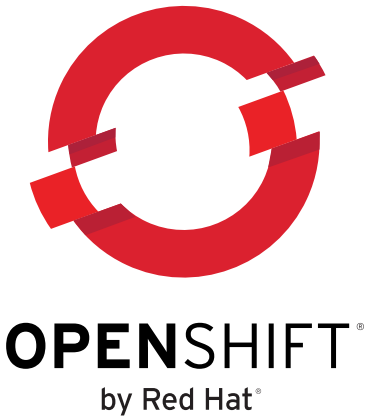

OpenShift Online All Versions REST API Guide

Using the OpenShift Online Representational State Transfer Application Programming Interface (REST API)

PressGang CCMS Build System

OpenShift Online All Versions REST API Guide

Using the OpenShift Online Representational State Transfer Application Programming Interface (REST API)

PressGang CCMS Build System
Red Hat Engineering Content Services

Keywords

Abstract

A guide to using the OpenShift Online REST API for developers

Table of Contents

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the Liberation Fonts set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **`Buttons`** tab, click the **`Left-handed mouse`** check box and click **`Close`** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **`gedit`** file, choose **Applications** → **Accessories** →

> **Character Map** from the main menu bar. Next, choose **Search → Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit → Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.
>
> The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.
>
> To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. Getting Help

2.1. Do You Need Help?

If you experience difficulty with a procedure or other information described in this documentation, visit the Red Hat Customer Portal at http://access.redhat.com where you can:

- search or browse through a knowledgebase of technical support articles about Red Hat products
- submit a support case to Red Hat Global Support Services (GSS)
- access other product documentation

You can also access the OpenShift web site at https://openshift.redhat.com/ to find blogs, FAQs, forums, and other sources of information.

Red Hat also hosts a large number of electronic mailing lists for discussion of Red Hat software and technology. You can find a list of publicly available mailing lists at https://www.redhat.com/mailman/listinfo. Click the name of any mailing list to subscribe to that list or to access the list archives.

2.2. We Need Feedback!

If you find a typographical or any other error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: http://bugzilla.redhat.com/ against the product **OpenShift Online.**

When submitting a bug report, be sure to mention the manual's identifier: *REST_API_Guide*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Chapter 1. Introduction

1.1. About OpenShift API

OpenShift provides a Representational State Transfer (REST) Application Programming Interface (API). Applications created for OpenShift can access the API using the standard Hypertext Transfer Protocol (HTTP). The OpenShift API is structured as a resource, and provides links to all children and any related resources.

Note

The legacy API has been removed, and is no longer supported. All customers are advised to use the current OpenShift REST API.

Report a bug

1.2. About Representational State Transfer (REST)

Representational State Transfer (REST) is a design architecture for networked applications or systems. In the REST design architecture a client progresses through an application by selecting links, also known as state transitions. Each link selected by the client returns a *representation* of the selected resource. Also, with each resource representation the client application transfers state, and results in a usable web page being transferred and rendered.

In the REST design architecture, a resource is created for every service that an application provides, with each resource identified by a URL. A client can reference the resource using the URL. The returned representation of the resource is further linked to more information, allowing the client to drill down as far as necessary to get more detailed information. The client can access and perform operations on available resources with standard HTTP methods, such as **GET**, **POST**, **PUT**, and **DELETE**.

Report a bug

Chapter 2. General API Information

2.1. Authentication

The OpenShift API uses *Basic Authentication*. A client is required to send the username and password, separated by a colon, with all requests to correctly authenticate. For example, send the credentials as *username*:*password*. This string is encoded with Base64 algorithm, and transmitted in the HTTP authorization header in the formats shown below.

Using Ruby

```
require 'base64'
base64string = Base64.encode64("#{username}:#{password}").strip
headers = { "Authorization" => "Basic #{base64string}" }
```

Using Python

```
import base64
base64string = base64.encodestring('%s:%s' % (username, password))[:-1]
request.add_header("Authorization", "Basic %s" % base64string)
```

Using cURL

The cURL library supports basic authentication using the **--user** option, as shown in the example below.

```
$ curl -k https://openshift.redhat.com/broker/rest/user --user
"[UserName]:[Password]"
```

Report a bug

2.2. Version

Every OpenShift REST API call returns the current API version, and other versions that are supported. The example below shows how to use the **cURL** command to return the API version.

```
$ curl -k "https://openshift.redhat.com/broker/rest/api"
```

The response for this command is shown below in both XML and JSON syntax.

XML Response

```
<response>
  <status>ok</status>
  <version>1.4</version>
  <supported-api-versions>
    <supported-api-version>1.0</supported-api-version>
    <supported-api-version>1.1</supported-api-version>
    <supported-api-version>1.2</supported-api-version>
    <supported-api-version>1.3</supported-api-version>
    <supported-api-version>1.4</supported-api-version>
  </supported-api-versions>
</response>
```

JSON Response

```
{
   "supported_api_versions": [
        1.0,
        1.1,
        1.2,
        1.3,
        1.4
    ],
    "version": "1.4"
}
```

To request a specific API version, the client must include the HTTP header with the response request. Examples in both XML and JSON syntax are provided below.

To Request Specific API Version in XML Response

```
Accept: application/xml; version=1.4
```

To Request Specific API Version in JSON Response

```
Accept: application/json; version=1.4
```

If the version requested by the client is not supported, the server responds with the HTTP status code 406, as shown in the examples below in XML and JSON syntax.

XML Response for Unsupported Version

```
<response>
  <messages>
    <message>
      <text>Requested API version 2.0 is not supported.  Supported versions are
1.0, 1.1, 1.2, 1.3, 1.4</text>
      <severity>error</severity>
    </message>
  </messages>
  <version>1.4</version>
  <supported-api-versions>
    <supported-api-version>1.0</supported-api-version>
    <supported-api-version>1.1</supported-api-version>
    <supported-api-version>1.2</supported-api-version>
    <supported-api-version>1.3</supported-api-version>
    <supported-api-version>1.4</supported-api-version>
  </supported-api-versions>
  <status>not_acceptable</status>
</response>
```

JSON Response for Unsupported Version

```
{
    "data": null,
    "messages": [
        {
            "exit_code": null,
            "field": null,
            "severity": "error",
            "text": "Requested API version 2.0 is not supported. Supported
versions are 1.0, 1.1, 1.2, 1.3, 1.4"
        }
    ],
    "status": "not_acceptable",
    "supported_api_versions": [
        1.0,
        1.1,
        1.2,
        1.3,
        1.4
    ],
    "type": null,
    "version": "1.4"
}
```

Report a bug

2.3. Response Type

Although OpenShift supports both XML and JSON response formats, the default server response is the JSON syntax. Include the following HTTP header to receive the response in XML:

```
Accept: application/xml
```

Using Ruby

```
headers = {"Accept" => "application/xml"}
```

Using Python

```
request.add_header("Accept", "application/xml")
```

Using cURL

```
curl -k "https://openshift.redhat.com/broker/rest/api" -H "Accept: application/xml"
```

Report a bug

2.4. Link Navigation

The OpenShift REST API implements the Hypermedia as the Engine of Application State, or *HATEOAS*, design principle of the REST application architecture. This principle implies that the interaction between a client and a network application happens entirely through links provided dynamically by the application server. No prior knowledge, beyond a generic understanding of REST and HTTP protocol, is required from the REST client on how to interact with any particular application or server. Entry to the REST application by a REST client is through a simple fixed URL. All future actions the client takes are discovered within resource representations returned from the server. The client selects the links within these resources to navigate to the required resource.

The **nolinks** parameter can be included with all supported APIs, where it can be set to ***true*** or ***false***. If the **nolinks** parameter is not included, it automatically defaults to ***false***. If the **nolinks** parameter is included and set to ***true***, the link navigation information is excluded from the API response resulting in a concise output and improved general processing speed.

The cURL command example below shows the API response for a **GET** method without the **nolinks** parameter:

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/user --user
"[UserName]:[Password]"
{
    "data": {
        "capabilities": {
            "gear_sizes": [
                "small",
                "medium"
            ],
            "plan_upgrade_enabled": true,
            "private_ssl_certificates": true,
            "subaccounts": false,
            "max_storage_per_gear": 5
        },
        "consumed_gears": 5,
        "created_at": "2013-02-07T22:48:58Z",
        "id": "51142f5adbd93ce16a0005b3",
        "links": {
            "LIST_KEYS": {
                "href": "https://openshift.redhat.com/broker/rest/user/keys",
                "method": "GET",
                "optional_params": [],
                "rel": "Get SSH keys",
                "required_params": []
            },
            "ADD_KEY": {
                "href": "https://openshift.redhat.com/broker/rest/user/keys",
                "method": "POST",
                "optional_params": [],
                "rel": "Add new SSH key",
                "required_params": [
                    {
                        "description": "Name of the key",
                        "invalid_options": [],
                        "name": "name",
                        "type": "string",
                        "valid_options": []
                    },
                    {
                        "description": "Type of Key",
                        "invalid_options": [],
                        "name": "type",
                        "type": "string",
                        "valid_options": [
                            "ssh-rsa",
                            "ssh-dss",
                            "ecdsa-sha2-nistp256-cert-v01@openssh.com",
                            "ecdsa-sha2-nistp384-cert-v01@openssh.com",
                            "ecdsa-sha2-nistp521-cert-v01@openssh.com",
                            "ssh-rsa-cert-v01@openssh.com",
                            "ssh-dss-cert-v01@openssh.com",
                            "ssh-rsa-cert-v00@openssh.com",
                            "ssh-dss-cert-v00@openssh.com",
                            "ecdsa-sha2-nistp256",
                            "ecdsa-sha2-nistp384",
                            "ecdsa-sha2-nistp521"
                        ]
                    },
                    {
                        "description": "The key portion of an rsa key (excluding
```

```
ssh-rsa and comment)",
                        "invalid_options": [],
                        "name": "content",
                        "type": "string",
                        "valid_options": []
                    }
                ]
            }
        },
        "login": "user@myemail.com",
        "max_gears": 16,
        "plan_id": "silver",
        "plan_state": "ACTIVE",
        "usage_account_id": "2526383"
    },
    "messages": [],
    "status": "ok",
    "type": "user",
}
```

The cURL example below shows the API response for a **GET** method with the **nolinks** parameter set to ***true***:

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/user?nolinks=true --
user "[UserName]:[Password]"
{
    "data": {
        "capabilities": {
            "gear_sizes": [
                "small",
                "medium"
            ],
            "plan_upgrade_enabled": true,
            "private_ssl_certificates": true,
            "subaccounts": false,
            "max_storage_per_gear": 5
        },
        "consumed_gears": 5,
        "created_at": "2013-02-07T22:48:58Z",
        "id": "51142f5adbd93ce16a0005b3",
        "login": "user@myemail.com",
        "max_gears": 16,
        "plan_id": "silver",
        "plan_state": "ACTIVE",
        "usage_account_id": "2526383"
    },
    "messages": [],
    "status": "ok",
    "type": "user",
}
```

The cURL command example below shows the API response for a **PUT** method with the **nolinks** parameter set to ***true***:

```
$ curl -k -X PUT https://openshift.redhat.com/broker/rest/user --user
"[UserName]:[Password]" --data "plan_id=free" --data "nolinks=true"
{
    "data": {
        "capabilities": {
            "plan_upgrade_enabled": true,
            "subaccounts": false,
            "gear_sizes": [
                "small"
            ]
        },
        "consumed_gears": 0,
        "created_at": "2013-05-29T23:18:16Z",
        "id": "51a68cb836905d42c3000016",
        "login": "user",
        "max_gears": 3,
        "plan_id": "free",
        "plan_state": "ACTIVE",
        "usage_account_id": "2223379"
    },
    "errors": {},
    "messages": [],
    "status": "ok",
    "type": "account",
}
```

API Entry Point

The entry to the OpenShift API is at the following URL: **`https://openshift.redhat.com/broker/rest/api`**.

Each link returned by the OpenShift API contains the **`HREF`**, **`METHOD`**, **`PARAMS`**, and **`REL`** elements. Each of these elements is described in the table below.

Table 2.1. Resource Elements

Element Name	Description
`HREF`	URL for resource link
`METHOD`	HTTP method to use with resource link: **`GET`**, **`PUT`**, **`POST`**, or **`DELETE`**
`PARAMS`	Optional or required parameters for the resource link
`REL`	Description of resource link

An example of an API representation in both XML and JSON syntax is shown in the examples below.

XML Representation

```
<link>
      <optional-params/>
      <required-params>
        <param>
          <type>string</type>
          <valid-options/>
          <name>id</name>
          <description>Name of the domain</description>
        </param>
      </required-params>
      <href>https://openshift.redhat.com/broker/rest/domains</href>
      <rel>Create new domain</rel>
      <method>POST</method>
    </link>
```

JSON Representation

```
{
  "required_params": [
    {
      "type": "string",
      "valid_options": [],
      "description": "Name of the domain",
      "name": "id"
    }
  ],
  "method": "POST",
  "optional_params": [],
  "href": "https://openshift.redhat.com/broker/rest/domains",
  "rel": "Create new domain"
}
```

Report a bug

2.5. OpenShift API Status Codes

The OpenShift REST API attempts to return standard *HTTP status codes*, with the more common status codes shown in the table below along with a brief description of each.

Table 2.2. HTTP Status Codes

Code	Text	Description
200	OK	Standard response for successful HTTP requests.
201	Created	The resource was successfully created.
204	No content	The requested delete operation was successful.
301	Moved Permanently	The resource has moved, and all future requests should be made to the new URI.
400	Bad Request	Invalid request due to bad syntax.
401	Unauthorized	Authentication has failed, or was not provided.
403	Forbidden	The request is understood, but server is refusing to respond.
404	Not Found	The requested resource cannot be found.
406	Not Acceptable	The content from the requested resource is not acceptable according to the Accept headers. Possibly due to version requested, or it no longer being supported.
409	Conflict	The request could not be processed because of conflict in the request.
410	Gone	The resource is no longer available, and will not be available again.
422	Unprocessable Entity	The request was well formed, but was not followed due to semantic errors.
500	Internal Server Error	A generic error message when something is broken.
502	Bad Gateway	Server was acting as a gateway or proxy, and received an invalid response.
503	Service Unavailable	The server is currently unavailable; possibly down for maintenance.
504	Gateway Timeout	The server was acting as a gateway or proxy and did not receive a timely response.

