccountId}/attributes", "rel": "self"}
  ]
}
```

22.3.2.1 Parameters

list - *list*
A collection of dictionaries containing information about specific attribute

name - *string*

Descriptive name of the attribute

value - *string*
Value of the attribute

NOTE: The "partner_account_id" attribute will not be returned in this list.

links - *list*
Hypermedia for this product resource. Links to the attribute.

links - *list*
Hypermedia for this resource. Links to self.

22.4 Examples

22.4.1 Sending unauthorized request

Request

```
GET /partnerAccounts/partner-account-id/attributes
```

Response

```
401 Unauthorized
```

22.4.2 Sending request without setting Content-Type

Request

```
GET /partnerAccounts/partner-account-id/attributes
Authorization: Basic base64_encode(username:password)
```

Response

```
415 Unsupported Media Type
```

22.4.3 Getting list of attributes of not existing partner

Request

```
GET /partnerAccounts/not-existing-partner-account-id/attributes
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
404 Not Found
```

22.4.4 Success scenario

Request

```
GET /partnerAccounts/partner-account-id/attributes
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
200 OK
```

```
{
    "list": [
        {
            "name": "test_attribute",
            "value": "123",
            "links": [{"href": "{{APIBaseURL}}/partnerAccounts/partner-account-id/attributes/test_attribute/", "rel": "related"}]
        }
    ],
    "links": [
        {
            "href": "{{APIBaseURL}}/partnerAccounts/partner-account-id/attributes/",
            "rel": "self"
        }
    ]
}
```

22.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

23 Accounts-accountId-attributes:POST

24 POST /accounts/*{accountId}*/attributes

Get attributes list.

24.1 Contents

- 1 POST /accounts/{accountId}/attributes
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
  - 1.3 Examples
    - 1.3.1 Sending unauthorized request
    - 1.3.2 Sending request without setting Content-Type
    - 1.3.3 Failure creating attribute for not existing account
    - 1.3.4 Success scenario
  - 1.4 See also

24.2 Request

`POST /accounts/{accountId}/attributes`

24.2.1 Parameters

accountId - *string*
: The Id of the Hostway account.

24.2.2 URI Parameters

None.

24.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Content-Type
: application/json

24.2.4 Request Body

```
{
  "name": "test_attribute",
  "value": "value"
}
```

24.2.4.1 Parameters

name - *string*
: Descriptive name of the attribute

value - *dictionary*
: Value of the attribute

24.3 Response

24.3.1 Status Code

201 Created
: Success.

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: Forbidden access to a user not authorized for such access.

404 Not Found
: The account Id or attribute were not found.

415 Unsupported Media Type
: The Content-Type header was not specified or the specified type is not supported. Use application/json.

24.4 Examples

24.4.1 Sending unauthorized request

Request

```
POST /accounts/account-id/attributes

{"name": "test_Account", "value": "test"}
```

Response

```
401 Unauthorized
```

24.4.2 Sending request without setting Content-Type

Request

```
POST /accounts/account-id/attributes
Authorization: Basic base64_encode(username:password)

{"name": "test_Account", "value": "test"}
```

Response

```
415 Unsupported Media Type
```

24.4.3 Failure creating attribute for not existing account

Request

```
POST /accounts/not-existing-account-id/attributes
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{"name": "test_Account", "value": "test"}
```

Response

```
404 Not Found
```

24.4.4 Success scenario

Request

```
POST /accounts/account-id/attributes
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{"name": "test_Account", "value": "test"}
```

Response

```
201 Created
```

24.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

25 PartnerAccounts-partnerAccountId-attributes:POST

26 POST /partnerAccounts/*{partnerAccountId}*/attributes

Get attributes list.

26.1 Contents

- 1 POST /partnerAccounts/{partnerAccountId}/attributes
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
  - 1.3 Examples
    - 1.3.1 Sending unauthorized request
    - 1.3.2 Sending request without setting Content-Type
    - 1.3.3 Failure creating attribute for not existing partner
    - 1.3.4 Success scenario
  - 1.4 See also

26.2 Request

`POST /partnerAccounts/{partnerAccountId}/attributes`

26.2.1 Parameters

partnerAccountId - *string*
: The Id of the partner account.

26.2.2 URI Parameters

None.

26.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Content-Type
: application/json

26.2.4 Request Body

```
{
  "name": "test_attribute",
  "value": "value"
}
```

26.2.4.1 Parameters

name - *string*
: Descriptive name of the attribute

value - *dictionary*
: Value of the attribute

26.3 Response

26.3.1 Status Code

201 Created
: Success.

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: Forbidden access to a user not authorized for such access.

404 Not Found
: The partner account Id or attribute were not found.

415 Unsupported Media Type
: The Content-Type header was not specified or the specified type is not supported. Use application/json.

26.4 Examples

26.4.1 Sending unauthorized request

Request

```
POST /partnerAccounts/partner-account-id/attributes

{"name": "test_Account", "value": "test"}
```

Response

```
401 Unauthorized
```

26.4.2 Sending request without setting Content-Type

Request

```
POST /partnerAccounts/partner-account-id/attributes
Authorization: Basic base64_encode(username:password)

{"name": "test_Account", "value": "test"}
```

Response

```
415 Unsupported Media Type
```

26.4.3 Failure creating attribute for not existing partner

Request

```
POST /partnerAccounts/not-existing-partner-account-id/attributes
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{"name": "test_Account", "value": "test"}
```

Response

```
404 Not Found
```

26.4.4 Success scenario

Request

```
POST /partnerAccounts/partner-account-id/attributes
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{"name": "test_Account", "value": "test"}
```

Response

```
201 Created
```

26.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

27 Accounts-accountId-attributes-attribute:GET

28 GET /accounts/*{accountId}*/attributes/*{attribute}*

Get attribute data.

28.1 Contents

- 1 GET /accounts/{accountId}/attributes/{attribute}
  - 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 Examples
    - 1.3.1 Sending unauthorized request
    - 1.3.2 Sending request without setting Content-Type
    - 1.3.3 Requesting not existing attribute
    - 1.3.4 Success scenario
  - 1.4 See also

28.2 Request

`GET /accounts/{accountId}/attributes/{attribute}`

28.2.1 Parameters

accountId - *string*
: The Id of the account.

28.2.2 URI Parameters

attribute - *string*

28.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Content-Type
: application/json

28.3 Response

28.3.1 Status Code

200 OK
: Success.

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: Forbidden access to a user not authorized for such access.

404 Not Found
: The account Id or attribute were not found.

415 Unsupported Media Type
: The Content-Type header was not specified or the specified type is not supported. Use application/json.

28.3.2 Response Body

```
{
   "name": "{name}",
   "value": {value},
   "links": [
      {
         "location": "{APIBaseURL}/accounts/{accountId}/attributes/{attribute}/",
         "rel": "self"
      }
   ]
}
```

28.3.2.1 Parameters

name - *string*
: Descriptive name of the attribute

value - *string*
: Value of the attribute

links - *list*
: Hypermedia for this resource. Links to self.

28.4 Examples

28.4.1 Sending unauthorized request

Request

```
GET /accounts/account-id/attributes/test_attribute
```

Response

```
401 Unauthorized
```

28.4.2 Sending request without setting Content-Type

Request

```
GET /accounts/account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)
```

Response

```
415 Unsupported Media Type
```

28.4.3 Requesting not existing attribute

Request

```
GET /accounts/account-id/attributes/not_existing_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
404 Not Found
```

28.4.4 Success scenario

Request

```
GET /accounts/account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
200 OK

{
   "name": "test_attribute",
   "value": "123",
   "links": [{"href": "{{APIBaseURL}}/accounts/account-id/attributes/test_attribute/", "rel": "related"}]
}
```

28.5 See also

- Partner Account Attributes Management
- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

29 PartnerAccounts-partnerAccountId-attributes-attribute:GET

30 GET /partnerAccounts/*{partnerAccountId}*/attributes/*{attribute}*

Get attribute data.

30.1 Contents

- 1 GET /partnerAccounts/{partnerAccountId}/attributes/{attribute}
  - 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 Examples
    - 1.3.1 Sending unauthorized request
    - 1.3.2 Sending request without setting Content-Type
    - 1.3.3 Requesting not existing attribute
    - 1.3.4 Success scenario
  - 1.4 See also

30.2 Request

`GET /partnerAccounts/{partnerAccountId}/attributes/{attribute}`

30.2.1 Parameters

partnerAccountId - *string*
: The Id of the partner account.

30.2.2 URI Parameters

attribute - *string*

30.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Content-Type
: application/json

30.3 Response

30.3.1 Status Code

200 OK
: Success.

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: Forbidden access to a user not authorized for such access.

404 Not Found
: The partner account Id or attribute were not found.

415 Unsupported Media Type
: The Content-Type header was not specified or the specified type is not supported. Use application/json.

30.3.2 Response Body

```
{
   "name": "{name}",
   "value": {value},
   "links": [
      {
         "location": "{APIBaseURL}/partnerAccounts/{partnerAccountId}/attributes/{attribute}/",
         "rel": "self"
      }
   ]
}
```

30.3.2.1 Parameters

name - *string*
: Descriptive name of the attribute

value - *string*
: Value of the attribute

links - *list*

Hypermedia for this resource. Links to self.

30.4 Examples

30.4.1 Sending unauthorized request

Request

```
GET /partnerAccounts/partner-account-id/attributes/test_attribute
```

Response

```
401 Unauthorized
```

30.4.2 Sending request without setting Content-Type

Request

```
GET /partnerAccounts/partner-account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)
```

Response

```
415 Unsupported Media Type
```

30.4.3 Requesting not existing attribute

Request

```
GET /partnerAccounts/partner-account-id/attributes/not_existing_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
404 Not Found
```

30.4.4 Success scenario

Request

```
GET /partnerAccounts/partner-account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
200 OK

{
   "name": "test_attribute",
   "value": "123",
   "links": [{"href": "{{APIBaseURL}}/partnerAccounts/partner-account-id/attributes/test_attribute/", "rel": "related"}]
}
```

30.5 See also

- Account Attribute Management
- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

31 Accounts-accountId-attributes-attribute:PUT

32 PUT /accounts/*{accountId}*/attributes/*{attribute}*

Updates attribute value.

32.1 Contents

- 1 PUT /accounts/{accountId}/attributes/{attribute}
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
  - 1.3 Examples
    - 1.3.1 Sending unauthorized request
    - 1.3.2 Sending request without setting Content-Type
    - 1.3.3 Failure updating not existing attribute
    - 1.3.4 Failure updating attribute with missing required value
    - 1.3.5 Success scenario
  - 1.4 See also

32.2 Request

`PUT /accounts/{accountId}/attributes/{attribute}`

32.2.1 Parameters

accountId - *string*
: The Id of the Hostway account.

32.2.2 URI Parameters

attribute - *string*

32.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Content-Type
: application/json

32.2.4 Request Body

```
{
    "value": "{value}"
}
```

32.2.4.1 Parameters

value - *string*

32.3 Response

32.3.1 Status Code

204 No Content
: Success.

400 Bad Request
: The format of the request body is invalid.

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: Forbidden access to a user not authorized for such access.

404 Not Found
: The account Id or attribute were not found.

415 Unsupported Media Type
: The Content-Type header was not specified or the specified type is not supported. Use application/json.

32.4 Examples

32.4.1 Sending unauthorized request

Request

```
PUT /accounts/account-id/attributes/test_attribute

{"value": "test"}
```

Response

```
401 Unauthorized
```

32.4.2 Sending request without setting Content-Type

Request

```
PUT /accounts/account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)

{"value": "test"}
```

Response

```
415 Unsupported Media Type
```

32.4.3 Failure updating not existing attribute

Request

```
PUT /accounts/account-id/attributes/not_existing_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{"value": "test"}
```

Response

```
404 Not Found
```

32.4.4 Failure updating attribute with missing required value

Request

```
PUT /accounts/account-id/attributes/not_existing_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{}
```

Response

```
400 Bad Request

{"value":"Required"}
```

32.4.5 Success scenario

Request

```
PUT /accounts/account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{"value": "test"}
```

Response

```
204 No Content
```

32.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

33 PartnerAccounts-partnerAccountId-attributes-attribute:PUT

34 PUT /partnerAccounts/*{partnerAccountId}*/attributes/*{attribute}*

Updates attribute value.

34.1 Contents

- 1 PUT /partnerAccounts/{partnerAccountId}/attributes/{attribute}
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
  - 1.3 Examples
    - 1.3.1 Sending unauthorized request
    - 1.3.2 Sending request without setting Content-Type
    - 1.3.3 Failure updating not existing attribute
    - 1.3.4 Failure updating attribute with missing required value
    - 1.3.5 Success scenario
  - 1.4 See also

34.2 Request

`PUT /partnerAccounts/{partnerAccountId}/attributes/{attribute}`

34.2.1 Parameters

partnerAccountId - *string*
: The Id of the partner account.

34.2.2 URI Parameters

attribute - *string*

34.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Content-Type
: application/json

34.2.4 Request Body

```
{
    "value": "{value}"
}
```

34.2.4.1 Parameters

value - *string*

34.3 Response

34.3.1 Status Code

204 No Content
: Success.

400 Bad Request
: The format of the request body is invalid.

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: Forbidden access to a user not authorized for such access.

404 Not Found
: The partner account Id or attribute were not found.

415 Unsupported Media Type
: The Content-Type header was not specified or the specified type is not supported. Use application/json.

34.4 Examples

34.4.1 Sending unauthorized request

Request

```
PUT /partnerAccounts/partner-account-id/attributes/test_attribute

{"value": "test"}
```

Response

```
401 Unauthorized
```

34.4.2 Sending request without setting Content-Type

Request

```
PUT /partnerAccounts/partner-account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)

{"value": "test"}
```

Response

```
415 Unsupported Media Type
```

34.4.3 Failure updating not existing attribute

Request

```
PUT /partnerAccounts/partner-account-id/attributes/not_existing_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{"value": "test"}
```

Response

```
404 Not Found
```

34.4.4 Failure updating attribute with missing required value

Request

```
PUT /partnerAccounts/partner-account-id/attributes/not_existing_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{}
```

Response

```
400 Bad Request

{"value":"Required"}
```

34.4.5 Success scenario

Request

```
PUT /partnerAccounts/partner-account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{"value": "test"}
```

Response

```
204 No Content
```

34.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

35 Accounts-accountId-attributes-attribute:DELETE

36 DELETE /accounts/*{accountId}*/attributes/*{attribute}*

Updates attribute value.

36.1 Contents

- 1 DELETE /accounts/{accountId}/attributes/{attribute}
  - 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 Examples
    - 1.3.1 Sending unauthorized request
    - 1.3.2 Sending request without setting Content-Type
    - 1.3.3 Failure updating not existing attribute
    - 1.3.4 Success scenario
  - 1.4 See also

36.2 Request

`DELETE /accounts/{accountId}/attributes/{attribute}`

36.2.1 Parameters

accountId - *string*
: The Id of the Hostway account.

36.2.2 URI Parameters

attribute - *string*

36.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Content-Type
: application/json

36.3 Response

36.3.1 Status Code

204 No Content
: Success.

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: Forbidden access to a user not authorized for such access.

404 Not Found
: The account Id or attribute were not found.

415 Unsupported Media Type
: The Content-Type header was not specified or the specified type is not supported. Use application/json.

36.4 Examples

36.4.1 Sending unauthorized request

Request

```
DELETE /accounts/account-id/attributes/test_attribute
```

Response

```
401 Unauthorized
```

36.4.2 Sending request without setting Content-Type

Request

```
DELETE /accounts/account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)
```

Response

```
415 Unsupported Media Type
```

36.4.3 Failure updating not existing attribute

Request

```
DELETE /accounts/account-id/attributes/not_existing_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
404 Not Found
```

36.4.4 Success scenario

Request

```
DELETE /accounts/account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
204 No Content
```

36.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

37 PartnerAccounts-partnerAccountId-attributes-attribute:DELETE

38 DELETE /partnerAccounts/*{partnerAccountId}*/attributes/*{attribute}*

Updates attribute value.

38.1 Contents

- 1 DELETE /partnerAccounts/{partnerAccountId}/attributes/{attribute}
  - 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 Examples
    - 1.3.1 Sending unauthorized request
    - 1.3.2 Sending request without setting Content-Type
    - 1.3.3 Failure updating not existing attribute
    - 1.3.4 Success scenario
  - 1.4 See also

38.2 Request

`DELETE /partnerAccounts/{partnerAccountId}/attributes/{attribute}`

38.2.1 Parameters

partnerAccountId - *string*
: The Id of the partner account.

38.2.2 URI Parameters

attribute - *string*

38.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Content-Type
: application/json

38.3 Response

38.3.1 Status Code

204 No Content
: Success.

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: Forbidden access to a user not authorized for such access.

404 Not Found
: The partner account Id or attribute were not found.

415 Unsupported Media Type
: The Content-Type header was not specified or the specified type is not supported. Use application/json.

38.4 Examples

38.4.1 Sending unauthorized request

Request

```
DELETE /partnerAccounts/partner-account-id/attributes/test_attribute
```

Response

```
401 Unauthorized
```

38.4.2 Sending request without setting Content-Type

Request

```
DELETE /partnerAccounts/partner-account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)
```

Response

```
415 Unsupported Media Type
```

38.4.3 Failure updating not existing attribute

Request

```
DELETE /partnerAccounts/partner-account-id/attributes/not_existing_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
404 Not Found
```

38.4.4 Success scenario

Request

```
DELETE /partnerAccounts/partner-account-id/attributes/test_attribute
Authorization: Basic base64_encode(username:password)
Content-Type: application/json
```

Response

```
204 No Content
```

38.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

39 Accounts-accountId-Contacts:GET

40 GET /accounts/*{accountId}*/contacts

Get contact details for an account.

40.1 Contents

- 1 GET /accounts/{accountId}/contacts
  - 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 Examples
    - 1.3.1 Successful Request
    - 1.3.2 Request for non-existent account Id
  - 1.4 See also

40.2 Request

`GET /accounts/{accountId}/contacts`

40.2.1 Request Parameters

accountId - *string*
: The unique account Id.

40.2.2 URI Parameters

None.

40.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

40.3 Response

40.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The request was sent without an *Authorization* header, or the user represented by the value of this header was not properly authenticated.

403 Forbidden
: The user authenticated for this request is not allowed to access the resource.

404 Not Found
: The specified account Id does not exist or is purged.

40.3.2 Response Body

```
{
  "contactInfo": {
    "regular": {
      "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}",
        "email2": "{email2}",
        "emailVerified": {emailverified}
      }
    },
    "billing": {
      ...
    },
    "administrator": {
      ...
    },
    "technical": {
      ...
    }
  },
  "links": [
    {
      "href": "https://api.hostway.com/accounts/{accountId}/contacts/",
      "rel": "self"
    }
  ]
}
```

40.3.2.1 Parameters

Note, there are 4 types of contact information that must be returned with the response: "regular", "billing", "administrator" and "technical".

city - *string*
: May be up to 50 characters long. May contain only alphanumerics, white space and the following characters: -.'

company - *string*
: *(OPTIONAL)*
May be up to 200 characters long. May contain only alphanumerics, white space and the following characters: -.'

countryCode - *string*
: ISO 3166-1 alpha-2[2] country code.

email1 - *string*
: May be up to 100 characters long. Must contain a valid e-mail address.

email2 - *string*
: *(OPTIONAL)*
May be up to 100 characters long. If provided, must contain a valid e-mail address.

emailVerified - *integer*
: Provides information whether the **email** is verified for a Zendesk access. Valid values are **0** or **1**.

fax - *string*
: *(OPTIONAL)*
May be up to 20 characters long. If provided, must contain ITU-T E.164[3] formatted phone number.

firstName - *string*
: May be up to 50 characters long. May contain only alphanumerics, white space and the following characters: -.'

lastName - *string*
: May be up to 50 characters long. May contain only alphanumerics, white space and the following characters: -.'

middleName - *string*
: *(OPTIONAL)*
May be up to 50 characters long. If provided, may contain only alphanumerics, white space and the following characters: -.'

phone1 - *string*
: May be up to 20 characters long. Must contain ITU-T E.164[4] formatted phone number.

phone2 - *string*
: *(OPTIONAL)*
May be up to 20 characters long. If provided, must contain ITU-T E.164[5] formatted phone number.

postalCode - *string*
: *(OPTIONAL)*
May be up to 30 characters long. May contain only alphanumerics, white space and the following characters: -.' This value is *required* if the value of *countryCode* is "US" or "CA". If the value of *countryCode* is "US", then *postalCode* must be in format "#####" or "#####-####" (where "#" represents any numeric digit). If the value of *countryCode* is "CA", then *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).

salutation - *string*
: *(OPTIONAL)*
May contain one of the salutations provided by GET /salutations

stateOrProvince - *string*
: *(OPTIONAL)*
May be up to 20 characters long. May contain only alphanumerics, white space and the following characters: -.' This value is *required* if the value of *countryCode* is "US" or "CA". If the value of *countryCode* is "US", then *stateOrProvince* must be a valid two-letter US state abbreviation. If the value of *countryCode* is "CA", then *stateOrProvince* must be a valid two-letter Canadian province abbreviation.

street1 - *string*
: May be up to 100 characters long. May contain only alphanumerics, white space and the following characters: -.'

street2 - *string*
(OPTIONAL)
May be up to 100 characters long. If provided, may contain only alphanumerics, white space and the following characters: -.'

40.4 Examples

40.4.1 Successful Request

Request

```
GET /accounts/{accountId}/contacts
Authorization: Basic {Base64-encoded username & password string}
```

Response

