
SpotCloud Buyer API Documentation

Release 1.2

Enomaly

May 05, 2011

CONTENTS

1	API Authentication	1
1.1	Python Authentication Example	1
1.2	Java Authentication Example	4
1.3	PHP Authentication Example	6
2	Instances	11
2.1	Listing Instances	11
2.2	Retrieving Instances	12
2.3	Creating Instances	14
2.4	Deleting Instances	16
3	Appliances	17
3.1	Listing Appliances	17
3.2	Listing Public Appliances	18
3.3	Retrieving Appliances	19
3.4	Updating Appliances	20
3.5	Downloading Appliances	21
3.6	Deleting Appliances	21
4	Providers	23
4.1	Listing Providers	23
4.2	Retrieving Providers	24
5	Hardware	27
5.1	Listing Hardware	27
5.2	Retrieving Hardware	28
6	Glossary	31
	Index	33

API AUTHENTICATION

The SpotCloud API uses *OAuth* to authenticate requests.

1.1 Python Authentication Example

Below is a Python example that illustrates how to authenticate with the *SpotCloud* API using the *OAuth* protocol.

```
1  """SpotCloud API authentication example using oauth."""
2
3  import os
4  import uuid
5  import httplib
6  from urllib import urlencode
7  from urlparse import urljoin
8
9  from oauth import oauth
10
11 OAUTH_GET_REQUEST_TOKEN_PATH    = '/_ah/OAuthGetRequestToken'
12 OAUTH_AUTHORIZE_TOKEN_PATH     = '/_ah/OAuthAuthorizeToken'
13 OAUTH_GET_ACCESS_TOKEN_PATH    = '/_ah/OAuthGetAccessToken'
14
15 URLENCODED_TYPE = 'application/x-www-form-urlencoded'
16
17 CHARSET = 'utf-8'
18
19 is_file = lambda o: hasattr(o, 'read')
20
21 def to_str(s, errors='strict'):
22     if not isinstance(s, basestring):
23         try:
24             return str(s)
25         except UnicodeEncodeError:
26             return unicode(s).encode(CHARSET, errors)
27     elif isinstance(s, unicode):
28         return s.encode(CHARSET, errors)
29     else:
30         return s
31
32 def field_generator(boundary, name, value):
33     yield '--' + boundary
34     yield 'Content-Disposition: form-data; name="%s"' % to_str(name)
35     yield ''
```

```
36     yield to_str(value)
37
38 def file_generator(boundary, name, file):
39     name = to_str(name)
40     filename = to_str(os.path.basename(file.name))
41
42     yield '--' + boundary
43     yield 'Content-Disposition: form-data; name="%s"; filename="%s"' % (name, filename)
44     yield 'Content-Type: application/octet-stream'
45     yield ''
46     yield file.read()
47
48 def multipart_generator(boundary, fields):
49     for key, value in fields.items():
50         if is_file(value):
51             for chunk in file_generator(boundary, key, value):
52                 yield chunk
53         else:
54             for chunk in field_generator(boundary, key, value):
55                 yield chunk
56     yield '--' + boundary + '--'
57     yield ''
58
59 def encode_multipart(data):
60     boundary = str(uuid.uuid4())
61     body = '\r\n'.join(multipart_generator(boundary, data))
62     headers = {
63         'Content-Type': 'multipart/form-data; boundary=%s' % boundary,
64         'Content-Length': len(body),
65     }
66     return headers, body
67
68
69 class ClientBase(object):
70
71     def __init__(self, host):
72         self.connection = httplib.HTTPSConnection(host)
73
74     def sign_request(self, method, path, data, body, headers):
75         pass
76
77     def request(self, method, path, data=None, headers=None):
78         body = None
79         headers = headers.copy() if headers else {}
80
81         if data:
82             if method in ('GET', 'DELETE'):
83                 path += '?' + urlencode(data, doseq=True)
84             elif method in ('POST', 'PUT'):
85                 content_headers, body = encode_multipart(data)
86                 headers.update(content_headers)
87             else:
88                 raise ValueError('%s method is not supported' % method)
89
90         self.sign_request(method, path, data, body, headers)
91
92         self.connection.request(method, path, body, headers)
93         response = self.connection.getresponse()
```

```
94         return response.read()
95
96
97     def get(self, path, data=None, headers=None):
98         return self.request('GET', path, data, headers)
99
100    def post(self, path, data=None, headers=None):
101        return self.request('POST', path, data, headers)
102
103    def put(self, path, data=None, headers=None):
104        return self.request('PUT', path, data, headers)
105
106    def delete(self, path, data=None, headers=None):
107        return self.request('DELETE', path, data, headers)
108
109
110    class SpotCloud(ClientBase):
111        host = 'spotcloud.appspot.com'
112
113        def __init__(self, consumer, access_token=None):
114            super(SpotCloud, self).__init__(self.host)
115
116            self.consumer = consumer
117            self.request_token = None
118            self.access_token = access_token
119
120        def fetch_oauth_request_token(self):
121            response = self.get(OAUTH_GET_REQUEST_TOKEN_PATH)
122            self.request_token = oauth.OAuthToken.from_string(response)
123            return self.request_token
124
125        def get_oauth_authorize_url(self):
126            if self.request_token is None:
127                raise RuntimeError('You should get request token first')
128
129            url = urljoin('https://%s/' % self.host, OAUTH_AUTHORIZE_TOKEN_PATH)
130
131            oauth_request = oauth.OAuthRequest.from_token_and_callback(
132                token = self.request_token,
133                http_url = url,
134            )
135            return oauth_request.to_url()
136
137        def fetch_oauth_access_token(self):
138            if self.request_token is None:
139                raise RuntimeError('You should get request token first')
140
141            response = self.get(OAUTH_GET_ACCESS_TOKEN_PATH)
142            self.access_token = oauth.OAuthToken.from_string(response)
143            return self.access_token
144
145        def sign_request(self, method, path, data, body, headers):
146            url = urljoin('https://%s/' % self.host, path)
147
148            parameters = None
149            if body is None or headers.get('Content-Type') == URLENCODED_TYPE:
150                parameters = data
151
```

```
152     signature_method = oauth.OAuthSignatureMethod_HMAC_SHA1()
153
154     if self.access_token is not None:
155         token = self.access_token
156     elif self.request_token is not None:
157         token = self.request_token
158     else:
159         token = None
160
161     oauth_request = oauth.OAuthRequest.from_consumer_and_token(
162         oauth_consumer = self.consumer,
163         token = token,
164         http_url = url,
165         http_method = method,
166         parameters = parameters,
167     )
168     oauth_request.sign_request(signature_method, self.consumer, token)
169     headers.update(oauth_request.to_header())
170
171 if __name__ == '__main__':
172
173     consumer = oauth.OAuthConsumer('key', 'secret')
174
175     client = SpotCloud(consumer)
176
177     #Get request token.
178     client.fetch_oauth_request_token()
179
180     #Authorize your request token.
181     authorization_url = client.get_oauth_authorize_url()
182     print 'Go to "%s", confirm token, come back here and press Enter.' % authorization_url
183     raw_input()
184
185     #Get your access token.
186     client.fetch_oauth_access_token()
187     access_token = client.access_token
188     print 'Got access token: oauth.OAuthToken(%r, %r)' % \
189         (client.access_token.key, client.access_token.secret)
```