Report a bug

Chapter 3. API Entry Point

Description

Interaction with the OpenShift API begins with a request to the URL for the API entry point. The entry point provides navigation links to resources for a client to manage an OpenShift cloud environment.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/api

Request Parameters

N/A

Response

The OpenShift API resources are returned.

cURL Command Example

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/api
```

Sample JSON Response

The sample JSON output below shows the API response. Unnecessary information has been left out for brevity.

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/api
{
    "data": {
        "API": {
            "href": "https://openshift.redhat.com/broker/rest/api",
            "method": "GET",
            "optional_params": [],
            "rel": "API entry point",
            "required_params": []
        },
        "GET_ENVIRONMENT": {
            "href": "https://openshift.redhat.com/broker/rest/environment",
            "method": "GET",
            "optional_params": [],
            "rel": "Get environment information",
            "required_params": []
        },
        "GET_USER": {
            "href": "https://openshift.redhat.com/broker/rest/user",
            "method": "GET",
            "optional_params": [],
            "rel": "Get user information",
            "required_params": []
        },
        "LIST_DOMAINS": {
            "href": "https://openshift.redhat.com/broker/rest/domains",
            "method": "GET",
            "optional_params": [],
            "rel": "List domains",
            "required_params": []
        },
        "ADD_DOMAIN": {
            "href": "https://openshift.redhat.com/broker/rest/domains",
            "method": "POST",
            "optional_params": [],
            "rel": "Create new domain",
            "required_params": [
                {
                    "description": "Name of the domain",
                    "invalid_options": [
                        "amentra",
                        "aop",
                        "apiviz",
                        "arquillian",
.......
                    ],
                    "name": "id",
                    "type": "string",
                    "valid_options": []
                }
            ]
        },
        "LIST_CARTRIDGES": {
            "href": "https://openshift.redhat.com/broker/rest/cartridges",
            "method": "GET",
            "optional_params": [],
            "rel": "List cartridges",
            "required_params": []
        },
        "LIST_AUTHORIZATIONS": {
```

```
            "href": "https://openshift.redhat.com/broker/rest/user/authorizations",
            "method": "GET",
            "optional_params": [],
            "rel": "List authorizations",
            "required_params": []
        },
        "SHOW_AUTHORIZATION": {
            "href":
"https://openshift.redhat.com/broker/rest/user/authorizations/:id",
            "method": "GET",
            "optional_params": [],
            "rel": "Retrieve authorization :id",
            "required_params": [
                {
                    "description": "Unique identifier of the authorization",
                    "invalid_options": [],
                    "name": ":id",
                    "type": "string",
                    "valid_options": []
                }
            ]
        },
        "ADD_AUTHORIZATION": {
            "href": "https://openshift.redhat.com/broker/rest/user/authorizations",
            "method": "POST",
            "optional_params": [
                {
                    "default_value": "userinfo",
                    "description": "Select one or more scopes that this
authorization will grant access to:\n\n*  session\n   Grants a client the authority
to perform all API actions against your account. Valid for 1 day.\n*  read\n
Allows the client to access resources you own without making changes. Does not
allow access to view authorization tokens. Valid for about 1 month.\n*  userinfo\n
Allows a client to view your login name, unique id, and your user capabilities.
Valid for about 1 month.",
                    "name": "scope",
                    "type": "string",
                    "valid_options": [
                        "session",
                        "read",
                        "userinfo"
                    ]
                },
                {
                    "default_value": null,
                    "description": "A description to remind you what this
authorization is for.",
                    "name": "note",
                    "type": "string",
                    "valid_options": []
                },
                {
                    "default_value": -1,
                    "description": "The number of seconds before this
authorization expires. Out of range values will be set to the maximum allowed
time.",
                    "name": "expires_in",
                    "type": "integer",
                    "valid_options": []
                },
```

```
                {
                    "default_value": false,
                    "description": "Attempt to locate and reuse an authorization
that matches the scope and note and has not yet expired.",
                    "name": "reuse",
                    "type": "boolean",
                    "valid_options": [
                        true,
                        false
                    ]
                }
            ],
            "rel": "Add new authorization",
            "required_params": []
        },
        "LIST_QUICKSTARTS": {
            "href": "https://www.openshift.com/api/v1/quickstarts/promoted.json",
            "method": "GET",
            "optional_params": [],
            "rel": "List quickstarts",
            "required_params": []
        },
        "SHOW_QUICKSTART": {
            "href": "https://www.openshift.com/api/v1/quickstarts/:id",
            "method": "GET",
            "optional_params": [],
            "rel": "Retrieve quickstart with :id",
            "required_params": [
                {
                    "description": "Unique identifier of the quickstart",
                    "invalid_options": [],
                    "name": ":id",
                    "type": "string",
                    "valid_options": []
                }
            ]
        },
        "SEARCH_QUICKSTARTS": {
            "href": "https://www.openshift.com/api/v1/quickstarts.json",
            "method": "GET",
            "optional_params": [],
            "rel": "Search quickstarts",
            "required_params": [
                {
                    "description": "The search term to use for the quickstart",
                    "invalid_options": [],
                    "name": "search",
                    "type": "string",
                    "valid_options": []
                }
            ]
        }
    },
    "messages": [],
    "status": "ok",
    "type": "links",
}
```


Report a bug

Chapter 4. Authorizations

This chapter provides information on API resources to add, view, and manage authorization tokens. An authorization token is a secret value that allows access to an OpenShift account without having to supply login information each time. An authorization token can also be used to grant another user access to your account. Authorization tokens use scopes to determine the type of access a user is granted and what they can do with that token; for example full access versus limited access.

Authorization tokens are easily managed, and offer better security because you are not repeatedly supplying your login credentials. For example, if your password is ever compromised, the password must be reset. If a secret authorization token is compromised, that token can be revoked and another one created.

The table below describes each parameter associated with an OpenShift authorization token.

Name	Description
id	Unique OpenShift login that created this authorization token.
scope	Scope of the authorization token to determine type of access. Scopes that are supported by a server are described in the **ADD_AUTHORIZATION** resource link and may be different for each server.
note	A reminder description of what the authorization is for.
expires_in	Total time in seconds before this authorization expires. Out of range values will be set to the maximum allowed time.
expires_in_seconds	Remaining time in seconds before this authorization expires.
reuse	Attempt to locate and reuse an authorization that matches the scope and note and has not yet expired.
token	Authorization string that contains user credentials.

The table below describes the available **scope** options that determine the type of access a user is granted with an authorization.

Name	Description
session	Equivalent of logging into an account with normal login credentials such as user name and password.
read	Access to the API is read-only, while authorization endpoints cannot be read.
userinfo	Only read access to the **/user** API resource is provided.

Report a bug

4.1. List Authorizations

Description

The **LIST_AUTHORIZATIONS** resource lists all authorizations for the specified user account, and provides the client additional resource links to manage existing authorizations.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/user/authorizations

Request Parameters

Not applicable

Response

The API returns the authorizations resource with related resource links. No resource links are returned if the user account does not contain any authorizations. See Chapter 4, *Authorizations* for more information on all authorization parameters.

cURL Command Example

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/user/authorizations
--user "[Username]:[Password]"
```

Sample JSON Response

As shown in the sample response output below, the API returns a list of authorizations for the specified user account. The API also returns related resource links to manage existing authorizations, which have been left out for brevity.

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/user/authorizations
--user "user@myemail.com:password"
{
    "data": [
        {
            "created_at": "2013-06-28T03:07:58Z",
            "expires_in": 8000,
            "expires_in_seconds": 212,
            "id": "51ccfe0e2587c8e0b20000aa",
            "identity": "user@myemail.com",
            "note": null,
            "scopes": "session",
            "token":
"37496659dd6893c39409d2c259a38245869f0762a9d7586f7e568d1aa738f2ab"
        }
    ],
    "messages": [
        {
            "exit_code": 0,
            "field": null,
            "severity": "info",
            "text": "List authorizations"
        }
    ],
    "status": "ok",
    "type": "authorizations",
}
```

Report a bug

4.2. Add an Authorization

Description

The **ADD_AUTHORIZATION** resource allows a client to add an authorization to the specified user account.

Method and URL Structure

Method	URL Structure
POST	https://openshift.redhat.com/broker/rest/user/authorizations

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
scope	Scope of the authorization.	String	yes	***session***; ***read***; ***userinfo***	Not applicable
note	Reminder description of authorization.	String	Yes	Not applicable	Not applicable
expires_in	Number of seconds before authorization expires.	String	Yes	Not applicable	-1. For invalid values, the default value is determined by the server.
reuse	Attempt to locate and reuse an authorization matching scope and note and has not expired.	Boolean	Yes	***true*** or ***false***	***false***

Response

The API returns the authorizations resource with related resource links. See Chapter 4, *Authorizations* for more information on all authorization parameters.

cURL Command Example

```
$ curl -k -X POST https://openshift.redhat.com/broker/rest/user/authorizations
--user "[Username]:[Password]" --data "scope=[Scope_Type]&expires_in=[Time]"
```

Sample JSON Response

The sample output below shows the API response after successfully adding an authorization token to the specified user account. In this example the **scope** type is ***session*** and the authorization expires in ***7200*** seconds. The related resource links returned by the API have been left out for brevity.

```
$ curl -k -X POST https://openshift.redhat.com/broker/rest/user/authorizations
--user "user@myemail.com:password" --data "scope=session&expires_in=7200"
{
    "data": {
        "created_at": "2013-06-28T02:41:21Z",
        "expires_in": 7200,
        "expires_in_seconds": 7200,
        "id": "51ccf7d1dbd93c041b0002af",
        "identity": "user@myemail.com",
        "note": null,
        "scopes": "session",
        "token":
"7bac790f2eaf043ed08d9425065a3f7d6b218b6320751ea555e6f38d74be40c1"
    },
    "errors": {},
    "messages": [
        {
            "exit_code": 0,
            "field": null,
            "severity": "info",
            "text": "Create authorization"
        }
    ],
    "status": "created",
    "type": "authorization",
}
```

Report a bug

4.3. Show Authorization

Description

The **SHOW_AUTHORIZATION** resource allows a client that knows the unique identifier of a resource to directly substitute the **id** of an authorization for the replacement variable ***:id*** in the resource URL. The client does not have to retrieve the authorizations list and scan for a known **id**.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/user/authorizations/:id

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
id	Unique identifier of the authorization	String	No	Not applicable	Not applicable

Response

The API returns the authorization resource with related resource links. See Chapter 4, *Authorizations* for more information on all authorization parameters.

cURL Command Example

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/user/authorizations/:id --user
"[Username]:[Password]"
```

Sample JSON Response

The sample response output below shows information about an authorization. The related resource links returned by the API have been left out for brevity.

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/user/authorizations/51ccfe0e2587c8e0b2
0000aa --user "user@myemail.com:password"
{
    "data": {
        "created_at": "2013-07-01T04:23:26Z",
        "expires_in": 8000,
        "expires_in_seconds": 7527,
        "id": "51d1043e2587c881910000af",
        "identity": "user@myemail.com",
        "note": null,
        "scopes": "session",
        "token":
"40a241ed42b73e8a88f5a3a5498d62b0dc01186ea13ce5383f776785ceb9e088"
    },
    "messages": [
        {
            "exit_code": 0,
            "field": null,
            "severity": "info",
            "text": "Display authorization"
        }
    ],
    "status": "ok",
    "type": "authorization",
}
```

The API response below indicates that the specified authorization **id** was not found. In this case, be sure that the authorization **id** has been created. See Section 4.2, “Add an Authorization” for more information about adding an authorization to a user account.

```
{
    "data": null,
    "messages": [
        {
            "exit_code": 129,
            "field": null,
            "severity": "error",
            "text": "Authorization :id not found"
        }
    ],
    "status": "not_found",
    "type": null,
}
```

Report a bug

4.4. Get Authorization Information

Description

Get information about the specified authorization.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/user/authorizations/*[Auth_ID]*

Request Parameters

Not applicable

Response

The API returns related resource links to manage the specified authorization. See Chapter 4, *Authorizations* for more information on all authorization parameters.

cURL Command Example

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/user/authorizations/[Auth_ID] --user
"[Username]:[Password]"
```

Sample JSON Response

The sample response output below shows information about the specified authorization. The related resource links returned by the API have been left out for brevity.

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/user/authorizations/51ccfe0e2587c8e0b2
0000aa --user "user@myemail.com:password"
{
    "data": {
        "created_at": "2013-07-01T04:23:26Z",
        "expires_in": 8000,
        "expires_in_seconds": 7527,
        "id": "51d1043e2587c881910000af",
        "identity": "user@myemail.com",
        "note": null,
        "scopes": "session",
        "token":
"40a241ed42b73e8a88f5a3a5498d62b0dc01186ea13ce5383f776785ceb9e088"
    },
    "messages": [
        {
            "exit_code": 0,
            "field": null,
            "severity": "info",
            "text": "Display authorization"
        }
    ],
    "status": "ok",
    "type": "authorization",
}
```

The API response below indicates that the specified authorization **id** was not found. In this case, be sure that the authorization **id** has been created. See Section 4.2, “Add an Authorization” for more information about adding an authorization to a user account.

```
{
    "data": null,
    "messages": [
        {
            "exit_code": 129,
            "field": null,
            "severity": "error",
            "text": "Authorization :id not found"
        }
    ],
    "status": "not_found",
    "type": null,
}
```

Report a bug

4.5. Update an Authorization

Description

Allows a client to update an existing authorization for the specified user account. However, currently this operation only supports updating the **note** parameter of an existing authorization.

Method and URL Structure

Method	URL Structure
PUT	https://openshift.redhat.com/broker/rest/user/authorizations/*[Auth_ID]*

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
note	Reminder description of authorization.	String	No	Not applicable	Not applicable

Response

The API returns the authorizations resource with related resource links. See Chapter 4, *Authorizations* for more information on all authorization parameters.

cURL Command Example

```
$ curl -k -X PUT
https://openshift.redhat.com/broker/rest/user/authorizations/[Auth_ID] --user
"[Username]:[Password]" --data "note=[Note_Text]"
```

Sample JSON Response

The sample output below shows the API response after successfully updating the **note** parameter of

the specified authorization. The related resource links and unnecessary information returned by the API have been left out for brevity.

```
$ curl -k -X PUT
https://openshift.redhat.com/broker/rest/user/authorizations/51d1043e2587c8819
10000af --user "user@myemail.com:password" --data "note=My note for
authorization"
{
    "data": {
        "id": "51d1043e2587c881910000af",
        "note": "My note for authorization",
        "scopes": "session",
    },
    "messages": [
        {
            "text": "Change authorization"
        }
    ],
    "status": "ok",
    "type": "authorization",
}
```

Report a bug

4.6. Delete All Authorizations

Description

Allows a client to delete all existing authorizations for the specified user account.