```
200 OK

{
  "contactInfo": {
    "regular": {
      "name": {
        "salutation": "Mr.",
        "firstName": "John",
        "middleName": "",
        "lastName": "Smith",
        "company": "TestCo, Inc."
      },
      "address": {
        "street1": "100 Main St.",
        "street2": "",
        "city": "Chicago",
        "stateOrProvince": "IL",
        "postalCode": "60001",
        "countryCode": "US"
      },
      "contactMedia": {
        "phone1": "555-555-5501",
        "phone2": "",
        "fax": "",
        "email1": "test@test.com",
        "email2": "",
        "emailVerified": 1
      }
    },
    "billing": {
      "name": {
        "salutation": "Mr.",
        "firstName": "John",
        "middleName": "",
        "lastName": "Smith",
        "company": "TestCo, Inc."
      },
      "address": {
        "street1": "100 Main St.",
        "street2": "",
        "city": "Chicago",
        "stateOrProvince": "IL",
        "postalCode": "60001",
        "countryCode": "US"
      },
      "contactMedia": {
        "phone1": "555-555-5501",
        "phone2": "",
        "fax": "",
        "email1": "test@test.com",
        "email2": "",
        "emailVerified": 1
      }
    },
    "administrator": {
      "name": {
        "salutation": "Mr.",
        "firstName": "John",
        "middleName": "",
        "lastName": "Smith",
        "company": "TestCo, Inc."
      },
      "address": {
        "street1": "100 Main St.",
        "street2": "",
        "city": "Chicago",
        "stateOrProvince": "IL",
        "postalCode": "60001",
        "countryCode": "US"
      },
      "contactMedia": {
        "phone1": "555-555-5501",
        "phone2": "",
        "fax": "",
        "email1": "test@test.com",
        "email2": "",
        "emailVerified": 1
      }
    },
    "technical": {
      "name": {
        "salutation": "Mr.",
        "firstName": "John",
        "middleName": "",
        "lastName": "Smith",
        "company": "TestCo, Inc."
      },
      "address": {
```

```
        "street1": "100 Main St.",
        "street2": "",
        "city": "Chicago",
        "stateOrProvince": "IL",
        "postalCode": "60001",
        "countryCode": "US"
      },
      "contactMedia": {
        "phone1": "555-555-5501",
        "phone2": "",
        "fax": "",
        "email1": "test@test.com",
        "email2": "",
        "emailVerified": 1
      }
    }
  },
  "links": [
    {
      "href": "https://api.hostway.com/accounts/{accountId}/contacts/",
      "rel": "self"
    }
  ]
}
```

40.4.2 Request for non-existent account Id

Request

```
GET /accounts/{fake_account_number}/contacts
Authorization: Basic {Base64-encoded username & password string}
```

Response

```
404 Not Found

{
   "itemNotFound": {
      "guid": {identifier},
      "message": "/accounts/fake-account-id/contacts",
      "code": 404,
      "details": ""
   }
}
```

40.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

41 Accounts-accountId-Contacts-contactType:GET

42 GET /accounts/*{accountId}*/contacts/*{contactType}*

Get specific contact type for an account

42.1 Contents

- 1 GET /accounts/{accountId}/contacts/{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 Examples
    - 1.3.1 Successful Request
    - 1.3.2 Request for non-existent contact type
  - 1.4 See also

42.2 Request

`GET /accounts/{accountId}/contacts/{contactType}`

42.2.1 Request Parameters

accountId - *string*
: The unique account Id.

contactType - *string*
: Contact type. One of 'regular', 'billing', 'administrator', 'technical'

42.2.2 URI Parameters

None.

42.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

42.3 Response

42.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The request was sent without an *Authorization* header, or the user represented by the value of this header was not properly authenticated.

403 Forbidden
: The user authenticated for this request is not allowed to access the resource.

404 Not Found
: Contact type does not match allowed values or the specified account Id does not exist or is purged.

42.3.2 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}",
      "email2": "{email2}",
      "emailVerified": {emailverified}
```

```
    }
  },
  "links": [
    {
      "href": "https://api.hostway.com/accounts/{accountId}/contacts/{contactType}/",
      "rel": "self"
    }
  ]
}
```

42.3.2.1 Parameters

city - *string*
: May be up to 50 characters long. May contain only alphanumeric, white space and the following characters: -.'

company - *string*
: *(OPTIONAL)*
May be up to 200 characters long. May contain only alphanumeric, white space and the following characters: -.'

countryCode - *string*
: ISO 3166-1 alpha-2[2] country code.

email1 - *string*
: May be up to 100 characters long. Must contain a valid e-mail address.

email2 - *string*
: *(OPTIONAL)*
May be up to 100 characters long. If provided, must contain a valid e-mail address.

emailVerified - *integer*
: Provides information whether the **email** is verified for a Zendesk access. Valid values are **0** or **1**.

fax - *string*
: *(OPTIONAL)*
May be up to 20 characters long. If provided, must contain ITU-T E.164[3] formatted phone number.

firstName - *string*
: May be up to 50 characters long. May contain only alphanumeric, white space and the following characters: -.'

lastName - *string*
: May be up to 50 characters long. May contain only alphanumeric, white space and the following characters: -.'

middleName - *string*
: *(OPTIONAL)*
May be up to 50 characters long. If provided, may contain only alphanumeric, white space and the following characters: -.'

phone1 - *string*
: May be up to 20 characters long. Must contain ITU-T E.164[4] formatted phone number.

phone2 - *string*
: *(OPTIONAL)*
May be up to 20 characters long. If provided, must contain ITU-T E.164[5] formatted phone number.

postalCode - *string*
: *(OPTIONAL)*
May be up to 30 characters long. May contain only alphanumeric, white space and the following characters: -.' This value is *required* if the value of *countryCode* is "US" or "CA". If the value of *countryCode* is "US", then *postalCode* must be in format "#####" or "#####-####" (where "#" represents any numeric digit). If the value of *countryCode* is "CA", then *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).

salutation - *string*
: *(OPTIONAL)*
May contain one of the salutations provided by GET /salutations

stateOrProvince - *string*
: *(OPTIONAL)*
May be up to 20 characters long. May contain only alphanumeric, white space and the following characters: -.' This value is *required* if the value of *countryCode* is "US" or "CA". If the value of *countryCode* is "US", then *stateOrProvince* must be a valid two-letter US state abbreviation. If the value of *countryCode* is "CA", then *stateOrProvince* must be a valid two-letter Canadian province abbreviation.

street1 - *string*
: May be up to 100 characters long. May contain only alphanumeric, white space and the following characters: -.'

street2 - *string*
: *(OPTIONAL)*
May be up to 100 characters long. If provided, may contain only alphanumeric, white space and the following characters: -.'

42.4 Examples

42.4.1 Successful Request

Request

```
GET /accounts/{accountId}/contacts/regular
Authorization: Basic {Base64-encoded username & password string}
```

Response

```
200 OK
```

```
{
  "contactInfo": {
    "name": {
      "salutation": "Mr.",
      "firstName": "John",
      "middleName": "",
      "lastName": "Smith",
      "company": "TestCo, Inc."
    },
    "address": {
      "street1": "100 Main St.",
      "street2": "",
      "city": "Chicago",
      "stateOrProvince": "IL",
      "postalCode": "60001",
      "countryCode": "US"
    },
    "contactMedia": {
      "phone1": "555-555-5501",
      "phone2": "",
      "fax": "",
      "email1": "test@test.com",
      "email2": "",
      "emailVerified": 1
    }
  },
  "links": [
    {
      "href": "https://api.hostway.com/accounts/{accountId}/contacts/regular/",
      "rel": "self"
    }
  ]
}
```

42.4.2 Request for non-existent contact type

Request

```
GET /accounts/{accountId}/contacts/fake
Authorization: Basic {Base64-encoded username & password string}
```

Response

```
404 Not Found

{
   "itemNotFound": {
      "guid": {identifier},
      "message": "Resource not found",
      "code": 404,
      "details": ""
   }
}
```

42.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

43 Accounts-accountId-Contacts:PUT

44 PUT /accounts/*{accountId}*/contacts

Set the contact details for an account

44.1 Contents

- 1 PUT /accounts/{accountId}/contacts
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 Examples
    - 1.3.1 Successful update of contact information
  - 1.4 See also

44.2 Request

`PUT /accounts/{accountId}/contacts`

44.2.1 Parameters

accountId - *string*
: The Id of the account.

44.2.2 URI Parameters

None.

44.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

44.2.4 Request Body

```
{
  "regular": {
    "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}",
      "email2": "{email2}"
    }
  },
  "billing": {
    ...
  },
  "administrator": {
    ...
  },
  "technical": {
    ...
  }
}
```

44.2.4.1 Parameters

Note, there are 4 types of contact information that must be sent with the request: "regular", "billing", "administrator" and "technical". All 4 are required.

city - *string*
: *(REQUIRED)*
May be up to 50 characters long. May contain only alphanumerics, white space and the following characters: -.'

company - *string*
(OPTIONAL)
May be up to 200 characters long. May contain only alphanumerics, white space and the following characters: -.'

countryCode - *string*
(REQUIRED)
ISO 3166-1 alpha-2[2] country code.

email1 - *string*
(REQUIRED)
May be up to 100 characters long. Must contain a valid e-mail address.

email2 - *string*
(OPTIONAL)
May be up to 100 characters long. If provided, must contain a valid e-mail address.

fax - *string*
(OPTIONAL)
May be up to 20 characters long. If provided, must contain ITU-T E.164[3] formatted phone number.

firstName - *string*
(REQUIRED)
May be up to 50 characters long. May contain only alphanumerics, white space and the following characters: -.'

lastName - *string*
(REQUIRED)
May be up to 50 characters long. May contain only alphanumerics, white space and the following characters: -.'

middleName - *string*
(OPTIONAL)
May be up to 50 characters long. If provided, may contain only alphanumerics, white space and the following characters: -.'

phone1 - *string*
(REQUIRED)
May be up to 20 characters long. Must contain ITU-T E.164[4] formatted phone number.

phone2 - *string*
(OPTIONAL)
May be up to 20 characters long. If provided, must contain ITU-T E.164[5] formatted phone number.

postalCode - *string*
(OPTIONAL)
May be up to 30 characters long. May contain only alphanumerics, white space and the following characters: -.' This value is *required* if the value of *countryCode* is "US" or "CA". If the value of *countryCode* is "US", then *postalCode* must be in format "#####" or "#####-####" (where "#" represents any numeric digit). If the value of *countryCode* is "CA", then *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).

salutation - *string*
(OPTIONAL)
May contain one of the salutations provided by GET /salutations

stateOrProvince - *string*
(OPTIONAL)
May be up to 20 characters long. May contain only alphanumerics, white space and the following characters: -.' This value is *required* if the value of *countryCode* is "US" or "CA". If the value of *countryCode* is "US", then *stateOrProvince* must be a valid two-letter US state abbreviation. If the value of *countryCode* is "CA", then *stateOrProvince* must be a valid two-letter Canadian province abbreviation.

street1 - *string*
(REQUIRED)
May be up to 100 characters long. May contain only alphanumerics, white space and the following characters: -.'

street2 - *string*
(OPTIONAL)
May be up to 100 characters long. If provided, may contain only alphanumerics, white space and the following characters: -.'

44.3 Response

44.3.1 Status Code

204 No Content
Successfully updated contact information

400 Bad Request
Contact information was not updated because either the request body was malformed, or some element of the request body did not meet the requirements.

401 Unauthorized
The request was sent without an *Authorization* header, or the user represented by the value of this header was not properly authenticated.

403 Forbidden
The user authenticated for this request is not allowed to access the resource.

404 Not Found
The account Id was not found.

44.3.2 Response Body

None.

44.4 Examples

44.4.1 Successful update of contact information

Request

```
PUT /accounts/{accountId}/contacts

Authorization: Basic {Base64-encoded username & password string}
```

```
{
  "regular": {
    name: {
      "salutation": "Mr.",
      "firstName": "John",
      "middleName": "",
      "lastName": "Smith",
      "company": "TestCo, Inc."
    },
    "address": {
      "street1": "100 Main St.",
      "street2": "",
      "city": "Chicago",
      "stateOrProvince": "IL",
      "postalCode": "60001",
      "countryCode": "US"
    },
    "contactMedia": {
      "phone1": "555-555-5501",
      "phone2": "",
      "fax": "",
      "email1": "test@test.com",
      "email2": ""
    }
  },
  "billing": {
    name: {
      "salutation": "Mr.",
      "firstName": "John",
      "middleName": "",
      "lastName": "Smith",
      "company": "TestCo, Inc."
    },
    "address": {
      "street1": "100 Main St.",
      "street2": "",
      "city": "Chicago",
      "stateOrProvince": "IL",
      "postalCode": "60001",
      "countryCode": "US"
    },
    "contactMedia": {
      "phone1": "555-555-5501",
      "phone2": "",
      "fax": "",
      "email1": "test@test.com",
      "email2": ""
    }
  },
  "administrator": {
    name: {
      "salutation": "Mr.",
      "firstName": "John",
      "middleName": "",
      "lastName": "Smith",
      "company": "TestCo, Inc."
    },
    "address": {
      "street1": "100 Main St.",
      "street2": "",
      "city": "Chicago",
      "stateOrProvince": "IL",
      "postalCode": "60001",
      "countryCode": "US"
    },
    "contactMedia": {
      "phone1": "555-555-5501",
      "phone2": "",
      "fax": "",
      "email1": "test@test.com",
      "email2": ""
    }
  },
  "technical": {
    name: {
      "salutation": "Mr.",
      "firstName": "John",
      "middleName": "",
      "lastName": "Smith",
      "company": "TestCo, Inc."
    },
    "address": {
      "street1": "100 Main St.",
      "street2": "",
      "city": "Chicago",
      "stateOrProvince": "IL",
      "postalCode": "60001",
      "countryCode": "US"
```

```
    },
    "contactMedia": {
      "phone1": "555-555-5501",
      "phone2": "",
      "fax": "",
      "email1": "test@test.com",
      "email2": ""
    }
  }
}
```

Response

```
204 No Content
```

44.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

45 Accounts-accountId-Contacts-contactType:PUT

46 PUT /accounts/*{accountId}*/contacts/*{contactType}*

Set specific contact type for an account

46.1 Contents

- 1 PUT /accounts/{accountId}/contacts/{contactType}
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 Examples
    - 1.3.1 Successful update of contact information
    - 1.3.2 Request for updating non-existent contact type
    - 1.3.3 Request for updating contact type with not valid data
    - 1.3.4 Request for updating contact type with missing or empty required data
  - 1.4 See also

46.2 Request

`PUT /accounts/{accountId}/contacts/{contactType}`

46.2.1 Parameters

accountId - *string*
: The Id of the account.

contactType - *string*
: Contact type. One of 'regular', 'billing', 'administrator', 'technical'

46.2.2 URI Parameters

None.

46.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

46.2.4 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}",
    "email2": "{email2}"
  }
}
```

46.2.4.1 Parameters

city - *string*
: *(REQUIRED)*
: May be up to 50 characters long. May contain only alphanumeric, white space and the following characters: -.'

company - *string*
: *(OPTIONAL)*
: May be up to 200 characters long. May contain only alphanumeric, white space and the following characters: -.'

countryCode - *string*
: *(REQUIRED)*
: ISO 3166-1 alpha-2[2] country code.

email1 - *string*
: *(REQUIRED)*

May be up to 100 characters long. Must contain a valid e-mail address.

email2 - *string*
(OPTIONAL)
May be up to 100 characters long. If provided, must contain a valid e-mail address.

fax - *string*
(OPTIONAL)
May be up to 20 characters long. If provided, must contain ITU-T E.164[3] formatted phone number.

firstName - *string*
(REQUIRED)
May be up to 50 characters long. May contain only alphanumeric, white space and the following characters: -.'

lastName - *string*
(REQUIRED)
May be up to 50 characters long. May contain only alphanumeric, white space and the following characters: -.'

middleName - *string*
(OPTIONAL)
May be up to 50 characters long. If provided, may contain only alphanumeric, white space and the following characters: -.'

phone1 - *string*
(REQUIRED)
May be up to 20 characters long. Must contain ITU-T E.164[4] formatted phone number.

phone2 - *string*
(OPTIONAL)
May be up to 20 characters long. If provided, must contain ITU-T E.164[5] formatted phone number.

postalCode - *string*
(OPTIONAL)
May be up to 30 characters long. May contain only alphanumeric, white space and the following characters: -.' This value is *required* if the value of *countryCode* is "US" or "CA". If the value of *countryCode* is "US", then *postalCode* must be in format "#####" or "#####-####" (where "#" represents any numeric digit). If the value of *countryCode* is "CA", then *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).

salutation - *string*
(OPTIONAL)
May contain one of the salutations provided by GET /salutations

stateOrProvince - *string*
(OPTIONAL)
May be up to 20 characters long. May contain only alphanumeric, white space and the following characters: -.' This value is *required* if the value of *countryCode* is "US" or "CA". If the value of *countryCode* is "US", then *stateOrProvince* must be a valid two-letter US state abbreviation. If the value of *countryCode* is "CA", then *stateOrProvince* must be a valid two-letter Canadian province abbreviation.

street1 - *string*
(REQUIRED)
May be up to 100 characters long. May contain only alphanumeric, white space and the following characters: -.'

street2 - *string*
(OPTIONAL)
May be up to 100 characters long. If provided, may contain only alphanumerics, white space and the following characters: -.'

46.3 Response

46.3.1 Status Code

204 No Content
Successfully updated contact information

400 Bad Request
Contact information was not updated because either the request body was malformed, or some element of the request body did not meet the requirements.

401 Unauthorized
The request was sent without an *Authorization* header, or the user represented by the value of this header was not properly authenticated.

403 Forbidden
The user authenticated for this request is not allowed to access the resource.

404 Not Found
The account Id was not found.

46.3.2 Response Body

None.

46.4 Examples

46.4.1 Successful update of contact information

Request

```
PUT /accounts/{accountId}/contacts/regular

Authorization: Basic {Base64-encoded username & password string}

{
  "name": {
    "salutation": "Mr.",
    "firstName": "John",
    "middleName": "",
    "lastName": "Smith",
    "company": "TestCo, Inc."
  },
  "address": {
    "street1": "100 Main St.",
    "street2": "",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60001",
    "countryCode": "US"
  },
  "contactMedia": {
    "phone1": "555-555-5501",
    "phone2": "",
    "fax": "",
    "email1": "test@test.com",
    "email2": ""
  }
}
```

Response

```
204 No Content
```

46.4.2 Request for updating non-existent contact type

Request

```
PUT /accounts/{accountId}/contacts/fake

Authorization: Basic {Base64-encoded username & password string}

{
  "name": {
    "salutation": "Mr.",
    "firstName": "John",
    "middleName": "",
    "lastName": "Smith",
    "company": "TestCo, Inc."
  },
  "address": {
    "street1": "100 Main St.",
    "street2": "",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60001",
    "countryCode": "US"
  },
  "contactMedia": {
    "phone1": "555-555-5501",
    "phone2": "",
    "fax": "",
    "email1": "test@test.com",
    "email2": ""
  }
}
```

Response

```
404 Not Found

{
   "itemNotFound": {
      "guid": {identifier},
      "message": "Resource not found",
      "code": 404,
      "details": ""
   }
}
```

46.4.3 Request for updating contact type with not valid data

Request

```
PUT /accounts/{accountId}/contacts/regular

Authorization: Basic {Base64-encoded username & password string}

{
  "name": {
    "salutation": "Mr.",
    "firstName": "John",
    "middleName": "",
    "lastName": "Smith",
    "company": "TestCo, Inc."
  },
```

```
  "address": {
    "street1": "100 Main St.",
    "street2": "",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60001",
    "countryCode": "USA"
  },
  "contactMedia": {
    "phone1": "555-555-5501",
    "phone2": "",
    "fax": "",
    "email1": "test@test.com",
    "email2": ""
  }
}
```

Response

```
400 Bad Request

{
    "badRequest": {
        "guid": {identifier},
        "message": "POST data error",
        "code": 400,
        "details": {
            "countryCode": "Invalid Country Code"
        }
    }
}
```

46.4.4 Request for updating contact type with missing or empty required data

Request

```
PUT /accounts/{accountId}/contacts/regular

Authorization: Basic {Base64-encoded username & password string}

{
  "name": {
    "salutation": "Mr.",
    "firstName": "",
    "middleName": "",
    "lastName": "Smith",
    "company": "TestCo, Inc."
  },
  "address": {
    "street1": "100 Main St.",
    "street2": "",
    "city": "Chicago",
    "stateOrProvince": "IL",
    "postalCode": "60001"
  },
  "contactMedia": {
    "phone1": "555-555-5501",
    "phone2": "",
    "fax": "",
    "email1": "test@test.com",
    "email2": ""
  }
}
```

Response

```
400 Bad Request

{
    "badRequest": {
        "guid": {identifier},
        "message": "POST data error",
        "code": 400,
         "details": {
            "countryCode": "Required",
            "firstName": "Required"
        }
    }
}
```

46.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

47 Initiate-Email-Verification-Action:POST

48 POST /accounts/{account_number}/contacts/action

Apply an action on contacts email e.g. **initiate email verification**

48.1 Contents

- 1 POST /accounts/{account_number}/contacts/action
  - 1.1 Request
    - 1.1.1 Request Parameters
    - 1.1.2 Request Headers
    - 1.1.3 Request Body
  - 1.2 Response
    - 1.2.1 Expected Response Codes
  - 1.3 Examples
    - 1.3.1 Initiate email verification request
  - 1.4 See also

48.2 Request

```
POST /accounts/{account_number}/contacts/action
```

48.2.1 Request Parameters

account_number - *string*
: The user account that owning the email.

48.2.2 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

48.2.3 Request Body

```
{
  "action": "verifyEmail",
  "actionData": "{contactinfo_type}"
}
```

action - *string*
: The action that should be performed. Allowed values are:
  verifyEmail - email verification.

actionData - *string*
: Data to be send with the action. This must be a contact information type.
  contactinfo_type - There are 4 types of contact information that can be used with the request: **regular**, **billing**, **administrator** or **technical**.

48.3 Response

48.3.1 Expected Response Codes

202 Accepted
: Success

400 Bad Request
: The supplied action is invalid.

401 Unauthorized
: The supplied credentials are invalid.

403 Forbidden
: The authorized user does not have permissions to access the resource. The only user(s) who has access to the contacts email must be a sales agent, the account owner, or the account technical admin.

404 Not Found
: The contacts email for the provided contact info type does not exist.

409 Conflict
: The contacts email is already verified.

48.4 Examples

48.4.1 Initiate email verification request

Request