1.2 Java Authentication Example

Below is a Java example that illustrates how to authenticate with the *SpotCloud* API using the *OAuth* protocol using the Scribe library. Please refer to the *Scribe* documentation for details on how to configure Scribe.

```
1  /* SpotCloud API authentication example using oauth. */
2
3  import org.scribe.oauth.*;
4  import org.scribe.builder.*;
5  import org.scribe.utils.*;
6  import org.scribe.builder.api.*;
7  import org.scribe.model.*;
8  import org.scribe.exceptions.*;
9  import org.apache.commons.codec.binary.*;
10 import java.io.*;
11 import java.util.*;
12
```

```

13 class java_oauth_authentication{
14
15     public static String host = "https://spotcloud.appspot.com";
16
17     public static OAuthService service = new ServiceBuilder()
18         .provider(SpotCloudApi.class)
19         .apiKey("anonymous")
20         .apiSecret("anonymous")
21         .build();
22
23     public static String get(Token token, String path, HashMap params, String respType) {
24
25         String paramString = "";
26
27         if(params != null) {
28             Set set = params.entrySet();
29             Iterator it = set.iterator();
30             while(it.hasNext()) {
31                 Map.Entry entry = (Map.Entry)it.next();
32                 paramString += "&" + entry.getKey() + "=" + entry.getValue();
33             }
34             paramString = paramString.substring(1);
35         }
36
37         OAuthRequest request = new OAuthRequest(Verb.GET,
38             host + path + "." + respType + "?" + paramString);
39
40         service.signRequest(token, request);
41         Response response = request.send();
42         return response.getBody();
43     }
44
45     public static Verifier getVerifier(Token requestToken) {
46
47         // Print Authorization URL
48         String authorization_url = service.getAuthorizationUrl(requestToken);
49         System.out.println("To retrieve your verifier go to:");
50         System.out.println(authorization_url + "\n");
51
52         // Read verifier
53         String sVerifier = null;
54         BufferedReader reader;
55         reader = new BufferedReader(new InputStreamReader(System.in));
56
57         System.out.println("Enter your verifier: ");
58
59         try {
60             sVerifier = reader.readLine();
61         } catch (IOException ioe) {
62             System.out.println("IO error occured attempting to read verifier.");
63             System.exit(1);
64         }
65
66         Verifier verifier = new Verifier(sVerifier);
67         return verifier;
68     }
69 }
70