Method and URL Structure

Method	URL Structure
DELETE	https://openshift.redhat.com/broker/rest/user/authorizations

Request Parameters

Not applicable

Response

The API returns confirmation of a successful **DELETE** operation.

cURL Command Example

```
$ curl -k -X DELETE
https://openshift.redhat.com/broker/rest/user/authorizations --user
"[Username]:[Password]"
```

Sample JSON Response

The sample output below shows the API response after successfully deleting all authorizations for the specified user account.

```
$ curl -k -X DELETE
https://openshift.redhat.com/broker/rest/user/authorizations --user
"user@myemail.com:password"
{
    "data": null,
    "errors": {},
    "messages": [
        {
            "exit_code": 0,
            "field": null,
            "severity": "info",
            "text": "All authorizations for 51142f5adbd93ce16a0005b3 are revoked."
        }
    ],
    "status": "ok",
    "type": null,
}
```

Report a bug

4.7. Delete an Authorization

Description

Allows a client to delete the specified authorization. An authorization can be deleted by passing either the ***id*** of the authorization, or the ***token*** string as *[Auth_ID]*.

Method and URL Structure

Method	URL Structure
DELETE	https://openshift.redhat.com/broker/rest/user/authorizations/*[Auth_ID]* or *[token]*

Request Parameters

Not applicable

Response

The API returns confirmation of a successful **DELETE** operation.

cURL Command Example

```
$ curl -k -X DELETE
https://openshift.redhat.com/broker/rest/user/authorizations/[Auth_ID] --user
"[Username]:[Password]"
```

Sample JSON Response

The sample output below shows the API response after successfully deleting the specified authorization.

```
$ curl -k -X DELETE
https://openshift.redhat.com/broker/rest/user/authorizations/51d123eadbd93cb662
000049 --user "user@myemail.com:password"
{
    "data": null,
    "errors": {},
    "messages": [
        {
            "exit_code": 0,
            "field": null,
            "severity": "info",
            "text": "Authorization 51d123eadbd93cb662000049 is revoked."
        }
    ],
    "status": "ok",
    "type": null,
}
```

Report a bug

Chapter 5. Domains

This chapter provides information on API resources to add, update, and manage OpenShift user domains. A domain must be created before an OpenShift application can be created. Domain names on OpenShift are non-strict, meaning there is no preceding period, and form part of the application name. Therefore, the syntax for the application name is ***ApplicationName-DomainName.rhcloud.com***.

Each username can only support a single domain, but multiple applications can be created within a domain. If multiple domains are required, multiple OpenShift accounts using different usernames must be created.

The table below describes each parameter associated with an OpenShift domain.

Name	Description
id	Name of the domain
suffix	Domain suffix

Report a bug

5.1. List Domains

Description

Get a list of all domains for an OpenShift user account.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/domains

Request Parameters

Not applicable

Response

The API returns a list of all domains for the specified user account and related resource links.

cURL Command Example

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/domains --user
"[UserName]:[Password]"
```

Sample JSON Response

The sample response output below shows a list of all domains that exist under the specified user account. The related resource links returned by the API have been left out for brevity.

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/domains --user
"user@myemail.com:password"
{
    "data": {
        "id": "MyDomain",
 "suffix": "rhcloud.com"
```

Report a bug

5.2. Create a Domain

Description

Create a new domain for an OpenShift user account. You must first create a domain before you can create OpenShift applications.

Method and URL Structure

Method	URL Structure
POST	https://openshift.redhat.com/broker/rest/domains

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
id	Name of domain	String	No	Not applicable	Not applicable

Response

The API returns the domain resource with related resource links. See Chapter 5, *Domains* for more information on all domain parameters.

cURL Command Example

```
$ curl -k -X POST https://openshift.redhat.com/broker/rest/domains/ --user
"[Username]:[Password]" --data "id=[Domain_ID]"
```

Sample JSON Response

The sample response output below shows successful creation of the domain named *MyDomain*, and it's suffix. The related resource links returned by the API have been left out for brevity.

```
$ curl -k -X POST https://openshift.redhat.com/broker/rest/domains --user
"user@myemail.com:password" --data "id=MyDomain"
{
    "data": {
        "id": "MyDomain",
 "suffix": "rhcloud.com"
 "status": "created",
```

Report a bug

5.3. Get Domain Information

Description

Get information about an existing domain.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*

Request Parameters

Not applicable

Response

The API returns the domain resource with related resource links. See Chapter 5, *Domains* for more information on all domain resource parameters.

cURL Command Example

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/domains/[Domain_ID]
--user "[UserName]:[Password]"
```

Sample JSON Response

The sample response output below shows information about the domain named *MyDomain* and it's suffix. The related resource links returned by the API have been left out for brevity.

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/domains/MyDomain --
user "user@myemail.com:password"
{
    "data": {
        "id": "MyDomain",
 "suffix": "rhcloud.com"
 "status": "ok",
 "type": "domain",
```

Report a bug

5.4. Update Domain Name

Description

Update the name of an existing domain.

Method and URL Structure

Method	URL Structure
PUT	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
id	Name of domain	String	No	Not applicable	Not applicable

Response

The API returns the domain resource with related resource links. See Chapter 5, *Domains* for more information on all domain parameters.

cURL Command Example

```
$ curl -k -X PUT
https://openshift.redhat.com/broker/rest/domains/[Old_Domain_ID] --user
"[Username]:[Password]" --data "id=[New_Domain_ID]"
```

Sample JSON Response

The sample JSON output below shows the name of the domain has been updated to *MyNewDomain*. The related resource links returned by the API have been left out for brevity.

```
$ curl -k -X PUT https://openshift.redhat.com/broker/rest/domains/MyDomain --
user "user@myemail.com:password" --data "id=MyNewDomain"
{
    "data": {
        "id": "MyNewDomain",
 "suffix": "rhcloud.com"
 "status": "ok",
 "type": "domain",
```

Report a bug

5.5. Delete a Domain

Description

Delete a domain for an OpenShift user account.

Note

The API will exit with an error message if there are applications within the domain you want to delete. You must either delete all applications before you can delete the domain, or set the **`force`** parameter to ***`true`***. Setting the **`force`** parameter to ***`true`*** will automatically delete all applications under that domain, and then delete the domain.

Warning

Deleting a domain with the **`force`** parameter set to ***`true`*** will delete all applications created within that domain. This operation cannot be reversed.

Method and URL Structure

Method	URL Structure
`DELETE`	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value

force	Force delete domain	Boolean	Yes	***true*** or ***false***	***false***

Response

No content is returned from a successful **DELETE** operation.

cURL Command Example

```
$ curl -k -X DELETE
https://openshift.redhat.com/broker/rest/domains/[Domain_ID] --user
"[Username]:[Password]"
```

Sample JSON Response

No content is returned if the domain is deleted successfully. However, if there are applications within the domain you want to delete, the API will exit with an error message, as shown in the sample response output below.

```
$ curl -k -X DELETE https://openshift.redhat.com/broker/rest/domains/MyDomain
--user "user@myemail.com:password"
{
    "status": "bad_request",
    "messages": [
        {
            "severity": "error",
            "text": "Domain contains applications. Delete applications first or set
force to true.",
            "exit_code": 128,
            "field": null
        }
}
```

In this case, you must delete all applications that exist under the domain before you can delete the domain. Alternatively, you can set the **force** parameter to ***true*** to automatically delete the applications as part of the domain delete process. The example below shows how to delete a domain and all applications that exist under that domain.

```
$ curl -k -X DELETE https://openshift.redhat.com/broker/rest/domains/MyDomain
--user "user@myemail.com:password" --data "force=true"
```

Report a bug

Chapter 6. Applications

This chapter provides information on API resources that allow a client to create and manage OpenShift applications. OpenShift supports a number of application frameworks, such as PHP, JBoss, and Ruby.

The table below describes each parameter associated with an OpenShift application.

Name	Description
name	Name of the application.
framework	Application framework. For example, JBoss, PHP, or Ruby.
domain_id	The domain ID of the application.
embedded	List of cartridges that have been added to this application.
creation_time	Time the application was created.
scalable	Whether application is scaled or not scaled. The values are either ***true*** or ***false***.
gear_count	Number of gears for this application.
gear_profile	The gear size for this application. For example, **small** .
aliases	Application server aliases, if applicable.
app_url	The URL to access this application.
git_url	The URL to access the Git repository for this application.
ssh_url	The URL to access this application using an SSH terminal.
health_check_path	The URL to check if the application is running.
uuid	Unique identifier for this application.
initial_git_url	The URL that was used to initialize the Git repository for this application.

Report a bug

6.1. List Applications

Description

List all applications for the specified domain.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications

Request Parameters

Not applicable

cURL Command Example

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications --
user "[UserName]:[Password]"
```

Response

The API returns a list of all applications that exist under the specified domain and related resource links. See Chapter 6, *Applications* for a description of each response parameter associated with an application.

Sample JSON Response

The sample response output below shows a list of all applications that exist under the domain named *MyDomain*. The related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications --user
"user@myemail.com:password"
{
    "data": [
        {
            "aliases": [],
            "app_url": "http://myapp-MyDomain.rhcloud.com/",
            "build_job_url": null,
            "building_app": null,
            "building_with": null,
            "creation_time": "2012-10-15T01:22:32Z",
            "domain_id": "MyDomain",
            "embedded": {},
            "framework": "php-5.3",
            "gear_count": 1,
            "gear_profile": "small",
            "git_url": "ssh://5ec82f67c1404f319d5518423b88ec84@myapp-
MyDomain.rhcloud.com/~/git/myapp.git/",
            "health_check_path": "health_check.php",
            "initial_git_url": "",
            "links": {
                "GET": {
                    "href":
"https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp",
                    "method": "GET",
                    "optional_params": [],
                    "rel": "Get application",
                    "required_params": []
                },
            },
            "name": "myapp",
            "scalable": false,
            "ssh_url": "ssh://5ec82f67c1404f319d5518423b88ec84@myapp-
MyDomain.rhcloud.com",
            "uuid": "5ec82f67c1404f319d5518423b88ec84"
        },
        {
            "aliases": [],
            "app_url": "http://myapp-MyDomain.rhcloud.com/",
            "build_job_url": null,
            "building_app": null,
            "building_with": null,
            "creation_time": "2012-10-18T23:44:21Z",
            "domain_id": "MyDomain",
            "embedded": {
                "haproxy-1.4": {},
                "mysql-5.1": {
                    "connection_url":
"mysql://$OPENSHIFT_MYSQL_DB_HOST:$OPENSHIFT_MYSQL_DB_PORT/",
                    "database_name": "myapp",
                    "password": "zF2MfdIdGdMk",
                    "username": "admin",
                    "info": "Connection URL:
mysql://$OPENSHIFT_MYSQL_DB_HOST:$OPENSHIFT_MYSQL_DB_PORT/"
                }
            },
            "framework": "php-5.3",
            "gear_count": 2,
            "gear_profile": "small",
```

```
            "git_url": "ssh://dfd34495f6ab404e819d2f74ebd4cb50@myapp-
MyDomain.rhcloud.com/~/git/myapp.git/",
            "health_check_path": "health_check.php",
            "initial_git_url": "",
            "links": {
                "GET": {
                    "href":
"https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp",
                    "method": "GET",
                    "optional_params": [],
                    "rel": "Get application",
                    "required_params": []
                },

            },
            "name": "myapp",
            "scalable": true,
            "ssh_url": "ssh://dfd34495f6ab404e819d2f74ebd4cb50@myapp-
MyDomain.rhcloud.com",
            "uuid": "dfd34495f6ab404e819d2f74ebd4cb50"
        }
    ],
    "status": "ok",
    "type": "applications",
}
```


Report a bug

6.2. List Applications and Cartridges

Description

There may be cases where you wish to list all your applications and all embedded cartridges at the same time. In such cases, this REST API call lists all applications and the embedded cartridges.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications

Request Parameters

Not applicable

cURL Command Example

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications?
include=cartridges --user "[UserName]:[Password]"
```

Response

The API returns a list of all applications and embedded cartridges, with related resource links for applications and cartridges. See Chapter 6, *Applications* for a description of each response parameter associated with an application.

Sample JSON Response

The sample response output below shows a list of all applications and embedded cartridges. The related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications?
include=cartridges --user "user@myemail.com:password"
{
    "data": [
        {
            "aliases": [],
            "app_url": "http://myapp-MyDomain.rhcloud.com/",
            "build_job_url": null,
            "building_app": null,
            "building_with": null,
            "cartridges": [
                {
                    "additional_gear_storage": 0,
                    "base_gear_storage": 1,
                    "collocated_with": [],
                    "current_scale": 1,
                    "description": "MySQL is a multi-user, multi-threaded SQL
database server.",
                    "display_name": "MySQL Database 5.1",
                    "gear_profile": "small",
                    "help_topics": {},
                    "license": "ASL 2.0",
                    "license_url": "",
                    "links": {
                    },
                    "name": "mysql-5.1",
                    "scales_from": 1,
                    "scales_to": 1,
                    "scales_with": "haproxy-1.4",
                    "status_messages": null,
                    "supported_scales_from": 1,
                    "supported_scales_to": 1,
                    "tags": [
                        "service",
                        "database",
                        "embedded"
                    ],
                    "type": "embedded",
                    "usage_rates": [],
                    "version": 5.1,
                    "website": "http://www.mysql.com"
                },
                {
                    "additional_gear_storage": 0,
                    "base_gear_storage": 1,
                    "collocated_with": [
                        "php-5.3"
                    ],
                    "current_scale": 1,
                    "description": "Acts as a load balancer for your web cartridge
and will automatically scale up to handle incoming traffic. Is automatically added
to scaled applications when they are created and cannot be removed or added to an
application after the fact.",
                    "display_name": "OpenShift Web Balancer",
                    "gear_profile": "small",
                    "help_topics": {},
                    "license": "unknown",
                    "license_url": "",
                    "links": {
```

```
                    },
                    "name": "haproxy-1.4",
                    "properties": [],
                    "scales_from": 1,
                    "scales_to": 1,
                    "scales_with": "haproxy-1.4",
                    "status_messages": null,
                    "supported_scales_from": 1,
                    "supported_scales_to": 1,
                    "tags": [
                        "web_proxy",
                        "scales",
                        "embedded"
                    ],
                    "type": "embedded",
                    "usage_rates": [],
                    "version": 1.4,
                    "website": "http://haproxy.1wt.eu/"
                },
                {
                    "additional_gear_storage": 0,
                    "base_gear_storage": 1,
                    "collocated_with": [
                        "haproxy-1.4"
                    ],
                    "current_scale": 1,
                    "description": "PHP is a general-purpose server-side
scripting language originally designed for Web development to produce dynamic Web
pages. Popular development frameworks include: CakePHP, Zend, Symfony, and Code
Igniter.",
                    "display_name": "PHP 5.3",
                    "gear_profile": "small",
                    "help_topics": {},
                    "license": "The PHP License, version 3.0",
                    "license_url": "http://www.php.net/license/3_0.txt",
                    "links": {
                    },
                    "name": "php-5.3",
                     "type": "standalone",
                    "usage_rates": [],
                    "version": 5.3,
                    "website": "http://www.php.net"
                }
            ],
            "creation_time": "2012-10-18T23:44:21Z",
            "domain_id": "MyDomain",
            "embedded": {
                "haproxy-1.4": {},
                "mysql-5.1": {
                    "connection_url":
"mysql://$OPENSHIFT_MYSQL_DB_HOST:$OPENSHIFT_MYSQL_DB_PORT/",
                    "database_name": "myapp",
                    "password": "zF2MfdIdGdMk",
                    "username": "admin",
                    "info": "Connection URL:
mysql://$OPENSHIFT_MYSQL_DB_HOST:$OPENSHIFT_MYSQL_DB_PORT/"
                }
            },
            "framework": "php-5.3",
            "gear_count": 2,
```

```
            "gear_profile": "small",
            "git_url": "ssh://dfd34495f6ab404e819d2f74ebd4cb50@myapp-
MyDomain.rhcloud.com/~/git/myapp.git/",
            "health_check_path": "health_check.php",
            "initial_git_url": "",
            "links": {
            },
            "name": "myapp",
            "scalable": false,
            "ssh_url": "ssh://5ec82f67c1404f319d5518423b88ec84@myapp-
MyDomain.rhcloud.com",
            "uuid": "5ec82f67c1404f319d5518423b88ec84"
        }
    ],

    "status": "ok",
}
```

Report a bug

6.3. Check DNS Availability

Description

Check to see if DNS is created with an actual DNS nameserver lookup that is not subject to caching. When you check DNS availability with the client tools, the value gets cached for approximately 30 seconds. This REST API call allows you to check the DNS availability by directly querying the DNS servers.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/dns_resolvable

Request Parameters

Not applicable

cURL Command Example

```
$ curl GET
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/dns_resolvable --user "[UserName]:[Password]"
```

Response

If the DNS is resolvable, the API returns the HTTP status code 200 and the response shown below. If the DNS is not resolvable, the API returns the HTTP status code 404 with an error message.

Sample JSON Response

The sample response output shows successful DNS resolution for the *myapp* application.

```
$ curl GET
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/d
ns_resolvable --user "user@myemail.com:password"
{
    "data": true,
    "messages": [],
    "status": "ok",
    "type": "boolean",
}
```

Report a bug

6.4. Create an Application

Description

Create a new OpenShift application. You must first create a domain before you can create an application. See Section 5.2, “Create a Domain” for more information on how to create a domain.

Method and URL Structure

Method	URL Structure
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
name	Name of application	String	No	Not applicable	Not applicable
cartridge	Framework-type, e.g: php-5.3	Array	No	***nodejs-0.6***; ***jbossas-7***; ***python-2.6***; ***jenkins-1.4***; ***ruby-1.8***; ***diy-0.1***; ***php-5.3***; ***perl-5.10***	Not applicable
scale	Mark application as scalable	Boolean	Yes	***true*** or ***false***	***false***
gear_profile	Size of the gear	String	Yes	***small***; ***medium***	***small***
initial_git_url	URL to Git source repository	String	Yes	Not applicable	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications --
user "[UserName]:[Password]" --data "name=[AppName]&cartridge=php-
5.3&scale=false"
```

Response

The API returns information about the newly created application, with related resource links. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample response output shows successful creation of an OpenShift application. In this example, a PHP application named *myapp* has been created under the domain named *MyDomain*. The related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications --user
"user@myemail.com:password" --data "name=myapp&cartridge=php-5.3&scale=true"
{
    "data": {
        "aliases": [],
        "app_url": "http://myapp-MyDomain.rhcloud.com/",
        "build_job_url": null,
        "building_app": null,
        "building_with": null,
        "creation_time": "2013-04-22T03:12:13Z",
        "domain_id": "MyDomain",
        "embedded": {
            "haproxy-1.4": {}
        },
        "framework": "php-5.3",
        "gear_count": 1,
        "gear_profile": "small",
        "git_url": "ssh://5174aa8ddbd93c117a0001dc@myapp-
MyDomain.rhcloud.com/~/git/myapp.git/",
        "health_check_path": "health_check.php",
        "initial_git_url": null,
        "links": {
            "GET": {
                "href":
"https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp",
                "method": "GET",
                "optional_params": [],
                "rel": "Get application",
                "required_params": []
            },
     "name": "myapp",
        "scalable": true,
        "ssh_url": "ssh://5174aa8ddbd93c117a0001dc@myapp-MyDomain.rhcloud.com",
        "uuid": "5174aa8ddbd93c117a0001dc"
    },
    "errors": {},
    "messages": [
        {
            "exit_code": null,
            "field": null,
            "severity": "info",
            "text": "Application myapp was created."
        },
        {
            "exit_code": 0,
            "field": "current_ip",
            "severity": "info",
            "text": "184.72.206.227"
        },
        {
            "exit_code": 0,
            "field": "result",
            "severity": "info",
            "text": ""
        }
    ],
    "status": "created",
    "type": "application",
}
```

Report a bug

6.5. Get Application Information

Description

Get information about an existing OpenShift application.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*

Request Parameters

Not applicable

cURL Command Example

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name] --user "[UserName]:[Password]"
```

Response

The API returns information about the specified application and related resource links. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample response output shows the information returned by the API for the application named *myapp*. The HAProxy 1.4 cartridge has been added automatically because *myapp* is a scaled application. The related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp -
-user "user@myemail.com:password"
{
    "data": {
        "aliases": [],
        "app_url": "http://myapp-MyDomain.rhcloud.com/",
        "build_job_url": null,
        "building_app": null,
        "building_with": null,
        "creation_time": "2012-10-18T23:44:21Z",
        "domain_id": "MyDomain",
        "embedded": {
            "haproxy-1.4": {},
            "mysql-5.1": {
                "connection_url":
"mysql://$OPENSHIFT_MYSQL_DB_HOST:$OPENSHIFT_MYSQL_DB_PORT/",
                "database_name": "myapp",
                "password": "zF2MfdIdGdMk",
                "username": "admin",
                "info": "Connection URL:
mysql://$OPENSHIFT_MYSQL_DB_HOST:$OPENSHIFT_MYSQL_DB_PORT/"
            }
        },
        "framework": "php-5.3",
        "gear_count": 2,
        "gear_profile": "small",
        "git_url": "ssh://dfd34495f6ab404e819d2f74ebd4cb50@myapp-
MyDomain.rhcloud.com/~/git/myapp.git/",
        "health_check_path": "health_check.php",
        "initial_git_url": "",
 "name": "myapp",
        "scalable": true,
        "ssh_url": "ssh://dfd34495f6ab404e819d2f74ebd4cb50@myapp-
MyDomain.rhcloud.com",
        "uuid": "dfd34495f6ab404e819d2f74ebd4cb50"
    },
    "messages": [],
    "status": "ok",
    "type": "application",
}
```

Report a bug

6.6. Get Application and Cartridge Information

Description

Get information about an existing OpenShift application and it's embedded cartridges.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*

Request Parameters

Not applicable

cURL Command Example

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]?include=cartridges --user "[UserName]:[Password]"
```

Response

The API returns information about the specified application, it's embedded cartridges, and related resource links for applications and cartridges. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample response output shows the information returned by the API for the application named *myapp*. The HAProxy 1.4 cartridge has been added automatically because *myapp* is a scaled application. The related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp -
-user "user@myemail.com:password"
{
    "data": {
        "aliases": [],
        "app_url": "http://myapp-MyDomain.rhcloud.com/",
        "build_job_url": null,
        "building_app": null,
        "building_with": null,
        "creation_time": "2012-10-18T23:44:21Z",
        "domain_id": "MyDomain",
        "embedded": {
            "haproxy-1.4": {
                "info": {}
            },
            "mysql-5.1": {
                "info": {}
            }
        },
        "framework": "php-5.3",
        "gear_count": 3,
        "gear_profile": "small",
        "git_url": "ssh://dfd34495f6ab404e819d2f74ebd4cb50@myapp-
MyDomain.rhcloud.com/~/git/myapp.git/",
        "health_check_path": "health_check.php",
        "initial_git_url": "",
        "name": "myapp",
        "scalable": true,
        "ssh_url": "ssh://dfd34495f6ab404e819d2f74ebd4cb50@myapp-
MyDomain.rhcloud.com",
        "uuid": "dfd34495f6ab404e819d2f74ebd4cb50"
    },
    "messages": [],
    "status": "ok",
}
```

Report a bug

6.7. Start Application

Description

Start an application that is not running.

Method and URL Structure

Method	URL Structure
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/events

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
event	Event	String	No	***start***	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name]/events --user "[UserName]:[Password]" --data "event=start"
```

Response

The API returns the application resource. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample response output below shows the information returned by the API after starting the *myapp* application. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/events --user "user@myemail.com:password" --data "event=start"
{
    "type": "application",
    "status": "ok",
    "messages": [
        {
            "severity": "INFO",
            "field": null,
            "text": "Application myapp has started",
            "exit_code": null
        }
........
```

Report a bug

6.8. Stop Application

Description

Stop a running application.

Method and URL Structure

Method	URL Structure
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/events

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
event	Event	String	No	***stop***	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name]/events --user "[UserName]:[Password]" --data "event=stop"
```

Response

The API returns the application resource. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample response output below shows the information returned by the API after stopping the *myapp* application. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/events --user "user@myemail.com:password" --data "event=stop"
{
    "status": "ok",
    "data": {
    "messages": [
        {
            "field": null,
            "text": "Application myapp has stopped",
            "severity": "INFO",
            "exit_code": null
        }
    ]
}
```

Report a bug

6.9. Force Stop Application

Description

Force a running application to stop.

Method and URL Structure

Method	URL Structure
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/events

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
event	Event	String	No	***force-stop***	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name]/events --user "[UserName]:[Password]" --data "event=force-stop"
```

Response

The API returns the application resource. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample response output below shows the information returned by the API after forcing the *myapp* application to stop. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/events --user "user@myemail.com:password" --data "event=force-stop"
{
    "type": "application",
    "status": "ok",
    "messages": [
        {
            "severity": "INFO",
            "field": null,
            "text": "Application myapp has forcefully stopped",
            "exit_code": null
        }
    ],
}
```

Report a bug

6.10. Restart Application

Description

Restart a running application.

Method and URL Structure

Method	URL Structure
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/events

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
event	Event	String	No	***restart***	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name]/events --user "[UserName]:[Password]" --data "event=restart""
```

Response

The API returns the application resource. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample response output below shows the information returned by the API after restarting the *myapp* application. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/events --user "user@myemail.com:password" --data "event=restart"
{
    "status": "ok",
"messages": [
        {
            "field": null,
            "text": "Application myapp has restarted",
            "severity": "INFO",
            "exit_code": null
        }
    ]
}
```

Report a bug

6.11. Delete Application

Description

Delete an OpenShift application.

Method and URL Structure

Method	Resource URL
DELETE	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*

Request Parameters

Not applicable

cURL Command Example

```
$ curl -k -X DELETE
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name] -user "[UserName]:[Password]"
```

Response

No content is returned from a successful **DELETE** operation.

Sample JSON Response

No content is returned from a successful **DELETE** operation.

Report a bug

6.12. Scale Up Application

Description

Scale up an application that was created with the scaling function enabled.

Method and URL Structure

Method	Resource URL
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/events

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
event	Event	String	No	***scale-up***	Not applicable

cURL Command Syntax

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name]/events --user "[UserName]:[Password]" --data "event=scale-up"
```

Response

The API returns the application resource. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample response output below shows the information returned by the API after the application *myapp* has been scaled up. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/e
vents --user "user@myemail.com:password" --data "event=scale-up"
{
    "type": "application",
    "status": "ok",
     "messages": [
        {
            "severity": "INFO",
            "field": null,
            "text": "Application myapp has scaled up",
            "exit_code": null
        }
}
```

The API returns an error message if a user has reached the maximum number of gears allowed for their account, as shown in the sample response output below.

```
{
    "status": "unprocessable_entity",
    "messages": [
        {
            "field": null,
            "text": "user@myemail.com has already reached the gear limit of 3",
            "severity": "error",
            "exit_code": 104
        }
    ]
}
```

Report a bug

6.13. Scale Down Application

Description

Scale down an application that was created with the scaling function enabled.

Method and URL Structure

Method	Resource URL
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/events

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
event	Event	String	No	***scale-down***	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/events --user "[UserName]:[Password]" --data "event=scale-down"
```

Response

The API returns the application resource. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample response output below shows the information returned by the API after the application *myapp* has been scaled down. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/e
vents --user "user@myemail.com:password" --data "event=scale-down"
{
    "type": "application",
    "status": "ok",
    "messages": [
        {
            "severity": "INFO",
            "field": null,
            "text": "Application myapp has scaled down",
            "exit_code": null
        }
    ],
}
```

The API returns an error message if an application cannot be scaled down any further, as shown in the sample response output below.

```
{
    "status": "unprocessable_entity",
    "messages": [
        {
            "field": null,
            "text": "Failed to add event scale-down to application myapp due to:
Cannot scale below minimum gear requirements for group '1'",
            "severity": "error",
            "exit_code": 1
        }
    ]
}
```

Report a bug

Chapter 7. Application Aliases and SSL Certificates

This chapter provides information on API resources that allow a client to manage application aliases and SSL certificates for OpenShift applications.