```
POST /accounts/acct515393/contacts/action
Content-Type:application/json; Charset=UTF-8
Authorization: Basic {Base64 string}

{
  "action": "verifyEmail",
```

```
  "actionData": "regular"
}
```

Response

```
202 Accepted
```

48.5 See also

- Account Management API

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

49 ZenDesk-SSO-Link-Per-Contact-Type:GET

50 GET /accounts/{accountId}/contacts/{contactType}/supportSSOLink

Get contact details for an account.

50.1 Contents

- 1 GET /accounts/{accountId}/contacts/{contactType}/supportSSOLink
  - 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 Examples
    - 1.3.1 Successful Request
    - 1.3.2 Request for SSO link of non-verified email
    - 1.3.3 Request for non-existent account Id or contact Type
  - 1.4 See also

50.2 Request

`GET /accounts/{accountId}/contacts/{contactType}/supportSSOLink`

50.2.1 Request Parameters

accountId - *string*
: The unique account Id.

contactType - *string*
: The contact type that needs support SSO link.
Note, there are 4 available types of contact information and you can request SSO link for one of them per request: "regular", "billing", "administrator" and "technical".

50.2.2 URI Parameters

None.

50.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

50.3 Response

50.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The request was sent without an *Authorization* header, or the user represented by the value of this header was not properly authenticated.

403 Forbidden
: The email address of the user authenticated for this request is not verified.

404 Not Found
: The specified account Id or contact type for this account Id does not exist or is purged.

50.3.2 Response Body

```
{
  "ssoLink" : "{ssoLink}",
  "links" : [ {
    "href" : "https://api.hostway.com/accounts/{accountId}/contacts/{contactType}/supportSSOLink/",
    "rel" : "self"
  } ]
}
```

50.3.2.1 Parameters

ssoLink - *string*
: Contains a link for SSO login to ZenDesk platform.

links - *array*
: Hypermedia for this resource.

50.4 Examples

50.4.1 Successful Request

Request

```
GET /accounts/{accountId}/contacts/{contactType}/supportSSOLink
Authorization: Basic {Base64-encoded username & password string}
```

Response

```
200 OK

{
  "ssoLink" : "https://hostway.zendesk.com/access/jwt?jwt={authString}",
  "links" : [ {
    "href" : "https://api.hostway.com/accounts/{accountId}/contacts/{contactType}/supportSSOLink/",
    "rel" : "self"
  } ]
}
```

50.4.2 Request for SSO link of non-verified email

Request

```
GET /accounts/{AccountId}/contacts/{ContactType}/supportSSOLink
Authorization: Basic {Base64-encoded username & password string}
```

Response

```
403 Forbidden

{
  "forbidden" : {
    "guid" : "{identifier}",
    "message" : "Email address is not verified",
    "code" : 403,
    "details" : ""
  }
}
```

50.4.3 Request for non-existent account Id or contact Type

Request

```
GET /accounts/{fakeAccountId}/contacts/{fakeContactType}/supportSSOLink
Authorization: Basic {Base64-encoded username & password string}
```

Response

```
404 Not Found

{
  "itemNotFound" : {
    "guid" : "{identifier}",
    "message" : "Resource not found",
    "code" : 404,
    "details" : ""
  }
}
```

50.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

51 Accounts-accountId-users:GET

52 GET /accounts/*{accountId}*/users

Returns a list of users in an account

52.1 Contents

- 1 GET /accounts/{accountId}/users
  - 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 See Also

52.2 Request

`GET /accounts/{accountId}/users`

52.2.1 Request Parameters

accountId - *string*
: The Id of the account

52.2.2 URI Parameters

None

52.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

52.3 Response

52.3.1 Status Code

200 OK
: Success

304 Not changed
: The list of users hasn't changed since the last request; the client should use its locally cached copy instead.

401 Unauthorized
: The authorization header contained an invalid value

403 Forbidden
: Successfully authenticated, but client is not allowed to view the list of users. Access is only permitted for agents and account owners of this account. An account owner is a user with the "sitecontrol_account_owner" role.

404 Not Found
: The account in question does not exist

52.3.2 Response Body

```
{
  "list": [
    {
      "userName": "{userName}",
      "details": "https://api.hostway.com/accounts/{accountId}/users/{userName}"
    },
    ...
  ],
  "links": [
    {"href": "https://api.hostway.com/accounts/{accountId}/users", "rel": "self"}
  ]
}
```

52.3.2.1 Parameters

accountNumber
: the Id of the account

userName
: the username of the user

52.4 See Also

- Account
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

53 Accounts-accountId-users:POST

54 POST /accounts/{accountId}/users

Create a new user under an account

54.1 Contents

- 1 POST /accounts/{accountId}/users
  - 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
  - 1.3 Allowed Security Questions
  - 1.4 See also

54.2 Request

```
POST /accounts/{accountId}/users
```

54.2.1 Request Parameters

accountId
: The ID of the account

54.2.2 URI Parameters

None

54.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

54.2.4 Request Body

```
{
  "userName": "{userName}",
  "password": "{password}",
  "securityQuestion": "{securityQuestionCode}",
  "securityAnswer": "{securityAnswer}"
}
```

54.2.4.1 Parameters

userName
: *Required.* A name for the user that's about to be created. This needs to be unique within the entire security realm (not just within the account).

password
: *Required.* A password for the user. This needs to be between 8 and 50 characters long inclusive, must contain at least one alpha *and* at least one numeric character, and must not contain any of the following characters: & ` ' " \ / < > $

securityQuestionCode
: *Required.* The reference code for a question that the user can use to recover the password in case (s)he forgot it. This must be selected from a pre-approved list of questions.

securityAnswer
: *Required.* An answer to the security question. If the user provides the correct answer during password recovery, the password will be reset and (s)he will be allowed to access the account. The API will automatically normalize the security answer by removing starting and ending spaces, lowercase the security answer and replace each sequence of whitespaces with just one single space character.

54.3 Response

54.3.1 Status Code

201 Created

The user was successfully created

400 Bad Request
: The request body was missing a required field or one of the fields did not comply with the requirements (in this case, the response body will contain hints for what went wrong)

401 Unauthorized
: Authentication failed

403 Forbidden
: Authentication succeeded, but the logged-in user is not allowed to create users in this account. The only users allowed to create users in this account are sales agents and account owners of *this* account. An account owner is any account user with the "sitecontrol_account_owner" role.

404 Not Found
: The account does not exist

409 Conflict
: The user cannot be created because another user exists with the same username.

54.3.2 Response Headers

Location
: The URI for the newly created user

54.3.3 Response Body

None

54.4 Allowed Security Questions

TBD

54.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

55 Accounts-accountId-users-userName:GET

56 GET /accounts/{accountId}/users/{userName}

Return the details of one specific user

56.1 Contents

- 1 GET /accounts/{accountId}/users/{userName}
  - 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 See also

56.2 Request

```
GET /accounts/{accountId}/users/{userName}
```

56.2.1 Request Parameters

accountId - *string*
: The Id of the account

userName - *string*
: The username of the user to retrieve

56.2.2 URI Parameters

None

56.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

56.3 Response

56.3.1 Status Code

200 OK
: Success

304 Not changed
: The user details haven't changed since the last request; the client should use its locally cached copy

401 Unauthorized
: Authentication failed

403 Forbidden
: Authentication succeeded, but the authenticated user is not allowed to view this user resource. The only authenticated users allowed to view this user resource are *this very user*, the account owner(s) of *this* account, and sales/support agents, the user which authorization credentials match the requested user's credentials. An account owner is a user within the account that has the role "sitecontrol_account_owner".

404 Not found
: Either the account or the requested user does not exist.

56.3.2 Response Body

```
{
  "userName": "{userName}",
  "securityQuestion": "{securityQuestionCode}",
  "roles": "https://api.hostway.com/accounts/{accountId}/users/{userName}/roles",
  "links": [
    {"href": "https://api.hostway.com/accounts/{accountId}/users/{userName}", "rel": "self"}
  ]
}
```

56.3.2.1 Parameters

accountId
: The ID of the account

userName
: The username of this user

securityQuestionCode
: The security question code of this user (this will be used to refer to a list of security questions)

56.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

57 Accounts-accountId-users-userName:PUT

58 PUT /accounts/{accountId}/users/{userName}

Modify the password and/or security question/answer of a user

58.1 Contents

- 1 PUT /accounts/{accountId}/users/{userName}
  - 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
  - 1.3 See also

58.2 Request

```
PUT /accounts/{accountId}/users/{userName}
```

58.2.1 Request Parameters

accountId
: The ID of the account

userName
: The username of the user

58.2.2 URI Parameters

None

58.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

58.2.4 = Request Body

```
{
  "userName": "{userName}",
  "password": "{password}",
  "securityQuestion": "{securityQuestionCode}",
  "securityAnswer": "{securityAnswer}"
}
```

58.2.4.1 Parameters

userName
: *Required.* A name for the user that's about to be created. This needs to be unique within the entire security realm (not just within the account).

password
: *Required only if securityQuestion and securityAnswer were not provided.* A password for the user. This needs to be between 8 and 50 characters long inclusive, must contain at least one alpha *and* at least one numeric character, and must not contain any of the following characters: & ` ' " \ / < > $

securityQuestionCode
: *Required if password was not provided.* The reference code for a question that the user can use to recover the password in case (s)he forgot it. This must be selected from a pre-approved list of questions. If securityQuestion was provided, a securityAnswer must also be provided.

securityAnswer
: *Required if password was not provided.* An answer to the security question. If the user provides the correct answer during password recovery, the password will be reset and (s)he will be allowed to access the account. If securityAnswer was provided, a securityQuestion must also be provided. The API will automatically normalize the security answer by removing starting and ending spaces, lowercase the security answer and replace each sequence of whitespaces with just one single space character.

58.3 Response

58.3.1 Status Code

204 No Content
: The user was successfully updated

400 Bad Request
: The request body was missing a required field or one of the fields did not comply with the requirements (in this case, the response body will contain hints for what went wrong)

401 Unauthorized
: Authentication failed

403 Forbidden
: Authentication succeeded, but the authenticated user is not allowed to view this user resource. The only authenticated users allowed to view this user resource are *this very user*, the account owner(s) of *this* account, and sales/support agents, the user which authorization credentials match the requested user's credentials. An account owner is a user within the account that has the role "sitecontrol_account_owner".

404 Not Found
: Either the account or user does not exist

58.3.2 Response Headers

None

58.3.3 Response Body

None

58.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

59 Accounts-accountId-user:PUT

60 PUT /accounts/*{accountId}*/user

Updates account user credentials.

60.1 Contents

- 1 PUT /accounts/{accountId}/user
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
  - 1.3 Examples
  - 1.4 See also

60.2 Request

`PUT /accounts/{accountId}/user`

60.2.1 Parameters

accountId - *string*
: The Id of the account.

60.2.2 URI Parameters

None.

60.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Content-Type
: application/json

60.2.4 Request Body

```
{
  "userName": "{username}",
  "password": "{password}"
  "securityQuestion": "{securityQuestionCode}",
  "securityAnswer": "{securityAnswer}"
}
```

60.2.4.1 Parameters

userName
: *Required.* A name for the user that's about to be created. This needs to be unique within the entire security realm (not just within the account).

password
: *Required only if securityQuestion and securityAnswer were not provided.* A password for the user. This needs to be between 8 and 50 characters long inclusive, must contain at least one alpha *and* at least one numeric character, and must not contain any of the following characters: & ` ' " \ / < > $

securityQuestionCode
: *Required if password was not provided.* The reference code for a question that the user can use to recover the password in case (s)he forgot it. This must be selected from a pre-approved list of questions. If securityQuestion was provided, a securityAnswer must also be provided.

securityAnswer
: *Required if password was not provided.* An answer to the security question. If the user provides the correct answer during password recovery, the password will be reset and (s)he will be allowed to access the account. If securityAnswer was provided, a securityQuestion must also be provided. The API will automatically normalize the security answer by removing starting and ending spaces, lowercase the security answer and replace each sequence of whitespaces with just one single space character.

60.3 Response

60.3.1 Status Code

204 No Content
: The user was successfully updated

400 Bad Request
: The request body was missing a required field or one of the fields did not comply with the requirements (in this case, the response body will contain hints for what went wrong)

401 Unauthorized
: Authentication failed

403 Forbidden
: Authentication succeeded, but the logged-in user is not allowed to update users in this account. The only users allowed to update this user resource are *this very user*, account owner(s) of *this* account, and sales/support agents. An account owner is any account user with the "sitecontrol_account_owner" role.

404 Not Found
: Either the account or user does not exist

415 Unsupported Media Type
: The Content-Type header was not specified or the specified type is not supported. Use application/json.

60.4 Examples

Request

```
PUT /accounts/test-account/user

{ "userName":"testuser", "password":"password12" }
```

Response

```
401 Unauthorized
```

Request

```
PUT /accounts/test-account/user
Authorization: Basic base64_encode(username:password)

{ "userName":"testuser", "password":"password12" }
```

Response

```
415 Unsupported Media Type

{"badMediaType": {"message": "Unsupported media", "code": 415, "details": ""}}
```

Request

```
PUT /accounts/test-account/user
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{ "userName":"notvalid-user", "password":"password12" }
```

Response

```
404 Not Found

{"itemNotFound": {"message": "Resource not found", "code": 404, "details": ""}}
```

Request

```
PUT /accounts/test-account/user
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{ "userName":"testuser", "password":"" }
```

Response

```
400 Bad Request

{"badRequest": {"message": "POST data error", "code": 400, "details": {"password": "Required"}}}
```

Request

```
PUT /accounts/test-account/user
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{ "userName":"testuser", "password":"pass" }
```

Response

```
400 Bad Request

{"computeFault": {"message": "The password should be at least 8 characters", "code": 400, "details": ""}}
```

Request

```
PUT /accounts/test-account/user
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{ "userName":"testuser", "password":"pa$$word" }
```

Response

```
400 Bad Request
```

```
{"computeFault": {"message": "The password must contains digits and letters", "code": 400, "details": ""}}
```

Request

```
PUT /accounts/test-account/user
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{ "userName":"testuser", "password":"password" }
```

Response

```
400 Bad Request

{"computeFault": {"message": "The password must contains at least one digit", "code": 400, "details": ""}}
```

Request

```
PUT /accounts/test-account/user
Authorization: Basic base64_encode(username:password)
Content-Type: application/json

{ "userName":"testuser", "password":"password12" }
```

Response

```
204 No Content
```

60.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

61 Accounts-accountId-users-userName:DELETE

62 DELETE /accounts/{accountId}/users/{userName}

Delete a user

62.1 Contents

- 1 DELETE /accounts/{accountId}/users/{userName}
  - 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 Headers
    - 1.2.3 Response Body
  - 1.3 See also

62.2 Request

```
DELETE /accounts/{accountId}/users/{userName}
```

62.2.1 Request Parameters

accountId - *string*
: The Id of the account

userName - *string*
: The username of the user to delete

62.2.2 URI Parameters

None

62.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

62.3 Response

62.3.1 Status Code

204 No Content
: User was successfully deleted

401 Unauthorized
: Authentication failed

403 Forbidden
: Authentication succeeded, but the authenticated user is not allowed to delete this user resource. The only authenticated users allowed to delete this user resource are the account owner(s) of *this* account, and sales/support agents. Account owners cannot delete themselves. An account owner is a user within the account that has the role "sitecontrol_account_owner".

404 Not found
: Either the account or the requested user does not exist.

62.3.2 Response Headers

None

62.3.3 Response Body

None

62.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

63 Accounts-accountId-users-userName-roles:GET

64 GET /accounts/{accountId}/users/{userName}/roles

Retrieve the roles of a user

64.1 Contents

- 1 GET /accounts/{accountId}/users/{userName}/roles
  - 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 See also

64.2 Request

```
GET /accounts/{accountId}/users/{userName}/roles
```

64.2.1 Request Parameters

accountId - *string*
: The Id of the account

userName - *string*
: The username of the user

64.2.2 URI Parameters

None

64.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

64.3 Response

64.3.1 Status Code

200 OK
: Success

304 Not changed
: The user's roles haven't changed since the last request; the client should use its locally cached copy

401 Unauthorized
: Authentication failed

403 Forbidden
: Authentication succeeded, but the authenticated user is not allowed to view this user's roles. The only authenticated users allowed to view these roles are *this very user*, the account owner(s) of *this* account, and sales/support agents. An account owner is a user within the account that has the role "sitecontrol_account_owner".

404 Not found
: Either the account or the requested user does not exist.

64.3.2 Response Body

```
{
  "list": [
    "role1",
    "role2"
  ],
  "links": [
    {"href": "https://api.hostway.com/accounts/{accountId}/users/{userName}/roles", "rel": "self"}
  ]
}
```

64.3.2.1 Parameters

accountId
: The ID of the account

userName
: The username of this user

64.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

65 Accounts-accountId-users-userName-roles:PUT

66 PUT /accounts/{accountId}/users/{userName}/roles

Modify the roles of a user.

66.1 Contents

- 1 PUT /accounts/{accountId}/users/{userName}/roles
  - 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 Allowed Roles
  - 1.2 Response
    - 1.2.1 Status Code
  - 1.3 See also

66.2 Request

```
PUT /accounts/{accountId}/users/{userName}/roles
```

66.2.1 Request Parameters

accountId
: The ID of the account

userName
: The username of the user

66.2.2 URI Parameters

None

66.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

66.2.4 Request Body

List of new roles to be set to the requested user

```
[
  "role1",
  "role2"
]
```

66.2.4.1 Allowed Roles

Can be retrieved with GET /roles call.

66.3 Response

66.3.1 Status Code

204 No Content
: The user's roles were successfully updated

400 Bad Request
: The request body was malformed, or the specified roles are not allowed, or the account owner was attempting to demote himself/herself which is not allowed. (in this case, the response body will contain hints for what went wrong)

401 Unauthorized
: Authentication failed

403 Forbidden
: Authentication succeeded, but the logged-in user is not allowed to update the roles of this user. The only users allowed to update the roles of this user are *this very user*, account owner(s) of *this* account, and sales/support agents. An account owner is any account user with the "sitecontrol_account_owner" role. An account owner may not remove their own "sitecontrol_account_owner" role, though they may do this to other account owners.

404 Not Found
: Either the account or user does not exist

66.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

67 Accounts-accountId-products:GET

68 GET /accounts/*{accountId}*/products

Retrieves a list of all products associated with a given account.

68.1 Contents

- 1 GET /accounts/{accountId}/products
  - 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 Alternative with partner Id
    - 1.3.1 Parameters
  - 1.4 See also

68.2 Request

`GET /accounts/{accountId}/products`

68.2.1 Parameters

accountId - *string*
: The Id of the account.

68.2.2 URI Parameters

filterStatus - *string*
: Optional. Only products with status matching the value of the parameter will be returned. Allowed values are **pending**, **open**, **suspended**, **closed**.

filterDisplayName - *string*
: Optional. Only products with displayName matching the value of the parameter will be returned.

filterPartnerId - *string*
: Optional. Only products with parnterId matching the value of the parameter will be returned.

68.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

68.3 Response

68.3.1 Status Code

200 OK
: Success

404 Not Found
: The account Id was not found.

403 Forbidden
: Access to **closed** and **suspended** accounts requires elevated permissions

68.3.2 Response Body


```
{
  "list": [
    {
      "name": "Product 1",
      "displayName": "My custom product name",
      "partnerId": "Partner Id",
      "productSpecification": "https://api.hostway.com/productSpecifications/{productSpecId1}",
      "bandwidth": "https://api.hostway.com/accounts/{accountId}/products/{productId1}/bandwidth/",
      "usage": "https://api.hostway.com/accounts/{accountId}/products/{productId1}/usage/",
      "status": "open",
      "billingCycle": 1,
      "links": [
        {"href": "https://api.hostway.com/accounts/{accountId}/{productId1}", "rel": "self"}
      ]
    },
    {
      "name": "Product 2",
```

```
      "displayName": "My custom product name",
      "partnerId": "Partner Id",
      "productSpecification": "https://api.hostway.com/productSpecifications/{productSpecId2}"
      "bandwidth": "https://api.hostway.com/accounts/{accountId}/products/{productId2}/bandwidth/",
      "usage": "https://api.hostway.com/accounts/{accountId}/products/{productId2}/usage/",
      "status": "open",
      "billingCycle": 1,
      "links": [
        {"href": "https://api.hostway.com/accounts/{accountId}/products/{productId2}", "rel": "self"}
      ],
    }
  ],
  "links": [
    {"href": "https://api.hostway.com/accounts/{accountId}/products", "rel": "self"}
  ]
}
```


68.3.2.1 Parameters

list - *list*
: A collection of dictionaries containing information about specific products

  status - *string*
  : Describes the status of a product. Valid values are **open**, **closed** or **suspended**

  name - *string*
  : Descriptive name of the product

  partnerId - *string*
  : Partner Id specified for this product.

  billingCycle - *integer*
  : Billing cycle (length of billing period in months) of the service. Allowed values are from **1** to **120**

  productSpecification - *string*
  : Hypermedia for this product resource. Link to the product's specification.

  bandwidth - *string*
  : Hypermedia for this product resource. Link to the bandwidth's information.

  usage - *string*
  : Hypermedia for this product resource. Link to the usage's information.

  links - *list*
  : Hypermedia for this product resource. Links to the product.

links - *list*
: Hypermedia for this resource. Links to self.

68.4 Alternative with partner Id

As an alternative to using the account Id the client can use a partner-specific Id set via the account attribute partner_account_id. The response to this request is the same as the response to a request using the account Id.

`GET /partnerAccounts/{partnerAccountId}/products`

68.4.1 Parameters

partnerAccountId - *string*
: Partner account Id set via the partner_account_id attribute.

68.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

69 Accounts-accountId-products-productId:GET

70 GET /accounts/*{accountId}*/products/{productId}

Get product information and (if requested) possible product changes.

70.1 Contents

- 1 GET /accounts/{accountId}/products/{productId}
  - 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 See also

70.2 Request

`GET /accounts/{accountId}/products/{productId}`

70.2.1 Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

70.2.2 URI Parameters

getTargetProductSpecifications - *boolean*
: Optional. If set to 1 the response will include valid product specification change targets list.

70.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

70.3 Response

70.3.1 Status Code

200 OK
: Success.

401 Unauthorized
: Authorization credentials not submitted or invalid.

404 Not Found
: The account or product Id not found.

70.3.2 Response Body

The response body includes product specification change targets list when the *getTargetProductSpecifications* URI parameter is set to 1.