```

```
71     public static void main(String[] args) {
72         // Get request token.
73         Token requestToken = service.getRequestToken();
74
75         // Get verifier to authorize your request token.
76         Verifier verifier = getVerifier(requestToken);
77
78         // Get your access token.
79         Token accessToken = service.getAccessToken(requestToken, verifier);
80         System.out.println("Got access token: OAuthToken(" +
81             accessToken.getToken() + ", " +
82             accessToken.getSecret() +
83             ")");
84     }
85 }
86
87
88
89
90 }
```

You will need to add the following *SpotCloudApi.java* file in the Scribe API directory in order to create the *SpotCloud OAuth* Service.

```
1  package org.scribe.builder.api;
2
3  import org.scribe.model.Token;
4
5  public class SpotCloudApi extends DefaultApi10a
6  {
7      private static final String AUTHORIZE_URL =
8          "https://spotcloud.appspot.com/_ah/OAuthAuthorizeToken?oauth_token=%s";
9
10     @Override
11     public String getAccessTokenEndpoint()
12     {
13         return "http://spotcloud.appspot.com/_ah/OAuthGetAccessToken";
14     }
15
16     @Override
17     public String getRequestTokenEndpoint()
18     {
19         return "http://spotcloud.appspot.com/_ah/OAuthGetRequestToken";
20     }
21
22     @Override
23     public String getAuthorizationUrl(Token requestToken)
24     {
25         return String.format(AUTHORIZE_URL, requestToken.getToken());
26     }
27 }
```

1.3 PHP Authentication Example

Below is a PHP example that illustrates how to authenticate with the *SpotCloud* API using the *OAuth* protocol.

```
1 <?php
2
3 require_once("OAuth.php");
4
5 class SpotCloudClient {
6
7     public $base_url = "https://spotcloud.appspot.com";
8
9     public $REQUEST_TOKEN_PATH;
10    public $AUTHORIZE_TOKEN_PATH;
11    public $ACCESS_TOKEN_PATH;
12
13    public $sig_method;
14
15    public $consumer;
16    public $request_token;
17    public $access_token;
18    public $verifier;
19
20    function __construct() {
21
22        $this->REQUEST_TOKEN_PATH = $this->base_url . "/_ah/OAuthGetRequestToken";
23        $this->AUTHORIZE_TOKEN_PATH = $this->base_url . "/_ah/OAuthAuthorizeToken";
24        $this->ACCESS_TOKEN_PATH = $this->base_url . "/_ah/OAuthGetAccessToken";
25        $this->consumer = new OAuthConsumer("anonymous", "anonymous", NULL);
26        $this->sig_method = new OAuthSignatureMethod_HMAC_SHA1();
27
28    }
29
30    function getRequestToken(){
31
32        $req = OAuthRequest::from_consumer_and_token($this->consumer,
33            NULL, "GET", $this->REQUEST_TOKEN_PATH, NULL);
34
35        $req->sign_request($this->sig_method, $this->consumer, NULL);
36
37        $response = $this->get($req->to_url());
38
39        parse_str($response, $params);
40
41        $token = $params['oauth_token'];
42        $token_secret = $params['oauth_token_secret'];
43
44        $this->request_token = new OAuthConsumer($token, $token_secret, NULL);
45
46    }
47
48    function getAuthUrl() {
49
50        $url = $this->AUTHORIZE_TOKEN_PATH . "?oauth_token=" . $this->request_token->key;
51
52        print "To grant access to your account go to:\n";
53        print $url . "\n";
54
55        print "Press enter once you have granted access: ";
56
57        $this->verifier = trim(fgets(STDIN));
58
```

```
59     }
60
61     function getAccessToken() {
62
63         $consumer = $this->consumer;
64         $req = OAuthRequest::from_consumer_and_token($consumer,
65             $this->request_token, "GET", $this->ACCESS_TOKEN_PATH);
66         $req->sign_request($this->sig_method, $consumer, $this->request_token);
67
68         $response = $this->get($req->to_url());
69
70         parse_str($response, $param);
71
72         $token = $param['oauth_token'];
73         $token_secret = $param['oauth_token_secret'];
74
75         $this->access_token = new OAuthConsumer($token, $token_secret, NULL);
76
77     }
78
79     function SpotCloudRequest($req, $response_type, $params) {
80
81         if($params != NULL) {
82             foreach($params as $key=>$value) {
83                 $paramString .= $key . "=" . $value . "&";
84             }
85             $paramString = rtrim($paramString, '&');
86         }
87
88         $url = $this->base_url . $req . "." . $response_type . "?" . $paramString;
89         $req = OAuthRequest::from_consumer_and_token($this->consumer,
90             $this->access_token, "GET", $url);
91
92         $req->sign_request($this->sig_method, $this->consumer, $this->access_token);
93
94         $response = $this->get($req->to_url());
95
96         return $response;
97
98     }
99
100    function get($url) {
101
102        $curl = curl_init();
103        curl_setopt($curl, CURLOPT_URL, $url);
104        curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
105        $response = curl_exec($curl);
106        curl_close($curl);
107        return $response;
108
109    }
110
111 }
112
113 $Client = new SpotCloudClient();
114
115 // Get request Token
116 $Client->getRequestToken();
```

```
117
118 // Get Authorization URL
119 $Client->getAuthUrl();
120
121 // Get Access Token
122 $Client->getAccessToken();
123 print "Got Access Token: $Client->request_token\n";
124
125 ?>
```