The table below describes each parameter associated with an OpenShift application with aliases and SSL certificates added.

Name	Description
id	Name of application alias.
certificate_added_at	The date and time when the SSL certificate was added.
has_private_ssl_certificate	Indicates whether an SSL certificate has been added to the specified application.

Report a bug

7.1. List Application Aliases

Description

List all aliases and SSL certificates associated with the specified application.

Note

A GET request to an application also returns a list of associated aliases for that particular application . See Section 6.5, “Get Application Information” for more information.

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/aliases

Request Parameters

Not applicable

cURL Command Example

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications
/[App_Name]/aliases --user "[UserName]:[Password]"
```

Response

The API returns a list of all aliases and SSL certificates for the specified application with related resource links. See Chapter 7, *Application Aliases and SSL Certificates* for a description of each response parameter associated with application aliases.

Sample JSON Response

The sample response output below shows a list of all aliases and SSL certificates, if any, for the *myapp*

application. The related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/a
liases --user "user@myemail.com:password"
{
    "data": [
        {
            "certificate_added_at": null,
            "has_private_ssl_certificate": false,
            "id": "myalias",
    "messages": [],
    "status": "ok",
    "type": "aliases",
}
```

Report a bug

7.2. Add Application Alias

Description

Add an alias to an application, with the option to upload an SSL certificate. Adding an alias allows you to use your own domain name for your OpenShift application.

Method and URL Structure

Method	Resource URL
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/aliases

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
id	Alias for application	String	No	Not applicable	Not applicable
ssl_certificate	Content of SSL certificate	String	Yes	Not applicable	Not applicable
private_key	Required private key for SSL certificate	String	Yes	Not applicable	Not applicable
pass_phrase	Optional passphrase for private key	String	Yes	Not applicable	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/aliases --user "[UserName]:[Password]" --data "id=[Alias_Name]"
```

When you add an alias, you have the option of uploading an SSL certificate to allow secure HTTPS communication with your application. If you want to add an SSL certificate, include the required private key with or without the optional private key passphrase. To upload an SSL certificate and a private key, simply cut and paste the contents of the SSL certificate and the key.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/aliases --user "[UserName]:[Password]" --data "id=[Alias_Name]" --data-
urlencode "ssl_certificate=[SSL_Cert]" --data-urlencode
"private_key=[Private_Key]" --data "pass_phrase=[Key_Passphrase]"
```

Response

The API returns the aliases resource. See Chapter 7, *Application Aliases and SSL Certificates* for all parameters associated with application aliases.

Sample JSON Response

The sample response output below shows *racer* added as an application alias to the *myapp* application. Because an SSL certificate is being uploaded, the private key must be included. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/aliases --user "[UserName]:[Password]" --data "id=[Alias_Name]" --data-
urlencode "ssl_certificate=-----BEGIN CERTIFICATE----- certificate content ---
--END CERTIFICATE-----" --data-urlencode "private_key=-----BEGIN RSA PRIVATE
KEY----- private key content -----END RSA PRIVATE KEY-----"
{
    "data": {
        "certificate_added_at": "2013-05-16T00:00:00Z",
        "has_private_ssl_certificate": true,
        "id": "racer",
    "errors": {},
    "messages": [
        {
            "exit_code": 0,
            "field": "result",
            "severity": "info",
            "text": ""
        }
    ],
    "status": "created",
    "type": "alias",
}
```

Report a bug

7.3. Update Application Alias

Description

Update an existing application alias with a new SSL certificate, or remove an existing certificate.

Method and URL Structure

Method	Resource URL
PUT	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/aliases/*[Alias_Name]*

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
ssl_certificate	Content of SSL certificate	String	No	Not applicable	Not applicable
private_key	Required private key for SSL certificate	String	No	Not applicable	Not applicable
pass_phrase	Optional passphrase for private key	String	Yes	Not applicable	Not applicable

cURL Command Example

When updating an application alias to upload a new SSL certificate, include the contents of the SSL certificate and the associated private key with or without the optional private key passphrase.

```
$ curl -k -X PUT
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/aliases/[Alias_Name] --user "[UserName]:[Password]" --data-urlencode
"ssl_certificate=[SSL_Cert]" --data-urlencode "private_key=[Private_Key]" --
data "pass_phrase=[Key_Passphrase]"
```

To remove an existing SSL certificate, use the REST call without any parameters. See the cURL command example below.

```
$ curl -k -X PUT
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/aliases/[Alias_Name] --user "[UserName]:[Password]"
```

Response

The API returns the aliases resource. See Chapter 7, *Application Aliases and SSL Certificates* for all parameters associated with application aliases.

Sample JSON Response

The sample response output below shows the existing *racer* application alias has been updated with the new SSL certificate and private key. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X PUT
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/aliases/[Alias_Name] --user "[UserName]:[Password]" --data-urlencode
"ssl_certificate=-----BEGIN CERTIFICATE----- certificate content -----END
CERTIFICATE-----" --data-urlencode "private_key=-----BEGIN RSA PRIVATE KEY----
- private key content -----END RSA PRIVATE KEY-----"
{
    "data": {
        "certificate_added_at": "2013-05-20T00:00:00Z",
        "has_private_ssl_certificate": true,
        "id": "racer",
     "errors": {},
    "messages": [
        {
            "exit_code": null,
            "field": null,
            "severity": "info",
            "text": "Added racer to application myapp"
        },
        {
            "exit_code": 0,
            "field": "result",
            "severity": "info",
            "text": ""
        }
    ],
    "status": "ok",
    "type": "alias",
}
```

The sample response output below shows the SSL certificate has been removed from the *racer* application alias. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X PUT
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/aliases/[Alias_Name] --user "[UserName]:[Password]"
{
    "data": {
        "certificate_added_at": null,
        "has_private_ssl_certificate": false,
        "id": "racer",
    "errors": {},
    "status": "ok",
    "type": "alias",
}
```

Report a bug

7.4. Delete Application Alias

Description

Remove an alias from an application.

Method and URL Structure

Method	Resource URL
DELETE	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/aliases/*[Alias_Name]*

Request Parameters

Not applicable

cURL Command Example

```
$ curl -k -X DELETE
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name]/aliases/[Alias_Name] --user "[UserName]:[Password]"
```

Response

No content is returned from a successful DELETE operation.

Sample JSON Response

No content is returned from a successful DELETE operation.

Report a bug

Chapter 8. Cartridges

This chapter provides information on API resources that allow a client to manage OpenShift cartridges. Cartridges are the components of an OpenShift application, and include databases, build systems, and management capabilities. Adding a cartridge to an application provides the desired capability without requiring you to administer or update the included feature. There are two types of cartridges available with OpenShift: standalone and embedded.

Standalone Cartridges

Standalone cartridges allow you to choose from a variety of programming languages and frameworks for developing your application. Every application requires a framework cartridge. Examples include PHP, JBoss, and Ruby.

The table below describes each parameter associated with both standalone and embedded cartridges.

Name	Description
name	Name of the cartridge.
version	Version of the packaged software in the cartridge.
license	License of the packaged software in the cartridge.
license_url	URL of the license file for the packaged software in the cartridge.
website	Website URL for the packaged software in the cartridge.
help_topics	Map of topics and associated URLs that can provide help on how to use and configure the cartridge.
display_name	Formatted name of the cartridge for user interfaces.
description	Description of the cartridge for user interfaces.
current_scale	Current number of gears used to run the cartridge.
scales_from	Minimum number of gears that a cartridge can scale to; once reached, **scale_down** requests are rejected. Cannot be less than the **supported_scales_from** value.
scales_to	Maximum number of gears that a cartridge can scale to; once reached, **scale_up** requests are rejected. Cannot be greater than the **supported_scales_to** value.
scales_with	Names of other cartridges that scale with this cartridge and run on the same set of gears; currently only HAProxy-1.4 is available.
supported_scales_from	Minimum number of gears supported by the cartridge; the **scales_from** value cannot be less than this number.
supported_scales_to	Maximum number of gears supported by the cartridge; the **scales_to** value cannot be greater than this number.
tags	Array of tags associated with the cartridge.
usage_rates	Plan usage costs.
status_messages	Status messages returned back by the cartridge.

Note

The **scales_from** and **scales_to** parameters can be set separately, or in the same REST API call. If the **scales_from** value is greater than the **scales_to** value, the broker logic automatically interchanges the two values.

Embedded Cartridges

Embedded cartridges provide extended functionality to OpenShift applications. Examples include MySQL, MongoDB, and phpMyAdmin.

The table below describes each parameter associated with only embedded cartridges.

Name	Description
additional_gear_storage	Set additional filesystem storage in gigabytes for the gear profile that the cartridge is running on.
base_gear_storage	Default basic storage in gigabytes associated with the gear profile that the cartridge is running on.
collocated_with	Array of names of other cartridges that share the same gear(s) with the cartridge.
gear_profile	Size of the gears grouped in this profile that the cartridge is running on; all gears in a group will have the same **gear_profile** .
properties	List of environment variables and property values exposed by the cartridge and usable in application code.

Report a bug

8.1. List Cartridges

Description

Get a list of all available OpenShift cartridges.

Method and URL Structure

Method	Resource URL
GET	https://openshift.redhat.com/broker/rest/cartridges

Request Parameters

Not applicable

cURL Command Example

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/cartridges --user
"[UserName]:[Password]"
```

Response

The API returns the cartridge resource with a list of all available OpenShift cartridges. See Chapter 8, *Cartridges* for more information on all cartridge parameters.

Sample JSON Response

The sample JSON output below shows the API response of all available OpenShift cartridges. The example shows information about one standalone cartridge, and one embedded cartridge. Information on other available cartridges has been removed for brevity.

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/cartridges --user
"user@myemail.com:password"
{
    "data": [
        {
            "current_scale": 0,
            "description": "MongoDB is a scalable, high-performance, open source
NoSQL database.",
            "display_name": "MongoDB NoSQL Database 2.2",
            "help_topics": [
                {
                    "Building with MongoDB":
"https://openshift.redhat.com/community/developers/mongodb"
                }
            ],
            "license": "ASL 2.0",
            "license_url": "http://www.apache.org/licenses/LICENSE-2.0.txt",
            "name": "mongodb-2.2",
            "scales_from": 1,
            "scales_to": 1,
            "scales_with": "haproxy-1.4",
            "supported_scales_from": 1,
            "supported_scales_to": 1,
            "tags": [
                "service",
                "database",
                "nosql",
                "embedded"
            ],
            "type": "embedded",
            "usage_rate_usd": 0,
            "version": "2.2.0",
            "website": "http://www.10gen.com"
        },
........
  {
            "current_scale": 0,
            "description": "PHP is a general-purpose server-side scripting
language originally designed for Web development to produce dynamic Web pages.
Popular development frameworks include: CakePHP, Zend, Symfony, and Code
Igniter.",
            "display_name": "PHP 5.3",
            "help_topics": {
                "Developer Center":
"https://openshift.redhat.com/community/developers"
            },
            "license": "The PHP License, version 3.0",
            "license_url": "http://www.php.net/license/3_0.txt",
            "name": "php-5.3",
            "scales_from": 1,
            "scales_to": -1,
            "scales_with": "haproxy-1.4",
            "supported_scales_from": 1,
            "supported_scales_to": -1,
            "tags": [
                "service",
                "php",
                "web_framework"
            ],
            "type": "standalone",
```

```
            "usage_rate_usd": 0,
            "version": "5.3.2",
            "website": "http://www.php.net"
        },
 "messages": [],
 "status": "ok",
 "type": "cartridges",
}
```

Report a bug

8.2. Embedded Cartridges

Embedded cartridges can be added to an OpenShift application to extend functionality. For example, a MySQL cartridge can be added to an application to provide database support.

Report a bug

8.2.1. List Embedded Cartridges

Description

Get a list of embedded cartridges that have been added to an OpenShift application. For scaled applications, the API also returns the scaling properties of the cartridge.

Note

You can also get a list of all embedded cartridges with the REST API call that lists your applications. See Section 6.2, “List Applications and Cartridges” for more information.

Method and URL Structure

Method	Resource URL
GET	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/cartridges

Request Parameters

Not applicable

cURL Command Example

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name]/cartridges --user "[UserName]:[Password]"
```

Response

The API returns a list of all embedded cartridges that have been added to the specified application, and related resource links. See Chapter 8, *Cartridges* for more information on all cartridge parameters.

Sample JSON Response

The sample JSON output below shows the API response. The output shows the PHP cartridge for the

myapp application. Other added cartridges and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/c
artridges --user "user@myemail.com:password"
{
    "data": [
        {
            "additional_gear_storage": 2,
            "base_gear_storage": 1,
            "collocated_with": [
                "haproxy-1.4"
            ],
            "current_scale": 1,
            "description": "PHP is a general-purpose server-side scripting
language originally designed for Web development to produce dynamic Web pages.
Popular development frameworks include: CakePHP, Zend, Symfony, and Code
Igniter.",
            "display_name": "PHP 5.3",
            "gear_profile": "small",
            "help_topics": {},
            "license": "The PHP License, version 3.0",
            "license_url": "http://www.php.net/license/3_0.txt",
             "name": "php-5.3",
             "scales_from": 1,
            "scales_to": -1,
            "scales_with": "haproxy-1.4",
            "status_messages": null,
            "supported_scales_from": 1,
            "supported_scales_to": -1,
            "tags": [
                "service",
                "php",
                "web_framework"
            ],
            "type": "standalone",
            "usage_rates": [],
            "version": 5.3,
            "website": "http://www.php.net"
        }
    ],
    "messages": [],
    "status": "ok",
    "type": "cartridges",
}
```

Report a bug

8.2.2. Add Cartridge

Description

Add an embedded cartridge to an OpenShift application.

Method and URL Structure

Method	Resource URL

POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/cartridges

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
cartridge	Cartridge type, e.g: mysql-5.1	String	No	***mongodb-2.0***; ***cron-1.4***; ***mysql-5.1***; ***postgresql-8.4***; ***haproxy-1.4***; ***10gen-mms-agent-0.1***; ***phpmyadmin-3.4***; ***metrics-0.1***; ***rockmongo-1.1***; ***jenkins-client-1.4***	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name]/cartridges --user "[UserName]:[Password]" --data
"cartridge=[Framework_Type]"
```

Response

The API returns the embedded cartridge resource.

Sample JSON Response

The sample JSON output below shows the API response for adding a *mysql-5.1* cartridge to the *myapp* application. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/c
artridges --user "user@myemail.com:password" --data "cartridge=mysql-5.1"
{
    "type": "cartridge",
    "status": "created",
    "messages": [
        {
            "severity": "info",
            "field": null,
            "text": "Added mysql-5.1 to application myapp",
            "exit_code": null
        },
        {
            "severity": "info",
            "field": "result",
            "text": "\nMySQL 5.1 database added.  Please make note of these
credentials:\n\n   Root User: admin\n   Root Password: 1ZikWPZRMwlq\n   Database
Name: myapp\n\nConnection URL: mysql://adcea5ab1d-
MyDomain.rhcloud.com:41356/\nMySQL gear-local connection URL:
mysql://127.3.64.129:3306/\n\nYou can manage your new MySQL database by also
embedding phpmyadmin-3.4.\n",
            "exit_code": 0
        },
        {
            "severity": "info",
            "field": "appinfo",
            "text": "Connection URL: mysql://adcea5ab1d-
MyDomain.rhcloud.com:41356/\n",
            "exit_code": 0
        }
    ],
    "data": {
        "type": "embedded",
        "tags": [
            "cartridge",
            "database"
        ],
        "license_url": "",
        "links": {
........
"license": "ASL 2.0",
        "website": "http://www.mysql.com",
        "version": 5.1,
        "name": "mysql-5.1",
}
```

Report a bug

8.2.3. Get Cartridge Information

Description

Get information for the specified cartridge.

Method and URL Structure

Method	Resource URL

GET	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/cartridges/*[Cartridge_Name]*

Request Parameters

Not applicable

cURL Command Syntax

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/cartridges/[Cartridge_Name] --user "[UserName]:[Password]"
```

Response

The API returns the embedded cartridge resource and related resource links.

Sample JSON Response

The sample JSON output below shows the API response. In this example, the API returns information about the *mysql-5.1* cartridge added to the *myapp* application. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/c
artridges/mysql-5.1 --user "user@myemail.com:password"
{
    "data": {
        "additional_gear_storage": 0,
        "base_gear_storage": 1,
        "collocated_with": [],
        "current_scale": 1,
        "description": "MySQL is a multi-user, multi-threaded SQL database
server.",
        "display_name": "MySQL Database 5.1",
        "gear_profile": "small",
        "help_topics": {},
        "license": "ASL 2.0",
        "license_url": "",
        "name": "mysql-5.1",
        "scales_from": 1,
        "scales_to": 1,
        "scales_with": "haproxy-1.4",
        "status_messages": null,
        "supported_scales_from": 1,
        "supported_scales_to": 1,
        "tags": [
            "service",
            "database",
            "embedded"
        ],
        "type": "embedded",
        "usage_rate_usd": 0,
        "version": 5.1,
        "website": "http://www.mysql.com"
    },
}
```

Report a bug

8.2.4. Get Cartridge Status

Description

Retrieve the **status_messages** string as an array for the specified cartridge in an application.

Method and URL Structure

Method	Resource URL
GET	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/cartridges/*[Cartridge_Name]*

Request Parameters

Not applicable

cURL Command Syntax

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/cartridges/[Cartridge_Name]?include=status_messages --user
"[UserName]:[Password]"
```

Response

The API returns the **status_messages** string as an array for the specified cartridge in an application.

Sample JSON Response

The sample JSON output below shows the API response for the *mysql-5.1* cartridge added to the *myapp* application. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/c
artridges/mysql-5.1?include=status_messages --user "user@myemail.com:password"
"status_messages": [
            {
                "gear_id": "51142f5adbd93ce16a0005c4",
                "message": "MySQL is running\n"
            }
        ],
```

Report a bug

8.2.5. Get Gear Groups

Description

Get a list of gear groups where each group represents a list of gears that share scaling and storage policies. Gears that are part of a group are of the same size, and have the same scaling limit in terms of maximum number of gears. A gear group is automatically created when you create a scaled application,

or when you add a cartridge to a scaled application. The API also returns a list of all cartridges belonging to that particular gear group. For an application that is not scalable, only one gear group gets created.

Note

You can also view scaling and storage policies for scaled cartridges as detailed in Section 8.2.1, "List Embedded Cartridges".

Method and URL Structure

Method	URL Structure
GET	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications*[App_Name]*/gear_groups

Request Parameters

Not applicable

cURL Command Example

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name]/gear_groups --user "[UserName]:[Password]"
```

Response

The API returns a list of all gear groups for the specified user. See Chapter 8, *Cartridges* for more information on all cartridge parameters.

Note

If you set the **scales_from** value higher than the **max_gear** limit for your subscription plan, you will get an error message. This is because OpenShift will attempt to scale up to the **scales_from** value. If your application demands more resources, and you require a higher minimum scaling limit that exceeds your current subscription plan, you must upgrade your subscription plan. Log onto your account at www.openshift.com, and go to the **Your Plan** page to view your upgrade options.

Sample JSON Response

The sample response output shows the **gear_groups** information returned by the API for the application named *myapp*.

```
$ curl -k -X GET
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/g
ear_groups --user "user@myemail.com:password"
{
    "data": [
        {
            "additional_gear_storage": 0,
            "base_gear_storage": 1,
            "cartridges": [
                {
                    "database_name": "myapp",
                    "username": "admin",
                    "password": "zF2MfdIdGdMk",
                    "connection_url": "mysql://b8fb6dafd3-
MyDomain.rhcloud.com:62251/",
                    "name": "mysql-5.1",
                    "display_name": "MySQL Database 5.1",
                    "tags": [
                        "service",
                        "database",
                        "embedded"
                    ]
                }
            ],
            "gear_profile": "small",
            "gears": [
                {
                    "id": "b8fb6dafd3e74eb299d5a91e03c92537",
                    "state": "idle"
                }
            ],
            "name": "51142f5adbd93ce16a0005c2",
            "scales_from": null,
            "scales_to": null,
            "uuid": "51142f5adbd93ce16a0005c2"
        },
        {
            "additional_gear_storage": 0,
            "base_gear_storage": 1,
            "cartridges": [
                {
                    "name": "haproxy-1.4",
                    "display_name": "OpenShift Web Balancer",
                    "tags": [
                        "web_proxy",
                        "scales",
                        "embedded"
                    ]
                },
                {
                    "name": "php-5.3",
                    "display_name": "PHP 5.3",
                    "tags": [
                        "service",
                        "php",
                        "web_framework"
                    ]
                }
            ],
            "gear_profile": "small",
```

```
            "gears": [
                {
                    "id": "dfd34495f6ab404e819d2f74ebd4cb50",
                    "state": "idle"
                },
                {
                    "id": "3600019c904a4574b930aca80136c211",
                    "state": "idle"
                }
            ],
            "name": "51142f5adbd93ce16a0005c6",
            "scales_from": 1,
            "scales_to": -1,
            "uuid": "51142f5adbd93ce16a0005c6"
        }
    ],
    "messages": [],
    "status": "ok",
}
```


Report a bug

8.2.6. Start Cartridge

Description

Start an application's embedded cartridge that is not running.

Method and URL Structure

Method	Resource URL
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/cartridges/*Cartridge_Name*/events

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
event	Event	String	No	***start***	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App_Name]/cartridges/[Cartridge_Name]/events --user "[UserName]:[Password]" --data "event=start"
```

Response

The API returns the application resource. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample JSON output below shows the API response to stop the *mysql-5.1* cartridge for the *myapp* application. Unnecessary information and related resource links returned by the API have been removed

for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/c
artridges/mysql-5.1/events --user "user@myemail.com:password" --data
"event=start"
{
    "status": "ok",
    "messages": [
        {
            "field": null,
            "text": "Added start on mysql-5.1 for application myapp",
            "severity": "info",
            "exit_code": null
        }
    ]
}
```

Report a bug

8.2.7. Stop Cartridge

Description

Stop an application's embedded cartridge that is running.

Method and URL Structure

Method	Resource URL
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/cartridges/*Cartridge_Name*/events

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
event	Event	String	No	***stop***	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/cartridges/[Cartridge_Name]/events --user "[UserName]:[Password]" --data
"event=stop"
```

Response

The API returns the application resource. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample JSON output below shows the API response to stop the *mysql-5.1* cartridge for the *myapp* application. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/c
artridges/mysql-5.1/events --user "user@myemail.com:password" --data
"event=stop"
{
    "status": "ok",
    "messages": [
        {
            "field": null,
            "text": "Added stop on mysql-5.1 for application myapp",
            "severity": "info",
            "exit_code": null
        }
    ]
}
```

Report a bug

8.2.8. Restart Cartridge

Description

Restart a running embedded cartridge.

Method and URL Structure

Method	Resource URL
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/cartridges/*Cartridge_Name*/events

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
event	Event	String	No	***restart***	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/cartridges/[Cartridge_Name]/events --user "[UserName]:[Password]" --data
"event=restart"
```

Response

The API returns the application resource. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample JSON output below shows the API response to restart the *mysql-5.1* cartridge for the *myapp* application. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/c
artridges/mysql-5.1/events --user "user@myemail.com:password" --data
"event=restart"
{
    "messages": [
        {
            "severity": "info",
            "field": null,
            "text": "Added restart on mysql-5.1 for application myapp",
            "exit_code": null
        }
    ],
}
```

Report a bug

8.2.9. Reload Cartridge

Description

Reload the embedded cartridge configuration if it has been modified.

Method and URL Structure

Method	Resource URL
POST	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/cartridges/*Cartridge_Name*/events

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
event	Event	String	No	***reload***	Not applicable

cURL Command Example

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/cartridges/[Cartridge_Name]/events --user "[UserName]:[Password]" --data
"event=reload"
```

Response

The API returns the application resource. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample JSON output below shows the API response to reload the *mysql-5.1* cartridge for the *myapp* application. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/c
artridges/mysql-5.1/events --user "user@myemail.com:password" --data
"event=reload"
{
    "status": "ok",
    "messages": [
        {
            "field": null,
            "text": "Added reload on mysql-5.1 for application myapp",
            "severity": "info",
            "exit_code": null
        }
    ]
}
```

Report a bug

8.2.10. Update Cartridge Configuration

Description

Update the configuration of an existing cartridge. You can set the minimum and maximum scaling factors, and set additional filesystem storage in gigabytes on each gear that contains the specified cartridge.

Note

You must be on an upgraded plan to configure additional gear storage. If you are using OpenShift Online, go to *www.openshift.com* and log into your account to view plan upgrade options. If you are using OpenShift Enterprise, contact your system administrator for more information.

Method and URL Structure

Method	Resource URL
PUT	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/cartridges/*[Cartridge_Name]*

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
additional_gear_storage	Additional filesystem storage (GB)	Integer	Yes	Not applicable	Not applicable
scales_from	Min number of gears	Integer	Yes	Not applicable	Not applicable
scales_to	Max number of gears	Integer	Yes	Not applicable	Not applicable

cURL Command Syntax

```
$ curl -k -X PUT
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/cartridges/[Cartridge_Name] --user "[UserName]:[Password]" --data
"[parameter_name]=[value]"
```

Response

The API returns the embedded cartridge resource with related resource links, and the updated values for the parameters you changed. See Chapter 8, *Cartridges* for more information on all cartridge parameters.

Sample JSON Response

The sample JSON output below shows the API response where the additional gear storage has been increased to 3GB. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X PUT
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/c
artridges/mysql-5.1 --user "user@myemail.com:password" --data
"additional_gear_storage=3"
{
    "data": {
        "additional_gear_storage": 3,
        "base_gear_storage": 1,
        "collocated_with": [
            "haproxy-1.4"
        ],
        "current_scale": 1,
        "display_name": "PHP 5.3",
        "gear_profile": "small",
        "help_topics": {},
        "license": "The PHP License, version 3.0",
        "license_url": "http://www.php.net/license/3_0.txt",
        "name": "php-5.3",
        "scales_from": 1,
        "scales_to": 3,
        "scales_with": "haproxy-1.4",
        "status_messages": null,
        "supported_scales_from": 1,
        "supported_scales_to": 3,
        "tags": [
            "service",
            "php",
            "web_framework"
        ],
        "type": "standalone",
        "usage_rates": [],
        "version": 5.3,
        "website": "http://www.php.net"
    },
    "errors": {},
    "messages": [],
    "status": "ok",
    "type": "cartridge",
}
```

The API returns an error message if your plan does not allow additional gear storage, as shown in the

example below.

```
$ curl -k -X PUT
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/c
artridges/php-5.3 --user "user@myemail.com:password" --data
"additional_gear_storage=2"
{
    "data": null,
    "errors": {},
    "messages": [
        {
            "exit_code": 164,
            "field": null,
            "severity": "error",
            "text": "User is not allowed to change storage quota"
        }
    ],
    "status": "unprocessable_entity",
}
```

Report a bug

8.2.11. Delete Cartridge

Description

Delete an embedded cartridge from an application.

Method and URL Structure

Method	Resource URL
DELETE	https://openshift.redhat.com/broker/rest/domains/*[Domain_ID]*/applications/*[App_Name]*/cartridges/*Cartridge_Name*

Request Parameters

Not applicable

cURL Command Example

```
$ curl -k -X DELETE
https://openshift.redhat.com/broker/rest/domains/[Domain_ID]/applications/[App
_Name]/cartridges/[Cartridge_Name] --user "[UserName]:[Password]"
```

Response

The API returns the application resource. See Chapter 6, *Applications* for more information on all application parameters.

Sample JSON Response

The sample JSON output below shows the API response to delete the *mysql-5.1* cartridge from the *myapp* application. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X DELETE
https://openshift.redhat.com/broker/rest/domains/MyDomain/applications/myapp/c
artridges/mysql-5.1 --user "user@myemail.com:password"
{
    "messages": [
        {
            "severity": "info",
            "field": null,
            "text": "Removed mysql-5.1 from application myapp",
            "exit_code": null
        }
    ],
}
```

Report a bug

Chapter 9. Quickstarts

This chapter provides information on API resources for OpenShift quickstart applications. Quickstarts provide quick access to new technology with code and libraries preconfigured, but you are responsible for updating the core libraries for security updates.

The table below describes each parameter associated with an OpenShift quickstart application.

Name	Description
id	Unique identifier of the quickstart.
search	The search term to use for the quickstart.