```
{
  "status": "open",
  "name": "Product Name",
  "displayName": "My custom product name",
  "partnerId": "Partner Id",
  "billingCycle": 12,
  "productSpecification": "https://api.hostway.com/productSpecifications/{productSpecificationId}",
  "changes": [
    {
      "type": "upgrade",
      "id": "{targetProductSpecificationId}",
      "href": "https://api.hostway.com/productSpecifications/{targetProductSpecificationId}"
    },
  ]
  "links": [
    {"href": "https://api.hostway.com/accounts/{accountId}/products/{productId}/", "rel": "self"}
  ]
}
```

70.3.2.1 Parameters

status - *string*
: Current status of the product. Allowed values are **new**, **pending**, **open**, **suspended**, **close_pending**, **closed** and **purge_pending**.

name - *string*
: Name of the product.

billingCycle - *integer*

Billing cycle of the product in months.

productSpecification - *string*
Link to the specification of the current product.

changes - *list*
A list with the valid product change targets. Returned if *getTargetProductSpecifications=1* query parameter is passed

displayName - *string*
A custom display name for the product.

partnerId - *string*
Partner Id specified for this product.

links - *list*
Hypermedia for the product resource

70.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

71 Accounts-accountId-products-productId:PUT

72 PUT /accounts/*{accountId}*/products/*{productId}'*

Updates the status of an existing product.

72.1 Contents

- 1 PUT /accounts/{accountId}/products/{productId}'
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 See also

72.2 Request

`PUT /accounts/{accountId}/products/{productId}`

72.2.1 Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

72.2.2 URI Parameters

None.

72.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

72.2.4 Request Body

```
{"status": "new_status"}
```

72.2.4.1 Parameters

status - *string*
: Can be one of the following statuses: **open**, **suspended**, **closed**, **purged**.

72.3 Response

72.3.1 Status Code

204 No Content
: The status was successfully changed.

400 Bad Request

The request body was invalid.

401 Unauthorized
: The authorization credentials were not supplied or are invalid.

404 Not Found
: The account or product Ids were not found.

409 Conflict
: The current status of the product does not allow for this change.

72.3.2 Response Body

None.

72.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

73 Accounts-accountId-products-productId-action:POST

74 POST /accounts/*{accountId}*/products/*{productId}*/action

Upgrades/downgrades existing product.

74.1 Contents

- 1 POST /accounts/{accountId}/products/{productId}/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
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 See also

74.2 Request

`POST /accounts/{accountId}/products/{productId}/action`

74.2.1 Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

74.2.2 URI Parameters

None.

74.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

74.2.4 Request Body

```
{
  "action": "change",
  "actionData": {
    "targetProductSpecification": "https://api.hostway.com/productSpecifications/{productSpecificationId}"
  }
}
```

74.2.4.1 Parameters

action - *string*
: The type of the requested action. Currently supported actions:
  change - change the product specification (upgrade/downgrade).

actionData - *dictionary*
: Contains action-specific data.
  change action:
    targetProductSpecification - the URI of the new product specification.

74.3 Response

74.3.1 Status Code

204 No Content
: The product update was successfully submitted.

400 Bad Request
: Request body is invalid.

401 Unauthorized
: The authorization credentials were not supplied or are invalid.

404 Not Found
: The account or product Ids were not found.

74.3.2 Response Body

None.

74.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

75 Accounts-accountId-products-productId-services:GET

76 GET /accounts/*{accountId}*/products/*{productId}*/services

Retrieves a list of all services associated with a given product.

76.1 Contents

- 1 GET /accounts/{accountId}/products/{productId}/services
  - 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 See also

76.2 Request

```
GET /accounts/{accountId}/products/{productId}/services
```

76.2.1 Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

76.2.2 URI Parameters

None.

76.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

76.3 Response

76.3.1 Status Code

200 OK
: Success

404 Not Found
: The account Id or product Id was not found.

403 Forbidden
: Access to **closed** and **suspended** accounts requires elevated permissions

76.3.2 Response Body

```
{
  "list": [
    {
      "name": "Product 1",
      "serviceSpecification": "https://api.hostway.com/serviceSpecifications/{serviceSpecId1}",
      "status": "open",
      "billingCycle": 1,
      "links": [
        {"href": "https://api.hostway.com/accounts/{accountId}/products/{productId}/services/{serviceId1}", "rel": "self"}
      ]
    },
    {
      "name": "Product 2",
      "serviceSpecification": "https://api.hostway.com/serviceSpecifications/{serviceSpecId2}"
      "status": "open",
      "billingCycle": 1,
      "links": [
        {"href": "https://api.hostway.com/accounts/{accountId}/products/{productId}/services/{serviceId2}", "rel": "self"}
      ],
    }
  ],
  "links": [
    {"href": "https://api.hostway.com/accounts/{accountId}/products/{productId}/services", "rel": "self"}
  ]
}
```

76.3.2.1 Parameters

list - *list*
: A collection of dictionaries containing information about specific services

  status - *string*
  : Describes the status of a service. Valid values are **open**, **closed**, **pending**, or **suspended**

  name - *string*
  : Descriptive name of the service

billingCycle - *integer*
Billing cycle (length of billing period in months) of the service. Allowed values are from **1** to **120**

serviceSpecification - *string*
Hypermedia for this service resource. Link to the service's specification.

links - *list*
Hypermedia for this service resource. Links to the service.

links - *list*
Hypermedia for this resource. Links to self.

76.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

77 Accounts-accountId-products-productId-services-serviceId:GET

78 GET /accounts/*{accountId}*/products/*{productId}*/services/*{serviceId}*

Get service information.

78.1 Contents

- 1 GET /accounts/{accountId}/products/{productId}/services/{serviceId}
  - 1.1 Request
    - 1.1.1 Parameters
    - 1.1.2 Request Headers
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 See also

78.2 Request

`GET /accounts/{accountId}/products/{productId}/services/{serviceId}`

78.2.1 Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

serviceId - *string*
: The Id of the service.

78.2.2 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

78.3 Response

78.3.1 Status Code

200 OK
: Success.

401 Unauthorized
: Authorization credentials not submitted or invalid.

404 Not Found
: The account, product Id or service Id not found.

78.3.2 Response Body

```
{
  "status": "open",
  "name": "Service Name",
  "displayName": "service-related-domain.com",
  "billingCycle": 12,
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/{serviceSpecificationId}",
  "links": [
    {
      "href": "https://api.hostway.com/accounts/{accountId}/products/{productId}/services/{serviceId}",
      "rel": "self"
    },
    {
      "href": "https://api.hostway.com/{detailsLink}",
      "rel": "manage"
    }
  ]
}
```

78.3.2.1 Parameters

status - *string*
: Current status of the service. Allowed values are **new**, **pending**, **open**, **suspended**, **close_pending**, **closed** and **purge_pending**.

name - *string*
: Name of the service.

displayName - *string*
: Domain name for the service if any exists.

billingCycle - *integer*
: Billing cycle of the service in months.

serviceSpecification - *string*
: Link to the specification of the current service.

links - *list*
: *self* - Hypermedia for the service resource

manage - Hypermedia for some service types. *detailsLink* format is:
for Domain type: *domains/{domainName}/details*. Please see: Domains-domainName-details:GET
for Email type: *accounts/{accountId}/email/{domainName}*. Please see: Accounts-account-email-domain:GET
for Exchange type: *accounts/{accountId}/exchange/{domainName}/mailboxes*. Please see: GET Exchange Mailboxes list
for Dedicated Server type: *accounts/{accountId}/dedicatedServers/{serverId}*. Please see:
Accounts-account-dedicatedServers-server:GET

78.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

79 Accounts-accountId-products-productId-services-serviceId:PUT

80 PUT /accounts/*{accountId}*/products/*{productId}*/services/*{serviceId}'*

Updates the status of an existing service.

80.1 Contents

- 1 PUT /accounts/{accountId}/products/{productId}/services/{serviceId}'
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 See also

80.2 Request

`PUT /accounts/{accountId}/products/{productId}/services/{serviceId}`

80.2.1 Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

serviceId- *string*
: The Id of the service.

80.2.2 URI Parameters

None.

80.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

80.2.4 Request Body

```
{"status": "new_status"}
```

80.2.4.1 Parameters

status - *string*
: Can be one of the following statuses: **open**, **suspended**, **closed**, **purged**.

80.3 Response

80.3.1 Status Code

204 No Content
: The status was successfully changed.

400 Bad Request

The request body was invalid.

401 Unauthorized
: The authorization credentials were not supplied or are invalid.

404 Not Found
: The account or product or service Ids were not found.

409 Conflict
: The current status of the product or service does not allow for this change.

80.3.2 Response Body

None.

80.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

ccounts-accountId-products-productId-services-serviceId-getTargetServiceSpecifications:

82 GET /accounts/*{accountId}*/products/*{productId}*/services/*{serviceId}*/?getTargetServiceSpe

Get allowed PTC changes

82.1 Contents

- 1 GET /accounts/{accountId}/products/{productId}/services/{serviceId}/?getTargetServiceSpecifications=1
  - 1.1 Request
    - 1.1.1 Parameters
    - 1.1.2 Request Headers
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 See also

82.2 Request

`GET /accounts/{accountId}/products/{productId}/services/{serviceId}/?getTargetServiceSpecifications=1`

82.2.1 Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

serviceId - *string*
: The Id of the service.

82.2.2 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

82.3 Response

82.3.1 Status Code

200 OK
: Success.

401 Unauthorized
: Authorization credentials not submitted or invalid.

404 Not Found
: The account, product Id or service Id not found.

82.3.2 Response Body

```
{
  "status": "open",
  "name": "Service Name",
  "displayName": "service-related-domain.com",
  "billingCycle": 12,
  "serviceSpecification": "https://api.hostway.com/serviceSpecifications/{serviceSpecificationId}",
  "links": [
    {
      "href": "https://api.hostway.com/accounts/{accountId}/products/{productId}/services/{serviceId}",
      "rel": "self"
    },
    {
      "href": "https://api.hostway.com/{detailsLink}",
      "rel": "manage"
    }
  ],
  "changes": [
    {
      "href": "https://api.hostway.com/serviceSpecifications/{serviceSpecificationId}",
      "type": "{type}",
      "name": "Target Service Name",
      "billingCycle": 1
    }
  ]
}
```

82.3.2.1 Parameters

status - *string*
: Current status of the service. Allowed values are **new**, **pending**, **open**, **suspended**, **close_pending**, **closed** and **purge_pending**.

name - *string*
: Name of the service.

displayName - *string*

Domain name for the service if any exists.

billingCycle - *integer*
Billing cycle of the service in months.

serviceSpecification - *string*
Link to the specification of the current service.

type - *string*
Type of service change. Might be upgrade, downgrade

links - *list*
self - Hypermedia for the service resource
manage - Hypermedia for some service types. *detailsLink* format is:
for Domain type: *domains/{domainName}/details*. Please see: Domains-domainName-details:GET
for Email type: *accounts/{accountId}/email/{domainName}*. Please see: Accounts-account-email-domain:GET
for Exchange type: *accounts/{accountId}/exchange/{domainName}/mailboxes*. Please see: GET Exchange Mailboxes list
for Dedicated Server type: *accounts/{accountId}/dedicatedServers/{serverId}*. Please see: Accounts-account-dedicatedServers-server:GET

82.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

83 Accounts-accountId-products-productId-services-serviceId-action:POST

84 POST /accounts/*{accountId}*/products/*{productId}*/services/*{serviceId}*/action

Upgrades/downgrades existing service.

84.1 Contents

- 1 POST /accounts/{accountId}/products/{productId}/services/{serviceId}/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
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 See also

84.2 Request

`POST /accounts/{accountId}/products/{productId}/services/{serviceId}/action`

84.2.1 Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

serviceId - *string*
: The Id of the service.

84.2.2 URI Parameters

None.

84.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

84.2.4 Request Body

```
{
  "action": "change",
  "actionData": {
    "targetServiceSpecification": "https://api.hostway.com/serviceSpecifications/{serviceSpecificationId}"
  }
}
```

84.2.4.1 Parameters

action - *string*
: The type of the requested action. Currently supported actions:
  cost - show the available fees for PTC change
  validate - validate if the service specification change is allowed
  change - change the service specification (upgrade/downgrade)
  changeQuantityCost - get the cost for changing a service quantity
  changeQuantity - change the service quantity.

actionData - *dictionary*
: Contains action-specific data.
  validate action:
    targetServiceSpecification - the URI of the new service specification.
  change action:
    targetServiceSpecification - the URI of the new service specification
    carryOverDiscounts - indicates whether current active discounts should be preserved (optional).
  changeQuantityCost action:
    quantity - the target quantity
    carryOverDiscounts - indicates whether current active discounts should be preserved (optional).
  changeQuantity action:
    quantity - the target quantity
    carryOverDiscounts - indicates whether current active discounts should be preserved (optional).

84.3 Response

Response for getting the fees

```
{
        "links": [{
                "href": "http://api.hostway.com/accounts/{accountId}/products/{productId}/services/{serviceId}/action/",
                "rel": "self"
        }],
        "oneTimeFee": {
                "currency": "USD",
                "amount": "9.95"
        },
```

```
        "taxes": {
                "amount": "0"
        },
        "total": {
                "currency": "USD",
                "amount": "19.90"
        },
        "subTotal": {
                "currency": "USD",
                "amount": "19.90"
        }
}
```

Response for the validation

```
{
        "links": [{
                "href": "http://api.hostway.com/accounts/{accountId}/products/{productId}/services/{serviceId}/action/",
                "rel": "self"
        }],
        "allowed": 1
}
```

84.3.1 Status Code

200 No Content
: The service update was successfully validated.

204 No Content
: The service update was successfully submitted.

400 Bad Request
: Request body is invalid.

401 Unauthorized
: The authorization credentials were not supplied or are invalid.

404 Not Found
: The account, product Id or service Id not found.

84.3.2 Response Body

None.

84.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

85 Accounts-accountId-products-productId-bandwidth-daily:GET

86 GET /accounts/*{accountId}*/products/*{productId}*/bandwidth/daily?filterType=*{filterType}*&fr

Get a report on the daily bandwidth usage of the product

Each bandwidth usage is filtered by the traffic type(www, ftp), the start date and the end date in ISO format(eg. 2013-02-20)

86.1 Contents

- 1 GET /accounts/{accountId}/products/{productId}/bandwidth/daily?filterType={filterType}&from={filterStartDate}&to={filterEndDate}
  - 1.1 Request
    - 1.1.1 Request Parameters
    - 1.1.2 URI Parameters
    - 1.1.3 Request Headers
    - 1.1.4 Request Body
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 GET call
  - 1.4 See also

86.2 Request

`GET /accounts/{accountId}/products/{productId}/bandwidth/daily?filterType={filterType}&from={filterStartDate}&to={filterEn`

86.2.1 Request Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

86.2.2 URI Parameters

filterType(optional) - *string* - *optional*
: Should be one of either **www** or **ftp**. In case the total(www + ftp) bandwidth is needed, this parameter can be omitted. If it's omitted, the response will include separate entries for the different types of bandwidth (it will not sum them up for each date).

filterStartDate - *string* - *required*
: Start date of the bandwidth usage report. Should be in **YYYY-MM-DD** format. The usage returned includes usage with dates from **filterStartDate 00:00:00 UTC**. *filterStartDate* must be older or equal to *filterEndDate*.

filterEndDate - *string* - *required*
: Start date of the bandwidth usage report. Should be in **YYYY-MM-DD** format. The bandwidth usage returned includes usage with dates to **filterEndDate 23:59:59 UTC**. *filterEndDate* must be older than today's date in UTC.

showBlanks - *optional*
: When the showBlanks parameter is passed, the response body will include values for all dates and types, even if there is no associated traffic.

86.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

86.2.4 Request Body

None

86.3 Response

86.3.1 Status Code

200 OK
: Success.

400 Bad Request
: A request is considered *bad* any of the following situation is available:
- **filterType** is not one of **www**, **ftp**;
- **filterStartDate** or **filterEndDate** URI parameter is missing;
- **filterEndDate** is older than **filterStartDate**;
- either **filterEndDate** or **filterStartDate** is today's date or a future date;
- **filterEndDate** is more than 100 days after **filterStartDate**

403 Forbidden
: Forbidden access to the account.

404 Not Found
: The account Id or product Id was not found.

86.3.2 Response Body

```
{
 "list" : [
 {
      "date": {dateISOFormat},
      "type": {trafficType},
      "domain": {domainName},
      "bytes": {numberOfBytes}
  } ,...,
  {
      "date": {dateISOFormat},
      "type": {trafficType},
      "domain": {domainName},
      "bytes": {numberOfBytes}
  } ],
  "links" : [ {
    "href" : "{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/daily?from=2012-01-01
    "rel" : "self"
  } ]
}
```

86.3.2.1 Parameters

list - *list*
: a list of all bandwidth usage records. Each record consists of:

  dateISOFormat - *string*
  : the date for which the bandwidth is collected

  trafficType - *string*
  : type of the traffic(web or ftp)

  domainName - *string*
  : the domain name, for which the bandwidth data is collected

  numberOfBytes - *integer*
  : the bandiwdth calculated in bytes

links - *list*
: Hypermedia for this resource. Links to self.

86.4 Examples

86.4.1 GET call

Request

GET
{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/daily?filterType=www&from=2012-01-0

Response

```
HTTP/1.1 200 OK
```

```
{
  "list" : [ {
      "date": "2012-01-01",
      "type": "www",
      "domain": "smithwidgets.com",
      "bytes": 12345678
  }, {
     "date": "2012-01-02",
     "type": "www",
     "domain": "smithwidgets.com" ,
     "bytes": 1048576
  }, {
    "date" : "2012-01-03",
    "type" : "www",
    "domain" : "smithwidgets.com",
    "bytes" : 2097152
  }],
  "links" : [ {
    "href" : "{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/daily?from=2012-01-02
    "rel" : "self"
  } ]
}
```

86.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

87 Accounts-accountId-products-productId-bandwidth-weekly:GET

88 GET /accounts/*{accountId}*/products/*{productId}*/bandwidth/weekly?filterType=*{filterType}*&

Get a report on the weekly bandwidth usage of the product

Each bandwidth usage is filtered by the traffic type(www, ftp), the start date and the end date in ISO format(eg. 2013-02-20)

88.1 Contents

- 1 GET /accounts/{accountId}/products/{productId}/bandwidth/weekly?filterType={filterType}&from={filterStartDate}&to={filterEndDate}
  - 1.1 Request
    - 1.1.1 Request Parameters
    - 1.1.2 URI Parameters
    - 1.1.3 Request Headers
    - 1.1.4 Request Body
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 GET call
  - 1.4 See also

88.2 Request

`GET /accounts/{accountId}/products/{productId}/bandwidth/weekly?filterType={filterType}&from={filterStartDate}&to={filterE`

88.2.1 Request Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

88.2.2 URI Parameters

filterType(optional) - *string* - *optional*
: Should be one of either **www** or **ftp**. In case the total(www + ftp) bandwidth is needed, this parameter can be omitted. If it's omitted, the response will include separate entries for each bandwidth type; they will not be summed for each week.

filterStartDate - *string* - *required*
: Start date of the bandwidth usage report. Should be in **YYYY-MM-DD**(ISO) format. The usage returned includes usage with dates from **filterStartDate 00:00:00 UTC**. *filterStartDate* must be older or equal to *filterEndDate*.

filterEndDate - *string* - *required*
: End date of the bandwidth usage report. Should be in **YYYY-MM-DD**(ISO) format. The bandwidth usage returned includes all dates to **filterEndDate 23:59:59 UTC**. *filterEndDate* must be older than today's date in UTC.

showBlanks - *optional*
: When the showBlanks parameter is passed, the response body will include values for all weeks and types, even if there is no associated traffic.

88.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

88.2.4 Request Body

None

88.3 Response

88.3.1 Status Code

200 OK
: Success.

400 Bad Request
: A request is considered *bad* any of the following situation is available:
  - **filterType** is not one of **www**, **ftp**;
  - **filterStartDate** or **filterEndDate** URI parameter is missing;
  - **filterEndDate** is older than **filterStartDate**;
  - either **filterEndDate** or **filterStartDate** are today's date or a future date;
  - **filterEndDate** is more than 100 weeks after **filterStartDate**

403 Forbidden
: Forbidden access to the account.

404 Not Found
: The account Id or product Id was not found.

88.3.2 Response Body

```
{
 "list" : [
 {
      "week": {weekNumber},
      "year": {yearNumber},
      "type": {trafficType},
      "domain": {domainName},
      "bytes": {numberOfBytes}
  } ,...,
  {
      "week": {weekNumber},
      "year": {yearNumber},
      "type": {trafficType},
      "domain": {domainName},
      "bytes": {numberOfBytes}
  } ],
  "links" : [ {
    "href" : "{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/weekly?from=2012-01-0
    "rel" : "self"
  } ]
}
```

88.3.2.1 Parameters

list - *list*
: a list of all bandwidth usage records. Each record consists of:

  week - *integer*
  : the week of the year for which the bandwidth is collected. This is a number from 1 to 52 or 53. "Week 1" is whichever week contains the date of January 4 of that year, starting with Monday and ending with Sunday. Any dates within the year that precede this week are considered to be within the last week of the previous year.

  year - *integer*
  : the year for which the bandwidth is collected

  trafficType - *string*
  : type of the traffic(www or ftp)

  domainName - *string*
  : the domain name, for which the bandwidth data is collected

  bytes - *integer*
  : the bandiwdth calculated in bytes

links - *list*
: Hypermedia for this resource. Links to self.

88.4 Examples

88.4.1 GET call

Request

GET
{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/weekly?filterType=www&from=2012-06

Response