INSTANCES

2.1 Listing Instances

Returns a list of instances owned by the *authenticated user*. By default, all instances are returned. The returned instances may be filtered by *state*, *appliance*, or *provider*. Each filter parameter further refines the resulting instance list.

2.1.1 Request

```
GET /api/v1/buyer/instances/list.{json|xml}
```

2.1.2 Request Parameters

- **state** - Filter the returned instances by state (*running* | *pending*).
- **appliance** - Filter the returned instances by *appliance* key.
- **provider** - Filter the returned instances by *provider* key.

2.1.3 Response Fields

See the retrieving instances *response fields*

2.1.4 Response Codes

- **200** - The list of instances was successfully returned.

2.1.5 JSON Response

```
[  
  {  
    "id": <id>,  
    "key": <key>,  
    "ip": <ip>,  
  }  
]
```

```
    "uri": <uri>,  
    "state": <state>,  
    "deadline": <deadline>,  
    "cpu": <cpu>,  
    "memory": <memory>,  
    "cost": <cost>,  
    "provider_key": <provider_key>,  
    "appliance_key": <appliance_key>,  
    "hardware_key": <hardware_key>,  
    "provider_uri": <provider_uri>,  
    "appliance_uri": <appliance_uri>,  
    "hardware_uri": <hardware_uri>  
  },  
  ...  
]
```

2.1.6 XML Response

```
<list>  
  <object>  
    <id>{id}</id>  
    <key>{key}</key>  
    <ip>{ip}</ip>  
    <uri>{uri}</uri>  
    <state>{state}</state>  
    <deadline>{deadline}</deadline>  
    <cpu>{cpu}</cpu>  
    <memory>{memory}</memory>  
    <cost>{cost}</cost>  
    <provider_key>{provider_key}</provider_key>  
    <appliance_key>{appliance_key}</appliance_key>  
    <hardware_key>{hardware_key}</hardware_key>  
    <provider_uri>{provider_uri}</provider_uri>  
    <appliance_uri>{appliance_uri}</appliance_uri>  
    <hardware_uri>{hardware_uri}</hardware_uri>  
  </object>  
</list>
```

2.2 Retrieving Instances

Returns a specific instance by *key*. The instance *key* is part of the *URI*. Depending on the state of the instance, the *deadline* field may not be set. Any instance will not have a deadline until it is in a *running* state. The deadline is when the instance will terminate itself and is based on the maximum runtime specified in the hardware profile.

2.2.1 Request

GET /api/v1/buyer/instance/{key} . {json|xml}

2.2.2 URI Parameters

- **key** - The *SpotCloud* instance key.

2.2.3 Response Fields

- **id** - The human-readable instance ID.
- **key** - The *SpotCloud* instance key.
- **ip** - The IP address of this instance.
- **uri** - The API *URI* of this instance.
- **state** - The current state of this instance (*running* | *pending*).
- **deadline** - The date/time in which this instance will be automatically destroyed.
- **cpu** - The number of CPUs the instance hardware has.
- **memory** - The amount of memory the instance hardware has.
- **cost** - The hourly cost of running this instance.
- **provider_key** - The *SpotCloud provider* key used to create this instance.
- **appliance_key** - The *SpotCloud appliance* key used to create this instance.
- **hardware_key** - The *SpotCloud hardware* key used to create this instance.
- **provider_uri** - The API *URI* for the *provider* used to create this instance.
- **appliance_uri** - The API *URI* for the *appliance* used to create this instance.
- **hardware_uri** - The API *URI* for the hardware used to create this instance.