Report a bug

9.1. List Quickstarts

Description

The **LIST_QUICKSTARTS** resource lists all available quickstarts. The client will only see this resource if there are quickstarts available, and it will be absent if there are none. Unlike other REST API calls, the following guidelines apply when retrieving a list of quickstarts:

- API versioning is not supported
- Only JSON is supported
- The body of the API response is different from other API responses
- Parse errors or unexpected data values must be handled by omitting the entry

Method and URL Structure

Method	URL Structure
GET	https://www.openshift.redhat.com/api/v1/quickstarts/promoted.json

Request Parameters

Not applicable

Response

The API returns a list of all quickstarts that are available. See Chapter 9, *Quickstarts* for more information on all quickstart parameters.

cURL Command Example

```
$ curl -k -X GET https://www.openshift.com/api/v1/quickstarts/promoted.json --
user "[Username]:[Password]"
```

Sample JSON Response

The sample response output below shows an example of one quickstart from the list of available quickstarts. The remaining quickstarts have been left out for brevity.

```
$ curl -k -X GET https://www.openshift.com/api/v1/quickstarts/promoted.json --
user "user@myemail.com:password"
{
    "data": [
        {
            "quickstart": {
                "id": "13145",
                "href": "https://www.stg.openshift.com/quickstarts/drupal-7",
                "name": "Drupal 7",
                "updated": "1365011911",
                "summary": "An open source content management platform written in
PHP powering millions of websites and applications. It is built, used, and
supported by an active and diverse community of people around the world. This
quickstart will download and install the most recent stable version of Drupal and
then generate a new site for you.  Your administrative username and password will
default to admin/openshift_changeme, so don't forget to alter them once you log
in!\n\nWithout sharing a filesystem, Drupal can't be web scaled, but the README.md
describes a workaround that will allow you to scale if you don't need direct file
upload into Drupal.\n\nCreating this quickstart may take several minutes.  You may
need to restart the application once the database is configured. NOTE: If you want
to run the Drupal cron tasks, please install the cron cartridge.",
                "body": "<p>An open source content management platform written in
PHP powering millions of websites and applications. It is built, used, and
supported by an active and diverse community of people around the world. This
quickstart will download and install the most recent stable version of Drupal and
then generate a new site for you.  Your administrative username and password will
default to admin/openshift_changeme, so don't forget to alter them once you log
in!</p>\n\n<p>Without sharing a filesystem, Drupal can't be web scaled, but the <a
href=\"https://github.com/openshift/drupal-
quickstart/blob/master/README.md\">README.md</a> describes a workaround that will
allow you to scale if you don't need direct file upload into
Drupal.</p>\n\n<p>Creating this quickstart may take several minutes.  You may need
to restart the application once the database is configured. NOTE: If you want to run
the Drupal cron tasks, please install the cron cartridge.</p>",
                "cartridges": "php-*, mysql-*",
                "website": "http://drupal.org/",
                "tags": "cms, drupal, instant_app, not_scalable, php",
                "language": "PHP",
                "initial_git_url": "https://github.com/openshift/drupal-
quickstart.git",
                "provider": "openshift"
            }
        },
    ]
}
```

Report a bug

9.2. Show Quickstart

Description

The **SHOW_QUICKSTART** resource allows a client that knows the unique identifier of a resource to directly substitute the id of a quickstart for the replacement variable **:id** in the resource URL. The client does not have to retrieve the quickstarts list and scan for a known id.

Method and URL Structure

Method	URL Structure
GET	https://www.openshift.redhat.com/api/v1/quickstarts/:id

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
id	Unique identifier of the quickstart	String	No	Not applicable	Not applicable

Response

The API returns information about the quickstart identified with the ***:id*** parameter.

cURL Command Example

```
$ curl -k -X GET https://www.openshift.com/api/v1/quickstarts/:id --user
"[Username]:[Password]"
```

Sample JSON Response

The sample response output below shows information about the quickstart retrieved using the ***:id*** parameter.

```
$ curl -k -X GET https://www.openshift.com/api/v1/quickstarts/12724 --user
"user@myemail.com:password"
{
    "data": [
        {
            "quickstart": {
                "id": "12724",
                "href": "https://www.stg.openshift.com/quickstarts/wordpress-3x",
                "name": "WordPress 3.x",
                "updated": "1365011887",
                "summary": "A semantic personal publishing platform written in PHP
with a MySQL back end, focusing on aesthetics, web standards, and usability.
Currently using version 3.5.1.\n\nThe first time you access the app you'll be asked
to set a username and password and give your blog a name.  Be sure to track
security updates from upstream.",
                "body": "<p>A semantic personal publishing platform written in PHP
with a MySQL back end, focusing on aesthetics, web standards, and usability.
Currently using version 3.5.1.</p>\n\n<p>The first time you access the app you'll
be asked to set a username and password and give your blog a name.  Be sure to
track security updates from upstream.</p>",
                "cartridges": "php-*, mysql-*",
                "website": "http://wordpress.org",
                "tags": "blog, cms, instant_app, not_scalable",
                "language": "PHP",
                "initial_git_url": "git://github.com/openshift/wordpress-
example.git",
                "provider": "openshift"
            }
        }
    ]
}
```

Report a bug

9.3. Search Quickstarts

Description

The **SEARCH_QUICKSTARTS** resource allows a client to search for a quickstart using a search term.

Method and URL Structure

Method	URL Structure
GET	https://www.openshift.redhat.com/api/v1/quickstarts.json?search=*[Search_Term]*

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
search	Search term to use for the quickstart.	String	No	Not applicable	Not applicable

Response

The API returns information about all quickstarts that match the specified search string. See Chapter 9, *Quickstarts* for more information on all quickstart parameters.

cURL Command Example

```
$ curl -k -X GET https://www.openshift.com/api/v1/quickstarts.json?
search=[Search_Term] --user "[Username]:[Password]"
```

Sample JSON Response

The sample response output below shows the API response using the search term *wordpress*.

```
$ curl -k -X GET https://www.openshift.com/api/v1/quickstarts.json?
search=wordpress --user "user@myemail.com:password"
{
    "data": [
        {
            "quickstart": {
                "id": "12724",
                "href": "https://www.stg.openshift.com/quickstarts/wordpress-3x",
                "name": "WordPress 3.x",
                "updated": "1365011887",
                "summary": "A semantic personal publishing platform written in PHP
with a MySQL back end, focusing on aesthetics, web standards, and usability.
Currently using version 3.5.1.\n\nThe first time you access the app you'll be asked
to set a username and password and give your blog a name.  Be sure to track
security updates from upstream.",
                "body": "<p>A semantic personal publishing platform written in PHP
with a MySQL back end, focusing on aesthetics, web standards, and usability.
Currently using version 3.5.1.</p>\n\n<p>The first time you access the app you'll
be asked to set a username and password and give your blog a name.  Be sure to
track security updates from upstream.</p>",
                "cartridges": "php-*, mysql-*",
                "website": "http://wordpress.org",
                "tags": "blog, cms, instant_app, not_scalable",
                "language": "PHP",
                "initial_git_url": "git://github.com/openshift/wordpress-
example.git",
                "provider": "openshift"
            }
        }
    ]
}
```

Report a bug

Chapter 10. User Information

This chapter provides information on API resources that allow a client to manage OpenShift user account information.

The table below describes each parameter associated with a user account.

Name	Description
capabilities	Map of user capabilities. See the table below for user capabilities.
consumed_gears	Total number of gears consumed by all applications owned by user.
login	Account user name.
max_gears	Maximum number of gears available to the specified user.
plan_id	Subscription plan of the specified user.
plan_state	State of the account for the specified user.

The table below further describes each available capability for the user.

Name	Description
plan_upgrade_enabled	Indicates whether the user is on an upgraded plan.
subaccounts	Indicates whether the user has the ability to create subaccounts.
gear_sizes	Available gear sizes depending on the type of plan.
max_storage_per_gear	Maximum storage in gigabytes available per gear to the specified user.
private_ssl_certificates	Subscription plan of the specified user.

Report a bug

10.1. View User Information

Description

Provides resource links to view user information, and manage user SSH keys.

Method and URL Structure

Method	Resource URL
GET	https://openshift.redhat.com/broker/rest/user

Request Parameters

N/A

cURL Command Example

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/user --user
"[UserName]:[Password]"
```

Response

The API returns the user information resource. See Chapter 10, *User Information* for more information on all user information parameters.

Sample JSON Response

The sample JSON output below shows the API response. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/user --user
"user@myemail.com:password"
{
    "data": {
        "capabilities": {
            "subaccounts": false,
            "gear_sizes": [
                "small",
                "medium"
            ],
            "plan_upgrade_enabled": true,
            "max_storage_per_gear": 10
        },
        "consumed_gears": 3,
        "created_at": "2013-02-07T22:48:58Z",
        "id": "51142f5adbd93ce16a0005b3",
         "login": "user@myemail.com",
        "max_gears": 50,
        "plan_id": "free",
        "plan_state": "ACTIVE",
        "usage_account_id": null
    },
    "messages": [],
    "status": "ok",
}
```

Report a bug

10.2. About Parent and Child Accounts

Parent accounts that have the subaccounts capability enabled can contain child user accounts. See Chapter 10, *User Information* for more information on all user account parameters and capabilities.

Report a bug

10.2.1. Delete Child Account

Description

Delete the specified child user account, if one exists.

Note

Parent accounts cannot be deleted. Therefore, the API returns an error message if you attempt to delete a parent account.

Method and URL Structure

Method	Resource URL

DELETE	https://openshift.redhat.com/broker/rest/user

Request Parameters

N/A

cURL Command Example

```
$ curl -k -X DELETE https://openshift.redhat.com/broker/rest/user -d --user
"[Child-UserName]:[Password]"
```

Response

No content is returned from a successful **DELETE** operation.

Sample JSON Response

No content is returned from a successful **DELETE** operation.

Report a bug

Chapter 11. Subscription Plans

This chapter provides information on API resources that allow a client to view and modify OpenShift subscription plans.

The table below describes each parameter associated with a subscription plan.

Name	Description
id	Plan identifier that can be used to get more information about that particular plan. Currently the **free** and **silver** plans are offered.
capabilities	Map of user capabilities. See the table below for plan capabilities.
name	Plan name.
plan_no	Plan number associated with the plan name, and is the third party billing provider plan number.
usage_rates	Plan usage costs.

The table below further describes each available capability based on the subscription plan.

Name	Description
subaccounts	Indicates whether the user has the ability to create subaccounts.
max_gears	Maximum number of available gears for the subscribed plan.
gear_sizes	Available gear sizes depending on the type of plan.
max_untracked_addtl_storage_per_gear	Maximum additional filesystem storage per gear that will NOT be tracked for the user.
max_tracked_addtl_storage_per_gear	Maximum additional filesystem storage per gear that will be tracked for the user.
plan_upgrade_enabled	Indicates whether the user is on an upgraded plan.
private_ssl_certificates	Subscription plan of the specified user.

Report a bug

11.1. List Plans

Description

Get a list of all supported subscription plans with details of each plan. Each plan is priced differently and provides certain capabilities that are assigned when you subscribe to a particular plan. Currently the free and silver plans are offered, with each offering different capabilities that determine what type of resources are available to you. Go to *www.openshift.com* for more information on available subscription plans.

Method and URL Structure

Method	Resource URL
GET	https://openshift.redhat.com/broker/rest/plans

Request Parameters

N/A

cURL Command Example

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/plans
```

Response

The API returns the plans resource. See Chapter 11, *Subscription Plans* for more information on all plan parameters.

Sample JSON Response

The sample JSON output below shows the API response. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/plans
{
    "data": [
        {
            "capabilities": {
                "subaccounts": false,
                "max_gears": 3,
                "gear_sizes": [
                    "small"
                ],
                "plan_upgrade_enabled": true
            },
            "id": "free",
            "name": "Free",
            "plan_no": 10332742,
            "usage_rates": null
        },
        {
            "capabilities": {
                "subaccounts": false,
                "max_gears": 16,
                "gear_sizes": [
                    "small",
                    "medium"
                ],
                "max_untracked_addtl_storage_per_gear": 5,
                "max_tracked_addtl_storage_per_gear": 0,
                "plan_upgrade_enabled": true,
                "private_ssl_certificates": true
            },
            "id": "silver",
            "name": "Silver",
            "plan_no": 10332743,
            "usage_rates": {
                "gear": {
                    "small": {
                        "usd": 0.04,
                        "cad": 0.04,
                        "eur": 0.03,
                        "duration": "hour"
                    },
                    "medium": {
                        "usd": 0.1,
                        "cad": 0.1,
                        "eur": 0.08,
                        "duration": "hour"
                    }
                },
                "storage": {
                    "gigabyte": {
                        "usd": 0,
                        "cad": 0,
                        "eur": 0,
                        "duration": "month"
                    }
                },
                "cartridge": {
                    "jbosseap-6.0": {
                        "usd": 0.03,
                        "cad": 0.03,
```

```
                        "eur": 0.02,
                        "duration": "hour"
                    }
                }
            }
        }
    ],
    "messages": [],
    "status": "ok",
    "type": "plans",
}
```


Report a bug

11.2. Get Plan Information

Description

Get information about a specific plan with the plan identifier.

Method and URL Structure

Method	Resource URL
GET	https://openshift.redhat.com/broker/rest/plans/*[Plan_ID]*

Request Parameters

N/A

cURL Command Example

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/plans/[Plan_ID]
```

Response

The API returns detailed information about the specified plan. See Chapter 11, *Subscription Plans* for more information on all plan parameters.

Sample JSON Response

The sample JSON output below shows the API response for the **silver** plan, and shows resources and capabilities available to the specified user.

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/plans/silver
{
    "data": {
        "capabilities": {
            "subaccounts": false,
            "max_gears": 16,
            "gear_sizes": [
                "small",
                "medium"
            ],
            "max_untracked_addtl_storage_per_gear": 5,
            "max_tracked_addtl_storage_per_gear": 0,
            "plan_upgrade_enabled": true,
            "private_ssl_certificates": true
        },
        "id": "silver",
        "name": "Silver",
        "plan_no": 10332743,
        "usage_rates": {
            "gear": {
                "small": {
                    "usd": 0.04,
                    "cad": 0.04,
                    "eur": 0.03,
                    "duration": "hour"
                },
                "medium": {
                    "usd": 0.1,
                    "cad": 0.1,
                    "eur": 0.08,
                    "duration": "hour"
                }
            },
            "storage": {
                "gigabyte": {
                    "usd": 0,
                    "cad": 0,
                    "eur": 0,
                    "duration": "month"
                }
            },
            "cartridge": {
                "jbosseap-6.0": {
                    "usd": 0.03,
                    "cad": 0.03,
                    "eur": 0.02,
                    "duration": "hour"
                }
            }
        }
    },
    "messages": [],
    "status": "ok",
    "type": "plan",
}
```

Report a bug

11.3. Change Plan

Description

Change to a different subscription plan. Regardless of what plan you are currently on, all capabilities of the new plan become available after you execute this REST API call.

Note

An upgraded OpenShift Online account is required to upgrade your account to the **silver** plan with this REST API call. Go to *www.openshift.com* for more information on available subscription plans.

Method and URL Structure

Method	Resource URL
PUT	https://openshift.redhat.com/broker/rest/user

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
plan_id	Plan identifier; for example, **silver**	String	No	**free** or **silver**	Not applicable

cURL Command Example

```
$ curl -k -X PUT https://openshift.redhat.com/broker/rest/user --user
"[UserName]:[Password]" --data "plan_id=[Plan_ID]"
```

Response

The API returns the user information resource and related resource links. See Chapter 10, *User Information* for more information on all user information parameters.

Sample JSON Response

The sample JSON output below shows the API response when a plan is changed to **silver** plan.

```
$ curl -k -X PUT https://openshift.redhat.com/broker/rest/user --user
"user@myemail.com:password" --data "plan_id=silver"
{
    "data": {
        "capabilities": {
            "plan_upgrade_enabled": true,
            "subaccounts": false,
            "gear_sizes": [
                "small",
                "medium"
            ],
            "private_ssl_certificates": true,
            "max_storage_per_gear": 5
        },
        "consumed_gears": 1,
        "created_at": "2013-05-16T23:43:25Z",
        "id": "51956f1d686f90a138000001",
        "login": "user@myemail.com",
        "max_gears": 16,
        "plan_id": "silver",
        "plan_state": "ACTIVE",
        "usage_account_id": "2540629"
    },
    "errors": {},
    "messages": [],
    "status": "ok",
    "type": "account",
}
```

Report a bug

Chapter 12. SSH Keys

This chapter provides information on API resources that allow a client to view and manage SSH keys.

The table below describes each parameter associated with SSH keys.

Name	Description
name	Name of the SSH key as specific by the user.
content	Content of the public SSH key.
type	Type of SSH key; for example, RSA or DSA. See Section A.1, “Types of SSH Keys Supported”.

Report a bug

12.1. List User SSH Keys

Description

Get a list of SSH keys for an OpenShift user account.

Method and URL Structure

Method	Resource URL
GET	https://openshift.redhat.com/broker/rest/user/keys

Request Parameters

N/A

cURL Command Example

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/user/keys --user
"[UserName]:[Password]"
```

Response

The API returns the key resource. See Chapter 12, *SSH Keys* for more information on all SSH key parameters.

Sample JSON Response

The sample JSON output below shows the API response to get a list of all SSH keys for the specified user account. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/user/keys --user
"user@myemail.com:password"
{
    "data": [
        {
            "content":
"AAAAB3NzaC1yc2EAAAABIwAAAQEA14PDPWsaZMDspZNK7ABsppzwy++Ih2tRwjBkxzC2KEcQi7v8IcyOD
b7qLJ72tgx3G90zRm7vQ6wuyy7rkYLIvTYiDncI4THYUsve7wuBuSCgFcHLUdon7xn7VrskjhMN4git6bja
Y1+o4Knpfm3N72+9q/6+T52QIWCE1+Ku6UYYuOGy8qTgoijy24bp4jGEKAXqTXcALuBoukC3uB+xujhZYH1
fbek6aEAQPYzO6sGqJqV1UoF0ascelzMbDJA4XOrKPr/5uJsPS+kGZguU16ykQb2k9K03TYHfvPP4rLe50
Q9G4dSZFbUOQXdC3n13CuioEVzizUGl0HyT8MhRqw==",
            "name": "mysshkey",
            "type": "ssh-rsa"
        }
    ],
    "status": "ok",
    "type": "keys",
}
```

Report a bug

12.2. Add SSH Key

Description

Add an SSH key to the specified user account.

Method and URL Structure

Method	Resource URL
POST	https://openshift.redhat.com/broker/rest/user/keys

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
name	Name of key	String	No	Not applicable	Not applicable
type	Type of SSH key	String	No	See Section A.1, "Types of SSH Keys Supported".	Not applicable
content	The key portion (excluding ssh-rsa and comment)	String	No	Not applicable	Not applicable

cURL Command Example

```
$ curl -k -X POST https://openshift.redhat.com/broker/rest/user/keys --user
"[UserName]:[Password]" --data "name=[Key_Name]&type=[Type_of_Key] --data-
urlencode "content=[Key_Content]"
```

Response

The API returns the key resource. See Chapter 12, *SSH Keys* for more information on all SSH key parameters.

Sample JSON Response

The sample JSON output below shows the API response for adding an SSH key to a user account. In this example, an SSH key named *mysshkey* has been added to a user account. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X POST https://openshift.redhat.com/broker/rest/user/keys --user
"user@myemail.com:password" --data "name=mysshkey&type=ssh-rsa" --data-
urlencode
"content=AAAAB3NzaC1yc2EAAAABIwAAAQEA14PDPWsaZMDspZNK7ABsppzwy++Ih2tRwjBkxzC2K
EcQi7v8IcyODb7qLJ72tgx3G90zRm7vQ6wuyy7rkYLIvTYiDncI4FEBSsve7wuBuSCgFcHLUdon7xn
7VrskjhMN4pae6bjaY1+o4Knpfm3N72+9q/6+T52QIWCE1+Ku6UYYuOGy8qWynddw24bp4jGEKAXqT
XcALuBoukC3uB+hrxvZYH1fbek6aEAQPYzO6sGqJqV1UoF0ascelzMbDJA4XOrKPr/5uJsPS+kGZg
uU16ykQb2k9K03JMSfvPP4rLe50Q9G4dSZFbUOQXdC3n13CqvsEVzizUGl0HyT8MhRqw=="
{
    "status": "ok",
    "data":
        {
             "type": "ssh-rsa",
            "content":
"AAAAB3NzaC1yc2EAAAABIwAAAQEA14PDPWsaZMDspZNK7ABsppzwy++Ih2tRwjBkxzC2KEcQi7v8IcyOD
b7qLJ72tgx3G90zRm7vQ6wuyy7rkYLIvTYiDncI4FEBSsve7wuBuSCgFcHLUdon7xn7VrskjhMN4pae6bja
Y1+o4Knpfm3N72+9q/6+T52QIWCE1+Ku6UYYuOGy8qWynddw24bp4jGEKAXqTXcALuBoukC3uB+hrxvZYH1
fbek6aEAQPYzO6sGqJqV1UoF0ascelzMbDJA4XOrKPr/5uJsPS+kGZguU16ykQb2k9K03JMSfvPP4rLe50
Q9G4dSZFbUOQXdC3n13CqvsEVzizUGl0HyT8MhRqw==",
            "name": "mysshkey"
        }
    ],
    "type": "keys",
}
```

Report a bug

12.3. Get SSH Key Information

Description

View the contents of an SSH key.

Method and URL Structure

Method	Resource URL
GET	https://openshift.redhat.com/broker/rest/user/keys/*[Key_Name]*

Request Parameters

N/A

cURL Command Example

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/user/keys/[Key_Name]
--user "[UserName]:[Password]"
```

Response

The API returns the key resource. See Chapter 12, *SSH Keys* for more information on all SSH key parameters.

Sample JSON Response

The sample JSON output below shows the API response to get information about an SSH key named *mysshkey*. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X GET https://openshift.redhat.com/broker/rest/user/keys/mysshkey --
user "user@myemail.com:password"
{
    "status": "ok",
    "data": {
 "type": "ssh-rsa",
        "content":
"AAAAB3NzaC1yc2EAAAABIwAAAQEA14PDPWsaZMDspZNK7ABsppzwy++Ih2tRwjBkxzC2KEcQi7v8IcyOD
b7qLJ72tgx3G90zRm7vQ6wuyy7rkYLIvTYiDncI4FEBSsve7wuBuSCgFcHLUdon7xn7VrskjhMN4pae6bja
Y1+o4Knpfm3N72+9q/6+T52QIWCE1+Ku6UYYuOGy8qWynddw24bp4jGEKAXqTXcALuBoukC3uB+hrxvZYH1
fbek6aEAQPYzO6sGqJqV1UoF0ascelzMbDJA4XOrKPr/5uJsPS+kGZguU16ykQb2k9K03JMSfvPP4rLe50
Q9G4dSZFbUOQXdC3n13CqvsEVzizUGl0HyT8MhRqw==",
        "name": "mysshkey"
    },
    "type": "key",
}
```

Report a bug

12.4. Update SSH Key

Description

Update the contents of an existing SSH key.

Method and URL Structure

Method	Resource URL
PUT	https://openshift.redhat.com/broker/rest/user/keys/*[Key_Name]*

Request Parameters

Name	Description	Type	Optional	Valid Options	Default Value
type	Type of key	String	No	See Section A.1, "Types of SSH Keys Supported".	Not applicable
content	The key portion of an rsa key (excluding ssh-rsa and comment)	String	No	Not applicable	Not applicable

cURL Command Example

```
$ curl -k -X PUT https://openshift.redhat.com/broker/rest/user/keys/[Key_Name]
--user "[UserName]:[Password]" --data "type=[Type_of_Key]" --data-urlencode
"content=[Key_Content]"
```

Note

It is recommended to URL encode the key contents because it may contain non alphanumeric characters.

Response

The API returns the key resource. See Chapter 12, *SSH Keys* for more information on all SSH key parameters.

Sample JSON Response

The sample JSON output below shows the API response for updating an SSH key. In this example, the contents of the SSH key named *mysshkey* have been successfully updated. Unnecessary information and related resource links returned by the API have been removed for brevity.

```
$ curl -k -X PUT https://openshift.redhat.com/broker/rest/user/keys/mysshkey --
user "user@myemail.com:password" --data "type=ssh-rsa" --data-urlencode
"content=AAAAB3NzaC1yc2EAAAADAQABAAABAQDDpC8I/85+IGS6jPsbmKDDbMKPi52+hbUGoyw1A
jFs3V+bIcABWUe3I2i/Lrr4BOYwHLUXOztSHFKbzcXdDGcNBo3buUznHbK5ta6NmlUyVAnW8aYEyWl
lZirW3mkEBdEfl43EMvsUgpGZ0+KhcE3XtjiUroFSHJmiSFYYQSZEGYYlCtF+tB7XUv8aU0bs+VPs
VJBHBIbIh1bURC8Nt9flLaOjWwLkCfyeiiPOrLW3UQNDpU1U1f3638iMWfVHdgWGYXspatt9nKMb8g
JNuvo1GtzSzvd6sQg0xY9DHTDgv2aMpIEK/Dd3xPUoB438PN/y2woydDI8GD76mF6RhIdp"
{
    "type": "key",
    "data": {
        "type": "ssh-rsa",
        "content":
"AAAAB3NzaC1yc2EAAAADAQABAAABAQDDpC8I/85+IGS6jPsbmKDDbMKPi52+hbUGoyw1AjFs3V+bIcAB
WUe3I2i/Lrr4BOYwHLUXOztSHFKbzcXdDGcNBo3buUznHbK5ta6NmlUyVAnW8aYEyWllZirW3mkEBdEfl
43EMvsUgpGZ0+KhcE3XtjiUroFSHJmiSFYYQSZEGYYlCtF+tB7XUv8aU0bs+VPsVJBHBIbIh1bURC8Nt9f
lLaOjWwLkCfyeiiPOrLW3UQNDpU1U1f3638iMWfVHdgWGYXspatt9nKMb8gJNuvo1GtzSzvd6sQg0xY9DH
TDgv2aMpIEK/Dd3xPUoB438PN/y2woydDI8GD76mF6RhIdp",
        "name": "mysshkey",
 "status": "ok",
    "messages": [
        {
            "field": null,
            "exit_code": null,
            "text": "Updated SSH key with name mysshkey for user
user@myemail.com",
            "severity": "info"
        }
    ],
}
```

Report a bug

12.5. Delete SSH Key

Description

Delete an SSH key from a user account.

Method and URL Structure

Method	Resource URL
DELETE	https://openshift.redhat.com/broker/rest/user/keys/*[Key_Name]*

Request Parameters

N/A

cURL Command Syntax

```
$ curl -k -X DELETE
https://openshift.redhat.com/broker/rest/user/keys/[Key_Name] --user
"[UserName]:[Password]"
```

Response

No content is returned from a successful **DELETE** operation.

Sample JSON Response

No content is returned from a successful **DELETE** operation.

However, if the key to be deleted does not exist, the API returns an error message, as shown in the sample JSON output below.

```
$ curl -k -X DELETE https://openshift.redhat.com/broker/rest/user/keys/mysshkey
--user "user@myemail.com:password"
{
    "status": "not_found",
    "type": null,
    "messages": [
        {
            "severity": "error",
            "exit_code": 118,
            "field": null,
            "text": "SSH key not found"
        }
    ],
}
```

Report a bug

Supported SSH Keys

A.1. Types of SSH Keys Supported

The table below lists all SSH key types that OpenShift currently supports.

ssh-rsa
ssh-dss
ecdsa-sha2-nistp256-cert-v01@openssh.com
ecdsa-sha2-nistp384-cert-v01@openssh.com
ecdsa-sha2-nistp521-cert-v01@openssh.com
ssh-rsa-cert-v01@openssh.com
ssh-dss-cert-v01@openssh.com
ssh-rsa-cert-v00@openssh.com
ssh-dss-cert-v00@openssh.com
ecdsa-sha2-nistp256
ecdsa-sha2-nistp384
ecdsa-sha2-nistp521

Go back to Chapter 12, *SSH Keys*

Go back to Section 12.2, “Add SSH Key”

Go back to Section 12.4, “Update SSH Key”

Report a bug

Revision History

Revision 1.0.31-0	**Tue Aug 06 2013**	**Bilhar Aulakh**

Add Authorizations resource.
Add Quickstarts resource.
BZ 987665: Fix URL in cURL examples in Add SSH Key section.

Revision 1.0.29-3	**Thu Jun 13 2013**	**Bilhar Aulakh**

Publish book for OpenShift Online.