```
HTTP/1.1 200 OK
```

```
{
  "list" : [ {
      "week": 23,
      "year": 2013
      "type": "www",
      "domain": "smithwidgets.com",
      "bytes": 12345678
  }, {
     "week": 24,
     "year": 2013
     "type": "www",
     "domain": "smithwidgets.com" ,
     "bytes": 1048576
  }, {
    "week" : 25,
    "year": 2013
    "type" : "www",
    "domain" : "smithwidgets.com",
    "bytes" : 2097152
  }],
  "links" : [ {
    "href" : "{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/daily?from=2013-06-03
    "rel" : "self"
  } ]
}
```

88.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

89 Accounts-accountId-products-productId-bandwidth-monthly:GET

90 GET /accounts/*{accountId}*/products/*{productId}*/bandwidth/monthly?filterType=*{filterType}*

Get a report on the mothly bandwidth usage of the product

Each bandwidth usage is filtered by the traffic type(www, ftp), the start date and the end date in ISO format(eg. 2013-02-20)

90.1 Contents

- 1 GET /accounts/{accountId}/products/{productId}/bandwidth/monthly?filterType={filterType}&from={filterStartDate}&to={filterEndDate}
  - 1.1 Request
    - 1.1.1 Request Parameters
    - 1.1.2 URI Parameters
    - 1.1.3 Request Headers
    - 1.1.4 Request Body
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 GET call
  - 1.4 See also

90.2 Request

`GET /accounts/{accountId}/products/{productId}/bandwidth/monthly?filterType={filterType}&from={filterStartDate}&to={filter`

90.2.1 Request Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

90.2.2 URI Parameters

filterType(optional) - *string* - *optional*
: Should be one of either **www** or **ftp**. In case the total(www + ftp) bandwidth is needed, this parameter can be omitted. If it's omitted, the response will contain separate entries for each bandwidth type for each month (the values will not be summed for each month).

filterStartDate - *string* - *required*
: Start date of the bandwidth usage report. Should be in **YYYY-MM-DD**(ISO) format. The usage returned includes usage with dates from **filterStartDate 00:00:00 UTC**. *filterStartDate* must be older or equal to *filterEndDate*.

filterEndDate - *string* - *required*
: End date of the bandwidth usage report. Should be in **YYYY-MM-DD**(ISO) format. The bandwidth usage returned includes all dates to **filterEndDate 23:59:59 UTC**. *filterEndDate* must be older than today's date in UTC.

showBlanks - *optional*
: When the showBlanks parameter is passed, the response body will include values for all months and types, even if there is no associated traffic.

90.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

90.2.4 Request Body

None

90.3 Response

90.3.1 Status Code

200 OK
: Success.

400 Bad Request
: A request is considered *bad* any of the following situation is available:
  - **filterType** is not one of **www**, **ftp**;
  - **filterStartDate** or **filterEndDate** URI parameter is missing;
  - **filterEndDate** is older than **filterStartDate**;
  - either **filterEndDate** or **filterStartDate** are today's date or a future date;
  - **filterEndDate** is more than 100 months after **filterStartDate**

403 Forbidden
: Forbidden access to the account.

404 Not Found
: The account Id or product Id was not found.

90.3.2 Response Body

```
{
 "list" : [
 {
      "month": {monthNumber},
      "year": {yearNumber},
      "type": {trafficType},
      "domain": {domainName},
      "bytes": {numberOfBytes}
  } ,...,
  {
      "month": {monthNumber},
      "year": {yearNumber}
      "type": {trafficType},
      "domain": {domainName},
      "bytes": {numberOfBytes}
  } ],
  "links" : [ {
    "href" : "{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/monthly?from=2012-01-
    "rel" : "self"
  } ]
}
```

90.3.2.1 Parameters

list - *list*
: a list of all bandwidth usage records. Each record consists of:

  monthNumber - *integer*
  : the number of the month for which the bandwidth is collected (January = 1, December = 12)

  yearNumber - *integer*
  : the year for which the bandwidth is collected

  trafficType - *string*
  : type of the traffic(www or ftp)

  domainName - *string*
  : the domain name, for which the bandwidth data is collected

  numberOfBytes - *integer*
  : the bandiwdth calculated in bytes

links - *list*
: Hypermedia for this resource. Links to self.

90.4 Examples

90.4.1 GET call

Request

GET
{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/monthly?filterType=www&from=2012-0

Response

```
HTTP/1.1 200 OK
```

```
{
  "list" : [ {
      "month": 1,
      "year": 2012,
      "type": "www",
      "domain": "smithwidgets.com",
      "bytes": 12345678
  }, {
      "month": 2,
      "year": 2012,
      "type": "www",
      "domain": "smithwidgets.com",
      "bytes": 12345678
  }, {
      "month": 3,
      "year": 2012,
      "type": "www",
      "domain": "smithwidgets.com",
      "bytes": 12345678
  }, {
      "month": 4,
      "year": 2012,
      "type": "www",
      "domain": "smithwidgets.com",
      "bytes": 12345678
  }, {
      "month": 5,
      "year": 2012,
      "type": "www",
      "domain": "smithwidgets.com",
      "bytes": 12345678
  },
```

```
],
  "links" : [ {
    "href" : "{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/monthly?filterType=ww
    "rel" : "self"
  } ]
}
```

90.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

91 Accounts-accountId-products-productId-bandwidth-yearly:GET

92 GET /accounts/*{accountId}*/products/*{productId}*/bandwidth/yearly?filterType=*{filterType}*&f

Get a report on the yearly bandwidth usage of the product

Each bandwidth usage is filtered by the traffic type(www, ftp), the start date and the end date in ISO format(eg. 2013-02-20)

92.1 Contents

- 1 GET /accounts/{accountId}/products/{productId}/bandwidth/yearly?filterType={filterType}&from={filterStartDate}&to={filterEndDate}
  - 1.1 Request
    - 1.1.1 Request Parameters
    - 1.1.2 URI Parameters
    - 1.1.3 Request Headers
    - 1.1.4 Request Body
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 GET call
  - 1.4 See also

92.2 Request

`GET /accounts/{accountId}/products/{productId}/bandwidth/yearly?filterType={filterType}&from={filterStartDate}&to={filterE`

92.2.1 Request Parameters

accountId - *string*
: The Id of the account.

productId - *string*
: The Id of the product.

92.2.2 URI Parameters

filterType(optional) - *string* - *optional*
: Should be one of either **www** or **ftp**. In case the total(www + ftp) bandwidth is needed, this parameter can be omitted. If it's omitted, the response will contain separate entries for each type for each year; the values will not be summed for each year.

filterStartDate - *string* - *required*
: Start date of the bandwidth usage report. Should be in **YYYY-MM-DD**(ISO) format. The usage returned includes usage with dates from **filterStartDate 00:00:00 UTC**. *filterStartDate* must be older or equal to *filterEndDate*.

filterEndDate - *string* - *required*
: End date of the bandwidth usage report. Should be in **YYYY-MM-DD**(ISO) format. The bandwidth usage returned includes all dates to **filterEndDate 23:59:59 UTC**. *filterEndDate* must be older than today's date in UTC.

showBlanks - *optional*
: When the showBlanks parameter is passed, the response body will include values for all years and types, even if there is no associated traffic.

92.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

92.2.4 Request Body

None

92.3 Response

92.3.1 Status Code

200 OK
: Success.

400 Bad Request
: A request is considered *bad* any of the following situation is available:
  - **filterType** is not one of **www**, **ftp**;
  - **filterStartDate** or **filterEndDate** URI parameter is missing;
  - **filterEndDate** is older than **filterStartDate**;
  - either **filterEndDate** or **filterStartDate** are today's date or a future date;
  - **filterEndDate** is more than 100 years after **filterStartDate**

403 Forbidden
: Forbidden access to the account.

404 Not Found
: The account Id or product Id was not found.

92.3.2 Response Body

```
{
 "list" : [
 {
      "year": {yearNumber},
      "type": {trafficType},
      "domain": {domainName},
      "bytes": {numberOfBytes}
  } ,...,
  {
      "year": {yearNumber}
      "type": {trafficType},
      "domain": {domainName},
      "bytes": {numberOfBytes}
  } ],
  "links" : [ {
    "href" : "{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/yearly?from=2012-01-0
    "rel" : "self"
  } ]
}
```

92.3.2.1 Parameters

list - *list*

a list of all bandwidth usage records. Each record consists of:

yearNumber - *integer*

the year for which the bandwidth is collected

trafficType - *string*

type of the traffic(www or ftp)

domainName - *string*

the domain name, for which the bandwidth data is collected

numberOfBytes - *integer*

the bandiwdth calculated in bytes

links - *list*

Hypermedia for this resource. Links to self.

92.4 Examples

92.4.1 GET call

Request

GET
{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/yearly?filterType=www&from=2012-01-

Response

```
HTTP/1.1 200 OK
```

```
{
  "list" : [ {
      "year": 2012,
      "type": "www",
      "domain": "smithwidgets.com",
      "bytes": 12345678
  }, {
      "year": 2013,
      "type": "www",
      "domain": "smithwidgets.com",
      "bytes": 12345678
  }
],
  "links" : [ {
    "href" : "{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/bandwidth/yearly?filterType=www
    "rel" : "self"
  } ]
}
```

92.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

93 Accounts-accountId-products-productId-usage:GET

94 GET /accounts/*{accountId}*/products/*{productId}*/usage?from=*{fromDate}*&to=*{toDate}*&rate

Get an aggregated services usage report of a product

Products that currently generate usage data:

FlexCloud Server
Exchange 2010 Mailbox Archiving
Exchange 2010 E-mail Encryption

94.1 Contents

- 1 GET /accounts/{accountId}/products/{productId}/usage?from={fromDate}&to={toDate}&rate=true
  - 1.1 Request
    - 1.1.1 Request Parameters
    - 1.1.2 URI Parameters
    - 1.1.3 Request Headers
    - 1.1.4 Request Body
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
      - 1.2.2.2 Usage Types
  - 1.3 Examples
    - 1.3.1 GET call
  - 1.4 See also

94.2 Request

`GET /accounts/{accountId}/products/{productId}/usage?from={fromDate}&to={toDate}&rate=true`

94.2.1 Request Parameters

accountId - *string*
The Id of the account.
productId - *string*
The Id of the product.

94.2.2 URI Parameters

fromDate - *string* - *required*
Start date of the usage report. Should be in **YYYY-MM-DD** format. The usage returned includes usage with dates from **fromDate 00:00:00 UTC**. *fromDate* must be older or equal to *toDate*.

toDate - *string* - *required*
Start date of the usage report. Should be in **YYYY-MM-DD** format. The usage returned includes usage with dates to **toDate 23:59:59 UTC**. *toDate* must be older than today's date in UTC.

rate - *string* - *optional*
If set to *true*, the response will include the price for each usage.

94.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
The Authentication credentials of the client application.

94.2.4 Request Body

None

94.3 Response

94.3.1 Status Code

200 OK
Success.

400 Bad Request
A request is considered *bad* any of the following situation is available:
- **fromDate** or **toDate** URI parameter is missing;
- **toDate** is older than **fromDate**;
- today is older than or equal to **toDate** or **fromDate**;

403 Forbidden
Forbidden access to regular accounts. Only Sales Agents have access to an *usage* resource.

404 Not Found
The account Id or product Id was not found.

94.3.2 Response Body

```
{
  "list": [
```

```
    {
      "instance": {instanceName},
      "amount" : {amount},
      "type": {usageType},
      "unit": {usageUnit},
      "ratedAmount": 0.95
    } ,...,
    {
      "instance": {instanceName},
      "amount": {amount},
      "type": {usageType},
      "unit": {usageUnit},
      "ratedAmount": 10.00
    }
  ],
  "links": [
    {
      "href": "{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/usage?from=2012-01-01&to=2012-
      "rel": "self"
    }
  ]
}
```

94.3.2.1 Parameters

list - *list*

a list of all services usage records. Each record consists of:

instanceName - *string*

name of a service instance

usageType - *string*

type of a resource used

amount - *integer*

used resouce amount

unit - *string*

measurement unit of the resource used

ratedAmount - *integer/float* - (optional)

the calculated price (charge) for the usage

links - *list*

Hypermedia for this resource. Links to self.

94.3.2.2 Usage Types

Usage types under monitoring are the following:

Type	Unit	Description
ipaddress	hour	Flexible IP address uptime
loadbalancer	hour	Load Balancer uptime
vm	hour	Virtual Machine uptime
vm.app.office.2007.pro	hour	Virtual Machine uptime for Office 2007 Professional
vm.app.office.2007.std	hour	Virtual Machine uptime for Office 2007 Standard
vm.app.plesk	hour	Virtual Machine uptime for Plesk Control Panel
vm.app.plesk.domain100	hour	Virtual Machine uptime for 100 Plesk domains
vm.app.plesk.powerpack	hour	Virtual Machine uptime for Plesk Powerpack
vm.app.plesk.powerpack	hour	Virtual Machine uptime for Plesk Powerpack
vm.app.sqlserver.std	hour	Virtual Machine uptime for SQL Server Standard Edition
vm.app.sqlserver.web	hour	Virtual Machine uptime for SQL Server Web Edition
vm.app.terminalservices.cal	hour	Virtual Machine uptime for 1 TS CAL
vm.cpu	hour	Virtual Machine CPU uptime
vm.disk.10gb	hour	Virtual Machine uptime for 10GB Disk Space
vm.memory.1gb	hour	Virtual Machine uptime for 1GB of Memory
vm.monitoring.port	hour	Virtual Machine uptime for Port Monitoring
vm.monitoring.url	hour	Virtual Machine uptime for URL Monitoring
vm.os.redhat	hour	Virtual Machine uptime for RedHat OS
vm.traffic.in	mb	Virtual Machine Incoming Traffic
vm.traffic.out	mb	Virtual Machine Outgoing Traffic
vm.traffic.total	mb	Virtual Machine Total Traffic
volume.disk.10gb	hour	Storage Volume uptime for 10GB of Space
vpn	hour	VPN uptime
vpn.remote5	hour	VPN uptime for 5 remote points
vpn.sitetosite	hour	VPN uptime for site-to-site access
hex2013.additionalquota *	mb	Additional quota for HEX 2013

Note: The additional quota unlike the other usage types is not directly retrieved, but rather it is calculated with the following formula:

```
CoSM = Count of standard mailboxes
QfSM = Quota for standard mailboxes
CoPM = Count of premium mailboxes
QfPM = Quota for premium mailboxes

Additional Quota = Quota - CoSM * QfSM -  CoPM * QfPM
```

94.4 Examples

94.4.1 GET call

Request

GET
{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/usage?from=2012-01-01&to=2012-12-31&rate=tru

Response

```
HTTP/1.1 200 OK
```


```
{
  "list" : [ {
    "instance" : "smith182278-Test-VM.1",
    "amount" : 24,
    "type" : "vm.disk.10gb",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM.0",
    "amount" : 24,
    "type" : "vm.disk.10gb",
    "unit" : "hour",
    "ratedAmount" : 0.95
  }, {
    "instance" : "smith182278-Test-VM.3",
    "amount" : 24,
    "type" : "vm.disk.10gb",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM.2",
    "amount" : 24,
    "type" : "vm.disk.10gb",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM.4",
    "amount" : 24,
    "type" : "vm.disk.10gb",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM.1",
    "amount" : 24,
    "type" : "vm.memory.1gb",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM.0",
    "amount" : 24,
    "type" : "vm.memory.1gb",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM.3",
    "amount" : 24,
    "type" : "vm.memory.1gb",
    "unit" : "hour",
    "ratedAmount" : 10.00
  }, {
    "instance" : "smith182278-Test-VM.2",
    "amount" : 24,
    "type" : "vm.memory.1gb",
    "unit" : "hour"
  }, {
    "instance" : "smith182278-Test-VM.5",
    "amount" : 24,
    "type" : "vm.memory.1gb",
    "unit" : "hour",
    "ratedAmount" : 11.05
  }, {
    "instance" : "smith182278-Test-VM.4",
    "amount" : 24,
    "type" : "vm.memory.1gb",
    "unit" : "hour",
    "ratedAmount" : 10.0
  }, {
    "instance" : "smith182278-Test-VM.7",
    "amount" : 24,
    "type" : "vm.memory.1gb",
    "unit" : "hour",
    "ratedAmount" : 10
  }, {
    "instance" : "smith182278-Test-VM.6",
    "amount" : 24,
    "type" : "vm.memory.1gb",
    "unit" : "hour",
    "ratedAmount" : 0.5
  }, {
    "instance" : "",
    "amount" : 6488,
    "type" : "vm.traffic.total",
```

```
    "unit" : "mb",
    "ratedAmount" : 0.99
  }, {
    "instance" : "smith182278-Test-VM",
    "amount" : 354,
    "type" : "vm.traffic.in",
    "unit" : "mb",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM",
    "amount" : 24,
    "type" : "vm.app.plesk",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM.1",
    "amount" : 24,
    "type" : "vm.cpu",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM.0",
    "amount" : 24,
    "type" : "vm.cpu",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM.3",
    "amount" : 24,
    "type" : "vm.cpu",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM.2",
    "amount" : 24,
    "type" : "vm.cpu",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM",
    "amount" : 24,
    "type" : "vm",
    "unit" : "hour",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM",
    "amount" : 6134,
    "type" : "vm.traffic.out",
    "unit" : "mb",
    "ratedAmount" : 0
  }, {
    "instance" : "smith182278-Test-VM",
    "amount" : 1024,
    "type" : "hex2013.additionalquota",
    "unit" : "mb",
    "ratedAmount" : 0
  } ],
  "links" : [ {
    "href" : "{{APIBaseURL}}/accounts/smith182278/products/smith182278-0000013534cbbc0dc7182cea000aa37f798db0/usage?from=2012-01-01&to=2012-1
    "rel" : "self"
  } ]
}
```


94.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

95 Accounts-accountId-paymentInfo:GET

96 GET /accounts/*{accountId}*/paymentInfo

Retrieves a list of all active payment methods.

96.1 Contents

- 1 GET /accounts/{accountId}/paymentInfo
  - 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 See also

96.2 Request

`GET /accounts/{accountId}/paymentInfo`

96.2.1 Parameters

accountId - *string*
: The Id of the account.

96.2.2 URI Parameters

None.

96.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

96.3 Response

96.3.1 Status Code

200 OK
: Success

403 Forbidden
: The user lacks the necessary permissions to access the resource

404 Not Found
: The account Id was not found.

96.3.2 Response Body


```
{
  "list": [{
    "paymentInfoType": "direct_debit",
    "routingNumber": "564738291",
    "links": [{"href": "https://api.hostway.com/accounts/{accountId}/paymentInfo/dd-100062441", "rel": "self"}],
    "accountNumber": "XXXXXXXXXXXX2332",
    "default": 0
  }, {
    "links": [{"href": "https://api.hostway.com/accounts/{accountId}/paymentInfo/cc-100069155", "rel": "self"}],
    "paymentInfoType": "credit_card",
    "creditCardType": "visa",
    "accountNumber": "XXXXXXXXXXXX1111",
    "expirationDate": "2017-04",
    "default": 1
  }],
  "default": [{"href": "https://api.hostway.com/accounts/{accountId}/paymentInfo/default/", "rel": "related"}],
  "links": [
    {"href": "https://api.hostway.com/accounts/{accountId}/paymentInfo/", "rel": "self"}
  ]
}
```


96.3.2.1 Parameters

list - *list*
: A collection of dictionaries containing information about specific products

  paymentInfoType- *string*
  : Describes whether the type of the payment method. Valid values are **direct_debit** or **credit_card**

  default - *integer*
  : Whether the payment method is the default. Allowed values are **0** and **1**

accountNumber - *integer*
: Available for direct debit and credit card. Contains the masked account number or the masked number of the credit card associated with the payment method

routingNumber - *string*
: *(OPTIONAL)*
Only available for direct debit. Contains the routing number of the payment method

creditCardType - *string*
: *(OPTIONAL)*
Only available for credit card. Describes the type of the card

expirationDate - *string*
: *(OPTIONAL)*
Only available for credit card. Contains the expiration date of the card

links - *list*
: Hypermedia for the individual payment method

links - *list*
: Hypermedia for this resource. Links to self

96.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

97 Accounts-accountId-paymentInfo:POST

98 POST /accounts/{accountId}/paymentInfo

Create a new payment information for a particular account

98.1 Contents

- 1 POST /accounts/{accountId}/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 Headers
    - 1.2.3 Response Body
  - 1.3 Examples
    - 1.3.1 Add an credit card payment info
    - 1.3.2 Add a direct debit payment info
  - 1.4 See also

98.2 Request

```
POST /accounts/{accountId}/paymentInfo
```

98.2.1 Request Parameters

accountId
: The ID of the account

98.2.2 URI Parameters

None

98.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

98.2.4 Request Body

For adding a credit card