2.2.4 Response Codes

- **200** - The requested instance was found and successfully returned.
- **404** - The requested instance doesn't exist.
- **401** - The requested instance isn't owned by the *authenticated client*.

2.2.5 JSON Response

```
{
  "id": <id>,
  "key": <key>,
  "ip": <ip>,
  "uri": <uri>,
  "state": <state>,
  "deadline": <deadline>,

```

```
"cpu": <cpu>,  
"memory": <memory>,  
"cost": <cost>,  
"provider_key": <provider_key>,  
"appliance_key": <appliance_key>,  
"hardware_key": <hardware_key>,  
"provider_uri": <provider_uri>,  
"appliance_uri": <appliance_uri>,  
"hardware_uri": <hardware_uri>  
}
```

2.2.6 XML Response

<object>

```
<id>{id}</id>  
<key>{key}</key>  
<ip>{ip}</ip>  
<uri>{uri}</uri>  
<state>{state}</state>  
<deadline>{deadline}</deadline>  
<cpu>{cpu}</cpu>  
<memory>{memory}</memory>  
<cost>{cost}</cost>  
<provider_key>{provider_key}</provider_key>  
<appliance_key>{appliance_key}</appliance_key>  
<hardware_key>{hardware_key}</hardware_key>  
<provider_uri>{provider_uri}</provider_uri>  
<appliance_uri>{appliance_uri}</appliance_uri>  
<hardware_uri>{hardware_uri}</hardware_uri>
```

</object>

2.3 Creating Instances

Creates a new instance on the specified provider. There are three required parameters for creating a new instance - *appliance*, *hardware*, and *provider*. All three are *keys*. You can retrieve the required parameters necessary to create a new instance using other API resources. For instance, the *appliance API* allows you to list all appliances belonging to the *authenticated user*. The *provider API* allows the *authenticated user* to list provider keys.

2.3.1 Request

```
POST /api/v1/buyer/instances/list.{json|xml}
```

2.3.2 Request Parameters

- **appliance** - The *SpotCloud* appliance key used to create this instance.
- **hardware** - The *SpotCloud* hardware key used to create this instance.
- **provider** - The *SpotCloud* provider key used to create this instance.

2.3.3 Response Fields

See the retrieving instances *response fields*

2.3.4 Response Codes

- **201** - The new instance was successfully created.
- **400** - One of the supplied create instance parameters was invalid.
- **401** - The appliance parameter isn't owned by the *authenticated client* or the client insufficient funds.

2.3.5 JSON Response

```
{
  "id": <id>,
  "key": <key>,
  "ip": <ip>,
  "uri": <uri>,
  "state": <state>,
  "deadline": <deadline>,
  "cpu": <cpu>,
  "memory": <memory>,
  "cost": <cost>,
  "provider_key": <provider_key>,
  "appliance_key": <appliance_key>,
  "hardware_key": <hardware_key>,
  "provider_uri": <provider_uri>,
  "appliance_uri": <appliance_uri>,
  "hardware_uri": <hardware_uri>
}
```

2.3.6 XML Response

```
<object>
  <id>{id}</id>
  <key>{key}</key>
  <ip>{ip}</ip>
  <uri>{uri}</uri>
  <state>{state}</state>
  <deadline>{deadline}</deadline>
  <cpu>{cpu}</cpu>
  <memory>{memory}</memory>
  <cost>{cost}</cost>
  <provider_key>{provider_key}</provider_key>
  <appliance_key>{appliance_key}</appliance_key>
  <hardware_key>{hardware_key}</hardware_key>
  <provider_uri>{provider_uri}</provider_uri>
  <appliance_uri>{appliance_uri}</appliance_uri>
  <hardware_uri>{hardware_uri}</hardware_uri>
</object>
```

2.4 Deleting Instances

Deletes an instance owned by the *authenticated client*.

2.4.1 Request

```
DELETE /api/v1/buyer/instance/{key}
```

2.4.2 URI Parameters

- **key** - The *SpotCloud* instance key.

2.4.3 Response Codes

- **204** - The requested instance was found and deleted.
- **404** - The requested instance does not exist.
- **401** - The requested instance isn't owned by the *authenticated client*.

APPLIANCES

3.1 Listing Appliances

Returns a list of uploaded appliance objects. The *SpotCloud* buyer must upload appliances using the appliance upload web interface.

3.1.1 Request

```
GET /api/v1/buyer/appliances/list.{json|xml}
```

3.1.2 Response Fields

See the retrieving appliances *response fields*.

3.1.3 Response Codes

- **200** - The list of appliance objects was successfully returned.