```
{
 "paymentInfoType": {paymentInfoType},
 "accountNumber": {accountNumber},
 "cardHolderName": {cardHolderName},
 "expirationDate": {expirationDate},
 "verificationNumber": {verificationNumber},
 "billingAddress": {
    "street1": {street1},
    "street2": {street2},
    "city": {city},
    "stateOrProvince": {stateOrProvince},
    "postalCode": {postalCode},
    "countryCode": {countryCode}
 },
 "routingNumber": {routingNumber},
 "accountOwner": {accountOwner},
 "accountName": {accountName},
 "bankName": {bankName},
 "bankCity": {bankCity},
 "bankCountry": {bankCountry}
}
```

98.2.4.1 Parameters

paymentInfoType - *string*
: *Required.* Type of the payment method. Allowed values are **credit_card** and **direct_debit**.

accountNumber - *string*
Required. Payment account number. This the credit card number or the bank account number.

cardHolderName - *string*
Required.(For credit cards only) Name of the card holder. Generally, it is the name that appears on the CC.

expirationDate - *string*
Required.(For credit cards only) Expiration date of the credit card. Should be in the **yyyy-m** format.

verificationNumber - *string*
Required.(For credit cards only) The CVV number (Card Verification Value) of the credit card.

billingAddress - *dictionary*
Required.(For credit cards only) The billing address of the credit card. It's important that the address provided matches the one with which the credit card was registered as the card will be validated through an AVS (Address Verification Service)
- street1 - *string*
  Required. The address line 1 of the billing address.
- street2 - *string*
  Optional. The address line 2 of the billing address.
- city - *string*
  Required. The city of the billing address.
- stateOrProvince - *string*
  Optional. The state Or province of the billing address.
- postalCode - *string*
  Optional. The postal code of the billing address.
- countryCode - *string*
  Optional. The country code of the billing address.

routingNumber - *string*
Required.(For direct debit only). The bank routing number.

accountOwner - *string*
Optional.(For direct debit only). Name as it Appears on the Check.

accountName - *string*
Optional.(For direct debit only). Name for the account.

bankName - *string*
Optional.(For direct debit only). The bank name.

bankCity - *string*
Optional.(For direct debit only). The bank city.

bankCountry - *string*
Optional.(For direct debit only). The bank country.

98.3 Response

98.3.1 Status Code

201 Created
The payment information was successfully created

400 Bad Request
The request body was missing a required field or one of the fields did not comply with the requirements (in this case, the response body will contain hints for what went wrong)

401 Unauthorized
Authentication failed

403 Forbidden
Authentication succeeded, but the logged-in user is not allowed to create users in this account. The only users allowed to create users in this account are sales agents and account owners of *this* account. An account owner is any account user with the "sitecontrol_account_owner" role.

404 Not Found
The account does not exist

409 Conflict
The payment information cannot be created because another payment account with the same information already exists for this user. **OR**
The credit card information has been rejected. Some reasons for CC validation failures are:
- Invalid or expired credit card;
- Invalid CVV number;
- Invalid credit card billing address;
- Black-listed credit card

98.3.2 Response Headers

Location
The URI for the newly created payment information

98.3.3 Response Body

None

98.4 Examples

98.4.1 Add an credit card payment info

```
POST http://<serverName>/accounts/<accountId>/paymentInfo

{
 "paymentInfoType": "credit_card",
 "accountNumber": "1111111111111111",
 "cardHolderName": "John Smith",
 "verificationNumber": "555",
 "creditCardType": "amex",
 "expirationDate": "2015-06",
 "billingAddress": {
    "street1": "Address Line 1",
    "street2": "Address Line 2",
    "countryCode": "US",
    "city": "Chicago",
    "postalCode": "60606",
    "stateOrProvince": "IL"
 }
}
```

Response headers

```
...
Location    http://<serverName>/accounts/<accountId>/paymentInfo/cc-<newly_created_id>
...
```

98.4.2 Add a direct debit payment info

```
POST http://<serverName>/accounts/<accountId>/paymentInfo

{
 "paymentInfoType": "direct_debit",
 "routingNumber": "123456789",
 "accountNumber": "1234",
 "accountName": "My Account Name",
 "accountOwner": "John Smith",
 "bankName": "My Bank",
 "bankCity": "Chicago",
 "bankCountry": "US"
}
```

Response headers

```
...
Location    http://<serverName>/accounts/<accountId>/paymentInfo/dd-<newly_created_id>
...
```

98.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

99 Accounts-accountId-paymentInfo-default:PUT

100 PUT /accounts/*{accountId}*/paymentInfo/default

Sets the default payment method

100.1 Contents

- 1 PUT /accounts/{accountId}/paymentInfo/default
  - 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
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 See also

100.2 Request

`PUT /accounts/{accountId}/paymentInfo/default`

100.2.1 Parameters

accountId - *string*
: The Id of the account.

100.2.2 URI Parameters

None.

100.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

100.2.4 Request Body

```
{
  "paymentInfo": "https://api.hostway.com/accounts/{accountId}/paymentInfo/{paymentId}"
}
```

100.2.4.1 Parameters

paymentInfo - *string*
: The new defalut payment method. Should contain a link to the payment info that will become the new default.

100.3 Response

100.3.1 Status Code

204 No Content
: Successful default payment change

400 Bad Request
: Request body was malformed, or an invalid link was submitted for the new default payment.

404 Not Found
: The account Id was not found.

100.3.2 Response Body

None.

100.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

101 Accounts-accountId-paymentInfo-default:DELETE

102 DELETE /accounts/*{accountId}*/paymentInfo/default

Unsets the default payment method. The default payment method is automatically set to "offline" payment. This method is applicable for those franchises where the "offline" payment is allowed.

102.1 Contents

- 1 DELETE /accounts/{accountId}/paymentInfo/default
  - 1.1 Request
    - 1.1.1 Parameters
    - 1.1.2 URI Parameters
    - 1.1.3 Request Headers
    - 1.1.4 Request Body
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 See also

102.2 Request

`DELETE /accounts/{accountId}/paymentInfo/default`

102.2.1 Parameters

accountId - *string*
: The Id of the account.

102.2.2 URI Parameters

None.

102.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

102.2.4 Request Body

None.

102.3 Response

102.3.1 Status Code

204 No Content
: Successful default payment change

404 Not Found
: The account Id was not found.

409 Conflict
: The "offline" payment method is not allowed

102.3.2 Response Body

None.

102.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

103 Accounts-accountId-paymentInfo-paymentInfoId:GET

104 GET /accounts/*{accountId}*/paymentInfo/*{paymentInfoId}*

Retrieves an individual active payment method.

104.1 Contents

- 1 GET /accounts/{accountId}/paymentInfo/{paymentInfoId}
  - 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 See also

104.2 Request

`GET /accounts/{accountId}/paymentInfo/{paymentInfoId}`

104.2.1 Parameters

accountId - *string*
: The Id of the account.

paymentInfoId - *string*
: The Id of the payment method.

104.2.2 URI Parameters

None.

104.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

104.3 Response

104.3.1 Status Code

200 OK
: Success

403 Forbidden
: The user lacks the necessary permissions to access the resource

404 Not Found
: The account or payment info was not found

104.3.2 Response Body

```
{
    "paymentInfoType": "direct_debit",
    "routingNumber": "564738291",
    "links": [{"href": "https://api.hostway.com/accounts/{accountId}/paymentInfo/dd-100062441", "rel": "self"}],
    "accountNumber": "XXXXXXXXXXXX2332",
    "default": 0
}
```

```
{
    "links": [{"href": "https://api.hostway.com/accounts/{accountId}/paymentInfo/cc-100069155", "rel": "self"}],
    "paymentInfoType": "credit_card",
    "creditCardType": "visa",
    "accountNumber": "XXXXXXXXXXXX1111",
    "expirationDate": "2017-04",
    "default": 1
}
```

104.3.2.1 Parameters

paymentType - *string*
: Describes whether the type of the payment method. Valid values are **direct_debit** or **credit_card**

default - *integer*
: Whether the payment method is the default. Allowed values are **0** and **1**

accountNumber - *integer*
: Available for direct debit and credit card. Contains the masked account number or the masked number of the credit card associated with the payment method

routingNumber - *string*
: *(OPTIONAL)*
Only available for direct debit. Contains the routing number of the payment method

creditCardType - *string*
(OPTIONAL)
Only available for credit card. Describes the type of the card

expirationDate - *string*
(OPTIONAL)
Only available for credit card. Contains the expiration date of the card

links - *list*
Hypermedia for this resource. Links to self

104.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

105 Accounts-accountId-paymentInfo-paymentInfoId:DELETE

106 DELETE /accounts/*{accountId}*/paymentInfo/*{paymentInfoId}*

Removes a given payment method. If the payment method is the current default, it cannot be removed

106.1 Contents

- 1 DELETE /accounts/{accountId}/paymentInfo/{paymentInfoId}
  - 1.1 Request
    - 1.1.1 Parameters
    - 1.1.2 URI Parameters
    - 1.1.3 Request Headers
    - 1.1.4 Request Body
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
  - 1.3 See also

106.2 Request

`DELETE /accounts/{accountId}/paymentInfo/{paymentInfoId}`

106.2.1 Parameters

accountId - *string*
: The Id of the account.

paymentInfoId - *string*
: The Id of the payment method.
Note: Default payment info can be unset only with agent credentials.

106.2.2 URI Parameters

None.

106.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

106.2.4 Request Body

None.

106.3 Response

106.3.1 Status Code

204 No Content
: Successful removal

403 Forbidden
: The current user is not allowed to delete this payment method

404 Not Found
: The account or payment info was not found

409 Conflict
: Cannot delete the current default payment method

106.3.2 Response Body

None.

106.4 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

107 Accounts-accountId-paymentInfo-paymentInfold-payment:POST

108 POST /accounts/{accountId}/paymentInfo/{paymentInfoId}/payment

Make a one-time payment toward the account balance. First, a "pending" payment transaction will be created, and the payment amount will be reflected in the effective balance but not in the actual balance. After the payment has been processed (asynchronously), the payment status will change to "active" and the payment amount will be reflected in the actual balance as well.

108.1 Contents

- 1 POST /accounts/{accountId}/paymentInfo/{paymentInfoId}/payment
  - 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
  - 1.3 Examples
    - 1.3.1 Make a payment
  - 1.4 See also

108.2 Request

```
POST /accounts/{accountId}/paymentInfo/{paymentInfoId}/payment
```

108.2.1 Request Parameters

accountId
: The ID of the account

paymentInfoId
: The ID of the payment method

108.2.2 URI Parameters

None

108.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

108.2.4 Request Body

```
{
 "amount": {paymentAmount}
}
```

108.2.4.1 Parameters

paymentAmount - *decimal*
: The amount of payment in the account's currency. Must be greater than zero.

108.3 Response

108.3.1 Status Code

204 No Content
: The payment was successfully submitted

400 Bad Request
: The request body was missing a required field or one of the fields did not comply with the requirements (in this case, the response body will contain hints for what went wrong)

401 Unauthorized
: Authentication failed

403 Forbidden
: Authentication succeeded, but the logged-in user is not allowed to create users in this account. The only users allowed to create users in this account are sales agents and account owners of *this* account. An account owner is any account user with the "sitecontrol_account_owner" role.

404 Not Found
: The account or the payment method does not exist

409 Conflict
: The payment cannot be submitted due to irregularities with the payment method. For example, this may happen if the credit card is expired, has reached its credit limit or is declined by the payment provider.

108.3.2 Response Headers

None

108.3.3 Response Body

None

108.4 Examples

108.4.1 Make a payment

```
POST https://api.hostway.com/accounts/smith123456/paymentInfo/cc-100075384/payment
```

```
{
  "amount": 100.00
}
```

108.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

109 Accounts-accountId-transactionsInfo:GET

110 GET /accounts/*{accountId}*/transactions

Retrieves a list of all account's transactions

110.1 Contents

- 1 GET /accounts/{accountId}/transactions
  - 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 Examples
    - 1.3.1 GET paginated transaction list call
  - 1.4 See also

110.2 Request

`GET /accounts/{accountId}/transactions`

110.2.1 Parameters

accountId - *string*
: The Id of the account.

110.2.2 URI Parameters

None.

110.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Accept - HTTP Accept header [2]
: Possible values are *application/json*, *text/csv*. If omitted json will be returned.

110.3 Response

110.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: The user lacks the necessary permissions to access the resource

404 Not Found
: The account Id was not found.

110.3.2 Response Body

```
{
  {
   "list": [{
      "startDate": "{startDate}",
      "endDate": "{endDate}",
      "amount": {amount},
      "description": "{description}",
      "currency": "{currency}",
      "date": "{date}",
      "productSpecification": "",
      "serviceNumber": "{serviceNumber}",
      "type": "{type}",
      "number": {number}
   }],
   "links" : [{
      "href" : "https://api.hostway.com/accounts/{account_id}/transactions",
      "rel" : "self"
  }]
}
```

110.3.2.1 Parameters

list - *list*
A collection of dictionaries containing information about transactions

startDate - *string*
Start date of transaction

endDate - *string*
End date of transaction

ammount - *integer*
Amount of transaction

description - *string*
Description of transaction

currency - *string*
Currency symbol of transaction

date - *string*
Date of transaction

productSpecification - *link*
URL link to the product specification

serviceNumber - *string*
Service number info

type - *string*
Type of transaction. Possible values are: 'charge', 'charge_back', 'payment', 'fee', 'refund', 'credit', 'cc_chargeback', 'charge_adjustment', 'discount', 'credit_adjustment', 'credit_proration', 'charge_proration', 'payment_adjustment', 'tax', 'cc_chargeback_reversal', 'usage_charge', 'write_off', 'undo_write_off', 'usage_discount'

number - *int*
Transaction's number

links - *list*
Hypermedia for the individual payment method

links - *list*
Hypermedia for this resource. Links to self

110.4 Examples

110.4.1 GET paginated transaction list call

Request

GET {{APIBaseURL}}/accounts/test731993/transactions?page=1&pageSize=10

Response

```
HTTP/1.1 200 OK
```


```
{
  "list" : [ {
      "startDate": "2012-04-23",
      "amount": 0.87,
      "endDate": "2012-05-18",
      "description": "Recurring fee Exchange 2010 Archiving, core.email.exchange2010.archiving  04/23/2012 through 05/18/2012",
      "currency": "USD",
      "date": "2012-04-23",
      "productSpecification": "{{APIBaseURL}}/productSpecifications/test731993-00000137a32daee9e2b08715000a30fe33d32c",
      "serviceNumber": "test731993-00000136dedf26dfd9ec22fd000a1ddba01da1",
      "type": "charge",
      "number": 101393524
  }],
  "links" : [
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/transactions?page=1&pageSize=10",
      "rel" : "self"
    },
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/transactions?page=2&pageSize=10",
      "rel" : "next"
    },
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/transactions?page=44&pageSize=10",
      "rel" : "last"
  }]
}
```


110.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

111 Accounts-accountId-transactionsInfo-filtered-type:GET

112 GET /accounts/*{accountId}*/transactions?filterType={filterType}

Gets a list of account's transactions filtered by transaction type

112.1 Contents

- 1 GET /accounts/{accountId}/transactions?filterType={filterType}
  - 1.1 Request
    - 1.1.1 Parameters
    - 1.1.2 Request Headers
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 GET paginated transaction list call
  - 1.4 See also

112.2 Request

`GET /accounts/{accountId}/transactions?filterType={filterType}`

112.2.1 Parameters

accountId - *string*
: The Id of the account.

filterType - *string*
: Type of transaction. Possible values are: 'charge', 'charge_back', 'payment', 'fee', 'refund', 'credit', 'cc_chargeback', 'charge_adjustment', 'discount', 'credit_adjustment', 'credit_proration', 'charge_proration', 'payment_adjustment', 'tax', 'cc_chargeback_reversal', 'usage_charge', 'write_off', 'undo_write_off', 'usage_discount'

112.2.2 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

112.3 Response

112.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: The user lacks the necessary permissions to access the resource

404 Not Found
: The accountId was not found.

112.3.2 Response Body

```
{
  {
   "list": [{
      "startDate": "{startDate}",
      "endDate": "{endDate}",
      "amount": {amount},
      "description": "{description}",
      "currency": "{currency}",
      "date": "{date}",
      "productSpecification": "",
      "serviceNumber": "{serviceNumber}",
      "type": "{type}",
      "number": {number}
   }],
   "links" : [{
      "href" : "https://api.hostway.com/accounts/{account_id}/transactions",
      "rel" : "self"
  }]
}
```

112.3.2.1 Parameters

list - *list*
: A collection of dictionaries containing information about transactions

  startDate - *string*
  : Start date of transaction

  endDate - *string*
  : End date of transaction

  ammount - *integer*
  : Amount of transaction

description - *string*
Description of transaction

currency - *string*
Currency symbol of transaction

date - *string*
Date of transaction

productSpecification - *link*
URL link to the product specification

serviceNumber - *string*
Service number info

type - *string*
Type of transaction. Possible values are: 'charge', 'charge_back', 'payment', 'fee', 'refund', 'credit', 'cc_chargeback', 'charge_adjustment', 'discount', 'credit_adjustment', 'credit_proration', 'charge_proration', 'payment_adjustment', 'tax', 'cc_chargeback_reversal', 'usage_charge', 'write_off', 'undo_write_off', 'usage_discount'

number - *int*
Transaction's number

links - *list*
Hypermedia for the individual payment method

links - *list*
Hypermedia for this resource. Links to self

112.4 Examples

112.4.1 GET paginated transaction list call

Request

GET {{APIBaseURL}}/accounts/test731993/transactions?filterType=payment&page=1&pageSize=10

Response

```
HTTP/1.1 200 OK
```


```
{
  "list" : [ {
      "startDate": "2012-04-23",
      "amount": 0.87,
      "endDate": "2012-05-18",
      "description": "Recurring fee Exchange 2010 Archiving, core.email.exchange2010.archiving  04/23/2012 through 05/18/2012",
      "currency": "USD",
      "date": "2012-04-23",
      "productSpecification": "{{APIBaseURL}}/productSpecifications/test731993-00000137a32daee9e2b08715000a30fe33d32c",
      "serviceNumber": "test731993-00000136dedf26dfd9ec22fd000a1ddba01da1",
      "type": "payment",
      "number": 101393524
  }],
  "links" : [
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/transactions?page=1&pageSize=10",
      "rel" : "self"
    },
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/transactions?page=2&pageSize=10",
      "rel" : "next"
    },
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/transactions?page=44&pageSize=10",
      "rel" : "last"
  }]
}
```


112.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

113 Accounts-accountId-transactionsInfo-sorted:GET

114 GET /accounts/*{accountId}*/transactions?sortField={sortField}&sortOrder={sortOrder}

Gets a list of account's transactions sorted by sortField param

114.1 Contents

- 1 GET /accounts/{accountId}/transactions?sortField={sortField}&sortOrder={sortOrder}
  - 1.1 Request
    - 1.1.1 Parameters
    - 1.1.2 Request Headers
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 GET sorted by amount value transaction list call
  - 1.4 See also

114.2 Request

`GET /accounts/{accountId}/transactions?sortField={sortField}&sortOrder={sortOrder}`

114.2.1 Parameters

accountId - *string*
: The Id of the account.

sortField - *string*
: sortField param. Possible values are: 'amount','type'.

sortOrder - *string*
: sortOrder param. Possible values are: 'asc','desc'.

114.2.2 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

114.3 Response

114.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: The user lacks the necessary permissions to access the resource

404 Not Found
: The accountId was not found.

114.3.2 Response Body

```
{
  {
   "list": [{
      "startDate": "{startDate}",
      "endDate": "{endDate}",
      "amount": {amount},
      "description": "{description}",
      "currency": "{currency}",
      "date": "{date}",
      "productSpecification": "",
      "serviceNumber": "{serviceNumber}",
      "type": "{type}",
      "number": {number}
   }],
   "links" : [{
      "href" : "https://api.hostway.com/accounts/{account_id}/transactions",
      "rel" : "self"
  }]
}
```

114.3.2.1 Parameters

list - *list*
: A collection of dictionaries containing information about transactions

  startDate - *string*
  : Start date of transaction

  endDate - *string*
  : End date of transaction

ammount - *integer*
Amount of transaction

description - *string*
Description of transaction

currency - *string*
Currency symbol of transaction

date - *string*
Date of transaction

productSpecification - *link*
URL link to the product specification

serviceNumber - *string*
Service number info

type - *string*
Type of transaction. Possible values are: 'charge', 'charge_back', 'payment', 'fee', 'refund', 'credit', 'cc_chargeback', 'charge_adjustment', 'discount', 'credit_adjustment', 'credit_proration', 'charge_proration', 'payment_adjustment', 'tax', 'cc_chargeback_reversal', 'usage_charge', 'write_off', 'undo_write_off', 'usage_discount'

number - *int*
Transaction's number

links - *list*
Hypermedia for the individual payment method

links - *list*
Hypermedia for this resource. Links to self

114.4 Examples

114.4.1 GET sorted by amount value transaction list call

Request

GET {{APIBaseURL}}/accounts/test731993/transactions?sortField=amount&sortOrder=asc

Response

```
HTTP/1.1 200 OK
```


```
{
  "list" : [
    {
      "startDate": "2012-04-23",
      "amount": 0.87,
      "endDate": "2012-05-18",
      "description": "Recurring fee Exchange 2010 Archiving, core.email.exchange2010.archiving  04/23/2012 through 05/18/2012",
      "currency": "USD",
      "date": "2012-04-23",
      "productSpecification": "{{APIBaseURL}}/productSpecifications/test731993-00000137a32daee9e2b08715000a30fe33d32c",
      "serviceNumber": "test731993-00000136dedf26dfd9ec22fd000a1ddba01da1",
      "type": "payment",
      "number": 101393524
    },
    {
      "startDate": "2013-05-04",
      "amount": 1.89,
      "endDate": "2013-06-18",
      "description": "Payment  - Notes: - Reference: VUGTI5F6NL0IRJSA17OR CC# 1111 (ProcessSOPayment)",
      "currency": "USD",
      "date": "2013-05-04",
      "productSpecification": "{{APIBaseURL}}/productSpecifications/test731993-00000737a32daee9e2b08715000a30fe13d32f",
      "serviceNumber": "test731993-00000136dedf26dfd9ec22fd000a1ddba02da2",
      "type": "payment",
      "number": 101395701
  }],
  "links" : [
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/transactions?sortField=amount&sortOrder=asc,
      "rel" : "self"
  }]
}
```


114.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

115 Accounts-accountId-invoicesInfo:GET

116 GET /accounts/*{accountId}*/invoices

Retrieves a list of all account's invoices

116.1 Contents

- 1 GET /accounts/{accountId}/invoices
  - 1.1 Request
    - 1.1.1 Parameters
    - 1.1.2 Request Headers
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 GET paginated invoices list call
  - 1.4 See also

116.2 Request

`GET /accounts/{accountId}/invoices`

116.2.1 Parameters

accountId - *string*
: The Id of the account.

116.2.2 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

116.3 Response

116.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: The user lacks the necessary permissions to access the resource

404 Not Found
: The account Id was not found.

116.3.2 Response Body

```
{
  {
   "list": [{
      "status": "{status}",
      "currency": "{currency}",
      "amount": "{amount}",
      "amountDue": "{amountDue}",
      "description": "{description}",
      "date": "{date}",
      "statusName": "{statusName}",
      "invoiceNumber": "{invoiceNumber}"
   }],
   "links" : [{
      "href" : "https://api.hostway.com/accounts/{account_id}/invoices",
      "rel" : "self"
  }]
}
```

116.3.2.1 Parameters

list - *list*
: A collection of dictionaries containing information about invoices

  status - *string*

Status of invoice

currency - *string*
Currency of invoice

ammount - *string*
Amount of invoice

amountDue - *string*
Amount due of invoice

description - *string*
Description of invoice

date - *string*
Date of invoice

statusName - *string*
Status of invoice

invoiceNumber - *string*
Invoice number

links - *list*
Hypermedia for the individual payment method

links - *list*
Hypermedia for this resource. Links to self

116.4 Examples

116.4.1 GET paginated invoices list call

Request

GET {{APIBaseURL}}/accounts/test731993/invoices?page=1&pageSize=3

Response

```
HTTP/1.1 200 OK
```

```
{
  "list" : [ {
      "status": "invoice_paid",
      "currency": "USD",
      "amount": "1023.7200",
      "amountDue": "0.0000",
      "description": "Invoice #SO-360704",
      "date": "2011-04-19",
      "statusName": "Invoice Paid",
      "invoiceNumber": "SO-360704"
  }],
  "links" : [
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/invoices?page=1&pageSize=3",
      "rel" : "self"
    },
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/invoices?page=2&pageSize=3",
      "rel" : "next"
    },
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/invoices?page=4&pageSize=3",
      "rel" : "last"
  }]
}
```

116.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

117 Accounts-accountId-invoicesInfo-filtered-status:GET

118 GET /accounts/*{accountId}*/invoice?filterStatus={filterStatus}

Gets a list of account's invoices filtered by status

118.1 Contents

- 1 GET /accounts/{accountId}/invoice?filterStatus={filterStatus}
  - 1.1 Request
    - 1.1.1 Parameters
    - 1.1.2 Request Headers
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 GET paginated invoices list call
  - 1.4 See also

118.2 Request

`GET /accounts/{accountId}/invoices?filterStatus={filterStatus}`

118.2.1 Parameters

accountId - *string*
: The Id of the account.

filterStatus - *string*
: Status of invoice. Possible values are: 'invoice_paid', 'invoice_unpaid', 'invoice_open'.

118.2.2 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

118.3 Response

118.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: The user lacks the necessary permissions to access the resource

404 Not Found
: The accountId was not found.

118.3.2 Response Body

```
{
  {
   "list": [{
      "status": "{status}",
      "currency": "{currency}",
      "amount": "{amount}",
      "amountDue": "{amountDue}",
      "description": "{description}",
      "date": "{date}",
      "statusName": "{statusName}",
      "invoiceNumber": "{invoiceNumber}"
   }],
   "links" : [{
      "href" : "https://api.hostway.com/accounts/{account_id}/invoices",
      "rel" : "self"
  }]
}
```

118.3.2.1 Parameters

list - *list*
: A collection of dictionaries containing information about invoices

  status - *string*
  : Status of invoice

  currency - *string*
  : Currency of invoice

  ammount - *string*
  : Amount of invoice

  amountDue - *string*
  : Amount due of invoice

description - *string*
Description of invoice

date - *string*
Date of invoice

statusName - *string*
Status of invoice

invoiceNumber - *string*
Invoice number

links - *list*
Hypermedia for this resource. Links to self

118.4 Examples

118.4.1 GET paginated invoices list call

Request

GET {{APIBaseURL}}/accounts/test731993/invoices?filterStatus=invoice_paid&page=1&pageSize=5

Response

```
HTTP/1.1 200 OK
```


```
{
  "list" : [ {
      "status": "invoice_paid",
      "currency": "USD",
      "amount": "1023.7200",
      "amountDue": "0.0000",
      "description": "Invoice #SO-360704",
      "date": "2011-04-19",
      "statusName": "Invoice Paid",
      "invoiceNumber": "SO-360704"
  }],
  "links" : [
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/invoices?page=1&pageSize=5",
      "rel" : "self"
    },
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/invoices?page=2&pageSize=5",
      "rel" : "next"
    },
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/invoices?page=44&pageSize=5",
      "rel" : "last"
  }]
}
```


118.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

119 Accounts-accountId-invoicesInfo-sorted:GET

120 GET /accounts/*{accountId}*/invoice?sortField={sortField}

Gets a list of account's invoices sorted by sortField

120.1 Contents

- 1 GET /accounts/{accountId}/invoice?sortField={sortField}
  - 1.1 Request
    - 1.1.1 Parameters
    - 1.1.2 Request Headers
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 GET paginated invoices list call sorted by status
  - 1.4 See also

120.2 Request

`GET /accounts/{accountId}/invoices?sortField={sortField}`

120.2.1 Parameters

accountId - *string*
: The Id of the account.

sortField - *string*
: sortField param. Possible value is 'status'.

sortOrder - *string*
: sortOrder param. Possible values are: 'asc','desc'.

120.2.2 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

120.3 Response

120.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: The user lacks the necessary permissions to access the resource

404 Not Found
: The accountId was not found.

120.3.2 Response Body

```
{
  {
   "list": [{
      "status": "{status}",
      "currency": "{currency}",
      "amount": "{amount}",
      "amountDue": "{amountDue}",
      "description": "{description}",
      "date": "{date}",
      "statusName": "{statusName}",
      "invoiceNumber": "{invoiceNumber}"
   }],
   "links" : [{
      "href" : "https://api.hostway.com/accounts/{account_id}/invoices",
      "rel" : "self"
  }]
}
```

120.3.2.1 Parameters

list - *list*
: A collection of dictionaries containing information about invoices

  status - *string*
  : Status of invoice

  currency - *string*
  : Currency of invoice

ammount - *string*
: Amount of invoice

amountDue - *string*
: Amount due of invoice

description - *string*
: Description of invoice

date - *string*
: Date of invoice

statusName - *string*
: Status of invoice

invoiceNumber - *string*
: Invoice number

links - *list*
: Hypermedia for this resource. Links to self

120.4 Examples

120.4.1 GET paginated invoices list call sorted by status

Request

GET {{APIBaseURL}}/accounts/test731993/invoices?sortField=status&sortOrder=asc

Response

```
HTTP/1.1 200 OK
```


```
{
  "list" : [
    {
      "status": "invoice_paid",
      "currency": "USD",
      "amount": "1023.7200",
      "amountDue": "0.0000",
      "description": "Invoice #SO-360704",
      "date": "2011-04-19",
      "statusName": "Invoice Paid",
      "invoiceNumber": "SO-360704"
    },
    {
      "status": "invoice_unpaid",
      "currency": "USD",
      "amount": "8692.6500",
      "amountDue": "8262.8600",
      "description": "Invoice #444547",
      "date": "2013-09-23",
      "statusName": "Invoice Unpaid",
      "invoiceNumber": "444547"
  }],
  "links" : [
    {
      "href" : "{{APIBaseURL}}/accounts/test731993/invoices?sortField=status&sortOrder=asc",
      "rel" : "self"
  }]
}
```


120.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

121 Accounts-accountId-invoicesInfo-invoiceId:GET

122 GET /accounts/*{accountId}*/invoces/*{invoiceId}*

Retrieves an individual invoice.

122.1 Contents

- 1 GET /accounts/{accountId}/invoces/{invoiceId}
  - 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 Examples
    - 1.3.1 GET invoice
    - 1.3.2 GET invoice PDF
  - 1.4 See also

122.2 Request

`GET /accounts/{accountId}/invoces/{invoiceId}`

122.2.1 Parameters

accountId - *string*
: The Id of the account.

invoiceId - *string*
: The Id of the invoice.

122.2.2 URI Parameters

None.

122.2.3 Request Headers

Authorization - *HTTP Authorization header* [1]
: The Authentication credentials of the client application.

Accept - *HTTP Accept header. Possible values are application/json, application/pdf. If omitted json will be returned* [2]

122.3 Response

122.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

403 Forbidden
: The user lacks the necessary permissions to access the resource

404 Not Found
: The account or invoice was not found

122.3.2 Response Body

```
{
    "status": "{status}",
    "currency": "{currency}",
    "amount": "{amount}",
    "amountDue": "{amountDue}",
    "description": "{description}",
    "links": [{
        "href": "https://api.hostway.com/accounts/{accountId}/invoices/{invoiceId}/",
        "rel": "self"
    }],
    "date": "{date}",
    "statusName": "{statusName}",
    "invoiceNumber": "{invoiceNumber}"
}
```

122.3.2.1 Parameters

status - *string*
: Status of invoice

currency - *string*
: Currency of invoice

ammount - *string*
: Amount of invoice

amountDue - *string*
: Amount due of invoice

description - *string*
: Description of invoice

date - *string*
: Date of invoice

statusName - *string*
: Status of invoice

invoiceNumber - *string*
: Invoice number

links - *list*
: Hypermedia for this resource. Links to self

122.4 Examples

122.4.1 GET invoice

Request

GET {{APIBaseURL}}/accounts/test731993/invoices/123456

Response

```
HTTP/1.1 200 OK
```

```
{
    "status": "invoice_paid",
    "currency": "USD",
    "amount": "1113.7500",
    "amountDue": "0.0000",
    "description": "Invoice #123456",
    "links": [
        {
            "href": "{{APIBaseURL}}/accounts/test731993/invoices/123456/",
            "rel": "self"
        }
    ],
    "date": "2011-11-10",
    "statusName": "Invoice Paid",
    "invoiceNumber": "123456"
}
```

122.4.2 GET invoice PDF

Request

GET {{APIBaseURL}}/accounts/test731993/invoices/123456 Accept: application/pdf

Response

```
HTTP/1.1 200 OK
Content-Type : application/pdf
```

```
%PDF-1.4
....
%%EOF
```

122.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

123 Accounts-accountId-invoices-invoiceId-transactions:GET

124 GET /accounts/{accountId}/invoices/{invoiceId}/transactions

Retrieves a list of transactions for a specific account and and invoice

124.1 Contents

- 1 GET /accounts/{accountId}/invoices/{invoiceId}/transactions
  - 1.1 Request
    - 1.1.1 Request Parameters
    - 1.1.2 URI Parameters
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 Success scenario
  - 1.4 See also

124.2 Request

```
GET /accounts/{accountId}/invoices/{invoiceId}/transactions
```

124.2.1 Request Parameters

accountId - *string*
: The user account owning the invoice

invoiceId - *string*
: The specific invoice for which the call will retrieve the list

124.2.2 URI Parameters

filterFrom - *date*
: Only transactions having startDate after the value of the parameter will be returned

filterTo - *date*
: Only transactions having startDate before the value of the parameter will be returned

filterType - *string*
: Only transactions having type matching the value of the parameter will be returned. Possible values are: 'charge', 'credit', 'discount'

page - *integer*
: Specifies which page should be displayed. Requires a pageSize parameter to also be provided

pageSize - *integer*
: Specifies the number of entries to be displayed on a page. Requires a page parameter to also be provided

sortField - *string*
: Specifies if the resulting list is to be sorted by a given field. Allowed values are **amount**, **from**, **to**

sortOrder - *string*
: The order in which the sorting is to be done. Allowed values are **asc** and **desc**

allowRollup - *boolean*
: Allows roll up of any charge or credit transactions in an invoice if the product associated with the transaction supports roll up. Default is not to allow roll up.

124.3 Response

124.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

404 Not Found
: The account or invoice does not exist.

124.3.2 Response Body

```
{
    "list": [
        {
            "startDate": "{startDate}",
            "endDate": {endDate},
            "currency": {currency},
            "amount": "{amount}",
            "description": "{description}",
            "date": "{date}",
            "productSpecification": "{productSpecification}",
            "serviceNumber": "{serviceNumber}",
            "type": "{type}",
            "number": "{number}",
            "isRolledUp": "{isRolledUp}"
        }
    ],
```

```
    "links": [
        {
            "href": "{href}",
            "rel": "{rel}"
        }
    ]
}
```

124.3.2.1 Parameters

startDate - *string*
: Start date of transaction

endDate - *string*
: End date of transaction

currency - *string*
: Currency of transaction

amount - *string*
: Amount of transaction

description - *string*
: Description of transaction

date - *string*
: Date of transaction

productSpecification - *string*
: Product specification of transaction

serviceNumber - *string*
: Service number of transaction

type - *string*
: Type of transaction. Possible values are: 'charge', 'credit', 'discount'

number - *string*
: Number of transaction

isRolledUp - *integer*
: Indicates if transaction is the aggregate of several rolled up transactions for the invoice. Possible values 1 if rolled up, 0 if not

href - *string*
: Link to other resources relevant to the mailboxes lists

rel - *string*
: Type of relation to the resource for the provided link

124.4 Examples

124.4.1 Success scenario

Request

```
GET /accounts/test-account/invoices/111111/transactions/
```

Response

```
{
   "list": [
      {
            "startDate": "2011-06-19",
            "amount": 34.65,
            "endDate": "2011-07-18",
            "description": "Usage Fee test@test.com Rate Type = Exchange mailbox usage, Usage amount = 7",
            "currency": "USD", "date": "2011-11-10",
            "productSpecification": "{{APIBaseURL}}/productSpecifications/test-acount-000001311f19185dc1df5d10000a000a000800",
            "serviceNumber": "test-account-0000013128ec0f17330329ef000a000a000800",
            "type": "charge",
            "number": 101378490,
            "isRolledUp": 0
      },
      ...
   ],
   "links": [
      {"href": "{{APIBaseURL}}/accounts/test-account/invoices/111111/transactions?page=1&pageSize=100", "rel": "self"},
      {"href": "{{APIBaseURL}}/accounts/test-account/invoices/111111/transactions?page=2&pageSize=100", "rel": "next"},
      {"href": "{{APIBaseURL}}/accounts/test-account/invoices/111111/transactions?page=5&pageSize=100", "rel": "last"}
   ]
}
```

124.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

125 Accounts-accountId-discount:GET

126 GET /accounts/{accountId}/discount

Get account level discount

126.1 Contents

- 1 GET /accounts/{accountId}/discount
  - 1.1 Request
    - 1.1.1 Request Parameters
  - 1.2 Response
    - 1.2.1 Status Code
    - 1.2.2 Response Body
      - 1.2.2.1 Parameters
  - 1.3 Examples
    - 1.3.1 Success scenario
  - 1.4 See also

126.2 Request

```
GET /accounts/{accountId}/discount
```

126.2.1 Request Parameters

accountId - *string*
: The user account on which the discount is applied

126.3 Response

126.3.1 Status Code

200 OK
: Success

401 Unauthorized
: The supplied credentials are invalid or do not provide permissions for this operation.

404 Not Found
: The account or invoice does not exist.

126.3.2 Response Body

```
    {
       "discountPercentage": "{discountPercentage}",
       "links":
       [
           {
                "href": "{{APIBaseURL}}/accounts/{accountId}/discount",
                "rel": "self"
           }
       ]
    }
```

126.3.2.1 Parameters

discountPercentage - *integer*
: Discount percentage

accountId - *string*
: The user account on which the discount is applied

links - *string*
: Hypermedia for this resource. Links to self

126.4 Examples

126.4.1 Success scenario

Request

```
GET /accounts/test-account/discount
```

Response

```
{
    {
       "discountPercentage": 30,
```

```
        "links":
        [
            {
                "href": "http://{{APIBaseURL}}/accounts/test-account/discount/",
                "rel": "self"
            }
        ]
    }
}
```

126.5 See also

- Account Management API
- Common Features

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

127 Accounts-accountId-os-security-groups:GET

This operation lists existing firewall groups.

128 GET /accounts/*{accountId}*/os-security-groups

This operation lists existing firewall groups.

128.1 Request

128.1.1 Request Parameters

accountId - *string*
: The selected account Id.

128.2 Response

128.2.1 JSON

```
{
   "securityGroups" : [
      {
         "displayName" : "{firewall_group_display_name}",
         "description" : "{firewall_group_description}",
         "links" : [
            {
               "href" : "https://api.hostway.com/accounts/{accountId}/os-security-groups/{firewall_group_id}/",
               "rel" : "self"
            }
         ],
         "rules" : [
            {
               "fromPort" : {start_port},
               "toPort" : {end_port},
               "protocol" : "{ip_protocol}",
               "cidr": "0.0.0.0/0",
               "id" : "145"
            }
         ]
         "servers" : ["{server_id}"],
         "accountNumber" : "{user_account_number}",
         "Id" : "{firewall_group_id}"
      }
   ]
}
```

128.2.1.1 security_groups

displayName - *string*
: The display name of the firewall group.

description - *string*
: Text describing the firewall group.

links - *list*
: Hypermedia for this resource.

rules - *list*
: List describing all defined security rules for the group.

fault - *dictionary*
: Might hold a fault information for the firewall group.

servers - *list*
: List of server ids this firewall group is assigned to.

accountNumber - *string*
: Identifies an account that owns the firewall group.

id - *string*
: The id of this firewall group.*(Unique identifier for this firewall group.)

128.3 Response Codes

Success
: *HTTP/1.1 200 OK*

Failure
: *HTTP/1.1 401 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.
HTTP/1.1 404 Not Found - The resource does not exist.

128.4 Examples

Successfully list firewall groups

: **GET** https://api.hostway.com/accounts/{accountId}/os-security-groups

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Response Code:

```
HTTP/1.1 200 OK
```

Response Body:


```
{
   "securityGroup" : [
      {
         "displayName": "Test Firewall Group 1",
         "description": "Test firewall group 1 description.",
         "links": [
            {
               "href": "https://api.hostway.com/accounts/{accountId}/os-security-groups/group2012May11_OK30Zs7Ju/",
               "rel": "self"
            }
         ],
         "rules": [
            {
               'toPort': 1111,
               'fromPort': 1111,
               'protocol': 'TCP',
               'id': 145,
               'cidr': '0.0.0.0/0'
            },
            {
               'toPort': 80,
               'fromPort': 80,
               'protocol': 'TCP',
               'id': 196,
               'cidr': '127.0.0.1/24'
            }
         ],
         "id": "group2012May11_OK30Zs7Ju",
         "accountNumber": "test_account_120761",
         "servers": ["test-server-0003"]
      },
      {
         "displayName": "Test Firewall Group",
         "description": "Test firewall group description.",
         "links": [
            {
               "href": "https://api.hostway.com/accounts/{accountId}/os-security-groups/group2012Apr30_9J5qDwOIo/",
               "rel": "self"
            }
         ],
         "rules": [
            {
               'toPort': 3389,
               'fromPort': 3389,
               'protocol': 'tcp',
               'id': 116,
               'cidr': '0.0.0.0/0'
            },
            {
               'toPort': 8443,
               'fromPort': 8443,
               'protocol': 'tcp',
               'id': 117,
               'cidr': '0.0.0.0/0'
            },
            {
               'toPort': 22,
               'fromPort': 22,
               'protocol': 'tcp',
               'id': 118,
               'cidr': '0.0.0.0/0'
            }
         ],
         "id": "group2012Apr30_9J5qDwOIo",
         "accountNumber": "test_account_901265",
         "servers": ["test-server-0011", "test-server-0024"]
      }
   }
}
```


Failure while trying to make unauthorized access

POST https://api.hostway.com/accounts/{accountId}/os-security-groups

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded fake_authentication_string
```

Response Code:

```
HTTP/1.1 401 Unauthorized
```

Response Body:

```
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.
```

Failure while trying to pass invalid account number

POST https://api.hostway.com/accounts/fakeId/os-security-groups

Response Code:

```
HTTP/1.1 404 Not Found
```

Response Body:

```
{
   "itemNotFound" : {
      "message": "Resource not found",
      "code": 404,
      "details": "",
      "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
}
```

128.5 Alternative URI /accounts/os-security-groups

An alternative way to execute this call is to omit the *accountId* portion of the URI. In this case the system assumes the request is executed for the account identified in the authorization header.

```
GET /accounts/os-security-groups
```

129 Accounts-accountId-os-security-groups:POST

A Security/Firewall Group defines firewall rules for your Server. You have the ability to configure your firewall connections and secure your network as required. You may have one or multiple servers behind a firewall group. Hostway provides a default firewall group, and typical firewall rules are already configured. You can create multiple firewall groups as needed.

130 POST /accounts/*{accountId}*/os-security-groups

This operation will create a new firewall group.

130.1 Request

130.1.1 Parameters

accountId - *string*
: The selected account Id.

130.1.2 JSON

```
{
   "securityGroup" : {
      "displayName" : "{firewall_group_display_name}",
      "description" : "{firewall_group_description}"
   }
}
```

130.1.2.1 securityGroup

displayName - *string*
: Display name for this firewall group.

description - *string*
: Descriptive information for this firewall group.

130.2 Response

130.2.1 JSON

```
{
   "securityGroup" : {
      "displayName": "{firewall_group_display_name}",
      "description": "{firewall_group_description}",
      "links": [
         {
            "href": "https://api.hostway.com/accounts/{accountId}/os-security-groups/{firewall_group_id}/",
            "rel": "self"
         }
      ],
      "rules": [],
      "id": "{firewall_group_id}",
      "accountNumber": "{user_account_number}",
      "servers": []
   }
}
```

130.2.1.1 securityGroup

displayName - *string*
: Display name for this firewall group.

description - *string*
: Descriptive information for this firewall group.

links - *list*
: Hypermedia for this resource.

rules - *list*
: List of all defined security rules for the group.

id - *string*
: The id of the firewall group.*(Unique identifier for this firewall group.)

accountNumber - *string*
: The account number this firewall group resource belongs to.

servers - *list*
: List of server IDs this security group is assigned to.