3.1.4 JSON Response

```
[  
  {  
    "key": <key>,  
    "name": <name>,  
    "size": <size>,  
    "uri": <uri>,  
    "download_uri": <download_uri>,  
    "logo_uri": <logo_uri>,  
    "logo_key": <logo_key>  
  },  
  ...  
]
```

3.1.5 XML Response

```
<list>

  <object>

    <key>{key}</key>
    <name>{name}</name>
    <size>{size}</size>
    <uri>{uri}</uri>
    <download_uri>{download_uri}</download_uri>
    <logo_uri>{logo_uri}</logo_uri>
    <logo_key>{logo_key}</logo_key>

  </object>

  ...

</list>
```

3.2 Listing Public Appliances

Returns a list of public appliance objects. Public appliances are created by Enomaly and available to all *SpotCloud* buyers.

3.2.1 Request

```
GET /api/v1/buyer/appliances/public.{json|xml}
```

3.2.2 Response Fields

See the retrieving appliances *response fields*.

3.2.3 Response Codes

- **200** - The list of appliance objects was successfully returned.

3.2.4 JSON Response

```
[

  {

    "key": <key>,
    "name": <name>,
    "size": <size>,
    "uri": <uri>,
    "download_uri": <download_uri>,
    "logo_uri": <logo_uri>,
    "logo_key": <logo_key>
```

```

    },
    ...
]

```

3.2.5 XML Response

```

<list>
  <object>
    <key>{key}</key>
    <name>{name}</name>
    <size>{size}</size>
    <uri>{uri}</uri>
    <download_uri>{download_uri}</download_uri>
    <logo_uri>{logo_uri}</logo_uri>
    <logo_key>{logo_key}</logo_key>
  </object>
  ...
</list>

```

3.3 Retrieving Appliances

Returns a specific appliance object with the specified *key*. The *key* is part of the *URI*. *Authenticated clients* can retrieve individual appliance objects to view the appliance *size*, view the *download URI*, and view the *logo URI*. The client must already have the appliance key. The appliance key may be retrieved by listing all appliances or referencing it from an *instance*.

3.3.1 Request

```
GET /api/v1/buyer/appliance/{key}.{json|xml}
```

3.3.2 URI Parameters

- **key** - The *SpotCloud* appliance key.

3.3.3 Response Fields

- **key** - The *SpotCloud* appliance key.
- **name** - The human-readable appliance name.
- **size** - The appliance size.
- **uri** - The appliance API *URI*.

- **download_uri** - The appliance download *URI*.
- **logo_uri** - The appliance logo *URI*.
- **logo_key** - The *SpotCloud* appliance logo key.

3.3.4 Response Codes

- **200** - The appliance object was found and successfully returned.
- **404** - The requested appliance does not exist.
- **401** - The requested appliance isn't owned by the *authenticated client*.

3.3.5 JSON Response

```
{  
  
  "key": <key>,  
  "name": <name>,  
  "size": <size>,  
  "uri": <uri>,  
  "download_uri": <download_uri>,  
  "logo_uri": <logo_uri>,  
  "logo_key": <logo_key>  
  
}
```

3.3.6 XML Response

```
<object>  
  
  <key>{key}</key>  
  <name>{name}</name>  
  <size>{size}</size>  
  <uri>{uri}</uri>  
  <download_uri>{download_uri}</download_uri>  
  <logo_uri>{logo_uri}</logo_uri>  
  <logo_key>{logo_key}</logo_key>  
  
</object>
```

3.4 Updating Appliances

Updates a specific appliance object.

3.4.1 Request

```
PUT /api/v1/buyer/appliance/{key}.{json|xml}
```

3.4.2 URI Parameters

- **key** - The *SpotCloud* appliance key.

3.4.3 Request Parameters

- **name** - The new appliance name.

3.4.4 Response Fields

See the retrieving appliances *response fields*.

3.4.5 Response Codes

- **200** - The appliance object was successfully updated.
- **404** - The requested appliance does not exist.
- **401** - The requested appliance isn't owned by the *authenticated client*.

3.5 Downloading Appliances

Downloads the raw appliance data of a specific appliance object. The appliance *key* is part of the *URI*. The appliance will continue to exist after downloading it.

3.5.1 Request

```
GET /api/v1/download/appliance/{key}
```

3.5.2 URI Parameters

- **key** - The *SpotCloud* appliance key.

3.5.3 Response Codes

- **200** - The appliance was found and should be downloaded successfully.
- **404** - The requested appliance does not exist.
- **401** - The requested appliance isn't owned by the *authenticated client*.

3.6 Deleting Appliances

Deletes an appliance owned by the *authenticated client*.

3.6.1 Request

DELETE /api/v1/buyer/appliance/{key}

3.6.2 URI Parameters

- **key** - The *SpotCloud* appliance key.

3.6.3 Response Codes

- **204** - The requested appliance object was found and deleted.
- **404** - The requested appliance object does not exist.
- **401** - The requested appliance object isn't owned by the *authenticated client* or it is in use and cannot be deleted.

PROVIDERS

Deprecated since version 1.2. Use the *hardware* API instead.

4.1 Listing Providers

Returns a list of *SpotCloud* provider objects. By default, all providers are returned. The *authenticated client* can filter the provider list by location (*continent*, *country*, *city*), by hardware requirements (*minimum CPU count*, *minimum memory*), and by *maximum hourly cost*.

4.1.1 Request

```
GET /api/v1/buyer/providers/list.{json|xml}
```

4.1.2 Request Parameters

- **continent** - Filter returned providers by continent.
- **country** - Filter returned providers by country.
- **city** - Filter returned providers by city.
- **min_cpu** - Filter returned providers by minimum hardware CPU count.
- **min_memory** - Filter returned providers by minimum hardware memory size.
- **max_cost** - Filter returned providers by maximum hardware cost.

4.1.3 Response Fields

See the retrieving providers *response fields*

4.1.4 JSON Response

```
[  
  {  
    "id": <id>,  
  }  
]
```

```
"key": <key>,  
"continent": <continent>,  
"country": <country>,  
"city": <city>,  
"rating": <rating>,  
"min_cost": <min_cost>,  
"max_cost": <max_cost>,  
"uri": <uri>  
  
},  
  
...  
  
]
```

4.1.5 XML Response

```
<list>  
  
  <object>  
  
    <id>{id}</id>  
    <key>{key}</key>  
    <continent>{continent}</continent>  
    <country>{country}</country>  
    <city>{city}</city>  
    <rating>{rating}</rating>  
    <min_cost>{min_cost}</min_cost>  
    <max_cost>{max_cost}</max_cost>  
    <uri>{uri}</uri>  
  
  </object>  
  
  ...  
  
</list>
```

4.2 Retrieving Providers

Returns a specific provider object using the specified *SpotCloud* provider *key*. The *key* is part of the *URI*. The provider object returned here has a *hardware* field. This field is a list of hardware profile objects offered by the provider.

4.2.1 Request

```
GET /api/v1/buyer/provider/{key}.{json|xml}
```

4.2.2 URI Parameters

- **key** - The *SpotCloud* provider key.

4.2.3 Response Fields

- **id** - The human-readable provider ID.
- **key** - The SpotCloud provider key.
- **continent** - The human-readable provider continent.
- **country** - The human-readable provider country.
- **city** - The human-readable provider city.
- **rating** - The SpotCloud provider rating.
- **min_cost** - The minimum hourly cost offered by the provider.
- **max_cost** - The maximum hourly cost offered by the provider.
- **uri** - The API URI for the provider.
- **hardware** - A list of hardware profiles offered by this provider.

Each hardware profile object in the *hardware* list has the following fields:

- **key** - The SpotCloud hardware profile key.
- **name** - The human-readable name given to the hardware profile.
- **cpu** - The number of CPUs this hardware profile has.
- **memory** - The amount of memory this hardware profile has.
- **max_hours** - The maximum runtime for instances created with this profile.
- **uri** - The API URI for this hardware profile.
- **cost** - The hourly cost of running instances using this hardware profile.

4.2.4 Response Codes

- **200** - The requested provider object was found and successfully returned.
- **404** - The requested provider object does not exist.
- **401** - The requested provider object is not owned by the *authenticated client*.

4.2.5 JSON Response

```
{
  "id": <id>,
  "key": <key>,
  "continent": <continent>,
  "country": <country>,
  "city": <city>,
  "rating": <rating>,
  "min_cost": <min_cost>,
  "max_cost": <max_cost>,
  "uri": <uri>
  "hardware": [
    {
```

```
    "key": <key>,  
    "name": <name>,  
    "cpu": <cpu>,  
    "memory": <memory>,  
    "max_hours": <max_hours>,  
    "uri": <uri>,  
    "cost": <cost>  
  },  
  ...  
]  
}
```

4.2.6 XML Response

<object>

```
<id>{id}</id>  
<key>{key}</key>  
<continent>{continent}</continent>  
<country>{country}</country>  
<city>{city}</city>  
<rating>{rating}</rating>  
<min_cost>{min_cost}</min_cost>  
<max_cost>{max_cost}</max_cost>  
<uri>{uri}</uri>  
<hardware>  
  <key>{key}</key>  
  <name>{name}</name>  
  <cpu>{cpu}</cpu>  
  <memory>{memory}</memory>  
  <max_hours>{max_hours}</max_hours>  
  <uri>{uri}</uri>  
  <cost>{cost}</cost>  
</hardware>
```

</object>

HARDWARE

5.1 Listing Hardware

Returns a list of available hardware offerings from the SpotCloud market.

5.1.1 Request

```
GET /api/v1/buyer/hardware/list.{json|xml}
```

5.1.2 Request Parameters

- **continent** - Filter hardware by continent code (*NA, SA, ...*)
- **country** - Filter hardware by country code (*CA, US, ...*)
- **city** - Filter hardware by city name.
- **min_cpu** - Find hardware with at least this many CPUs.
- **max_cpu** - Find hardware that has no more than this many CPUs.
- **min_memory** - Find hardware with at least this much memory (*bytes*).
- **max_memory** - Find hardware that has no more than this much memory (*bytes*).
- **max_cost** - Find hardware that costs no more than this (*USD*).

5.1.3 Response Fields

See the retrieving hardware *response fields*

5.1.4 Response Codes

- **200** - The list of hardware was successfully returned.

5.1.5 JSON Response

```
[
  {
    "key": <key>,
    "name": <name>,
    "cost": <cost>,
    "cpu": <cpu>,
    "memory": <memory>,
    "max_hours": <max_hours>,
    "continent": <continent>,
    "country": <country>,
    "city": <city>,
    "uri": <uri>
  },
  ...
]
```

5.1.6 XML Response

```
<list>
  <object>
    <key>{key}</key>
    <name>{name}</name>
    <cost>{cost}</cost>
    <cpu>{cpu}</cpu>
    <memory>{memory}</memory>
    <max_hours>{max_hours}</max_hours>
    <continent>{continent}</continent>
    <country>{country}</country>
    <city>{city}</city>
    <uri>{uri}</uri>
  </object>
</list>
```

5.2 Retrieving Hardware

Returns a specific hardware profile by *key*. The hardware *key* is part of the *URI*.

5.2.1 Request

```
GET /api/v1/buyer/hardware/{key}.{json|xml}
```

5.2.2 URI Parameters

- **key** - The *SpotCloud* hardware key.

5.2.3 Response Fields

- **key** - The hardware profile key.
- **name** - The name of the hardware profile.
- **cost** - The hourly cost of running an instance using this profile.
- **cpu** - The CPU count of this hardware.
- **memory** - The amount of memory this hardware has (*bytes*).
- **max_hours** - The maximum number of hours this hardware profile may run.
- **continent** - The provider continent code.
- **country** - The provider country code.
- **city** - The provider city name.
- **uri** - The URI of this hardware profile.

5.2.4 Response Codes

- **200** - The requested hardware profile was found and successfully returned.
- **404** - The requested hardware profile doesn't exist.

5.2.5 JSON Response

```
{
  "key": <key>,
  "name": <name>,
  "cost": <cost>,
  "cpu": <cpu>,
  "memory": <memory>,
  "max_hours": <max_hours>,
  "continent": <continent>,
  "country": <country>,
  "city": <city>,
  "uri": <uri>
}
```

5.2.6 XML Response

<object>

```
<key>{key}</key>
<name>{name}</name>
<cost>{cost}</cost>
```

```
<cpu>{cpu}</cpu>
<memory>{memory}</memory>
<max_hours>{max_hours}</max_hours>
<continent>{continent}</continent>
<country>{country}</country>
<city>{city}</city>
<uri>{uri}</uri>

</object>
```

GLOSSARY

OAuth [OAuth](#) is a protocol that allows a user to grant a third party limited permission to access a web application on her behalf, without sharing her credentials (username and password) with the third party.

SpotCloud [SpotCloud](#) is a clearing house for excess capacity in cloud service provider environments. This is the [SpotCloud documentation](#).

URI URI stands for *uniform resource identifier*. Everything in the SpotCloud API is a URI and may be referred to by this address. The terms *URI* and *URL* are interchangeable.

INDEX

A

Appliances, 16
Authentication, 1

C

Creating Instances, 14

D

Deleting Appliances, 21
Deleting Instances, 15
Downloading Appliances, 21

H

Hardware, 26

I

Instances, 9

J

Java Authentication Example, 4

L

Listing Appliances, 17
Listing Hardware, 27
Listing Instances, 11
Listing Providers, 23
Listing Public Appliances, 18

O

OAuth, 31

P

PHP Authentication Example, 6
Providers, 22
Python Authentication Example, 1

R

Retrieving Appliances, 19
Retrieving Hardware, 28
Retrieving Instances, 12

Retrieving Providers, 24

S

SpotCloud, 31

U

Updating Appliances, 20
URI, 31