130.3 Response Codes

Success
: *HTTP/1.1 201 Created*

Failure
: *HTTP/1.1 400 Bad Request*
HTTP/1.1 401 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.
HTTP/1.1 409 Conflict - The resource already exists.

130.4 Examples

Successfully create a new firewall group

: **POST** https://api.hostway.com/accounts/{accountId}/os-security-groups

Request Headers:

```
   Content-Type:application/json
   Charset=UTF-8
   Authorization:Base64-encoded username & password string
```

Request Body:

```
{
   "securityGroup" : {
      "displayName" : "Test Firewall Group",
      "description" : "Test firewall group description."
   }
}
```

Response Code:

```
HTTP/1.1 201 Created
```

Response Body:

```
{
   "securityGroup" : {
      "displayName": "Test Firewall Group",
      "description": "Test firewall group description.",
      "links": [
         {
            "href": "https://api.hostway.com/accounts/{accountId}/os-security-groups/group2012May11_OK30Zs7Ju/",
            "rel": "self"
         }
      ],
      "rules": [],
      "id": "group2012May11_OK30Zs7Ju",
      "accountNumber": "test_account_120761",
      "servers": []
   }
}
```

Failure while trying to create a new firewall group with invalid data

POST https://api.hostway.com/accounts/{accountId}/os-security-groups

Request Headers:

```
   Content-Type:application/json
   Charset=UTF-8
   Authorization:Base64-encoded username & password string
```

Request Body:

```
{
   Empty
}
```

Response Code:

```
HTTP/1.1 400 Bad Request
```

Response Body:

```
{
   "badRequest" : {
      "message": "Invalid",
      "code": 400,
      "details": "",
      "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
}
```

Failure while trying to create a new firewall group with an already existing firewall group display name

POST https://api.hostway.com/accounts/{accountId}/os-security-groups

Request Headers:

```
   Content-Type:application/json
   Charset=UTF-8
   Authorization:Base64-encoded username & password string
```

Request Body:

```
{
   "securityGroup" : {
      "displayName" : "Test Firewall Group",
      "description" : "Test firewall group description."
   }
}
```

Response Code:

```
HTTP/1.1 409 Conflict
```

Response Body:

```
{
   "computeFault" : {
      "message": "GroupNameError",
      "code": 400,
      "details": "Display name in use",
      "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
```

```
}
```

Failure while trying to make unauthorized access

POST https://api.hostway.com/accounts/{accountId}/os-security-groups

Request Headers:

```
   Content-Type:application/json
   Charset=UTF-8
   Authorization:Base64-encoded fake_authentication_string
```

Response Code:

```
HTTP/1.1 401 Unauthorized
```

Response Body:

```
    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.
```

Failure while trying to pass invalid account number

POST https://api.hostway.com/accounts/fakeId/os-security-groups

Response Code:

```
HTTP/1.1 404 Not Found
```

Response Body:

```
{
   "itemNotFound" : {
      "message": "Resource not found",
      "code": 404,
      "details": "",
      "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
}
```

130.5 Alternative URI /accounts/os-security-groups

An alternative way to execute this call is to omit the *accountId* portion of the URI. In this case the system assumes the request is executed for the account identified in the authorization header.

```
POST /accounts/os-security-groups
```

131 Accounts-accountId-os-security-groups-groupId:GET

This operation will list the details for the specified firewall group.

132 GET /accounts/*{accountId}*/os-security-groups/*{groupId}*

This operation will list the details for the specified firewall group.

132.1 Parameters

accountId - *string*
: The selected account Id.

groupId - *string*
: The unique identifier of the firewall group.

132.2 Response

132.2.1 JSON

```
{
   "securityGroup" : {
      "displayName" : "{firewall_group_display_name}",
      "description" : "{firewall_group_description}",
      "links" : [
         {
            "href" : "https://api.hostway.com/accounts/{accountId}/os-security-groups/{groupId}/",
            "rel" : "self"
         }
      ],
      "rules" : [
         {
            "fromPort" : "{start_port}",
            "toPort" : "{end_port}",
            "protocol" : "{ip_protocol}",
            "cidr": "0.0.0.0/0",
            "id" : "145"
         },
         {
            "fromPort" : "{start_port}",
            "toPort" : "{end_port}",
            "protocol" : "{ip_protocol}",
            "cidr": "127.0.0.1/0",
            "id" : "169"
         }
      ],
      "servers" : ["{server_id}"],
      "accountNumber" : "{user_account_number}",
      "id" : "{firewall_group_id}"
   }
}
```

132.2.1.1 securityGroup

displayName - *string*
: The display name of the firewall group.

description - *string*
: Text describing the firewall group.

links - *list*
: Hypermedia for this resource.

rules - *list*
: List describing all defined security rules for the group.

servers - *list*
: List of server ids this firewall group is assigned to.

accountNumber - *string*
: Identifies an account that owns the firewall group.

id - *string*
: The id of this firewall group.*(Unique identifier for this firewall group.)

132.3 Response Codes

Success
: *HTTP/1.1 200 OK*

Failure
: *HTTP/1.1 401 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.
HTTP/1.1 404 Not Found - The resource does not exist.

132.4 Examples

Successfully list details for specified firewall group

: **GET** https://api.hostway.com/accounts/{accountId}/os-security-groups/group2012May11_OK30Zs7Ju

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Response Code:

```
HTTP/1.1 200 OK
```

Response Body:

```
{
   "securityGroup" : {
      "displayName": "Test Firewall Group 1",
      "description": "Test firewall group 1 description.",
      "links": [
         {
            "href": "https://api.hostway.com/accounts/{accountId}/os-security-groups/group2012May11_OK30Zs7Ju/",
            "rel": "self"
         }
      ],
      "rules": [
         {
            'toPort': 1111,
            'fromPort': 1111,
            'protocol': 'TCP',
            'id': 145,
            'cidr': '0.0.0.0/0'
         },
         {
            'toPort': 80,
            'fromPort': 80,
            'protocol': 'TCP',
            'id': 196,
            'cidr': '127.0.0.1/24'
         }
      ],
      "id": "group2012May11_OK30Zs7Ju",
      "accountNumber": "test_account_120761",
      "servers": ["test-server-0003"]
   }
}
```

Failure while trying to request details for a firewall group that does not exist

GET https://api.hostway.com/accounts/{accountId}/os-security-groups/{fake_firewall_group_id}

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Response Code:

```
HTTP/1.1 404 Not Found
```

Response Body:

```
{
   "itemNotFound":
      {
         "message": "Resource not found",
         "code": 404,
         "details": "",
         "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
      }
}
```

Failure while trying to make unauthorized access

GET https://api.hostway.com/accounts/{accountId}/os-security-groups/group2012May11_OK30Zs7Ju

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded fake_authentication_string
```

Response Code:

```
HTTP/1.1 401 Unauthorized
```

Response Body:

```
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.
```

Failure while trying to pass invalid account number

GET https://api.hostway.com/accounts/fakeId/os-security-groups/group2012May11_OK30Zs7Ju

Response Code:

```
HTTP/1.1 404 Not Found
```

Response Body:

```
{
   "itemNotFound" : {
      "message": "Resource not found",
```

```
        "code": 404,
        "details": "",
        "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
    }
}
```

132.5 Alternative URI /accounts/os-security-groups/{groupId}

An alternative way to execute this call is to omit the *accountId* portion of the URI. In this case the system assumes the request is executed for the account identified in the authorization header.

```
GET /accounts/os-security-groups/{groupId}
```

133 Accounts-accountId-os-security-groups-groupId:PUT

This operation will update displayname and/or descriptin of the specified firewall group.

134 PUT /accounts/*{accountId}*/os-security-groups/*{groupId}*

134.1 Parameters

accountId - *string*
: The selected account Id.

groupId - *string*
: The unique identifier of the firewall group.

134.2 Request

134.2.1 JSON

```
{
   "securityGroup" : {
      "displayName" : "{firewall_group_display_name}",
      "description" : "{firewall_group_description}",
   }
}
```

134.3 Response

134.3.1 JSON

```
{
   "securityGroup" : {
      "displayName" : "{firewall_group_display_name}",
      "description" : "{firewall_group_description}",
      "links" : [
         {
            "href" : "https://api.hostway.com/accounts/{accountId}/os-security-groups/{groupId}/",
            "rel" : "self"
         }
      ],
      "rules" : [
         {
            "fromPort" : "{start_port}",
            "toPort" : "{end_port}",
            "protocol" : "{ip_protocol}",
            "cidr": "0.0.0.0/0",
            "id" : "145"
         },
         {
            "fromPort" : "{start_port}",
            "toPort" : "{end_port}",
            "protocol" : "{ip_protocol}",
            "cidr": "127.0.0.1/0",
            "id" : "169"
         }
      ],
      "servers" : ["{server_id}"],
      "accountNumber" : "{user_account_number}",
      "id" : "{firewall_group_id}"
   }
}
```

134.4 Response Codes

Success
: *HTTP/1.1 200 OK*

Failure
: *HTTP/1.1 401 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.
HTTP/1.1 404 Not Found - The resource does not exist.

134.5 Examples

Successfully edit details for specified firewall group

: **PUT** https://api.hostway.com/accounts/{accountId}/os-security-groups/group2012May11_OK30Zs7Ju

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Request Body:

```
{
    "securityGroup" : {
       "displayName": "New display name",
       "description": "New description",
}
```

Response Code:

```
HTTP/1.1 200 OK
```

Response Body:

```
 {
```

```
    "securityGroup" : {
       "displayName": "New display name",
       "description": "New description.",
       "links": [
          {
             "href": "https://api.hostway.com/accounts/{accountId}/os-security-groups/group2012May11_OK30Zs7Ju/",
             "rel": "self"
          }
       ],
       "rules": [
          {
             'toPort': 1111,
             'fromPort': 1111,
             'protocol': 'TCP',
             'id': 145,
             'cidr': '0.0.0.0/0'
          },
          {
             'toPort': 80,
             'fromPort': 80,
             'protocol': 'tcp',
             'id': 196,
             'cidr': '127.0.0.1/24'
          }
       ],
       "id": "group2012May11_OK30Zs7Ju",
       "accountNumber": "test_account_120761",
       "servers": ["test-server-0003"]
    }
 }
```

Failure while trying to edit a firewall group that does not exist

PUT https://api.hostway.com/accounts/{accountId}/os-security-groups/{fake_firewall_group_id}

Request Headers:

```
   Content-Type:application/json
   Charset=UTF-8
   Authorization:Base64-encoded username & password string
```

Response Code:

```
HTTP/1.1 404 Not Found
```

Response Body:

```
{
   "itemNotFound":
      {
         "message": "Resource not found",
         "code": 404,
         "details": "",
         "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
      }
}
```

135 Accounts-accountId-os-security-groups-groupId:DELETE

This operation will delete the specified firewall group.

136 DELETE /accounts/*{accountId}*/os-security-groups/*{groupId}*

This operation will delete the specified firewall group.

136.1 Parameters

accountId - *string*
: The selected account Id.

groupId - *string*
: The unique identifier of the firewall group.

136.2 Response Codes

Success
: *HTTP/1.1 204 No Content*

Failure
: *HTTP/1.1 401 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.
HTTP/1.1 404 Not Found - The resource does not exist.

136.3 Examples

Successfully delete the specified firewall group

DELETE https://api.hostway.com/accounts/{accountId}/os-security-groups/group2012May11_OK30Zs7Ju

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Response Code:

```
HTTP/1.1 204 No Content
```

Failure while trying to make unauthorized access

DELETE https://api.hostway.com/accounts/{accountId}/os-security-groups/group2012May11_OK30Zs7Ju

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded fake_authentication_string
```

Response Code:

```
HTTP/1.1 401 Unauthorized
```

Response Body:

```
 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.
```

Failure while trying to delete a non-existent firewall group

DELETE https://api.hostway.com/accounts/{accountId}/os-security-groups/{fake_firewall_group_id}

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Response Code:

```
HTTP/1.1 404 Not Found
```

Failure while trying to pass invalid account number

DELETE https://api.hostway.com/accounts/fakeId/os-security-groups/group2012May11_OK30Zs7Ju

Response Code:

```
HTTP/1.1 404 Not Found
```

Response Body:

```
{
   "itemNotFound" : {
      "message": "Resource not found",
      "code": 404,
      "details": "",
      "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
}
```

136.4 Alternative URI /accounts/os-security-groups/{groupId}

An alternative way to execute this call is to omit the *accountId* portion of the URI. In this case the system assumes the request is executed for the account identified in the authorization header.

```
DELETE /accounts/os-security-groups/{groupId}
```

137 Accounts-accountId-os-security-group-rules:POST

This operation will create a new firewall group rule.

138 POST /accounts/*{accountId}*/os-security-group-rules

This operation will create a new firewall group rule.

138.1 Request

138.1.1 Parameters

accountId - *string*
: The selected account Id.

138.1.2 JSON

```
{
   "securityGroupRule" : {
      "fromPort": {start_port},
      "toPort": {end_port},
      "protocol": "{ip_protocol}",
      "cidr": "127.0.0.1/24",
      "parentGroupId": "{parent_group_id}",
      "action": "{action}",
      "priority": {priority}
   }
}
```

138.1.2.1 securityGroupRule

fromPort - *int*
: Start port number.

toPort - *int*
: End port number.

protocol - *string*
: IP protocol used. Can be TCP, UDP, AH, ESP, GRE, AH+ESP, or ALL
AH, ESP, GRE, AH+ESP, ALL protocols don't support port - both should be 0

cidr - *string*
: CIDR notation IP address.

parentGroupId - *string*
: Parent group identifier for this rule.

action = *string*
: rule target action. One of ACCEPT, REJECT or DROP

priority = *integer*
: rule priority, starting from 0 lowest priority

138.2 Response

138.2.1 JSON

```
{
   "securityGroupRule" : {
      "fromPort": {start_port},
      "protocol": "{ip_protocol}",
      "links": [
         {
            "href": "https://api.hostway.com/accounts/{accountId}/os-security-group-rules/{firewall_group_rule_id}/",
            "rel": "self"
         }
      ],
      "toPort": {end_port},
      "parentGroupId": "{parent_firewall_group_name}",
      "cidr": "127.0.0.1/24"
      "id": {firewall_group_rule_id}
   }
}
```

138.2.1.1 securityGroupRule

fromPort - *int*
: Start port number.

protocol - *string*
: IP protocol used.

links - *list*
: Hypermedia for this resource.

toPort - *int*
: End port number.

parentGroupId - *string*
: Parent group identifier for this rule.

cidr - *string*
: IP range in CIDR notation.

id - *string*

Firewall group rule unique identifier.

138.3 Response Codes

Success

HTTP/1.1 201 Created

Failure

HTTP/1.1 400 Bad Request
HTTP/1.1 401 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.
HTTP/1.1 404 Not Found - The resource does not exist.

138.4 Examples

Successfully create a new firewall group rule

POST https://api.hostway.com/accounts/{accountId}/os-security-group-rules

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Request Body:

```
{
   "security_group_rule" : {
      "from_port": 80,
      "to_port": 80,
      "ip_protocol": "tcp",
      "cidr": "127.0.0.1/24",
      "parent_group_id": "group2012May11_NpCpS3995",
      "action": "ACCEPT",
      "priority": 1
   }
}
```

Response Code:

```
HTTP/1.1 201 Created
```

Response Body:

```
{
   "securityGroupRule" : {
      "fromPort": 80,
      "protocol": "tcp",
      "links": [
         {
            "href": "https://api.hostway.com/accounts/{accountId}/os-security-group-rules/214/",
            "rel": "self"
         }
      ],
      "toPort": 80,
      "parentGroupId": "group2012May11_NpCpS3995",
      "cidr": "127.0.0.1/24",
      "id": 214
   }
}
```

Successfully create a new firewall group rule with an IP without a netmask

POST https://api.hostway.com/accounts/{accountId}/os-security-group-rules

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Request Body:

```
{
   "securityGroupRule" : {
      "fromPort": 80,
      "toPort": 80,
      "protocol": "TCP",
      "cidr": "127.0.0.1",
      "parentGroupId": "group2012May11_NpCpS3995"
   }
}
```

Response Code:

```
HTTP/1.1 201 Created
```

Response Body:

```
{
   "securityGroupRule" : {
      "fromPort": 80,
      "protocol": "tcp",
      "links": [
         {
            "href": "https://api.hostway.com/accounts/{accountId}/os-security-group-rules/215/",
```

```
            "rel": "self"
         }
      ],
      "toPort": 80,
      "parentGroupId": "group2012May11_NpCpS3995",
      "cidr": "127.0.0.1/32",
      "id": 215
   }
}
```

Failure while trying to create a new firewall group rule with invalid data

POST https://api.hostway.com/accounts/{accountId}/os-security-group-rules

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Request Body:

```
{
   Empty
}
```

Response Code:

```
HTTP/1.1 400 Bad Request
```

Response Body:

```
{
   "badRequest" : {
      "message": "Invalid",
      "code": 400,
      "details": "",
      "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
}
```

Failure while trying to create a new rule with invalid from and to ports

POST https://api.hostway.com/accounts/{accountId}/os-security-group-rules

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Request Body:

```
{
   "securityGroupRule" : {
      "fromPort": 81,
      "toPort": 80,
      "protocol": "tcp",
      "cidr": "127.0.0.1/24",
      "parentGroupId": "group2012May11_NpCpS3995"
   }
}
```

Response Code:

```
HTTP/1.1 400 Bad Request
```

Response Body:

```
{
   "badRequest" : {
      "message": "FirewallRuleError",
      "code": 400,
      "details": "Invalid ports",
      "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
}
```

Failure while trying to create a new rule rule with an invalid protocol

POST https://api.hostway.com/accounts/{accountId}/os-security-group-rules

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Request Body:

```
{
   "securityGroupRule" : {
      "fromPort": 80,
      "toPort": 80,
      "protocol": "proto",
      "cidr": "127.0.0.1/24",
      "parentGroupId": "group2012May11_NpCpS3995"
   }
}
```

Response Code:

```
HTTP/1.1 400 Bad Request
```

Response Body:

```
{
   "badRequest" : {
      "message": "Invalid",
      "code": 400,
      "details": "",
      "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
}
```

Failure while trying to create a new firewall rule with an invalid IP address

POST http(s)://{api_server}/accounts/{accountId}/os-security-group-rules

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Request Body:

```
{
   "securityGroupRule" : {
      "fromPort": 80,
      "toPort": 80,
      "protocol": "tcp",
      "cidr": "ip_address",
      "parentGroupId": "group2012May11_NpCpS3995"
   }
}
```

Response Code:

```
HTTP/1.1 400 Bad Request
```

Response Body:

```
{
   "badRequest" : {
      "message": "FirewallRuleError",
      "code": 400,
      "details": "Source IP address is invalid",
      "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
}
```

Failure while trying to make unauthorized access

POST https://api.hostway.com/accounts/{accountId}/os-security-groups

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded fake_authentication_string
```

Response Code:

```
HTTP/1.1 401 Unauthorized
```

Response Body:

```
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.
```

Failure while trying to pass invalid account number

POST http(s)://{api_server}/accounts/fakeId/os-security-groups

Response Code:

```
HTTP/1.1 404 Not Found
```

Response Body:

```
{
   "itemNotFound" : {
      "message": "Resource not found",
      "code": 404,
      "details": "",
      "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
}
```

138.5 Alternative URI /accounts/os-security-group-rules

An alternative way to execute this call is to omit the *accountId* portion of the URI. In this case the system assumes the request is executed for the account identified in the authorization header.

```
POST /accounts/os-security-group-rules
```

139 Accounts-accountId-os-security-group-rules-ruleId:DELETE

This operation will delete a firewall group rule.

140 DELETE /accounts/*{accountId}*/os-security-group-rules/*{ruleId}*

This operation will delete a firewall group rule.

140.1 Parameters

accountId - *string*
: The selected account Id.

ruleId - *string*
: The unique firewall group rule identifier.

140.2 Response Codes

Success
: *HTTP/1.1 204 No Content*

Failure
: *HTTP/1.1 401 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.
HTTP/1.1 404 Not Found - The resource does not exist.

140.3 Examples

Successfully delete the specified firewall group rule

DELETE https://api.hostway.com/accounts/{accountId}/os-security-group-rules/{firewall_group_rule_id}

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Response Code:

```
HTTP/1.1 204 No Content
```

Failure while trying to delete a non-existent firewall group rule

DELETE https://api.hostway.com/accounts/{accountId}/os-security-group-rules/{fake_firewall_group_rule_id}

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded username & password string
```

Response Code:

```
HTTP/1.1 404 Not Found
```

Failure while trying to make unauthorized access

DELETE https://api.hostway.com/accounts/{accountId}/os-security-group-rules/{firewall_group_rule_id}

Request Headers:

```
Content-Type:application/json
Charset=UTF-8
Authorization:Base64-encoded fake_authentication_string
```

Response Code:

```
HTTP/1.1 401 Unauthorized
```

Response Body:

```
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.
```

Failure while trying to pass invalid account number

DELETE https://api.hostway.com/accounts/fakeId/os-security-group-rules/{firewall_group_rule_id}

Response Code:

```
HTTP/1.1 404 Not Found
```

Response Body:

```
{
   "itemNotFound" : {
      "message": "Resource not found",
      "code": 404,
      "details": "",
      "errorid": "59e0b3dd-2173-432a-8ff5-60efe16e83bf"
   }
}
```

140.4 Alternative URI /accounts/os-security-group-rules/{ruleId}

An alternative way to execute this call is to omit the *accountId* portion of the URI. In this case the system assumes the request is executed for the account identified in the authorization header.

```
DELETE /accounts/os-security-group-rules/{ruleId}
```

141 Common Features

All Hostway APIs support common features for authentication and authorization, pagination, references to related resources, etc.

- Authentication
- Hypermedia
- Pagination
- Server Response Codes
- Filtering

download this selection of articles as a PDF book

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus