Hewlett Packard
Enterprise

HPE Codar

Software Version: 1.60

API and CLI Reference

Document Release Date: January 2016
Software Release Date: January 2016

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HPE required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: **https://softwaresupport.hp.com**

This site requires that you register for an HPE Passport and sign in. To register for an HPE Passport ID, go to: **https://hpp12.passport.hp.com/hppcf/createuser.do**

Or click the **the Register** link at the top of the HPE Software Support page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

Support

Visit the HPE Software Support Online web site at: **https://softwaresupport.hp.com**

This web site provides contact information and details about the products, services, and support offered by HPE Software.

HPE Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as a Passport user and sign in. Many also require a support contract. To register for a Passport ID, go to:

https://hpp12.passport.hp.com/hppcf/createuser.do

To find more information about access levels, go to:

https://softwaresupport.hp.com/web/softwaresupport/access-levels

HPE Software Solutions Now accesses the HPSW Solution and Integration Portal web site. This site enables you to explore HPE product solutions to meet your business needs, includes a full list of integrations between HPE products, as well as a listing of ITIL processes. The URL for this web site is **http://h20230.www2.hp.com/sc/solutions/index.jsp**

Contents

Introduction

Welcome to the HPE Codar API and CLI Reference Guide.

Codar

Codar provides a model driven approach to DevOps. It facilitates continuous delivery where every change to a system is releasable and every code change can be deployed in production. Codar enables automation of continuous delivery. In Codar every code change triggers a build. This build is deployed, tested, and deployed to an environment automatically based on policies. Elements of the core Codar value proposition include the following:

- Modeling the application and infrastructure to deliver infrastructure-as-code (IaaC) for fulfillment
- Managing applications, versions, and packages across different life cycle stages
- Policy management for promotion of application packages through the life cycle stages on different environments consistently in a repeatable manner

For more information about Codar, see the *HP Codar Concepts Guide*.

Codar APIs

Codar provides a REST (REpresentational State Transfer) Application Programming Interface (API) that allows you to interact programmatically with Codar functionality.

This API Reference Guide is designed to help you learn the API by introducing the Codar capabilities that can be accessed programmatically,explaining how to access them, and walking through several examples. This guide assumes that you:

- Understand how to use a RESTful API.
- Have installed and set up Codar following the instructions in the *Codar Installation and Configuration Guide*.

Codar CLI

The Codar command line interface (CLI) commands provide access to Codar REST end points. You can perform all Codar specific functionality from your local machine using the CLI.

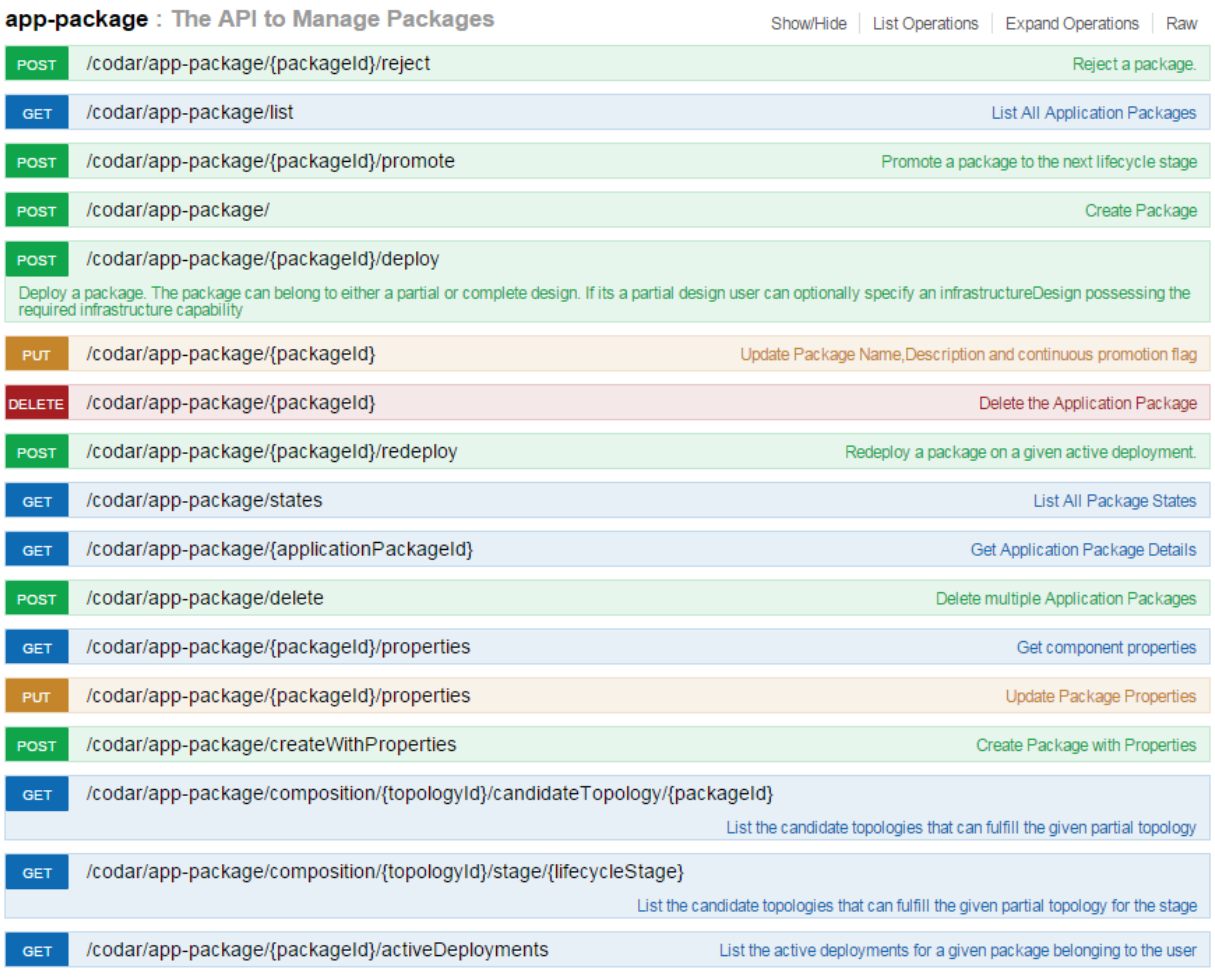
Use the `--help` option to view a list of available commands, and to read documentation about the functionality and options available for each command.

See "Codar CLI set up" on page 110 for more information on how to set up your system to run the Codar CLI commands.

Codar API documentation overview

You can exercise the Codar RESTful API calls programmatically and from an HTTP client. You can find more information on HTTP clients in "Retrieve information from Codar using RESTful calls" on page 13. Codar also includes a framework for describing, producing, consuming and exercising these RESTful services. In addition to providing reference API documentation, this content provides an interactive environment where you can exercise API calls and view responses from your server.

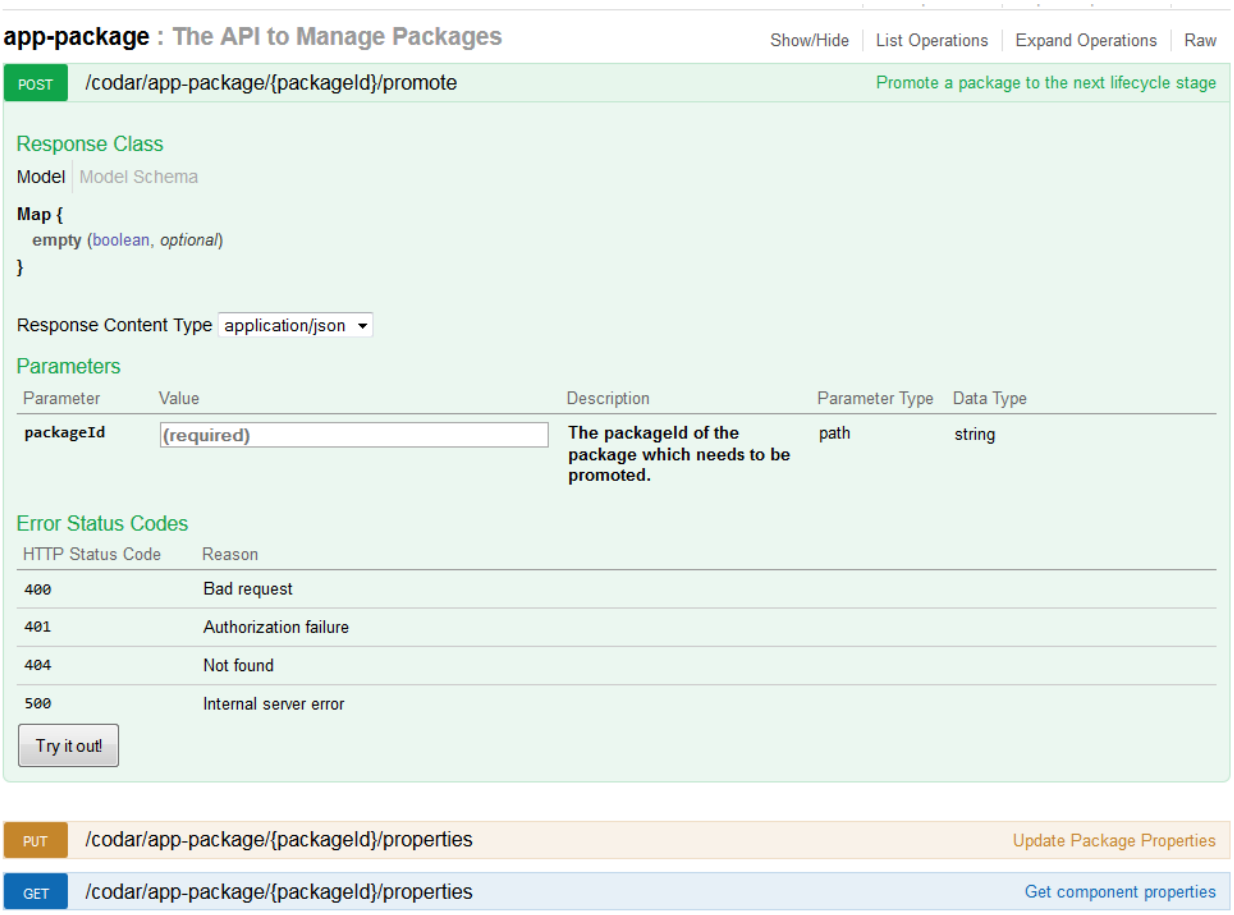
The following is an example of content for an API call:



You can click on any method listed to get detailed content for that call, and an interactive "Try it out!" feature. The detailed content provides developers quick and easy access the following for each API call:

- Syntax
- Method (operation)
- Parameter descriptions
- Model and schema information
- Interactive "Try it out!" dialogs that let you submit calls to your server, see the results, and then understand how you need to construct REST URLs and any payloads.

The following is an example of the detailed content for an API call:



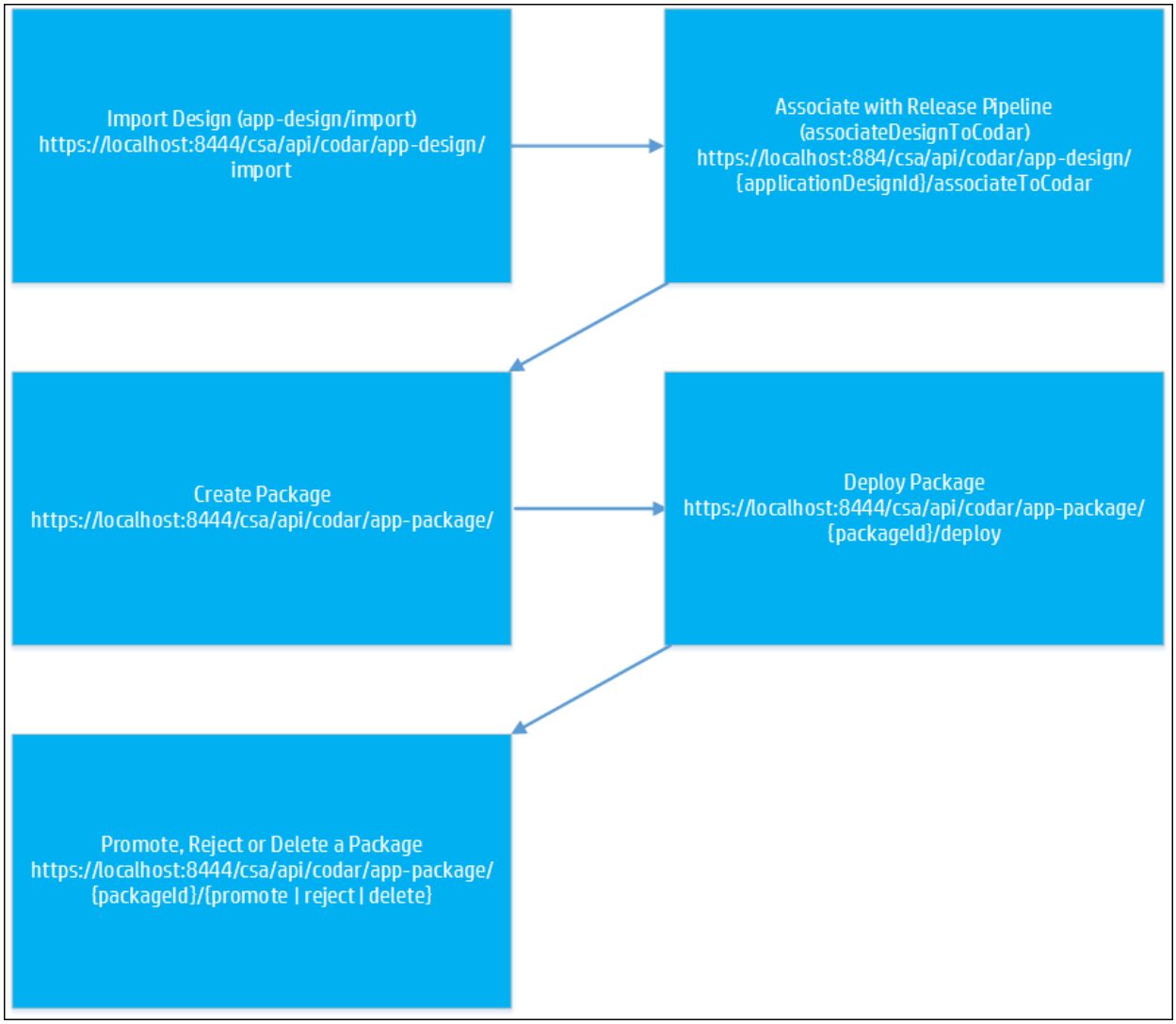
See "Retrieve information from Codar using RESTful calls" on page 13 for more information on executing an API call from this interactive content.

Flow of an API call example

The following steps describe an example flow using Codar APIs:

1. Use the "Import an application design" API to create an application design by providing the application design in JSON format as input. If the design already exists, an error is returned.
2. Use the "Associate a design to Codar" API to tag an application design with the Release Pipeline. This associates a design to the Codar pipeline management process.
3. Use the "Create a package" API to create a package for the application design. The package is only created if the Codar license is present and the design is associated to the Release Pipeline.
4. Once the package is created, use the "Deploy a package" API. You can also promote or reject packages using the APIs.

The following graphic describes this example flow:


Import Design (app-design/import)
https://localhost:8444/csa/api/codar/app-design/import
Associate with Release Pipeline (associateDesignToCodar)
https://localhost:884/csa/api/codar/app-design/{applicationDesignId}/associateToCodar
Create Package
https://localhost:8444/csa/api/codar/app-package/
Deploy Package
https://localhost:8444/csa/api/codar/app-package/{packageId}/deploy
Promote, Reject or Delete a Package
https://localhost:8444/csa/api/codar/app-package/{packageId}/{promote | reject | delete}

Retrieve information from Codar using RESTful calls

The following sections provide information about using Codar RESTful API calls.

Communication with Codar

Solution developers communicate with Codar over HTTP or HTTPS and parse the data structures returned by Codar. The default port for communication with Codar is port 8444.

API data is returned in JSON format. You need to set the HTTP headers `Content-Type:application/json` and `accept:application/json` for the call you are making.

Authentication with an integration account

Authentication is handled using HTTP basic authentication. The authentication value is provided in the API call via the Authorization header. The Codar Integration User role is used for integrating third party tools like Jenkins to access Codar for pipeline management. From the Organization tab of the Codar console, you can configure the user access control for Codar Integration User. Users of this role have permission to create, update, deploy, promote, delete, and reject packages.

Users of this role are commonly used in Jenkins integration to create packages and execute release gate actions during continuous package promotion. Administrator has to associate this role to the first stage in pipeline management to ensure packages are created and release gate actions are executed

When exercising an API call from an HTTP client or from the interactive API content, the Authorization header is automatically generated using the credentials you supply when you log into Codar as it will be required to get permission to make RESTful calls.

When exercising an API call from your application's code, you must create the Authorization header. Supply the Base64 encoded value of the <username>:<password> string for the user you will use for authentication.

Executing RESTful calls

Codar RESTful API calls can be exercised through an HTTP client as described in "Exercising API calls using an HTTP client" on the next page or programmatically from an application. The API calls can also be exercised through the interactive content as explained in "Exercising API calls using the interactive content" on page 16. Some calls have more strict permission requirements, such as administrator level permissions, so you must supply the appropriate credentials.

The base URL for the API is **https://<host>:<port>/csa/api/**, which is appended with the specific URI for the API call. For example, to access the API for managing application deployments, you would use the URL: `https://<host>:<port>/csa/api/codar/app-deploy`, substituting the host and port information appropriate for your Codar environment.

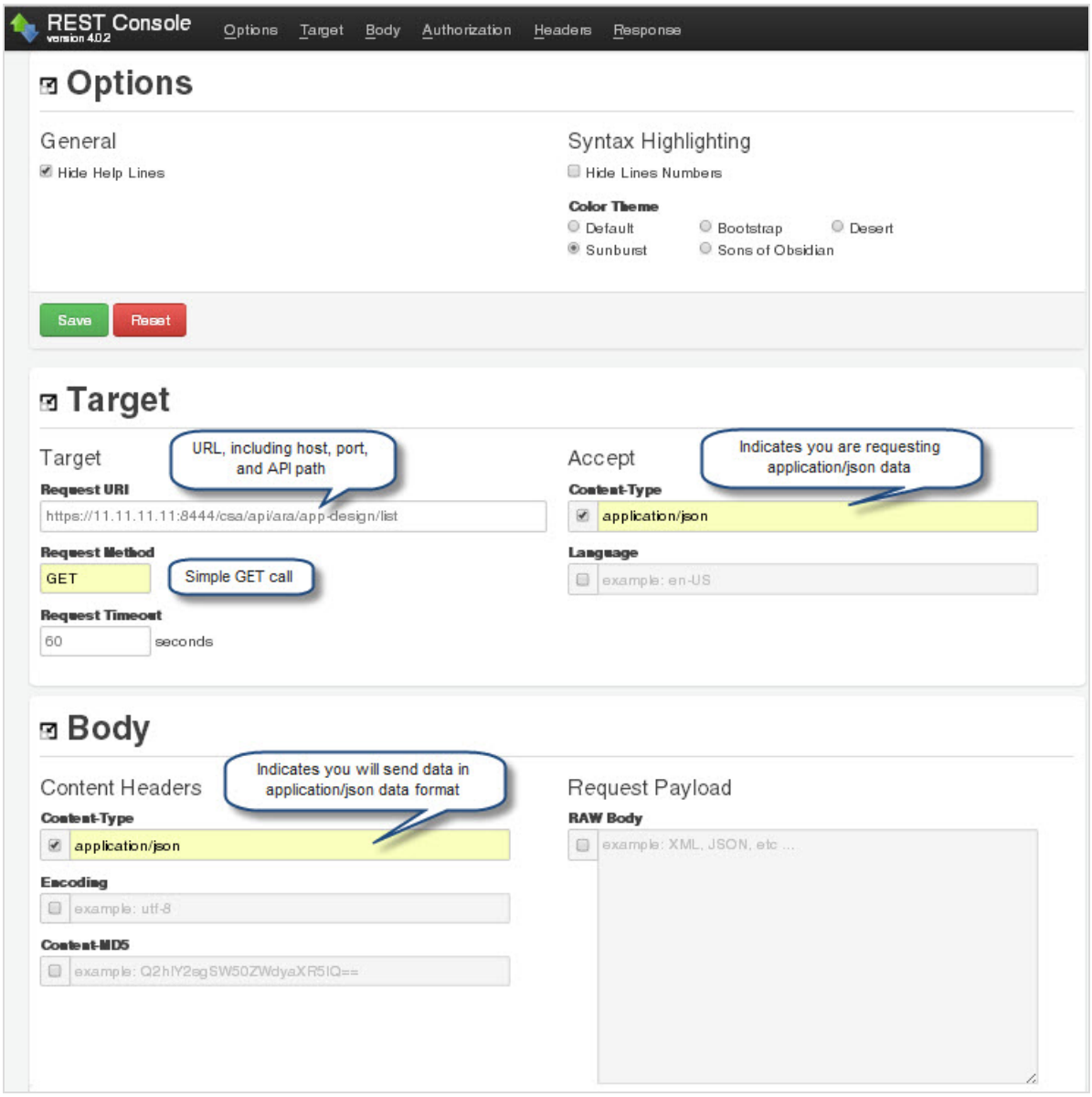
Exercising API calls using an HTTP client

Though you can issue RESTful calls through any typical HTTP client (browser), you will likely find it more convenient to use a client designed especially for developers making RESTful calls. These are often referred to as *REST clients*. A REST client organizes the information you will work with when making RESTful calls: headers, methods, request and response bodies, and so on. A REST client makes it easier to compose and submit requests to the Codar RESTful service APIs, as well as for viewing server responses.

A number of REST clients are available. You can add the REST Console plug-in for Google™ Chrome, for example, as follows:

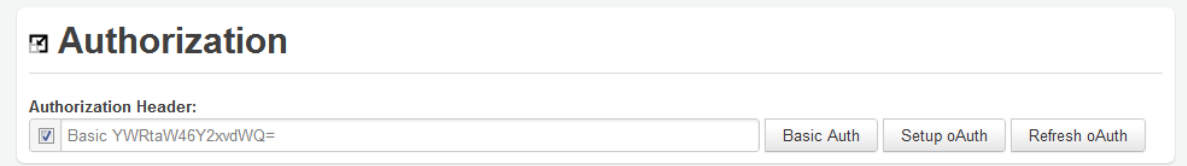
1. Start Google Chrome.
2. Open Chrome Web Store.
3. Use search box to search for REST Console.
4. Click the **+ Free** button, then click the **Add** button in the dialog box that appears.

Example configuration of REST Console plug-in:

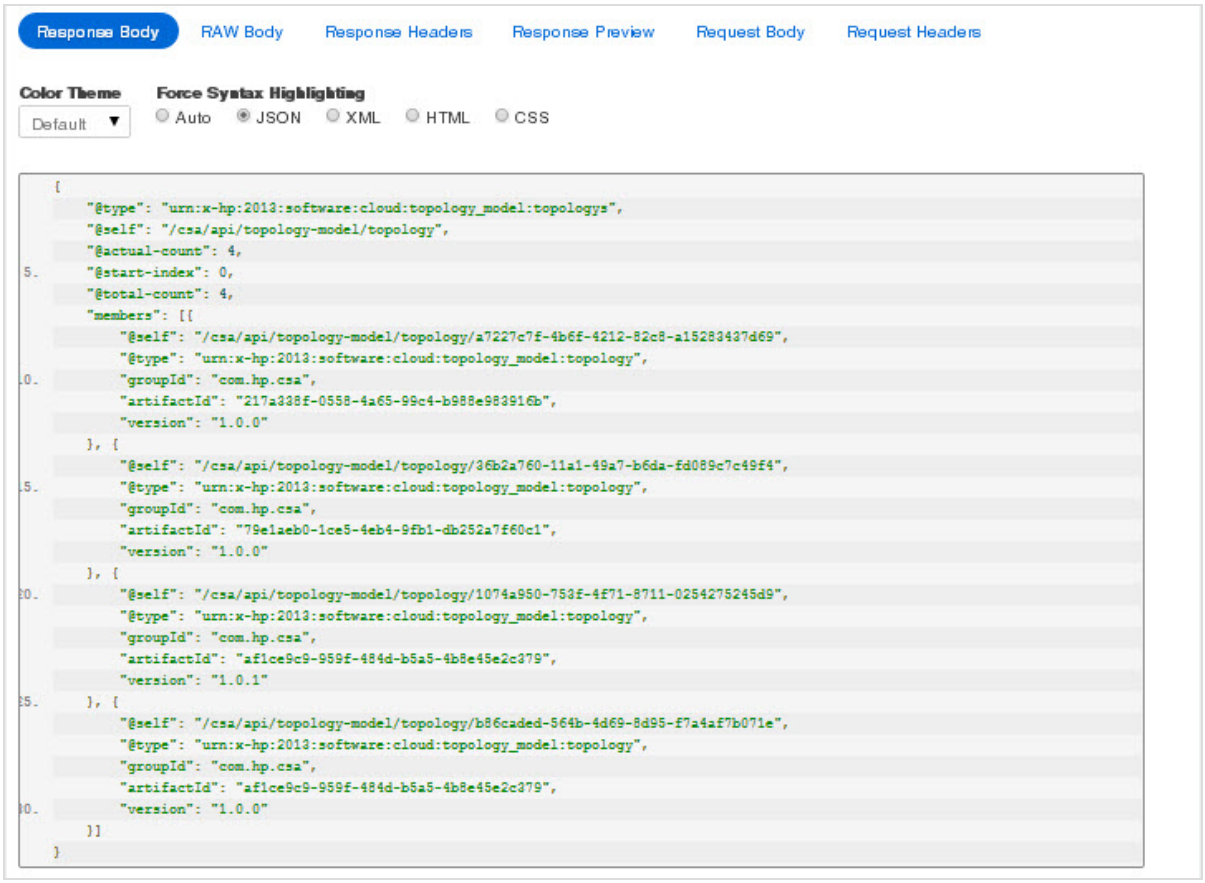


Configure the basic authorization in the Authorization tab using a username and password. See "Authentication with an integration account" on page 13 for more information. This user must have privileges to access Codar's RESTful APIs.

Example REST Console authorization:



If the plug-in is configured correctly, you will see output in the Response Body tab similar to the following when you issue a request:

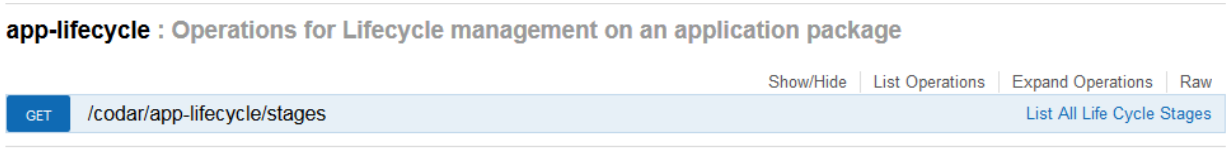


Exercising API calls using the interactive content

In addition to being exercised programmatically or through an HTTP client, the Codar API can be exercised through the “Try it out!” feature in the interactive API content. You will be prompted to log in to Codar to access this content. Basic authentication and authorization required to make RESTful calls will be configured as part of the login process.

The interactive content is presented in a web interface, and can be accessed at **https://<host>:<port>/csa/apidocs.jsp** from a browser, substituting the host and port information appropriate for your Codar environment.

The following is example content. The app-lifecycle call has been expanded in this example:



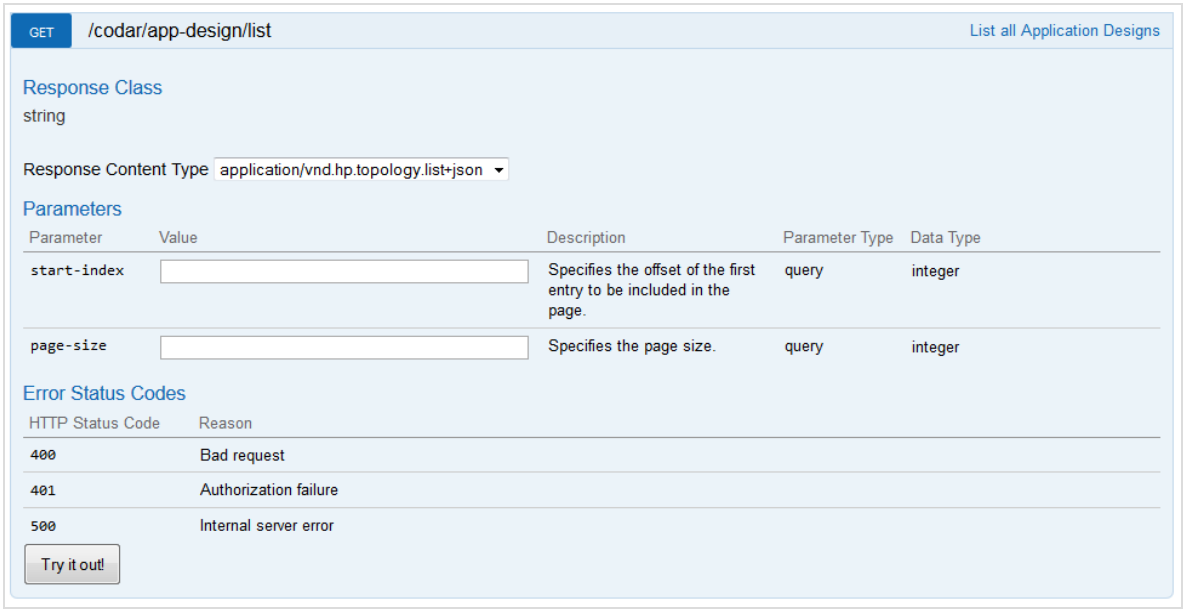
The content provides developers easy access to the following items:

- URI syntax for each call
- Required or optional query parameters

- The data type of each parameter
- Model and schema information, if applicable
- Interactive "Try it out!" dialogs that let you submit calls to your server, see the results, and understand how you need to construct your REST URIs and any request body.

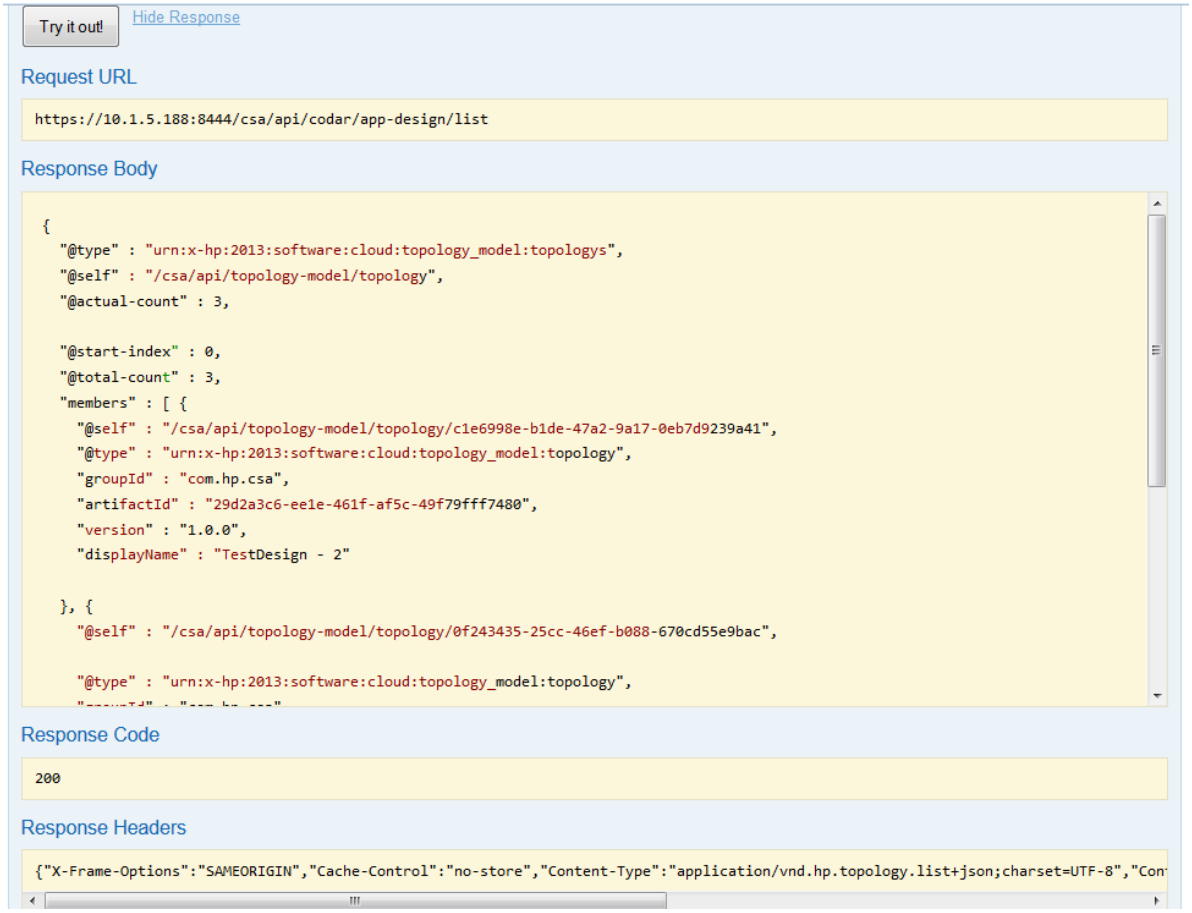
Perform the following steps to try an API call.

1. From an Codar instance, launch the interactive API content from a browser at **https://<host>:<port>/csa/apidocs.jsp**.
2. If you are not already logged into Codar, you will be prompted to log in. You must have appropriate authority to exercise Codar RESTful calls.
3. Locate the *app-design* API call. Click on the call title to expand it.
4. Click on the codar/app-design/list *GET* method to view the complete documentation for this call.



5. Enter values for the `start-index` and `page-size` parameters. By default, the response for this API is to list all application designs starting with the start index of zero.
6. Click on the **Try it out!** button.
7. You should now see the request URL that was sent, as well as the response body, response code and

response headers for your request as shown here.



Not every API call will be this simple to exercise, but this example demonstrates how the interactive API content is presented, and how RESTful calls can easily be made from this interface. In some situations you will need to make other API calls first to get information necessary for parameters or a request body.

Making API calls from an application

RESTful calls can be made from application code. Preparation for making these calls is similar to making them from an HTTP client. Authentication and authorization requirements as explained in "Communication with Codar " on page 13 must be met.

You also need to set the HTTP headers `Content-Type:application/json` and `accept:application/json` as appropriate for the call you are making.

API calls

The following public API calls provide access to Codar data and functionality. These calls are documented and can be exercised through a browser at: **https://<host>:<port>/csa/apidocs.jsp**. Use the host and port information appropriate for your environment. API calls can also be exercised programmatically or through an HTTP client. See "Retrieve information from Codar using RESTful calls" on page 13 for more information on how to exercise API calls.

Application design APIs

URI	Method	See also
/csa/api/codar/app-design/list	GET	"List application designs" on the next page
/csa/api/codar/app-design/{applicationDesignID}	GET	"Export an application design" on the next page
/csa/api/codar/app-design/import	POST	"Import an application design" on page 21
/csa/api/codar/app-design/importNew	POST	"Import a new application design" on page 23
/csa/api/codar/app-design/{applicationDesignId}/associateToCodar	POST	"Associate a design to Codar" on page 24
/csa/api/codar/app-design/{applicationDesignID}	PUT	"Update an application design" on page 25
/csa/api/codar/app-design/{applicationDesignID}	DELETE	"Delete an application

URI	Method	See also
		design" on page 26

List application designs

Retrieves the list of application designs to which the user has access.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-design/list	GET	None	200 - successful 400 - bad request 401 - not authorized 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-design/list
```

The following JSON was returned:

```
{
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topologys",
  "@self" : "/csa/api/topology-model/topology",
  "@actual-count" : 1,
  "@start-index" : 0,
  "@total-count" : 1,
  "members" : [ {
   "@self" : "/csa/api/topology-model/topology/22b4706c-0849-4c35-b552-4e42c90ae5e4",

    "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
    "groupId" : "com.hp.csa",
    "artifactId" : "81072ea2-2a7f-4fd7-b582-5e47b97fd0b9",
    "version" : "1.0.0"
    "displayName" : "Petclinic Application for AWS"
  } ]
}
```

Export an application design

Export an application design by providing the application design ID. The application design is returned in JSON format.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-design/	GET	Design ID	200 - successful

URI	Method	Parameters	Returns
{applicationDesignID}			400 - bad request 401 - not authorized 404 - not found 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-design/e57673db-0ee7-4061-8f9e-ffdcabc07b4c
```

The following JSON was returned:

```
{
  "@self" : "/csa/api/topology-model/topology/e57673db-0ee7-4061-8f9e-ffdcabc07b4c",
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
  "groupId" : "com.hp.csa",
  "artifactId" : "3eb835ae-750f-41c9-8861-de70a285ca40",
  "version" : "1.0.0",
  "displayName" : "Test_Design-1",
  "description" : "my design",
  "resources" : [ {
    "id" : "VcenterServer0001",
    "name" : "vCenter Server 1",
    ...
}
```

For complete output, see "Export an application design example" on page 133.

Import an application design

Import an application design by providing the application design in JSON format as input. If the design already exists, then an error is returned.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-design/import	POST	JSON	201 - successful 400 - bad request 401 - authorization failed 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-design/import
```

The following JSON was sent:

```
{
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
  "groupId" : "com.hp.csa",
  "artifactId" : "3eb835ae-750f-41c9-8861-de70a285ca43",
  "version" : "1.0.0",
  "displayName" : "Test_Design-2",
  "description" : "my design 02",
  "resources" : [ {
    "id" : "VcenterServer0001",
    "name" : "vCenter Server 1",
    "component" : {
      "@self" : "/csa/api/topology-model/component-type/b83f4e21-d8a9-4ff3-a76e-
3a69775d1d86"
    },
    ...
}
```

For complete output, see "Import an application design example" on page 139.

The following JSON was returned:

```
{
  "@self" : "/csa/api/topology-model/topology/e57673db-0ee7-4061-8f9e-ffdcabc07b4c",
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
  "groupId" : "com.hp.csa",
  "artifactId" : "3eb835ae-750f-41c9-8861-de70a285ca40",
  "version" : "1.0.0",
  "displayName" : "Test_Design-1",
  "description" : "my design",
  "resources" : [ {
    "id" : "VcenterServer0001",
    "name" : "vCenter Server 1",
    "component" : {
      "@self" : "/csa/api/topology-model/component-type/b83f4e21-d8a9-4ff3-a76e-
3a69775d1d86"
    },
    ...
}
```

For complete output, see "Import an application design example" on page 139.

Note: If you have a Scaling Group component in the design, you must change the ID and remove "@self" attribute as shown in the following example:

```
"id" : "0c66986b-a62f-4a99-ba1a-1c05e39b4295", ! Modify the ID
"name" : "SG",
"component" : {
  "@self" : "/csa/api/topology-model/component-type/283f9da2-44c7-4208-bcdd-
67f9fdddc02e", ! Remove this line
  "groupId" : "com.hp.csa.type.scalingGroup",
  "artifactId" : "ScalingGroupType_0c66986b-a62f-4a99-ba1a-1c05e39b4295", ! Modify
```

```
the ID
  "version" : "1"
},
```

Import a new application design

Import a new application design if it does not exist. If the design exists, no error is displayed and the original design is returned.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-design/importNew	POST	JSON	201 - successful 400 - bad request 401 - authorization failed 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-design/importNew
```

The following JSON was sent:

```
{
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
  "groupId" : "com.hp.csa",
  "artifactId" : "3eb835ae-750f-41c9-8861-de70a285ca43",
  "version" : "1.0.1",
  "displayName" : "Test_Design-2",
  "description" : "my design 02",
   ... (The remaining output is the same as "Import an application design example" on page 139.)
}
```

The following JSON was returned:

```
{
  "@self" : "/csa/api/topology-model/topology/e57673db-0ee7-4061-8f9e-ffdcabc07b4c",
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
  "groupId" : "com.hp.csa",
  "artifactId" : "3eb835ae-750f-41c9-8861-de70a285ca40",
  "version" : "1.0.1",
  "displayName" : "Test_Design-1",
  "description" : "my design",
  ... (The remaining output is the same as "Import an application design example" on page 139)
}
```

Note: If you have a Scaling Group component in the design, you must change the ID and remove "@self" attribute as shown in the following example:

```
"id" : "0c66986b-a62f-4a99-ba1a-1c05e39b4295", ! Modify the ID
"name" : "SG",
"component" : {
  "@self" : "/csa/api/topology-model/component-type/283f9da2-44c7-4208-bcdd-
67f9fdddc02e", ! Remove this line
  "groupId" : "com.hp.csa.type.scalingGroup",
  "artifactId" : "ScalingGroupType_0c66986b-a62f-4a99-ba1a-1c05e39b4295", ! Modify
the ID
  "version" : "1"
},
```

Associate a design to Codar

Associate an application design to the Codar pipeline management process by tagging the application design with the Release Pipeline.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-design/{applicationDesignId}/associateToCodar	POST	Application Design ID	200 - successful 400 - bad request 401 - not authorized 404 - Not found 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-design/ae058bcc-4ad3-4e9a-8eb8-
3ee98eef8d8f/associateToCodar
```

The following JSON was returned:

```
{
"associatedTagId": "8a818cf8ara15bc772b0145cb6efjhg99",
"designId": "ae058bcc-4ad3-4e9a-8eb8-3ee98eef8d8f",
"designName": "MyDesign",
"associatedTagName": "Codar Application"
}
```

Dissociate a design from Codar

Dissociate an application design from the Codar pipeline management process by removing the Release Pipeline from the application design.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-design/{applicationDesignId}/dissociateFromCodarApplication	POST	Application Design ID	200 - successful 400 - bad request 404 - not found

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-design/440af252-741b-4f65-addf-8239f6feb8dd/dissociateFromCodarApplication
```

The following JSON was returned:

```
{
"dissociatedTagId": "8a8185bb47b46bbe0147b4c4b8c00006",
"designId": "267fa378-7234-4e8d-b96d-97f62a448cbe",
"designName": "NewD",
"dissociatedTagName": "Codar Application"
}
```

Update an application design

Update an application design by providing an application design ID and an application design in JSON format.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-design/{applicationDesignID}	PUT	• Application Design ID • JSON	200 - successful 400 - bad request 401 - not authorized 404 - not found 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-design/e57673db-0ee7-4061-8f9e-ffdcabc07b4c
```

The following JSON was sent:

```
{
  "@self" : "/csa/api/topology-model/topology/e57673db-0ee7-4061-8f9e-ffdcabc07b4c",
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
  "groupId" : "com.hp.csa",
  "artifactId" : "3eb835ae-750f-41c9-8861-de70a285ca40",
  "version" : "1.0.0",
  "displayName" : "Test_Design-1",
  "description" : "my design updated",
```

... (The remaining output is the same as "Import an application design example" on page 139 with modified values.)

```
}
```

The following JSON was returned:

```
{
  "@self" : "/csa/api/topology-model/topology/e57673db-0ee7-4061-8f9e-ffdcabc07b4c",
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
  "groupId" : "com.hp.csa",
  "artifactId" : "3eb835ae-750f-41c9-8861-de70a285ca40",
  "version" : "1.0.2",
  "displayName" : "Test_Design-1",
  "description" : "my design updated",
```

... (The remaining output is the same as "Import an application design example" on page 139.)

```
}
```

Delete an application design

Delete an application design. Any package that is associated with this design is also deleted.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-design/{applicationDesignID}	DELETE	Application Design ID	200 - successful 204 - no content 400 - bad request 401 - not authorized 404 - not found 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-design/e57673db-0ee7-4061-8f9e-ffdcabc07b4c
```

Application environment APIs

The application environment APIs manage environments for an application design version.

URI	Method	Description
/csa/api/codar/app-environments	GET	"List environments associated with an application design version" below
/csa/api/codar/app-environments/{designId}/{lifecycleStage}/eligible	GET	"List environments associated with the lifecycle stage of an application design" on page 29
/csa/api/codar/app-environments/dissociate	GET	"List environments not associated with the lifecycle stage of an application design" on page 30
/csa/api/codar/app-environments/eligible	GET	"List environments eligible for deploying a package" on page 31
/csa/api/codar/app-environments	PUT	"Associate environments to the specified lifecycle stage of an application design" on page 32

List environments associated with an application design version

This API lists all environments associated to the application design.

The appDesignId (ID of the application design) parameter is required for executing this API. With appDesignId as the parameter, the API lists environments that are associated to all the life cycle stages of the specified application design.

Optionally, you can also provide lifecycleStage, StartIndex, and pageSize as parameters. With lifecycleStage as parameter along with appDesignId, the API returns environments that are associated to the specified life cycle stage of the application design.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/codar/app-environment	GET	• appDesignId - Id ofthe application design • (Optional) lifecycleStage - Filters the response by life cycle stage. • (Optional) startIndex - A number that specifies the offset of the first entry to be included in the page. • (Optional) pageSize - A number that specifies the page size.	200 - Successful 400 - Bad request 401 - Not authorized 404 - Not found 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-environments?appDesignId=980aa84e-2b04-48f3-af18-7b2595716abb
```

The following JSON was returned:


```
{
 "members": [
  {
   "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a5",
   "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
   "id": "84905d24-d92f-4916-b312-78af206d31a5",
   "name": "DEVELOPMENT",
   "displayName": "test stage1",
   "description": "test stage1 desc",
   "count": "1",
   "color": "#ffffff",
   "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
   "environments": [
    {
     "@self": "/csa/api/resource/environment/8a8186da513854bf01513d73dcb50c24",
     "@type": "urn:x-hp:2012:software:cloud:data_model:resource-environment",
     "id": "8a8186da513854bf01513d73dcb50c24",
     "name": "Testing environment",
     "description": "Testing environment",
     "icon": "/csa/images/library/other.png"
    }
   ]
  },
  {
   "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a8",
   "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
   "id": "84905d24-d92f-4916-b312-78af206d31a8",
   "name": "PRODUCTION",
   "displayName": "Production",
   "description": "The final stage in which the artifacts are deployed and can go live.",
   "count": "0",
   "color": "#ff887c",
   "icon": "/csa/api/blobstore/lifecycle_stage_production.png?tag=library",
   "environments": []
  },
  {
   "@self": "/csa/api/codar/app-lifecycle/stages/673b77f2-127b-45a5-81e5-25efbdfcd960",
   "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
   "id": "673b77f2-127b-45a5-81e5-25efbdfcd960",
   "name": "TEST_STAGE",
   "displayName": "test stage",
   "description": "test stage desc",
   "count": "2",
   "color": "#ffffff",
   "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
   "environments": [
    {
```

```
      "@self": "/csa/api/resource/environment/8a8186da513854bf01513d733be40c22",
      "@type": "urn:x-hp:2012:software:cloud:data_model:resource-environment",
      "id": "8a8186da513854bf01513d733be40c22",
      "name": "Application Development",
      "description": "Application Development",
      "icon": "/csa/images/library/software_application_tier.png"
     },
     {
      "@self": "/csa/api/resource/environment/8a8186da510f5cd80151340c64432359",
      "@type": "urn:x-hp:2012:software:cloud:data_model:resource-environment",
      "id": "8a8186da510f5cd80151340c64432359",
      "name": "vCenter",
      "icon": "/csa/images/library/other.png"
     }
    ]
   }
  ]
}
```


List environments associated with the lifecycle stage of an application design

List all environments that are associated with the lifecycle stage of an application design. designId (Id of application design) and LifecycleStage (name of life cycle stage) parameters are required for executing this API.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-environments/{designId}/{lifecycleStage}/eligible	GET	• designId • LifecycleStage	200 - Success 400 - bad request 401 - Authorization failure 404 - not found 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-environments/980aa84e-2b04-48f3-af18-7b2595716abb/DEVELOPMENT/eligible
```

The following JSON was returned:

```
{
 "@total_results": 1,
 "@self": "/csa/api/resource/environment/",
 "@type": "urn:x-hp:2012:software:cloud:data_model:resource-environment:collection",
 "members": [
  {
   "name": "Testing environment",
   "id": "8a8186da513854bf01513d73dcb50c24",
   "description": "Testing environment",
   "icon": "/csa/api/blobstore/other.png?tag=library"
  }
 ]
}
```


List environments not associated with the lifecycle stage of an application design

List all environments that are not associated with the lifecycle stage of an application design. appDesignId and LifecycleStage parameters are required for executing this API.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-environments/dissociate	GET	• appDesignId • LifecycleStage	200 - Success 400 - bad request 401 - Authorization failure 404 - not found 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-environments/dissociate?appDesignId=980aa84e-2b04-48f3-af18-7b2595716abb&lifecycleStage=DEVELOPMENT
```

The following JSON was returned:


```
{
 "members": [
  {
```

```
    "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a5",
    "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
    "id": "84905d24-d92f-4916-b312-78af206d31a5",
    "name": "DEVELOPMENT",
    "displayName": "test stage1",
    "description": "test stage1 desc",
    "count": "3",
    "color": "#ffffff",
    "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
    "environments": [
     {
      "@self": "/csa/api/resource/environment/8a8186da510f5cd80151340c64432359",
      "@type": "urn:x-hp:2012:software:cloud:data_model:resource-environment",
      "id": "8a8186da510f5cd80151340c64432359",
      "name": "vCenter",
      "icon": "/csa/images/library/other.png"
     },
     {
      "@self": "/csa/api/resource/environment/8a8186da513854bf01513862b01c007b",
      "@type": "urn:x-hp:2012:software:cloud:data_model:resource-environment",
      "id": "8a8186da513854bf01513862b01c007b",
      "name": "HDP",
      "icon": "/csa/images/library/other.png"
     },
     {
      "@self": "/csa/api/resource/environment/8a8186da513854bf01513d733be40c22",
      "@type": "urn:x-hp:2012:software:cloud:data_model:resource-environment",
      "id": "8a8186da513854bf01513d733be40c22",
      "name": "Application Development",
      "description": "Application Development",
      "icon": "/csa/images/library/software_application_tier.png"
     }
    ]
   }
  ]
}
```


List environments eligible for deploying a package

Lists all environments that are eligible for deploying a given package.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-environments/eligible	GET	packageId	200 - Success 400 - Bad request 404 - not

URI	Method	Parameters	Returns
			found

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-environments/eligible?packageId=d416df45-af08-4cac-981b-232853d2b0dd
```

The following JSON was returned:


```
{
 "@total_results": 4,
 "@self": "/csa/api/resource/environment/",
 "@type": "urn:x-hp:2012:software:cloud:data_model:resource-environment:collection",
 "members": [
  {
   "name": "vCenter",
   "id": "8a8186da510f5cd80151340c64432359",
   "icon": "/csa/api/blobstore/other.png?tag=library"
  },
  {
   "name": "HDP",
   "id": "8a8186da513854bf01513862b01c007b",
   "icon": "/csa/api/blobstore/other.png?tag=library"
  },
  {
   "name": "Application Development",
   "id": "8a8186da513854bf01513d733be40c22",
   "description": "Application Development",
   "icon": "/csa/api/blobstore/software_application_tier.png?tag=library"
  },
  {
   "name": "Testing environment",
   "id": "8a8186da513854bf01513d73dcb50c24",
   "description": "Testing environment",
   "icon": "/csa/api/blobstore/other.png?tag=library"
  }
 ]
}
```


Associate environments to the specified lifecycle stage of an application design

Associates environments to the specified lifecycle stage of an application design.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-environments	PUT	• appDesignId - Id of the application design • Body - list of environment ids for the specific lifecycle stage in Json format	200 - Successful 400 - Bad request 401 - Authorization failure 404 - Not found 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-environments?appDesignId=980aa84e-2b04-48f3-af18-7b2595716abb
```

The following JSON was sent:

```
{
    "name":"PRODUCTION",
    "environments": [
    {
        "id":"8a8186da513854bf01513862b01c007b"
    }]
}
```

The following JSON was returned:

```
no content
```

Application lifecycle APIs

This API provides different operations for the life cycle management on an application.

URI	Method	Description
/api/codar/app-lifecycle/stages	GET	"List application lifecycle stages" on the next page
/api/codar/app-lifecycle/stages/{stageId}/design	GET	"List design versions" on page 38
/api/codar/app-lifecycle/stages/template	GET	"List lifecycle stages of the default template" on

URI	Method	Description
		page 41
/api/codar/app-lifecycle/stages	POST	"Create a new lifecycle stage" on page 42
/api/codar/app-lifecycle/stages/{stageId}	PUT	"Update application lifecycle stage" on page 44
/api/codar/app-lifecycle/stages/template	PUT	"Update lifecycle stages of the default template" on page 45
/api/codar/app-lifecycle/stages/{stageId}	DELETE	"Delete application lifecycle stage" on page 46

List application lifecycle stages

Lists all lifecycle stages. This API is used by the Codar UI to retrieve packages grouped by life cycle. The default stage sequence is Development >Testing >Staging > Production.

You can optionally provide applicationDesignId (Id of an application design) as parameter. By providing applicationDesignId as parameter, this API lists the lifecycle stages of that particular application design.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-lifecycle/stages	GET	*(Optional)* applicationDesignId	200 - Successful to list the life cycle stages 400 - Bad request 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-lifecycle/stages
```

The following JSON was returned:

Tip: {
 "@total_results": 7,
 "@self": "/csa/api/codar/app-lifecycle/stages/",
 "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
 "members": [
  {
   "@self": "/csa/api/codar/app-lifecycle/stages/673b77f2-127b-45a5-81e5-25efbdfcd960",
   "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
   "name": "TEST_STAGE",
   "displayName": "test stage",
   "description": "test stage desc",
   "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
   "color": "#ffffff",
   "roles": [
    {
     "@self": "/csa/api/codar/role/4552b9da88a24767ac7e5b605f050df5",

```
        "@type": "urn:x-hp:2012:software:cloud:data_model:role",
        "name": "CODAR_APPLICATION_ARCHITECT",
        "displayName": "Application Architect",
        "description": "The Application Architect can embrace component, create, edit, delete Application and application version, deploy, create, edit and delete package in development stage. Application Architect cannot reject package in any stage.",
        "icon": "/csa/images/categories/role/application_architect.png",
        "readOnly": true
       }
     ]
   },
   {
     "@self": "/csa/api/codar/app-lifecycle/stages/ab0be9f6-ef4f-4bf7-9430-04c0c6776ac5",
     "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
     "name": "TEST_STAGE3",
     "displayName": "test stage3",
     "description": "test stage4 desc",
     "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
     "color": "#ffffff",
     "roles": [
       {
         "@self": "/csa/api/codar/role/4552b9da88a24767ac7e5b605f050df5",
         "@type": "urn:x-hp:2012:software:cloud:data_model:role",
         "name": "CODAR_APPLICATION_ARCHITECT",
         "displayName": "Application Architect",
         "description": "The Application Architect can embrace component, create, edit, delete Application and application version, deploy, create, edit and delete package in development stage. Application Architect cannot reject package in any stage.",
         "icon": "/csa/images/categories/role/application_architect.png",
         "readOnly": true
       }
     ]
   },
   {
     "@self": "/csa/api/codar/app-lifecycle/stages/02e4ffb1-3c72-4ffa-932b-a4827adabd08",
     "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
     "name": "TEST_STAGE8",
     "displayName": "test stage8",
     "description": "test stage8 desc",
     "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
     "color": "#ffffff",
     "roles": [
       {
         "@self": "/csa/api/codar/role/4552b9da88a24767ac7e5b605f050df5",
         "@type": "urn:x-hp:2012:software:cloud:data_model:role",
         "name": "CODAR_APPLICATION_ARCHITECT",
         "displayName": "Application Architect",
         "description": "The Application Architect can embrace component, create, edit, delete Application and application version, deploy, create, edit and delete package in development stage. Application
```

```
Architect cannot reject package in any stage.",
        "icon": "/csa/images/categories/role/application_architect.png",
        "readOnly": true
       }
     ]
    },
    {
     "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a5",
     "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
     "name": "DEVELOPMENT",
     "displayName": "test stage1",
     "description": "test stage1 desc",
     "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
     "color": "#ffffff",
     "roles": [
      {
       "@self": "/csa/api/codar/role/4552b9da88a24767ac7e5b605f050df5",
       "@type": "urn:x-hp:2012:software:cloud:data_model:role",
       "name": "CODAR_APPLICATION_ARCHITECT",
       "displayName": "Application Architect",
       "description": "The Application Architect can embrace component, create, edit, delete Application
and application version, deploy, create, edit and delete package in development stage. Application
Architect cannot reject package in any stage.",
       "icon": "/csa/images/categories/role/application_architect.png",
       "readOnly": true
      }
     ]
    },
    {
     "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a6",
     "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
     "name": "TESTING",
     "displayName": "Testing",
     "description": "The test cases are executed against the artifacts created in the development stage.
The validation and verification of the business and technical requirements are performed on the
application.",
     "icon": "/csa/api/blobstore/lifecycle_stage_testing.png?tag=library",
     "color": "#fbd75b",
     "roles": [
      {
       "@self": "/csa/api/codar/role/40faa9a2769b451f8fecd1104260d062",
       "@type": "urn:x-hp:2012:software:cloud:data_model:role",
       "name": "CODAR_APPLICATION_QA",
       "displayName": "Application QA",
       "description": "The Application QA can deploy, edit, delete and reject package in testing and
staging stages and promote package from testing stage to staging stage.",
       "icon": "/csa/images/categories/role/application_qa.png",
       "readOnly": true
      }
```

```
    ]
   },
   {
    "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a7",
    "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
    "name": "STAGING",
    "displayName": "Staging",
    "description": "The pre-production stage in which the artifacts are tested on production like
environment.",
    "icon": "/csa/api/blobstore/lifecycle_stage_staging.png?tag=library",
    "color": "#dbadff",
    "roles": [
     {
      "@self": "/csa/api/codar/role/6ff42967987c4a58ab5751817088607d",
      "@type": "urn:x-hp:2012:software:cloud:data_model:role",
      "name": "CODAR_APPLICATION_OPERATIONS_MANAGER",
      "displayName": "Application Operations Manager",
      "description": "The Application Operations Manager can deploy, reject, edit, delete package in
staging stage.",
      "icon": "/csa/images/categories/role/application_operations_manager.png",
      "readOnly": true
     },
     {
      "@self": "/csa/api/codar/role/5e78c029ace644ce8778f3a50f88873b",
      "@type": "urn:x-hp:2012:software:cloud:data_model:role",
      "name": "CODAR_APPLICATION_RELEASE_MANAGER",
      "displayName": "Application Release Manager",
      "description": "The Application Release Manager can deploy, reject, edit, delete package in staging
and production stages and promote package from staging stage to production stage.",
      "icon": "/csa/images/categories/role/application_releasemanager.png",
      "readOnly": true
     }
    ]
   },
   {
    "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a8",
    "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
    "name": "PRODUCTION",
    "displayName": "Production",
    "description": "The final stage in which the artifacts are deployed and can go live.",
    "icon": "/csa/api/blobstore/lifecycle_stage_production.png?tag=library",
    "color": "#ff887c",
    "roles": [
     {
      "@self": "/csa/api/codar/role/5e78c029ace644ce8778f3a50f88873b",
      "@type": "urn:x-hp:2012:software:cloud:data_model:role",
      "name": "CODAR_APPLICATION_RELEASE_MANAGER",
      "displayName": "Application Release Manager",
      "description": "The Application Release Manager can deploy, reject, edit, delete package in staging
```

```
and production stages and promote package from staging stage to production stage.",
        "icon": "/csa/images/categories/role/application_releasemanager.png",
        "readOnly": true
      }
    ]
  }
 ]
}
```

List design versions

Lists all application designs and their versions that are associated with the specified lifecycle stage.

URI	Method	Parameters	Returns
http://[host]:[port]/api/codar/app-lifecycle/stages/{stageId}/design	GET	stageId - Id of the lifecycle stage	200 - Successful to list life cycle stages found 400 - Bad request 500 - Internal Server error

Example

The following URL was sent:

`https://localhost:8444/csa/api/codar/app-lifecycle/stages/{stageId}/design`

The following JSON was returned:

```
https://localhost:8444/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a5/design
```

The following JSON was returned:

```
{
  "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a5",
  "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
  "name": "DEVELOPMENT",
  "displayName": "test stage1",
  "description": "test stage1 desc",
  "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
  "color": "#ffffff",
  "members": [
    {
      "@self": "/csa/csa/api/container/topology/8a8186da510f5cd801510f6868160155",
      "@type": "urn:x-hp:2012:software:cloud:data_model:topology:collection",
      "createdOn": "2015-11-16 13:57:59.383",
      "modified": "2015-11-16 13:57:59.383",
      "name": "PetClinic Application on Existing Servers",
      "description": "A two-tier PetClinic Application with Database component installed
on MySQL and Application component on Tomcat Server. These components are deployed on
```

```
Existing Servers.",
      "icon": "/csa/images/library/pets.png",
      "members": [
        {
          "@self": "/csa/csa/api/designs/design/f78a358e-28d0-4690-95ee-644697765bcb",
          "@type": "urn:x-hp:2012:software:cloud:data_model:topology",
          "createdOn": "2015-05-22 17:42:16.733",
          "description": "A two-tier PetClinic Application with Database component
installed on MySQL and Application component on Tomcat Server. These components are
deployed on Existing Servers.",
          "version": "1.50.00"
        }
      ]
    },
    {
      "@self": "/csa/csa/api/container/topology/8a8186da510f5cd80151347d4f7626ec",
      "@type": "urn:x-hp:2012:software:cloud:data_model:topology:collection",
      "createdOn": "2015-11-23 18:46:46.327",
      "modified": "2015-11-23 18:46:46.327",
      "name": "HDP Node Chat Application",
      "description": "HDP Node Chat Application",
      "icon": "/csa/images/library/Exchange_solutions_RGB_blue_NT.png",
      "members": [
        {
          "@self": "/csa/csa/api/designs/design/8611a60a-fc81-4005-9b1b-14c6ae5cbaa2",
          "@type": "urn:x-hp:2012:software:cloud:data_model:topology",
          "createdOn": "2015-11-23 18:46:46.433",
          "description": "HDP Node Chat Application",
          "version": "1.60.00"
        }
      ]
    },
    {
      "@self": "/csa/csa/api/container/topology/8a8186da510f5cd80151238b98aa02af",
      "@type": "urn:x-hp:2012:software:cloud:data_model:topology:collection",
      "createdOn": "2015-11-20 11:48:49.897",
      "modified": "2015-11-20 11:48:49.897",
      "name": "XI app",
      "description": "",
      "icon": "/csa/images/library/Service_Design.png",
      "members": [
        {
          "@self": "/csa/csa/api/designs/design/980aa84e-2b04-48f3-af18-7b2595716abb",
          "@type": "urn:x-hp:2012:software:cloud:data_model:topology",
          "createdOn": "2015-11-20 11:48:50.307",
          "description": "",
          "version": "1.0.0"
        }
      ]
    },
```

```
    {
      "@self": "/csa/csa/api/container/topology/8a8186da510f5cd801510f68522400fa",
      "@type": "urn:x-hp:2012:software:cloud:data_model:topology:collection",
      "createdOn": "2015-11-16 13:57:53.763",
      "modified": "2015-11-16 13:57:53.763",
      "name": "PetClinic Application with Load Balancer",
      "description": "A PetClinic Application with scalable stack and Apache Load
Balancer. This is a partial design which requires the Database and Application server
microservices for fulfillment.",
      "icon": "/csa/images/library/pets.png",
      "members": [
        {
          "@self": "/csa/csa/api/designs/design/d86f5445-e930-45ef-b8c8-5c9cae860541",
          "@type": "urn:x-hp:2012:software:cloud:data_model:topology",
          "createdOn": "2015-05-29 21:21:40.363",
          "description": "A PetClinic Application with scalable stack and Apache Load
Balancer. This is a partial design which requires the Database and Application server
microservices for fulfillment.",
          "version": "1.50.00"
        }
      ]
    },
    {
      "@self": "/csa/csa/api/container/topology/8a8186da510f5cd801510f6827920052",
      "@type": "urn:x-hp:2012:software:cloud:data_model:topology:collection",
      "createdOn": "2015-11-16 13:57:42.867",
      "modified": "2015-11-16 13:57:42.867",
      "name": "PetClinic Application on vCenter",
      "description": "A two-tier PetClinic Application with Database component installed
on MySQL and Application component on Tomcat Server. These components are deployed on
vCenter Servers.",
      "icon": "/csa/images/library/pets.png",
      "members": [
        {
          "@self": "/csa/csa/api/designs/design/cfc40fe6-7884-4e74-93c1-a8f91e02e422",
          "@type": "urn:x-hp:2012:software:cloud:data_model:topology",
          "createdOn": "2015-05-29 10:00:11.507",
          "description": "A two-tier PetClinic Application with Database component
installed on MySQL and Application component on Tomcat Server. These components are
deployed on vCenter Servers.",
          "version": "1.50.00"
        }
      ]
    },
    {
      "@self": "/csa/csa/api/container/topology/8a8186da510f5cd801510f682f3e006c",
      "@type": "urn:x-hp:2012:software:cloud:data_model:topology:collection",
      "createdOn": "2015-11-16 13:57:44.83",
      "modified": "2015-11-16 13:57:44.83",
      "name": "PetClinic Application",
```

```
      "description": "A partial PetClinic Application design requiring an infrastructure
with Database and Application Server for fulfillment.",
      "icon": "/csa/images/library/pets.png",
      "members": [
        {
          "@self": "/csa/csa/api/designs/design/c0929227-1fbb-4f9f-b8e4-e587938c42d7",
          "@type": "urn:x-hp:2012:software:cloud:data_model:topology",
          "createdOn": "2014-11-04 19:27:45.687",
          "description": "A partial PetClinic Application design requiring an
infrastructure with Database and Application Server for fulfillment.",
          "version": "1.50.00"
        }
      ]
    }
  ]
}
```


List lifecycle stages of the default template

Lists the lifecycle stages of the default template.

URI	Method	Parameters	Returns
http://[host]:[port]/api/codar/app-lifecycle/stages/template	GET		200 - Successful to list life cycle stages found 400 - Bad request 500 - Internal Server error

Example

The following URL was sent:

`https://localhost:8444/csa/api/codar/app-lifecycle/stages/template`

The following JSON was returned:


```
{
  "@total_results": 4,
  "@self": "/csa/api/codar/app-lifecycle/stages/",
  "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
  "members": [
    {
      "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a5",
      "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
      "name": "DEVELOPMENT",
      "displayName": "test stage1",
      "description": "test stage1 desc",
      "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
      "color": "#ffffff"
    },
```

```
    {
      "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a6",
      "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
      "name": "TESTING",
      "displayName": "Testing",
      "description": "The test cases are executed against the artifacts created in the
development stage. The validation and verification of the business and technical
requirements are performed on the application.",
      "icon": "/csa/api/blobstore/lifecycle_stage_testing.png?tag=library",
      "color": "#fbd75b"
    },
    {
      "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a7",
      "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
      "name": "STAGING",
      "displayName": "Staging",
      "description": "The pre-production stage in which the artifacts are tested on
production like environment.",
      "icon": "/csa/api/blobstore/lifecycle_stage_staging.png?tag=library",
      "color": "#dbadff"
    },
    {
      "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a8",
      "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
      "name": "PRODUCTION",
      "displayName": "Production",
      "description": "The final stage in which the artifacts are deployed and can go
live.",
      "icon": "/csa/api/blobstore/lifecycle_stage_production.png?tag=library",
      "color": "#ff887c"
    }
  ]
}
```


Create a new lifecycle stage

Creates a new lifecycle stage.

URI	Method	Parameters	Returns
http://[host]:[port]csa/api/codar/app-lifecycle/stages	POST		201 - Successfully created a new life cycle stage 400 - Bad request 401 - Authorization failure 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-lifecycle/stages
```

The following input JSON was sent:

```
 {
    "displayName":"test",
    "description":"test stage desc",
    "icon":"/csa/api/blobstore/Service_Design.png?tag=library",
    "color":"#ffffff",
    "roles":["4552b9da88a24767ac7e5b605f050df5","90d96588360da0c701360da0f2b900cb"]

 }
```

The following JSON was returned:

```
{
  "@self": "/csa/api/codar/app-lifecycle/stages/ab0be9f6-ef4f-4bf7-9430-04c0c6776ac5",
  "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
  "name": "TEST_STAGE",
  "displayName": "test",
  "description": "test stage desc",
  "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
  "color": "#ffffff",
  "roles": [
    {
      "@self": "/csa/api/codar/role/90d96588360da0c701360da0f2b900cb",
      "@type": "urn:x-hp:2012:software:cloud:data_model:role",
      "name": "CSA_ADMIN",
      "displayName": "Administrator",
      "description": "The Administrator has access to all functionality in the Cloud
Service Management Console which includes the Catalogs, Offerings, Designs, Components,
Resouces, Operations and Organizations content areas.",
      "icon": "/csa/images/categories/role/csa_admin.png",
      "readOnly": false
    },
    {
      "@self": "/csa/api/codar/role/4552b9da88a24767ac7e5b605f050df5",
      "@type": "urn:x-hp:2012:software:cloud:data_model:role",
      "name": "CODAR_APPLICATION_ARCHITECT",
      "displayName": "Application Architect",
      "description": "The Application Architect can embrace component, create, edit,
delete Application and application version, deploy, create, edit and delete package in
development stage. Application Architect cannot reject package in any stage.",
      "icon": "/csa/images/categories/role/application_architect.png",
      "readOnly": false
    }
```

```
  ]
}
```

Update application lifecycle stage

Updates an application lifecycle stage.

URI	Method	Parameters	Returns
http://[host]:[port]csa/api/codar/app-lifecycle/stages/{stageId}	PUT	stageId - Id of the lifecycle stage	200 - Successfully updated lifecycle stage 400 - Bad request 500 - Internal Server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-lifecycle/stages/{stageId}
```

The following JSON was sent to update an lifecycle stage:

```
 {
       "displayName":"test stage",
       "description":"test stage desc",
       "icon":"/csa/api/blobstore/Service_Design.png?tag=library",
       "color":"#ffffff",
       "roles":["4552b9da88a24767ac7e5b605f050df5","90d96588360da0c701360da0f2b900cb"]
 }
```

The following JSON was returned:

```
 {
       "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a5",
       "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
       "name": "DEVELOPMENT",
       "displayName": "test stage",
       "description": "test stage desc",
       "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
       "color": "#ffffff",
       "roles": [
         {
               "@self": "/csa/api/codar/role/90d96588360da0c701360da0f2b900cb",
```

```
            "@type": "urn:x-hp:2012:software:cloud:data_model:role",
            "name": "CSA_ADMIN",
            "displayName": "Administrator",
            "description": "The Administrator has access to all functionality in the
                            Cloud Service Management Console which includes the Catalogs,
                            Offerings, Designs, Components, Resouces, Operations and
                            Organizations content areas.",
            "icon": "/csa/images/categories/role/csa_admin.png",
            "readOnly": false
        },
        {
            "@self": "/csa/api/codar/role/4552b9da88a24767ac7e5b605f050df5",
            "@type": "urn:x-hp:2012:software:cloud:data_model:role",
            "name": "CODAR_APPLICATION_ARCHITECT",
            "displayName": "Application Architect",
            "description": "The Application Architect can embrace component, create, edit,
                            delete Application and application version, deploy, create,
                            edit and delete package in development stage.
                            Application Architect cannot reject package in any stage.",
            "icon": "/csa/images/categories/role/application_architect.png",
            "readOnly": false
        }
      ]
}
```


Update lifecycle stages of the default template

Updates lifecycle stages of the default template.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-lifecycle/stages/template	PUT		200 - Successfully updated lifecycle stage 400 - Bad request 500 - Internal Server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-lifecycle/stages/template
```

The following JSON was sent to update an lifecycle stages of the default template:

```
{
       "members":[
              {"id":"84905d24-d92f-4916-b312-78af206d31a5","stageOrder":1},
              {"id":"84905d24-d92f-4916-b312-78af206d3117","stageOrder":2},
              {"id":"84905d24-d92f-4916-b312-78af206d3113","stageOrder":3},
              {"id":"84905d24-d92f-4916-b312-78af206d31a8","stageOrder":4}
       ]
}
```

The following JSON was returned:

```
{
  "@total_results": 2,
  "@self": "/csa/api/codar/app-lifecycle/stages/",
  "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
  "members": [
    {
      "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a5",
      "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
      "name": "DEVELOPMENT",
      "displayName": "test stage1",
      "description": "test stage1 desc",
      "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
      "color": "#ffffff"
    },
    {
      "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a8",
      "@type": "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
      "name": "PRODUCTION",
      "displayName": "Production",
      "description": "The final stage in which the artifacts are deployed and can go
live.",
      "icon": "/csa/api/blobstore/lifecycle_stage_production.png?tag=library",
      "color": "#ff887c"
    }
  ]
}
```

Delete application lifecycle stage

Deletes an application lifecycle stage.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-lifecycle/stages/{stageId}	DELETE	StageId - Id of the lifecycle stage	204 - Deleted successfully and has response no content

URI	Method	Parameters	Returns
			400 - Bad request 401 - Authorization failure 500 - Internal Server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-lifecycle/stages/{stageId}
```

Artifact APIs

URI	Method	See also
/artifact/{organizationID}/group	GET	"List active groups associated with an organization" below
/artifact/{organizationID}/group	POST	"Add groups to an organization" on the next page
/artifact/{organizationID}/group/{groupID}	PUT	"Update group display name or distinguished name" on page 50
/artifact/{organizationID}/group/{groupID}	DELETE	"Delete or dissociate a group from an organization" on page 51

List active groups associated with an organization

List active groups associated with an organization

URI	Method	Parameters	Returns
http://[host]:[port]/csa/rest/artifact/{organizationID}/group	GET	• User ID - Required; the user ID you want to use as credentials for this API call. • Organization ID	200 - successful 401 - not authorized 404 - object not found 500 - server exception

Example

The following URL was sent:

```
https://localhost:8444/csa/rest/artifact/8a81818f3d02fb7e013d0308891d0003/group?userIden
tifier=90d96588360da0c701360da0f1d5f483
```

The following XML was returned:

```
 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GroupList>
  <count>1</count>
  <limit>1</limit>
  <group>
    <id>8a81818f3d02fb7e013d030af854000f</id>
    <isCriticalSystemObject>false</isCriticalSystemObject><isCriticalSystemObject>fal
se</isCriticalSystemObject>
    <name>sc_February 22, 2013 5:54:43 PM UTC</name>
    <displayName>CODAR_APPLICATION_DEVELOPER</displayName>
    <state>...</state>
    <artifactType>...</artifactType>
    <disabled>false</disabled>
    <distinguishedName>
      cn=CodarApplicationDeveloper,ou=CodarApplicationDeveloperGroup,ou=CodarGroups
    </distinguishedName>
    <role>
     ...
    </role>
  </group>
</GroupList>
```

Add groups to an organization

Add groups to an organization.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/rest/artifact/{organizationID} /group	POST	• Organization ID • User ID - Required; the user ID you want to use as credentials for this API call. • XML request body	200 - successful 401 - not authorized 404 - object not found 500 - internal server error

Note that the role must be specified for each group in the request body. The following are valid roles:

- CODAR_APPLICATION_ARCHITECT
- CODAR_APPLICATION_DEVELOPER
- CODAR_APPLICATION_QA
- CODAR_APPLICATION_RELEASE_MANAGER

Example

The following URL was sent:

```
https://localhost:port/csa/rest/artifact/8a81818f3d1421e7013d1423635a0003/group?userIden
tifier=90d96588360da0c701360da0f1d5f483
```

The following XML was sent:

```
<GroupList>
  <group>
   <displayName>My-Group-Name</displayName>
   <distinguishedName>
      cn=TestConsumer,ou=CodarApplicationDeveloper,ou=CodarGroups
    </distinguishedName>
    <role>
      <name>CODAR_APPLICATION_DEVELOPER</name>
 </role>
  </group>
  <group>
    <displayName>Another-Group-Name</displayName>
    <distinguishedName>
 cn=TestConsumer2,ou=CodarApplicationDeveloper,ou=CodarGroups
   </distinguishedName>
 <role>
 <name>CODAR_APPLICATION_DEVELOPER</name>
 </role>
 </group>
 </GroupList>
```

The following XML was returned:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GroupList>
  <count>2</count>
  <limit>2</limit>
  <group>
    <isCriticalSystemObject>false</isCriticalSystemObject>
    <name>My-Group-Name</name>
     <displayName>My-Group-Name</displayName>
    <disabled>false</disabled>
     <distinguishedName>
      cn=TestConsumer,ou=CodarApplicationDeveloper,ou=CodarGroups
     </distinguishedName>
     <role>
       <isCriticalSystemObject>false</isCriticalSystemObject>
       <name>CODAR_APPLICATION_DEVELOPER</name>
       <disabled>false</disabled>
     </role>
   </group>
```

```
   <group>
      <isCriticalSystemObject>false</isCriticalSystemObject>
     <name>Another-Group-Name</name>
     <displayName>Another-Group-Name</displayName>
     <disabled>false</disabled>
     <distinguishedName>
       cn=TestConsumer2,ou=CodarApplicationDeveloper,ou=CodarGroups
    </distinguishedName>
     <role>
       <isCriticalSystemObject>false</isCriticalSystemObject>
       <name>CODAR_APPLICATION_DEVELOPER</name>
       <disabled>false</disabled>
     </role>
   </group>
</GroupList>
```

Update group display name or distinguished name

Updates the group display name or distinguished name for the specified organization.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/rest/artifact/{organizationID}/group/{groupID}	PUT	• Organization ID • User ID - Required; the user ID you want to use as credentials for this API call.	200 - successful 401 - not authorized 404 - object not found 505 - server exception

Example

The following URL was sent:

```
https://localhost:8444/csa/rest/artifact/8a81818f3d02fb7e013d0308891d0003/group/8a81818f3d1437e2013d1795d41107ea?userIdentifier=90d96588360da0c701360da0f1d5f483
```

The following XML was sent:

```
<GroupList>
  <group>
    <displayName>My-New-Group-Name</displayName>
    <distinguishedName>
      cn=TestConsumer,ou=CodarApplicationDeveloper,ou=CodarGroups
    </distinguishedName>
  </group>
</GroupList>
```

The following XML was returned:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Group>
 <id>8a81818f3d1437e2013d1795d41107ea</id>
  <isCriticalSystemObject>false</isCriticalSystemObject>
  <name>sc_February 22, 2013 5:54:43 PM UTC</name>
 <displayName>My-New-Group-Name</displayName>
 <state>
   <id>90d96588360da0c701360da0ef470038</id>
    <isCriticalSystemObject>false</isCriticalSystemObject>
    <name>ACTIVE</name>
    <disabled>false</disabled>
  </state>
 <artifactType>
    <id>90d96588360da0c701360da0eeff002b</id>
   <isCriticalSystemObject>false</isCriticalSystemObject>
    <name>GROUP</name>
    <disabled>false</disabled>
  </artifactType>
  <disabled>false</disabled>
  <distinguishedName>
    cn=TestConsumer,ou=CodarApplicationDeveloper,ou=CodarGroups
 </distinguishedName>
 </Group>
```

Delete or dissociate a group from an organization

Deletes a group or dissociates it from an organization. If no organization is associated with this group, the group will be deleted. Otherwise, the group will be dissociated from the specified organization.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/rest/artifact/{organizationID}/group/{groupID}	DELETE	• Organization ID • Group ID • User ID - Required; the user ID you want to use as credentials for this API call.	200 - successful 401 - not authorized 404 - object not found 500 - server exception

Example

The following URL was sent:

```
https://localhost:8444/csa/rest/artifact/8a81818f3d1421e7013d1423635a0003/group/8a81818f3d1437e2013d1795d41107ea?userIdentifier=90d96588360da0c701360da0f1d5f483
```

The following XML was returned:

```
<messageList>
  <messages>Removed role association for My-New-Group-Name</messages>
</messageList>
```

Composition APIs

URI	Method	See also
csa/api/ui/topology-model/composition/{topologyID}/candidateTopology	GET	"List the candidate topologies that can fulfill the specified partial topology" below

List the candidate topologies that can fulfill the specified partial topology

Lists the candidate topologies that can fulfill the specified partial topology.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/ui/topology-model/composition/{topologyID}/candidateTopology	GET	Topology ID	400 - bad request 403 - not authorized 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/ui/topology-model/composition/9dc7a8f7-9a11-4d4f-a149-6e5b3aeb7618/candidateTopology
```

The following JSON was returned:

```
{
   "@count" : 1,
   "members" : [ {
      "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
      "@self" : "/csa/api/topology-model/topology/9dc7a8f7-9a11-4d4f-a149-
6e5b3aeb7618",
      "id" : "9dc7a8f7-9a11-4d4f-a149-6e5b3aeb7618",
      "iconUrl" : "/csa/designer/img/noimage-lightgray.svg",
      "displayName" : "Allin1",
      "displayNameWithVersion" : "Allin1 - 1.0.0"
      "description" : "",
      "version" : "1.0.0",
```

```
      "artifactId" : "f20b7a8b-0abe-454b-b6d4-cd45764e70c4",
      "groupId" : "com.hp.csa",
      "published" : false,
      "tagIds" : [ ]
  } ],
  ...
}
```

For complete output, see "List the candidate topologies that can fulfill the specified partial topology example" on page 151.)

Container APIs

URI	Method	See also
csa/api/container/	GET	"List existing containers" below
csa/api/container/{containerID}	GET	"List a container" on page 55
csa/api/container/topology/	GET	"List existing topology design containers" on page 55
csa/api/container/topology/{containerID}	GET	"List a service design container" on page 57
csa/api/container/	POST	"Create a container" on page 57
csa/api/container/filter	POST	"List containers matching a filter on tag and type" on page 58
csa/api/container/topology/filter	POST	"List topology design containers matching a filter on tag and type" on page 60
csa/api/container/useraccess/{containerID}	GET	"List user access details for a specified container" on page 61
csa/api/container/useraccess/{containerID}	POST	"Manage user access control on a container" on page 62
csa/api/container/{containerID}	PUT	"Update a container" on page 63
csa/api/container/{containerID}	DELETE	"Delete a container " on page 64

List existing containers

Returns a list of all existing containers.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/container	GET	• type - Specifies the type of containers to return. • start-index - Specifies the offset of the first entry to be included in the page.	200 - successful 403 -not

URI	Method	Parameters	Returns
		• page-size - Specifies the page size. • sort - Name of field to be used for ordering. Optionally followed by colon and "ascending" or "descending." • after - Filters members to those modified on or after the specified timestamp. This parameter uses the SimpleDateFormat ("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'") in UTC. • before - Filters members to those modified before this timestamp. Uses the SimpleDateFormat ("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'") in UTC.	authorized 404 - no containers found

Example

The following URL was sent:

```
https://localhost:8444/csa/api/container
```

The following JSON was returned:


```
{
  "@total_results": 1,
  "@start_index": 0,
  "@items_per_page": 1,
  "@self": "/csa/api/container/",
  "@type": "urn:x-hp:2012:software:cloud:data_model:metamodel:collection",
  "members": [
    {
      "@self": "/csa/api/container/72A82AD399D64A1FADABE348AA59DB90",
      "@type": "urn:x-hp:2012:software:cloud:data_model:metamodel",
      "global_id": "72A82AD399D64A1FADABE348AA59DB90",
      "name": "HP CSA",
      "description": "HP Cloud Service Automation Default Component Catalog",
      "icon": "/csa/api/blobstore/HP_Blue_RGB_150_MN.png?tag=library",
      "ext": {
        "csa_name_key": "HP_CSA",
        "csa_critical_system_object": false,
        "csa_artifact_container_type": "COMPONENT_PALETTE_SEQUENCE_BASED"
      },
      "tags": []
    }
  ],
  "@modified": null
}
```

List a container

Returns the existing container, versions, and number of packages for each life cycle stage for each version of the specified container ID.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/container/{containerID}	GET	Container ID	200 - successful 400 - bad values in the Container ID parameter 403 - not authorized 404 - container does not exist

Examples

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-container/8a81855b4cf9cdd6014cfa2981c40004
```

The following JSON was returned:

```
{
  "@self": "/csa/api/container/topology/8a81855b4cf9cdd6014cfa2981c40004",
  "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
  "@created": "2015-04-27T09:16:06.470Z",
  "@modified": "2015-04-29T10:41:54.603Z",
  "name": "design1",
  "description": "",
  "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
  "ext":
    {
     "csa_name_key": "design1",
     "csa_critical_system_object": false
    },
...
```

For complete output, see "List a container example" on page 153.

List existing topology design containers

Returns a list of all existing topology design containers.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/container/topology/	GET	• start-index - Specifies the offset of the first entry to be included in the page. • page-size - Specifies the page size. • sort - Name of field to be used for ordering. Optionally followed by colon and "ascending" or "descending." • after - Filters members to those modified on or after the specified timestamp. This parameter uses the SimpleDateFormat ("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'") in UTC. • before - Filters members to those modified before this timestamp. Uses the SimpleDateFormat ("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'") in UTC.	200 - successful 403 - not authorized 404 - no containers found

Example

The following URL was sent:

```
https://localhost:8444/csa/api/container/topology/?start-index=1&page-size=3
```

The following JSON was returned:


```
{
  "@total_results": 1,
  "@start_index": 0,
  "@items_per_page": 1,
  "@self": "/csa/api/container/topology/",
  "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
  "members": [
    {
      "@self": "/csa/api/container/topology/8a81848d4d47af63014d47b847230003",
      "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
      "@created": "2015-05-12T10:42:48.740Z",
      "@modified": "2015-05-12T10:42:49.390Z",
      "global_id": "8a81848d4d47af63014d47b847230003",
      "name": "Debian Infra",
      "description": "Debian Infra",
      "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
      "ext": {
        "csa_name_key": "e67b408c66d840a788593690a665e0d3",
        "csa_critical_system_object": false
      },

...
```


For complete output,see "List existing topology design containers example" on page 156.

List a service design container

Lists the service design container specified by Container ID parameter.

URI	Method	Parameters	
http://[host]:[port]/csa/api/container/topology/{containerID}	GET	Container ID	200 - successful 403 - not authorized 404 - no containers found

Example

The following URL was sent:

```
https://localhost:8444/csa/api/container/topology/8a81848d4d47af63014d47b847230003
```

The following JSON was returned:

```
{
  "@self": "/csa/api/container/topology/8a81848d4d47af63014d47b847230003",
  "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
  "@created": "2015-05-12T10:42:48.740Z",
  "@modified": "2015-05-12T10:42:49.390Z",
  "global_id": "8a81848d4d47af63014d47b847230003",
  "name": "Debian Infra",
  "description": "Debian Infra",
  "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
  "ext": {
    "csa_name_key": "e67b408c66d840a788593690a665e0d3",
    "csa_critical_system_object": false
  },
  ...
```

For complete output, see "List a service design container example" on page 157.

Create a container

Creates a container.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/container/	POST	body - JSON representation of an artifact container	200 - successful 400 - bad values in Container ID parameter 401 - not authorized 404 - the container does not exist

Example

The following is sample input for creating an artifact container:

```
{
    "name": "Sample Container",
    "description": "Sample Container Description",
    "icon": "/csa/api/blobstore/Arrow_02_48.png?tag=library",
    "tags": [
        {
            "@self": "/csa/api/tag/8a81848d4d6cd612014d6fdee47f0036"
        }
    ],
    "container_type": "topology_artifact_container"
}
```

The following JSON was returned:

```
{
  "@self": "/csa/api/container/8a81848d4d6cd612014d7044313d00c1",
  "@type": "urn:x-hp:2012:software:cloud:data_model:metamodel",
  "global_id": "8a81848d4d6cd612014d7044313d00c1",
  "name": "Sample Container",
  "description": "Sample Container Description",
  "icon": "/csa/api/blobstore/Arrow_02_48.png?tag=library",
  "ext": {
   "csa_name_key": "fb1e1294-6e5b-4323-a0cc-7414234c10a1",
   "csa_critical_system_object": false,
   "csa_artifact_container_type": "TOPOLOGY_DESIGN_BASED"
 },
 "container_type": "TOPOLOGY_ARTIFACT_CONTAINER",
 "tags": [
  {
   "@self": "/csa/api/tag/8a81848d4d6cd612014d6fdee47f0036",
   "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
   "name": null,
   "description": null,
   "icon": null,
   "color": null,
   "scopes": null
  }
 ]
}
```

List containers matching a filter on tag and type

Lists all containers for a specified tag and type.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/container/filter	POST	Body - query filter specification	200 - successful 400 - bad values in query parameter 403 - not authorized

Examples

The following URL was sent:

```
https://localhost:8444/csa/api/container/filter
```

The following input JSON was sent:

```
{
 "tag": {
 "@self": "/csa/api/tag/8a81848d4d6cd612014d6fdee47f0036"
}
 "container_type": "topology_artifact_container"
}
```

The following JSON was returned:

```
{
 "@total_results": 3,
 "@start_index": 0,
 "@items_per_page": 3,
 "@self": "/csa/api/container/topology/",
 "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
 "members": [
  {
   "@self": "/csa/api/container/topology/8a81848d4d6cd612014d6fdfcb590038",
   "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
   "@created": "2015-05-20T05:50:47.130Z",
   "@modified": "2015-05-20T05:50:48.420Z",
   "global_id": "8a81848d4d6cd612014d6fdfcb590038",
   "name": "Developer Design 1",
   "description": "Developer Design 1",
   "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
   "ext": {
    "csa_name_key": "307c9284-358d-4a5c-b6ee-41e14fa664c8",
    "csa_critical_system_object": false
  },
  "container_type": "TOPOLOGY_ARTIFACT_CONTAINER",
...
```

For complete output, see "List containers matching a filter on tag and type example" on page 159.

List topology design containers matching a filter on tag and type

Lists containers matching a filter on tag and type

URI	Method	Parameters	
http://[host]:[port]/csa/api/container/topology/filter	POST	Body in the following format: `{ "tag": { "@self": "/csa/api/tag/<tagid> " }, "container_type": "topology_artifact_container" }`	200 - successful 403 - not authorized 404 - no containers found

Example

The following URL was sent:

`https://localhost:8444/csa/api/container/topology/filter`

The following JSON request was sent:

```
{
"tag": {
  "@self": "/csa/api/tag/d22c213f7e644d1d8362b2830b2c93a9"
 },
"container_type": "topology_artifact_container"
}
```

The following JSON was returned:

```
{
  "@total_results": 12,
  "@start_index": 0,
  "@items_per_page": 12,
  "@self": "/csa/api/container/topology/",
  "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
  "members": [
    {
      "@self": "/csa/api/container/topology/8a81848d4d47af63014d47b847230003",
      "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
      "@created": "2015-05-12T10:42:48.740Z",
      "@modified": "2015-05-12T10:42:49.390Z",
      "global_id": "8a81848d4d47af63014d47b847230003",
      "name": "Debian Infra",
      "description": "Debian Infra",
```

```
      "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
      "ext": {
        "csa_name_key": "e67b408c66d840a788593690a665e0d3",
        "csa_critical_system_object": false
      },

...
```


For complete output, see "List topology design containers matching a filter on tag and type example" on page 162

List user access details for a specified container

Lists the user access for a specified container. Details about users and groups with access to the specified container are returned in the response.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/container/useraccess/{containerID}	GET	Container ID	200 - successful 400 - bad values in Container ID parameter 403 - not authorized 404 - the container does not exist

Example

The following URL was sent:

`https://localhost:8444/csa/api/container/useraccess/72A82AD399D64A1FADABE348AA59DB90`

The following JSON was returned:


```
{
  "name": "Container1",
  "displayName": "Container1",
  "description": "",
  "everyOneAccess": false,
  "members": [
    {
      "cn": "anbu",
      "dn": "CN=anbu,CN=Users",
      "emailid": "user1@hp.com",
      "type": "user",
      "roles": [
        {
          "role": "Application Developer"
        }
      ]
```

```
    },
    {
      "cn": "G1",
      "dn": "CN=G1,CN=Users",
      "type": "group",
      "roles": [
        {
          "role": "External Group"
        }
      ]
    }
  ]
}
```

Manage user access control on a container

Adds LDAP users and groups to the system and provides user access to a container specified by the Container ID parameter. To grant access on a container for all users, set the attribute "everyOneAccess" : "true".

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/container/useraccess/ {containerID}	POST	• Container ID • Body - A JSON containing the organization name, everyoneAccess attribute and group details.	200 - successful 400 - bad request 403 - not authorized 404 - not found

Example

The following URL was sent:

`https://localhost:8444/csa/api/container/useraccess/72A82AD399D64A1FADABE348AA59DB90`

The following JSON was provided in the RequestBody:

```
{
 "organizationName": "CSA-Provider",
 "everyOneAccess": "false",
 "members": [
   {
     "cn": "john",
     "dn": "CN=john,CN=Users",
     "emailid": "",
     "userAvatar": "",
```

```
     "memberOf": "CN=ApplicationDeveloper,CN=CSA_BLR_USERS",
     "type": "user"
   },
   {
    "cn" : "G1",
    "dn": "CN=G1,CN=Users",
    "emailid" : "",
    "type" : "group"
   }
 ]
}
```

There is no response output, and an empty JSON is returned.

Update a container

Updates an artifact container.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/container/{containerID}	PUT	• Container ID • body - a JSON representation of a service design container	200 - successful 400 - bad request 403 - not authorized

Example

The following URL was sent:

```
https://localhost:8444/csa/api/container/72A82AD399D64A1FADABE348AA59DB90
```

The following JSON was sent:

```
{
  "name": "Updated Sample Container",
  "description": "Updated Sample Container Description",
  "icon": "/csa/api/blobstore/Arrow_02_48.png?tag=library",
  "tags": [
    {"@self": "/csa/api/tag/8a81848d4d6cd612014d6fdee47f0036"}
],
"container_type": "topology_artifact_container"
}
```

The following JSON was returned:

```
{
 "@self":
"/csa/api/container/8a81848d4d6cd612014d7044313d00c1",
```

```
 "@type": "urn:x-hp:2012:software:cloud:data_model:metamodel",
 "@created": "2015-05-20T07:40:26.813Z",
 "@modified": "2015-05-20T07:40:26.813Z",
 "global_id": "8a81848d4d6cd612014d7044313d00c1",
 "name": "Updated Sample Container",
 "description": "Updated Sample Container Description",
 "icon": "/csa/api/blobstore/Arrow_02_48.png?tag=library",
 "ext": {
  "csa_name_key": "fb1e1294-6e5b-4323-a0cc-7414234c10a1",
  "csa_critical_system_object": false,
  "csa_artifact_container_type": "TOPOLOGY_DESIGN_BASED"
 },
 "container_type": "TOPOLOGY_ARTIFACT_CONTAINER",
 "tags": [
   {
    "@self": "/csa/api/tag/8a81848d4d6cd612014d6fdee47f0036",
    "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
    "name": "Developer Desings",
    "description": "Developer Desings",
    "icon": "/csa/api/blobstore/other.png?tag=library",
    "color": "#ffffff",
    "scopes": [
     "TOPOLOGY_ARTIFACT_CONTAINER"
   ]
  }
 ]
}
```


Delete a container

Deletes a container specified by the Container ID parameter.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/container/{containerID}	DELETE	Container ID	200 - successful 403 - not authorized 404 - not found 409 - invalid delete request

Example

The following URL was sent:

`https://localhost:8444/csa/api/container/72A82AD399D64A1FADABE348AA59DB90`

The response body is empty and the status code is 2.4.

LDAP APIs

URI	Method	See also
csa/api/ldap/organization/{orgName}/type/{type}	GET	"Get LDAP users and groups" below

Get LDAP users and groups

Returns a list of user and groups in the configured LDAP server at the organization provider level.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/ldap/organization/ {orgName}/type/{type}	POST	• orgName - Organization name. For example, CSA-Provider. • type - User or group. • body- JSON input body. Search based on cn and roles **Note:** This API supports a wild card search for the JSON input parameter cn. For example, cn with "app" as the search term returns all users and groups whose name starts with "app."	200 - successful 400 - bad request 401 - not authorized 404 - not found 501 - Internal server error

Example

The following JSON was sent:

```
{
"cn":"A",
"roles":["CODAR_APPLICATION_DEVELOPER","CODAR_APPLICATION_QA","CODAR_APPLICATION_
ARCHITECT"]
}
```

The following JSON was returned:

```
 {
 "@self": "/csa/api/ldap/",
 "@total_results": 3,
 "Organization": "/csa/CSA-Provider",
 "members": [
 {
 "cn": "AppArchitect",
 "dn": "CN=AppArchitect,CN=Users",
 "emailid": "",
```

```
"userAvatar": "https://10.1.9.115:8884/csa/thumbnails/apparchitect.jpg",
"memberOf": "CN=ApplicationArchitect",
"role":"Application Architect",
"type": "user"
},
{
"cn": "appdev",
"dn": "CN=appdev,CN=Users",
"emailid": "",
"userAvatar": "https://10.1.9.115:8884/csa/thumbnails/dev.jpg",
"memberOf": "CN=ApplicationDeveloper,CN=ARA Group",
"role": "Application Developer",
"type": "user"
},
{
"cn": "appdev1",
"dn": "CN=appdev1, CN=Users",
"emailid": "",
"userAvatar":"",
"memberOf": "CN=ApplicationDeveloper",
"role": "Application Developer",
"type": "user"
}
]
}
```


Package APIs

URI	Method	See also
/csa/api/codar/app-package/list	GET	"List packages" on the next page
/csa/api/codar/app-package/{applicationPackageID}	GET	"Get application package details" on page 68
/csa/api/codar/app-package/states	GET	"Get package states" on page 69
/csa/api/codar/app-package/{packageID}/properties	GET	"Get package properties" on page 70
/csa/api/codar/app-package/composition/{topologyID} /candidateTopology	GET	"List candidate designs" on page 71
/csa/api//codar/app-package/composition/{topologyId} /stage/{lifecycleStage}	GET	"List candidate topologies" on page 72
/csa/api/codar/app-package/{packageID} /activeDeployments	GET	"Get a list of active deployments" on page 72
/csa/api/codar/app-package	POST	"Create a package" on page 73

URI	Method	See also
/csa/api/codar/app-package/createWithProperties	POST	"Create package with properties" on page 74
/csa/api/codar/app-package/{packageID}/promote	POST	"Promote a package" on page 75
/csa/api/codar/app-package/{packageID}/reject	POST	"Reject a package" on page 76
/csa/api/codar/app-package/{packageID}/deploy	POST	"Deploy a package" on page 76
/csa/api/codar/app-package/{packageID}/redeploy	POST	"Redeploy a package" on page 78
/csa/api/codar/app-package/delete	POST	"Delete multiple packages" on page 79
/csa/api/codar/app-package/{packageID}	PUT	"Update package name and description" on page 80
/csa/api/codar/app-package/{packageID}/properties	PUT	"Update package component properties" on page 80
/csa/api/codar/app-package/{packageID}	DELETE	"Delete a package" on page 81

List packages

Retrieve a list of all packages for an application design. The Redeploy field indicates if the design can be redeployed. The Deployment State field indicates NEW when a package is created, DEPLOYED when a package is deployed and NO_ACTIVE_DEPLOYMENTS when there are no deployments.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/codar/app-package/list	GET	• applicationDesignID - Obtains the list of application packages for the specified application Design ID • lifecycleStage - Obtains the list of application packages for the specified life cycle stage. • startIndex - Specifies the offset of the first entry to be included in the page. • pageSize - Specifies the number of package deployment instances to be returned.	200 - successful 400 - bad request 401 - not authorized 404 - not found 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/list
```

The following JSON was returned:

```
{
  "@self": "/csa/api/package/",
  "@type": "urn:x-hp:2012:software:cloud:data_model:package",
  "redeploy": false,
  "members": [
   {
     "@type": "urn:x-hp:2012:software:cloud:data_model:package",
     "name": "DEVELOPMENT",
     "color": "#D1AF89",
     "image": "images/applications/dev.png",
     ...
}
```

For complete output, see "List packages example" on page 163.

Get application package details

Retrieves application package details including topology, deployed instance, and life cycle stage.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/codar/app-package/ {applicationPackageID}	GET	• ApplicationpackageID - Obtains the list of application packages for the specified application package ID. • pageSize - Specifies the number of package deployment instances to be returned. • startIndex - Specifies the offset of the first entry to be included in the page.	200 - successful 400 - bad request 401 - not authorized 404 - not found 500 - server exception

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/5975ad69-0519-484b-bde3-30b936dd31b
```

The following JSON was returned:

```
{
  "@self": "/csa/api/package/455274d8-80dc-4bff-a6cd-4d15408bffe0",
  "@type": "urn:x-hp:2012:software:cloud:data_model:package",
  "stage": "TESTING",
  "state": "ACTIVE",
  "deploymentState": "DEPLOYED",
  "icon": "images/applications/dev.png",
  "name": "package 3"
  "deployedInstanceCount": "1",
```

```
  "lastUpdated": "2015-03-06T04:34:59.087Z",
  "description": null,
  "topologyData": {
    "id": "b2c076af-7992-4301-bc88-b0a392a189f6",
    "partialDesign": false,
    "name": "vCenter Design",
    "iconurl": "/csa/images/library/HomeServer.png",
    "version": "1.0.0"
  },
  "instanceData": [
    {
      "statusDisplayName": "Deploying",
      "createdOn": "2015-03-09T06:09:30.425Z",
      "id": "8a8184694bee585b014bfd270f390004",
      "blueprintId": "8a8184694be3279e014be3921e4702d6",
      "createdBy": "admin"
      "status": "DEPLOYING",
      "iconUrl": "/csa/images/categories/service_instance_state/deploying.png",
      "deploymentStage": "TESTING"
      "displayName": "package 3:deploy"
    }
  ]
}
```


Get package states

Retrieves package states.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-package/states	GET	None	200 - successful 400 - bad request 401 - not authorized 404 - not found 500 - internal server error

Example

The following URL was sent:

`https://localhost:8444/csa/api/codar/app-package/states`

The following JSON was returned:


```
{
  "@total_results": 2,
  "@self": "/csa/api/package/states",
  "@type": "urn:x-hp:2012:software:cloud:data_model:packageState",
  "members": [
  {
```

```
    "name": "ACTIVE"
  },
  {
    "name": "REJECTED"
    ]
 }
```

Get package properties

Retrieve the properties that are parametrized in the package for each component of the application design.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-package/{packageID}/properties	GET	Package ID	200 - successful 400 - bad request 404 - not found

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/07c979d2-afe0-4cc8-a607-8cbf780ed725/properties
```

The following Parameter was sent:

```
packageId
```

The following JSON was returned:

```
{
 "members" : [ {
    "icon" : "/csa/designer//csa/api/blobstore/Tools1.png?tag=library",
    "@self" : "/csa/api/package/component/New_PetDB_855ff95b_fde7_432e_b4dd_
b7f7e8c2ba67_e6b6ca0914a34eec93438670d70e55e4__VERSION__1__GROUPID__
com.hp.csa.type0002",
    "description" : "Creates Pet Clinic Database on the database server",
    "@type" : "urn:x-hp:2012:software:cloud:data_model:package",
    "name" : "PetClinic DB Conf",
    "provider" : null,
    "properties" : [ {
      "modifiable" : false,
      "name" : "mysqlusername",
      "value" : "root",
      "enumeration" : null,
      "displayName" : "mysqlusername",
      "type" : "String",
```

```
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
     ...
 }
```

For complete output, see "Get package properties example" on page 165

List candidate designs

Retrieve the service designs for a given partial design (Topology Composition) of the application design.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-package/composition/{topologyID}/candidateTopology	GET	Topology ID	200 - successful 400 - bad request 404 - not found

Example

The following URL was sent:

`https://localhost:8444/csa/api/codar/app-package/composition/902442d3-700c-4b57-8abb-8d1a52ae3f3d/candidateTopology`

The following JSON was sent:

```
{
"count": 5,
"members" : [ {
"@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
"@self" : "/csa/api/topology-model/topology/7843ee06-8a5e-425e-ac0d-424e3a297d52",
"id" : "7843ee06-8a5e-425e-ac0d-424e3a297d52",

"iconUrl" : "/csa/designer/img/noimage-lightgray.svg",
"displayName" : "Infra Design",
"description" : "dfs",
...
}
```

For complete output, see "List candidate designs example" on page 173.

List candidate topologies

Lists the candidate topologies that can fulfill the given partial topology for the specified lifecycle stage. This API requires lifecycleSatge and topologyId as the parameters.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api//codar/app-package/composition/{topologyId}/stage/{lifecycleStage}	GET	• lifecycleStage - Obtains the list of application packages for the specified life cycle stage. • topologyId	200 - successful 400 - bad request 401 - not authorized 404 - not found 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/composition/bda96081-ca7e-4b19-829c-a6ee42b4e551/stage/%2084905d24-d92f-4916-b312-78af206d31a5
```

The following JSON was returned:

```
{
 "@count": 0,
 "members": []
}
```

Get a list of active deployments

Returns a list of active deployments that belong to the logged-in user that are in the same life cycle stage as the specified package.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-package/{packageID}/activeDeployments	GET	Package ID	200 - successful 400 - bad request 401 - not authorized 404 - not found 501 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/7843ee06-8a5e-425e-ac0d-424e3a297d52/activeDeployments
```

The following JSON was returned:

```
{
   "members" : [
   {       "id" : "8a81841f4b396569014b397bfb6600f5",
           "displayName" : "Pkg1:D2"
   },
   {        "id": "8a81841f4b396569014b396c51170016",
            "displayName": "Pkg2:D1"
   }
  ]
}
```

Create a package

Creates a new package for the specified application design. The package is only created if the Codar license is present and the design is associated to the Release Pipeline.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-package	POST	JSON body	201 - successful 400 - bad request 404 - not found

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package
```

The following JSON was sent:

```
{
 "name":"HelloWorld Package 1",
 "description":"This is my first Package",
 "applicationDesignId" : "30d4458b-e54c-4e91-a144-026d09289ae0"
}
```

The following JSON was returned:

```
{
   "@self": "/csa/api/package/09ead497-82d1-4045-8b17-73122c3cc6be",
   "@type": "urn:x-hp:2012:software:cloud:data_model:package",
```

```
   "lifecycleStage": "DEVELOPMENT",
   "packageState": "ACTIVE",
   "applicationDesignId": "30d4458b-e54c-4e91-a144-026d09289ae0",
   "name": "HelloWorld Package 1",
   "description": "This is my first Package"
}
```

Create package with properties

Create a new package for the given application design with properties. The package is only created if the Codar license is present and the design is associated to the Release Pipeline. Only the required and modifiable properties of the components in the application can be parameterized in the package.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-package/createWithProperties	POST	JSON body	200 - successful 400 - bad request 401 - not authorized 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/createWithProperties
```

The following JSON was sent:

```
{
  "designId":"02e8c0cc-1dd9-43c7-941e-1252b6619cee",
 "continuousPromote":false
  "packageName":"Package1",
  "packageDescription":"description",
  "members": [
      {
        "icon": "pluginResources/topology/icons/SERVER.svg",
        "@self": "/csa/api/package/component/VcenterServerType__VERSION__04.20.0000__
GROUPID__com.hp.csa.type0001",
        "description": "vCenter Server",
        "@type": "urn:x-hp:2012:software:cloud:data_model:package",
        "name": "vCenter Server",
        "provider": "VMWARE_VCENTER",
        "displayName": "vCenter Server",
        "properties" : [
          {
           "name" : "vmTemplateReference",
           "value" : "UbuntuTemp"
          },
           {
```

```
            "name" : "vmNamePrefix",
            "value" : "Test"
            }
            ]
            },
              "displayName": "vCenter Server"
    ]
}
```

The following JSON was returned:

```
{
packageId": "bc384c91-6345-4fe1-886f-e3c973280d09"
}
```

Promote a package

Promote a package to the next lifecycle stage if no release gate definition exists for the current lifecycle stage. The out-of-the-box lifecycle stages are Development, Testing, Staging, and Production. Only one active package can be promoted to production at any point of time. If you want to promote another package to production, you must reject the package that is already in production and then promote the other package.

Once a package reaches production, it cannot be promoted further. A package in the rejected state cannot be promoted. Once a package is promoted to production, the entire design is locked to prevent further modification of the application design.

If a release gate definition exists for the current lifecycle stage, the package is not promoted instantly; instead it goes through the release gate actions that are defined. After executing the release gate, if the final status is successful, then the package moves to the next lifecycle stage.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/package/{packageID}/promote	POST	• Package ID • comments	200 - updated 400 - bad request 404 - not found

Example

The following URL was sent:

`https://localhost:8444/csa/api/codar/package/d05bb6ec-23bf-425a-9c90-240fbc8f99a9/promote`

The following JSON was returned:

```
{
 "promotedToStage": "TESTING",
 "packageName": "Demo_package",
```

```
 "packageId": "d05bb6ec-23bf-425a-9c90-240fbc8f99a9" }
}
```

Reject a package

Moves a package to the rejected state. A package in the rejected state cannot be deployed or promoted.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-package/{packageID}/reject	POST	Package ID	200 - successful 400 - bad request 401 - not authorized 404 - not found 500 - internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/package/a7fa2df4-9f23-4ba7-b8df-0a14f9eba221/reject
```

The following XML was sent:

```
Example XML sent in request, if applicable
```

The following XML was returned:

```
{
 "packageName": "Promote Package 2",
 "packageState": "REJECTED",
 "packageId": "a7fa2df4-9f23-4ba7-b8df-0a14f9eba221"
}
```

Deploy a package

Deploys a package for a complete or partial application design. This API maps service designs to the requirements they can fulfill, and creates a topology design composed of the service designs that satisfy each requirement. When the design ID of the service design is null, the designToRequirementsMapping is used and the design ID of each mapped object is used to replace the requirement component in the composed design. When the design ID of the service design has a value, a single design is chosen for composition.

Use the "List the candidate topologies that can fulfill the specified partial topology" API to obtain the design IDs of the candidate designs.

This API validates the deployment of a package and allows all environments associated with a container and life cycle stage. If there is no association, this API deploys packages in any environment.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/codar/app-package/ {packageID}/deploy	POST	• Package ID • JSON with runname, environment ID, service design, and modifiable properties of the package. **Note:** Profile ID is not used.	200 - successful 400 - bad request 404 - not found

Examples

In the following example, a single service design is selected for composition.

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/d8482010-89b8-42ab-8b69-1396267223cd/deploy
```

The following JSON was returned:


```
{
    "name": "deployName",
    "environmentId": "6ba19933-76df-4c5c-b784-fc22a704ec2b",
    "modifiableProperties": [{
        "id": "VcenterServerType__VERSION__04.10.0000__GROUPID__com.hp.csa.type0001",
        "itemType": "NODE",
        "propertyName": "vmTemplateReference",
        "propertyType": "string",
        "propertyValue": "csata-rhelsa915"
    },
    {
        "id": "VcenterServerType__VERSION__04.10.0000__GROUPID__com.hp.csa.type0001",
        "itemType": "NODE",
        "propertyName": "customizationSpec",
        "propertyType": "string",
        "propertyValue": "Linux"
    }],
    "infrastructureDesign": {
        "designId": "99a19933-76df-4c5c-b784-fc22a704ec26"
    },
    "designToRequirementsMap": [{
        "designId": "f1f1a314-5d73-4fed-9f93-fcb4165f8404",
        "requirementIds": ["Server__VERSION__1__GROUPID__com.hp.csa.type0001"]
    },
    {
        "designId": "f1f1a314-5d73-4fed-9f93-fcb4165f8404",
        "requirementIds": ["Server__VERSION__1__GROUPID__com.hp.csa.type0002"]
    },
    {
        "designId": "4d7a843a-64e0-4538-882f-889379a92101",
```

```
        "requirementIds": ["ApplicationServer__VERSION__1__GROUPID__
com.hp.csa.type0001"]
    }]
}
```

In the following example, the designtoRequirementsMapping is used, and service designs are selected per requirement and shared between multiple components. All of the requirement IDs that share the service design are provided for the same design ID object.

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/d8482010-89b8-42ab-8b69-
1396267223cd/deploy
```

The following JSON was returned:

```
{
  "serviceInstanceId" : "8a8186e24d9a968a014d9f4488220134"
}
```

Redeploy a package

Redeploys a package on an existing active deployment (service instance) belonging to same life cycle stage and user.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/codar/app-package/ {packageID}/redeploy	POST	Path Parameter - Packge ID to redeploy RequestBody - A JSON containing the serviceInstanceId on which to redeploy the package, and a redeployProperties.	200 - successful 400 - bad request 401 - not authorized 501 - internal server error

Example

The following JSON was sent in the requestBody parameter:

```
{ "serviceInstanceId":"8a81841f4b733315014b76c191b70099",
  "redeployProperties":[{"id":"VcenterServerType__VERSION__04.20.0000__GROUPID__
com.hp.csa.type0001",
      "itemType":"NODE",
      "propertyName":
      "cpuCount",
      "propertyType":"string",
      "propertyValue":"4"},
   {"id":"VcenterServerType__VERSION__04.20.0000__GROUPID__com.hp.csa.type0001",
     "itemType":"NODE",
```

```
     "propertyName":"memorySize",
     "propertyType":"string",
     "propertyValue":"1024"}
   ]
}
```

The following JSON was returned:

```
{ "serviceInstanceId":"8a81841f4b733315014b76c191b70099 }
```

Delete multiple packages

Deletes multiple packages specified by the user.

Packages that are in production stage cannot be deleted. If multiple packages are selected to be deleted, and a selected package is in production stage, an error is displayed an no packages are deleted.

A user in the developer role cannot delete packages that are in the testing stage. If multiple packages are selected by a developer to be deleted, and any of the selected packages are in testing stage, an error is displayed and no packages are deleted.

At least one package must be selected. If no packages are selected to be deleted, an error message is displayed.

Packages that have deployed instances cannot be deleted. You must cancel and delete the deployments before you can delete the package.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-package/delete	POST	Path - A list of package IDs.	200 - successful 400 - bad request 401 - not authorized 404 - not found 501 - internal server error

Example

The following URL was sent:

`https://localhost:8444/csa/api/codar/app-package/delete`

The following JSON was sent:

```
{
"members": [ {
"packageId":"2090bf73-0f8c-4c34-9737-f5f569c04a4d"},
 {
"packageId":"2090bf73-0f8c-4c34-9737-f5f569c03a3f"}
 ]
}
```

The following response body was returned if the packages are successfully deleted:

```
No contents
```

Update package name and description

Updates a package name and description.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-package/{packageID}	PUT	• Package ID • JSON body with package name and description	200 - successful 400 - bad request 404 - not found

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/5975ad69-0519-484b-bde3-30b936dd31b
```

The following JSON was sent:

```
{"name":"package Name","description":"Package Description"}
```

The following JSON was returned:

```
{
  "@self": "/csa/api/package/update/5410dd5a-7b03-4ea3-932a-3dcb108a09a9",
  "@type": "urn:x-hp:2012:software:cloud:data_model:package",
  "name": "package Name",
  "description": "Package Description"
}
```

Update package component properties

Updates the component properties for a package.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-package/{packageID}/properties	PUT	• Package ID • JSON body	200 - successful 400 - bad request 404 - not found

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/5975ad69-0519-484b-bde3-
30b936dd31b/properties
```

The following JSON was sent:

```
{
"members": [
      {
              "icon": "pluginResources/topology/icons/SERVER.svg",
              "@self": "/csa/api/package/component/VcenterServerType__VERSION__
04.10.0000__GROUPID__com.hp.csa.type0001",
              "description": "vCenter Server",
              "@type": "urn:x-hp:2012:software:cloud:data_model:package",
              "name": "vCenter Server",
              "provider": "VMWARE_VCENTER",
              "displayName": "vCenter Server",
              "properties" : [
                     {
                            "name" : "vmTemplateReference",
                            "value" : "ubuntu_tempref"
                            },{
                            "name" : "customizationSpec",
                            "value" : "linux_spec"
                            },{
                            "name" : "vmNamePrefix",
                            "value" : "app_"
                     }
              ],
              "displayName": "vCenter Server"
      }
]
}
```

The following JSON was returned:

```
No Contents
```

Delete a package

Deletes a package.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/app-package/{packageID}	DELETE	Package ID	204 - successful 400 - bad request 404 - not found

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/5975ad69-0519-484b-bde3-30b936dd31b
```

The following JSON was returned:

```
No contents
```

Release gate APIs

URI	Method	See also
/csa/api/codar/release-gate/gate-action/contextproperties	GET	"List context properties" on the next page
/csa/api//codar/release-gate/gate-action/properties/	GET	"List OO flow Id component properties" on page 84
/csa/api/codar/release-gate/gate-action/{actionId}	GET	"Retrieve a release gate action based on a specified action ID" on page 85
/csa/api/codar/release-gate/gate-action	GET	"List all release gate actions" on page 86
/csa/api/codar/release-gate/gate-action/types	GET	"Lists all release gates action types" on page 90
/csa/api/codar/release-gate/gate-action/{action}	DELETE	"Delete release gate actions" on page 91
/csa/api/codar/release-gate/gate-action/deploy	POST	"Create a deploy gate action" on page 91
/csa/api/codar/release-gate/gate-action/custom	POST	"Creates a custom action" on page 93
/csa/api/codar/release-gate/gate-action/approval	POST	"Add approval gate action" on page 94
/csa/api/codar/release-gate/gate-action/deploy/{actionId}	PUT	"Update deploy action details" on page 95
/csa/api/codar/release-gate/gate-action/custom/{actionId}	PUT	"Update custom action details" on page 96
/csa/api/codar/release-gate/gate-action/approval/{actionId}	PUT	"Updates approval action details" on page 97
/csa/api/codar/release-gate/gate-action/	PUT	"Update release gate actions order" on page 99

List context properties

Lists the following context properties:

- SOURCE_LIFECYCLE_STAGE: Source lifecycle stage for a package
- DESTINATION_LIFECYCLE_STAGE: Destination lifecycle stage for package
- PACKAGE_NAME: Package name
- ACTION_INSTANCE_ID: Action instance ID for the request promoted.
- USER: Requested user name

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-action/contextproperties	GET		200 - Success 400 - Bad request 401 - Authorization failure 404 - Not found 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/release-gate/gate-action/contextproperties
```

The following JSON was returned:


```
{
  "members": [
    {
      "displayName": "Source Lifecycle Stage",
      "name": "SOURCE_LIFECYCLE_STAGE"
    },
    {
      "displayName": "Destination Lifecycle Stage",
      "name": "DESTINATION_LIFECYCLE_STAGE"
    },
    {
      "displayName": "Package Name",
      "name": "PACKAGE_NAME"
    },
    {
      "displayName": "Action Instance Id",
      "name": "ACTION_INSTANCE_ID"
    },
    {
      "displayName": "Request User",
      "name": "USER"
```

```
    }
  ]
}
```

List OO flow Id component properties

Lists component properties for the OO flow Id. This API requires ooFlowId and providerInstanceId as parameters.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api//codar/release-gate/gate-action/properties/	GET	• ooFlowId • providerInstanceId	200 - Successful 400 - Bad request 401 - Authorization failure 404 - Not found 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/release-gate/gate-action/properties/?ooFlowId=22e2317c-20d8-4efc-b572-332f3222a801&providerInstanceId=6ba7b81c-9dad-11d1-80b4-00c04fd430c8
```

The following JSON was returned:

```
 {
  "@self" : "/csa/codar/release-gate/gate-action/properties",
  "@type" : "urn:x-hp:2012:software:cloud:data_model:ReleaseGateProperty",
  "id" : "22e2317c-20d8-4efc-b572-332f3222a801",
  "name" : "ExecuteDevelopmentTest",
  "path" : "/Library/Integrations/Hewlett-Packard/Cloud Service
Automation/Components/HPOO/Dummy Application/1.0",
  "input_properties" : [ {
    "id" : "7817109d-b58e-40b8-ba34-17038ec90c5c",
    "name" : "OutputFilePath",
    "description" : "",
    "mandatory" : false
  }, {
    "id" : "a5f148fb-c6f7-491f-af26-726727201260",
    "name" : "LIFECYCLE_PHASE",
    "description" : "",
    "mandatory" : false
  } ],
```

```
  "output_properties" : [ {
    "id" : null,
    "name" : "Result",
    "description" : "",
    "mandatory" : false
  } ]
}
```

Retrieve a release gate action based on a specified action ID

Retrieves the details of a release gate action based on the action type for a specified action ID. This API accepts *actionId* as input. *actionId* is mandatory and is the UUID of the release gate action.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-action/{actionId}	GET	actionId	200 - Success 400 - Bad request 401 - Authorization failure 404 - Not found 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/release-gate/gate-action/5791d61b-4541-411b-b397-172e50fe7319
```

The following JSON was returned:

```
{
  "@self": "/csa/codar/release-gate/gate-action/5791d61b-4541-411b-b397-
172e50fe7319",
  "id": "5791d61b-4541-411b-b397-172e50fe7319",
  "name": "deploy2",
  "description": "deploy two",
  "type": "DEPLOY",
  "typeDisplayName": "Deploy Action",
  "order": 2,
  "enabled": true,
  "lifecycleStage": "DEVELOPMENT",
  "createdBy": "admin",
  "createdOn": "2015-12-22T04:31:58.697Z",
  "lastUpdated": "2015-12-22T04:31:58.697Z",
  "updatedBy": "admin",
  "rejectOnFailure": true,
```

```
  "notifyOnSuccess": false,
  "notifyOnFailure": false,
  "deployAction": {
    "deploymentName": "deploy2",
    "environmentId": "null",
    "cleanupOnSuccess": true,
    "cleanupOnFailure": true,
    "designType": "PARTIAL",
    "infraDesignParameters": [
      {
        "requirementId": "Complete",
        "modelId": "243c7c75-ec05-40cd-9f36-9fa45751a76c"
      }
    ]
  }
}
```


List all release gate actions

Lists all the release gate actions, which includes deploy, custom, and approval actions. This API accepts applicationDesignId and lifecycleStage as input. The applicationDesignId parameter is mandatory and is the UUID of the application design. The lifecycleStage parameter is optional. The applicationDesignId and lifecycleStage parameters must be valid.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-action	GET	• applicationDesignId • (Optional) lifecycleStage	200 - successful 400 - bad request 404 - not found 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/release-gate/gate-
action/?applicationDesignId=4ed560b1-a533-43c0-ae7b-eb635dd3a3cc
```

The following JSON was returned:


```
{
 "@total_results": "4",
 "@self": "/csa/codar/release-gate/gate-action/",
 "@type": "urn:x-hp:2012:software:cloud:data_model:ReleaseGateAction",
 "members": [
   {
     "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
```

```
78af206d31a5",
      "id": "84905d24-d92f-4916-b312-78af206d31a5",
      "name": "DEVELOPMENT",
      "displayName": "Development",
      "count": "8",
      "color": "#ffb878",
      "icon": "/csa/api/blobstore/lifecycle_stage_development.png?tag=library",
      "actions": [
        {
          "@self": "/csa/codar/release-gate/gate-action/6decf992-8d51-41d6-b994-
8f7b5f6a7425",
          "id": "6decf992-8d51-41d6-b994-8f7b5f6a7425",
          "name": "Deploy1",
          "description": null,
          "type": "DEPLOY",
          "typeDisplayName": "Deploy Action",
          "order": 0,
          "enabled": true,
          "lifecycleStage": "DEVELOPMENT",
          "createdBy": "admin",
          "createdOn": "2015-11-25T07:01:12.657Z",
          "rejectOnFailure": false
        },
        {
          "@self": "/csa/codar/release-gate/gate-action/1ce957c0-f49c-4048-9989-
fedcffb7bda8",
          "id": "1ce957c0-f49c-4048-9989-fedcffb7bda8",
          "name": "deploy3",
          "description": "Action description",
          "type": "DEPLOY",
          "typeDisplayName": "Deploy Action",
          "order": 1,
          "enabled": true,
          "lifecycleStage": "DEVELOPMENT",
          "createdBy": "admin",
          "createdOn": "2015-11-26T07:31:04.041Z",
          "rejectOnFailure": true
        },
        {
          "@self": "/csa/codar/release-gate/gate-action/80862c3f-dcb7-46b4-8c1d-
4d024cbbe2f8",
          "id": "80862c3f-dcb7-46b4-8c1d-4d024cbbe2f8",
          "name": "deploy2 kunal",
          "description": "",
          "type": "DEPLOY",
          "typeDisplayName": "Deploy Action",
          "order": 1,
          "enabled": true,
          "lifecycleStage": "DEVELOPMENT",
```

```
          "createdBy": "admin",
          "createdOn": "2015-11-26T06:35:48.686Z",
          "rejectOnFailure": false
        },
        {
          "@self": "/csa/codar/release-gate/gate-action/10c04bc8-2f8d-4ff0-bb8e-
97b029e83f20",
          "id": "10c04bc8-2f8d-4ff0-bb8e-97b029e83f20",
          "name": "Custom2",
          "description": "Custom test",
          "type": "CUSTOM",
          "typeDisplayName": "Custom Action",
          "order": 2,
          "enabled": true,
          "lifecycleStage": "DEVELOPMENT",
          "createdBy": "admin",
          "createdOn": "2015-12-02T09:13:56.241Z",
          "rejectOnFailure": false
        },
        {
          "@self": "/csa/codar/release-gate/gate-action/7ed36412-2289-4822-a66d-
70c85292dc49",
          "id": "7ed36412-2289-4822-a66d-70c85292dc49",
          "name": "Custom action3",
          "description": "Custom action for test",
          "type": "CUSTOM",
          "typeDisplayName": "Custom Action",
          "order": 3,
          "enabled": true,
          "lifecycleStage": "DEVELOPMENT",
          "createdBy": "admin",
          "createdOn": "2015-12-02T10:07:41.624Z",
          "rejectOnFailure": false
        },
        {
          "@self": "/csa/codar/release-gate/gate-action/aa34c2d0-8e67-4d9a-abcf-
5b8b50c77355",
          "id": "aa34c2d0-8e67-4d9a-abcf-5b8b50c77355",
          "name": "Approval",
          "description": "approval test",
          "type": "APPROVAL",
          "typeDisplayName": "Approval Action",
          "order": 4,
          "enabled": true,
          "lifecycleStage": "DEVELOPMENT",
          "createdBy": "admin",
          "createdOn": "2015-12-02T10:46:40.719Z",
          "rejectOnFailure": false
        },
```

```
        {
          "@self": "/csa/codar/release-gate/gate-action/6d3b3708-c417-47e0-8a49-
d3d90bcaf3bf",
          "id": "6d3b3708-c417-47e0-8a49-d3d90bcaf3bf",
          "name": "custom1",
          "description": "Action description",
          "type": "CUSTOM",
          "typeDisplayName": "Custom Action",
          "order": 1,
          "enabled": true,
          "lifecycleStage": "DEVELOPMENT",
          "createdBy": "admin",
          "createdOn": "2015-11-26T11:20:50.713Z",
          "rejectOnFailure": true
        },
        {
          "@self": "/csa/codar/release-gate/gate-action/d4b016a5-47f6-4db3-88ab-
6e494a2039e0",
          "id": "d4b016a5-47f6-4db3-88ab-6e494a2039e0",
          "name": "Approval1",
          "description": "approval test1",
          "type": "APPROVAL",
          "typeDisplayName": "Approval Action",
          "order": 5,
          "enabled": true,
          "lifecycleStage": "DEVELOPMENT",
          "createdBy": "admin",
          "createdOn": "2015-12-02T11:12:32.556Z",
          "rejectOnFailure": false
        }
      ]
    },
    {
      "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a6",
      "id": "84905d24-d92f-4916-b312-78af206d31a6",
      "name": "TESTING",
      "displayName": "Testing",
      "count": "0",
      "color": "#fbd75b",
      "icon": "/csa/api/blobstore/lifecycle_stage_testing.png?tag=library",
      "actions": []
    },
    {
      "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a7",
      "id": "84905d24-d92f-4916-b312-78af206d31a7",
      "name": "STAGING",
      "displayName": "Staging",
```

```
      "count": "0",
      "color": "#dbadff",
      "icon": "/csa/api/blobstore/lifecycle_stage_staging.png?tag=library",
      "actions": []
    },
    {
      "@self": "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a8",
      "id": "84905d24-d92f-4916-b312-78af206d31a8",
      "name": "PRODUCTION",
      "displayName": "Production",
      "count": "0",
      "color": "#ff887c",
      "icon": "/csa/api/blobstore/lifecycle_stage_production.png?tag=library",
      "actions": []
    }
  ]
}
```


Lists all release gates action types

Lists all release gate action types available.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api/codar/release-gate/gate-action/types	GET		200 - Successful 400 - Bad request 401 - Authorization failure 404 - Not found 500 - Internal server error

Example

The following URL was sent:

`https://localhost:8444csa/api/codar/release-gate/gate-action/types`

The following JSON was returned:


```
{
  "@total_results": "3",
  "@self": "/csa/codar/release-gate/gate-action/",
  "@type": "urn:x-hp:2012:software:cloud:data_model:ReleaseGateAction",
  "members": [
    {
      "name": "DEPLOY",
      "displayName": "Deploy Action"
    },
    {
      "name": "CUSTOM",
```

```
      "displayName": "Custom Action"
    },
    {
      "name": "APPROVAL",
      "displayName": "Approval Action"
    }
  ]
}
```

Delete release gate actions

Deletes release gate actions. This API takes the action ID as the parameter.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-action/{action}	DELETE	actionId	200 - updated 400 - bad request 404 - not found 500 - Server Exception

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/release-gate/action/6d3b3708-c417-47e0-8a49-d3d90bcaf3bf
```

Create a deploy gate action

Creates a deploy gate action.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-action/deploy	POST		200 - updated 400 - bad request 401 - Authorization failure 404 - not found 500 - Server Exception

Example

The following URL was sent for the deploy action:

```
https://localhost:8444/csa/api/codar/release-gate/gate-action/deploy
```

The following JSON was sent for creating the deploy action:

```
{
"name": "Test Action",
"description": "Action description" ,
"enabled": "true",
"order": "1",
"notifyOnSuccess": "true",
"notifyOnFailure": "true",
"applicationDesignId": "ce61114c-e22a-435f-b967-e04758242572",
"stage": "DEVELOPMENT",
"rejectOnFailure": "true",
"environmentId": "adf557f-bh5t-hj66-df88-a56sdf6",
"cleanupOnSuccess": true,
"cleanupOnFailure": true,
"infra design parameters": [
{"requirementId": "agh568t-bh5t-hj66-df88-a56sdf6","modelId": "hjdx567-tg66-iu69-rg87-ad546gg"},
{"requirementId": "sdfsf786sdf-bh5t-hj66-987sd-896sdfsdf89","modelId": "sdfds876-tg66-iu69-rg87-sds57s8d"}
]
}
```

Note:

- If you are deploying a package from a complete design, the value of infraDesign parameters (requirementId and modelId) should be null.

  Example: `"infraDesignParameters": []`

- If you are deploying a package from a partial design, you can provide Infradesign parameter values for either a single service design or microservices that meet the requirements of the design.

  Example for a single service design:

  ```
  "infraDesignParameters": [{
                  "requirementId": "Complete",
                  "modelId": "243c7c75-ec05-40cd-9f36-9fa45751a76c"
         }]
  }
  ```

  Example for microservices

  ```
  "infraDesignParameters": [{
                  "requirementId": "ce36905b-21d5-ef3d-8384-dc895d0c7fd4",
                  "modelId": "243c7c75-ec05-40cd-9f36-9fa45751a76c"
         }, {
                  "requirementId": "857590ab-1031-82db-208e-f50468ac4550",
  ```

```
                "modelId": "243c7c75-ec05-40cd-9f36-9fa45751a76c"
        }, {
                "requirementId": "8f8fc9a5-ea64-4671-6b0a-cdadcbc87885",
                "modelId": "243c7c75-ec05-40cd-9f36-9fa45751a76c"
        }]
  }
```

The following is the response JSON for the deploy action:

```
{
 "@self": "/csa/codar/release-gate/action/",
 "@type": "urn:x-hp:2012:software:cloud:data_model:ReleaseGateAction",
 "id": "9cc4b130-fcd3-4f9c-8af4-bf6e82e43136",
 "actionName": "XYZ",
 "actionDescription": "Action description",
 "actionType": "DEPLOY",
 "createdBy": "admin",
 "createdOn": "2015-09-15T08:50:48.522Z"
}
```

Creates a custom action

Creates a custom action for a specified lifecycle stage of an application design.

Ensure that you have created a deploy action before creating a custom action.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-action/custom	POST		200 - updated 400 - bad request 401 - Authorization failure 404 - not found 500 - Server Exception

Example

The following URL was sent for the custom action:

```
https://localhost:8444/csa/api/codar/release-gate/gate-action/custom
```

The following JSON was sent for the custom action:

```
{
"name": "Custom action3",
"description": "Custom action for test" ,
"enabled": "true",
"applicationDesignId": "4ed560b1-a533-43c0-ae7b-eb635dd3a3cc",
"stage": "DEVELOPMENT",
"notifyOnSuccess": "false",
"notifyOnFailure": "false",
"ignoreOnFailure": "false",
"rejectOnFailure": "false",
"deployActionId": "1ce957c0-f49c-4048-9989-fedcffb7bda8",
"engineProcessId": "22e2317c-20d8-4efc-b572-332f3222a801",
"engineProcessName": "ExecuteDevelopmentTest",
"properties": [
0: {name: "OutputFilePath", mandatory: false, description: ""}
1: {name: "LIFECYCLE_PHASE", mandatory: false, description: ""}
]
}
```

The following is the response JSON for the custom action:

```
{
  "@self" : "/csa/codar/release-gate/gate-action/7ed36412-2289-4822-a66d-70c85292dc49",
  "@type" : "urn:x-hp:2012:software:cloud:data_model:ReleaseGateAction",
  "id" : "7ed36412-2289-4822-a66d-70c85292dc49",
  "name" : "Custom action3",
  "description" : "Custom action for test",
  "type" : "CUSTOM",
  "createdBy" : "admin",
  "createdOn" : "2015-12-02T10:07:41.624Z"
}
```

Add approval gate action

Adds approval gate action to a specified lifecycle stage of an application design.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-action/approval	POST		200 - updated 400 - bad request 401 - Authorization failure 404 - not

URI	Method	Parameters	Returns
			found 500 - Server Exception

Example

The following URL was sent to add approval gate action:

```
https://localhost:8444/csa/api/codar/release-gate/gate-action/approval
```

The following JSON was sent for the approval action:

```
{
 "name": "Approval1",
 "description": "approval test1" ,
 "enabled": "true",
 "applicationDesignId": "4ed560b1-a533-43c0-ae7b-eb635dd3a3cc",
 "stage": "DEVELOPMENT",
 "templateType": "NAMED_APPROVER_TEMPLATE",
  "automation": "true",
  "waitTime": "0",
  "replyType": "Approved",
  "minimumApprovals": "1",
  "minimumDenials": "1",
  "approvers":[
{"personId": "8a818d5851613d3b0151624ced9c045c",
  "userName":"dharmaarchitect"}
]
}
```

The following is the response JSON:

```
{
  "@self": "/csa/codar/release-gate/gate-action/d4b016a5-47f6-4db3-88ab-6e494a2039e0",
  "@type": "urn:x-hp:2012:software:cloud:data_model:ReleaseGateAction",
  "id": "d4b016a5-47f6-4db3-88ab-6e494a2039e0",
  "name": "Approval1",
  "description": "approval test1",
  "type": "APPROVAL",
  "createdBy": "admin",
  "createdOn": "2015-12-02T11:12:32.556Z"
}
```

Update deploy action details

Updates deploy action details. This API takes actionId of the deploy action as the parameter.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-action/deploy/{actionId}	PUT	actionId	200 - updated 400 - bad request 404 - not found

Example

The following URL was sent to update the deploy action:

```
https://localhost:8444/csa/api//codar/release-gate/gate-action/deploy/6decf992-8d51-41d6-b994-8f7b5f6a7425
```

The following JSON was sent for the update deploy action:

```
{
"name":"Deploy1",
"description":"",
"enabled":"true",
"order":"2",
"notifyOnSuccess":"false",
"notifyOnFailure":"true",
"rejectOnFailure":"false",
"environmentId":"8a818d585147a8ac015147af240b0009",
}
```

following is the response JSON for the update deploy action:

```
{
  "@self": "/csa/codar/release-gate/gate-action/6decf992-8d51-41d6-b994-8f7b5f6a7425",
  "@type": "urn:x-hp:2012:software:cloud:data_model:ReleaseGateAction",
  "id": "6decf992-8d51-41d6-b994-8f7b5f6a7425",
  "name": "Deploy1",
  "description": "",
  "updatedBy": "admin",
  "lastUpdated": "2015-12-02T11:58:06.070Z"
}
```

Update custom action details

Updates custom action details. This API takes actionId of the custom action as the parameter.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-action/custom/{actionId}	PUT	actionId	200 - updated 400 - bad request 404 - not found

Example

The following URL was sent to update the custom action:

```
https://localhost:8444/csa/api/codar/release-gate/gate-action/custom/6d3b3708-c417-47e0-8a49-d3d90bcaf3bf
```

The following JSON was sent for the update custom action:

```
{
"name": "custom1",
"description": "Action description" ,
"enabled": "true",
"notifyOnSuccess": "true",
"notifyOnFailure": "true",
"ignoreOnFailure": "true",
"rejectOnFailure": "true",
"deployActionId": "80862c3f-dcb7-46b4-8c1d-4d024cbbe2f8",
"engineProcessId": "80862c3f-dcb7-46b4-8c1d-4d024cbbe2f8",
"engineProcessName": "Test App OO Flow"
}
```

The following is the response JSON for the update custom action:

```
{
  "@self" : "/csa/codar/release-gate/gate-action/6d3b3708-c417-47e0-8a49-d3d90bcaf3bf",
  "@type" : "urn:x-hp:2012:software:cloud:data_model:ReleaseGateAction",
  "id" : "6d3b3708-c417-47e0-8a49-d3d90bcaf3bf",
  "name" : "custom1",
  "description" : "Action description",
  "updatedBy" : "admin",
  "lastUpdated" : "2015-12-02T10:50:46.983Z"
}
```

Updates approval action details

Updates approval action details. This API takes actionId of the approval action as the parameter.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-action/approval/{actionId}	PUT	actionId	200 - updated 400 - bad request 404 - not found

Example

The following URL was sent to update the approval action:

```
https://localhost:8444/csa/api/coda/release-gate/gate-action/approval/d4b016a5-47f6-4db3-88ab-6e494a2039e0
```

The following JSON was sent for the update approval action:

```
{
"name": "Approval1",
"description": "approval test1" ,
"enabled": "true",
"applicationDesignId": "4ed560b1-a533-43c0-ae7b-eb635dd3a3cc",
"stage": "DEVELOPMENT",
"templateType": "NAMED_APPROVER_TEMPLATE",
 "automation": "true",
 "waitTime": "0",
 "replyType": "Approved",
 "minimumApprovals": "1",
 "minimumDenials": "1",
 "approvers":[
{"personId": "8a818d5851613d3b0151624ced9c045c",
 "userName":"dharmaarchitect"}
]
}
```

The following is the response JSON for the update approval action:

```
{
  "@self": "/csa/codar/release-gate/gate-action/d4b016a5-47f6-4db3-88ab-6e494a2039e0",
  "@type": "urn:x-hp:2012:software:cloud:data_model:ReleaseGateAction",
  "id": "d4b016a5-47f6-4db3-88ab-6e494a2039e0",
  "name": "Approval1",
  "description": "approval test1",
  "updatedBy": "admin",
```

```
  "lastUpdated": "2015-12-02T11:29:27.162Z"
}
```

Update release gate actions order

Modify the properties and order of release gate actions. With this API, you can update multiple actions at one time.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-action/	PUT		200 - Updated 400 - bad request 401 - Authorization failure 404 - not found 500 - Internal server error

Example

The following URL was sent to update the release gate action:

```
https://localhost:8444/csa/api/codar/release-gate/gate-action/
```

The following JSON was sent for the update action:

```
{
"members": [
{"id": "1ce957c0-f49c-4048-9989-fedcffb7bda8",
"name": "Deploy Action",
"enabled": "true",
"order":"1"
},
{"id": "6d3b3708-c417-47e0-8a49-d3d90bcaf3bf",
"name": "Custom Action",
"enabled": "true",
"order":"1"
}
]
```

The following is the response JSON for the update action:

```
{
 "@total_results": "2",
 "@self": "/csa/codar/release-gate/gate-action/",
 "@type": "urn:x-hp:2012:software:cloud:data_model:ReleaseGateAction",
 "members": [
  {
   "@self": "/csa/codar/release-gate/gate-action/1ce957c0-f49c-4048-9989-fedcffb7bda8",
   "id": "1ce957c0-f49c-4048-9989-fedcffb7bda8",
   "name": "Deploy Action",
   "description": "",
   "type": "DEPLOY",
   "typeDisplayName": "Deploy Action",
   "order": 1,
   "enabled": true,
   "lifecycleStage": "DEVELOPMENT",
   "createdBy": "admin",
   "createdOn": "2015-11-26T07:31:04.041Z"
  },
  {
   "@self": "/csa/codar/release-gate/gate-action/6d3b3708-c417-47e0-8a49-d3d90bcaf3bf",
   "id": "6d3b3708-c417-47e0-8a49-d3d90bcaf3bf",
   "name": "Custom Action",
   "description": null,
   "type": "CUSTOM",
   "typeDisplayName": "Custom Action",
   "order": 1,
   "enabled": true,
   "lifecycleStage": "DEVELOPMENT",
   "createdBy": "admin",
   "createdOn": "2015-11-26T11:20:50.713Z"
  }
 ]
}
```


Gate requests APIs

URI	Method	See also
/csa/api/codar/release-gate/gate-request/{packageId}	GET	"List all promotion requests" on the next page
/csa/api/codar/release-gate/gate-request/{releaseGateRequestId}/action-instances	GET	"List all action instances" on the next page
/csa/api/codar/release-gate/gate-request/{releaseGateRequestId}	POST	"Update status of release gate request" on page 102

URI	Method	See also
/csa/api/codar/release-gate/gate-request/{releaseGateRequestId}	DELETE	"Delete release gate actions" on page 91

List all promotion requests

Lists all promotion requests. This API takes the PackageId and (optional) lifecycleStage as parameters.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-request/{packageId}	GET	• PackageId • (*Optional*) lifecycleStage	200 - successful 400 - bad request 404 - not found 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/release-gate/gate-request/a3eb6d50-5b6a-4dad-ac7b-fc81032c81c2
```

The following JSON was returned:


```
{
  "@type": "urn:x-hp:2012:software:cloud:data_model:ReleaseGateRequest",
  "@total_results": "1",
  "members": [
    {
      "@self": "/csa/codar/release-gate/gate-request/48feb259-3d24-464b-be59-
1661b5fb7266",
      "id": "48feb259-3d24-464b-be59-1661b5fb7266",
      "sourceStage": "DEVELOPMENT",
      "destStage": "PRODUCTION",
      "lastUpdated": "2015-11-20T06:23:14.623Z",
      "startTime": "2015-11-20T06:23:14.623Z",
      "endTime": "2015-11-20T06:25:00.277Z",
      "message": "{\"packageId\":\"a3eb6d50-5b6a-4dad-ac7b-fc81032c81c2\"}",
      "createdBy": "admin",
      "userAllowed": true,
      "status": "COMPLETED"
    }
  ]
}
```


List all action instances

Lists all action instances. This API takes the releaseGateRequestId as the parameter.

URI	Method	Parameters	Returns
http://[host]:[port] /csa/api//codar/release-gate/gate-request/{releaseGateRequestId} /action-instances	GET	releaseGateRequestId	200 - successful 400 - bad request 404 - not found 500 - Internal server error

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/release-gate/gate-request/48feb259-3d24-464b-be59-1661b5fb7266/action-instances
```

The following JSON was returned:


```
{
  "@type": "urn:x-hp:2012:software:cloud:data_model:ReleaseGateActionInstance",
  "@total_results": "1",
  "members": [
    {
      "@self": "/csa/codar/release-gate/gate-request/action-instance/2869b382-ec18-4aa9-b380-d069a4c8e973",
      "ignoreOnFailure": false,
      "startTime": "2015-11-20T06:23:15.420Z",
      "endTime": "2015-11-20T06:25:00.280Z",
      "statusMessage": "Action Instance Completed Successfully",
      "actionName": "deploy1",
      "actionOrder": 0,
      "status": "COMPLETED"
    }
  ]
}
```


Update status of release gate request

Updates the status of release gate request. This API takes the releaseGateRequestId and state as the parameters. The value of states can be stop, resume, and pause.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api//codar/release-gate/gate-request/{releaseGateRequestId}	PUT	• releaseGateRequestId • state	200 - Ok 400 - Bad request 401- Authorization failure 404 - Not found 500 - Server Exception

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/release-gate/gate-request/48feb259-3d24-464b-be59-1661b5fb7266/?state=resume
```

The following JSON was sent to update a custom role:

```
{
  "requestId": "48feb259-3d24-464b-be59-1661b5fb7266"
}
```

Deletes release gate requests

Deletes release gate actions. This API takes the releaseGateRequestId as the parameter.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/release-gate/gate-request/{releaseGateRequestId}	DELETE	releaseGateRequestId	204 - Deleted 400 - bad request 401 - Authorization failure 404 - Not found 500 - Server Exception

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/release-gate/gate-request/48feb259-3d24-464b-be59-1661b5fb7266
```

Custom roles APIs

URI	Method	See also
/csa/api/codar/role	GET	"List all custom roles" on the next page

List all custom roles

Lists all the custom roles created in Codar. The response of the API contains the roles and lifecycle stage information associated with each role. The information retrieved through this API is specific to release pipeline management and does not contain the permissions and users who are a part of this role.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/role	GET		200 - updated 400 - bad request 404 - not found 500 - Server Exception

Example

The following URL was sent:

```
https://<codar server>:8444/csa/api/codar/role
```

The following JSON was returned:


```
{ "@total" : 5,
 "@self" : "/csa/api/codar/role/",
"@type" : "urn:x-hp:2012:software:cloud:data_model:role:collection",
"members" : [ {
"@self" : "/csa/api/codar/role/710e7e12cb8c4e8992bf9e7400a7301f",
"@type" : "urn:x-hp:2012:software:cloud:data_model:role",
"name" : "CODAR_APPLICATION_DEVELOPER",
 "displayName" : "Application Developer",
 "description" : "The Application Developer can create, edit, delete, deploy package
in development stage and promote package from development stage to testing stage.",
"icon" : "/csa/images/categories/role/service_consumer.png",
"editable" : false,
"permissions" : [ {
 "name" : "APPLICATION_PACKAGE_DEPLOY",
"displayName" : "Package deploy",
"description" : "Allows to view and deploy and undeploy application packages" }, {
"name" : "APPLICATION_PACKAGE_PROMOTE",
"displayName" : "Package promote",
"description" : "Allows to view and promote application packages" } ],
"stages" : [ {
"@self" : "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a5",
 "@type" : "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
"name" : "DEVELOPMENT",
"image" : "images/applications/dev.png",
```

```
"color" : "#D1AF89" } ] }, {
"@self" : "/csa/api/codar/role/4552b9da88a24767ac7e5b605f050df5",
 "@type" : "urn:x-hp:2012:software:cloud:data_model:role",
 "name" : "CODAR_APPLICATION_ARCHITECT",
 "displayName" : "Application Architect",
"description" : "The Application Architect can embrace component, create, edit,
delete Application and application version, deploy, create, edit and delete package
in development stage. Application Architect cannot reject package in any stage.",
"icon" : "/csa/images/categories/role/service_designer.png",
"editable" : false,
"permissions" : [ {
"name" : "APPLICATION_DESIGN_CREATE",
"displayName" : "Design create",
"description" : "Allow to embrace component, create, edit, delete application design
and versions." }, {
"name" : "APPLICATION_PACKAGE_DEPLOY",
"displayName" : "Package deploy",
"description" : "Allows to view and deploy and undeploy application packages" }, {
"name" : "APPLICATION_CUSTOM_ROLE_CREATE",
"displayName" : "Custom role create",
"description" : "Allows to create, edit and delete custom role for application
lifecycle management" } ],
"stages" : [ {
"@self" : "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a5",
 "@type" : "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
"name" : "DEVELOPMENT",
"image" : "images/applications/dev.png", "color" : "#D1AF89" } ] }, {
"@self" : "/csa/api/codar/role/5e78c029ace644ce8778f3a50f88873b",
"@type" : "urn:x-hp:2012:software:cloud:data_model:role",
"name" : "CODAR_APPLICATION_RELEASE_MANAGER",
"displayName" : "Application Release Manager",
"description" : "The Application Release Manager can deploy, reject, edit, delete
package in staging and production stages and promote package from staging stage to
production stage.", "icon" : "/csa/images/categories/role/service_business_
manager.png",
"editable" : false, "permissions" : [ {
"name" : "APPLICATION_PACKAGE_DEPLOY",
"displayName" : "Package deploy",
"description" : "Allows to view and deploy and undeploy application packages" }, {
"name" : "APPLICATION_PACKAGE_PROMOTE",
"displayName" : "Package promote",
"description" : "Allows to view and promote application packages" }, {
"name" : "APPLICATION_PACKAGE_REJECT",
"displayName" : "Package reject",
"description" : "Allows to view and reject application packages" }, {
"name" : "APPLICATION_DASHBOARD_VIEW", "displayName" : "Dashboard view",
"description" : "Allows to view the package application dashboard" }, {
"name" : "APPLICATION_CUSTOM_ROLE_CREATE",
"displayName" : "Custom role create",
"description" : "Allows to create, edit and delete custom role for application
```

```
lifecycle management" } ],
"stages" : [ {
 "@self" : "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a7", "@type" : "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
"name" : "STAGING", "image" : "images/applications/stage.png", "color" : "#E188CA" },
{ "@self" : "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-
78af206d31a8", "@type" : "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
"name" : "PRODUCTION", "image" : "images/applications/prod.png", "color" : "#FF7D6A"
} ] }, {
"@self" : "/csa/api/codar/role/40faa9a2769b451f8fecd1104260d062",
"@type" : "urn:x-hp:2012:software:cloud:data_model:role",
"name" : "CODAR_APPLICATION_QA",
"displayName" : "Application QA",
"description" : "The Application QA can deploy, edit, delete and reject package in
testing and staging stages and promote package from testing stage to staging stage.",

"icon" : "/csa/images/categories/role/service_consumer.png",
"editable" : false, "permissions" : [ {
"name" : "APPLICATION_PACKAGE_DEPLOY",
"displayName" : "Package deploy",
"description" : "Allows to view and deploy and undeploy application packages" }, {
"name" : "APPLICATION_PACKAGE_PROMOTE",
"displayName" : "Package promote",
"description" : "Allows to view and promote application packages" }, {
"name" : "APPLICATION_PACKAGE_REJECT",
"displayName" : "Package reject",
"description" : "Allows to view and reject application packages" } ],
"stages" : [ {
"@self" : "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a6",
 "@type" : "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
"name" : "TESTING", "image" : "images/applications/test.png", "color" : "#FFC300" } ]
 }, { "@self" : "/csa/api/codar/role/6ff42967987c4a58ab5751817088607d",
"@type" : "urn:x-hp:2012:software:cloud:data_model:role",
"name" : "CODAR_APPLICATION_Operations_MANAGER",
"displayName" : "Application Operations Manager",
"description" : "The Application Operations Manager can deploy, reject, edit, delete
package in staging stage.",
"icon" : "/csa/images/categories/role/service_staging_manager.png",
"editable" : false, "permissions" : [ {
"name" : "APPLICATION_PACKAGE_DEPLOY",
"displayName" : "Package deploy",
"description" : "Allows to view and deploy and undeploy application packages" }, {
"name" : "APPLICATION_PACKAGE_PROMOTE",
"displayName" : "Package promote",
"description" : "Allows to view and promote application packages" }, {
"name" : "APPLICATION_PACKAGE_REJECT",
"displayName" : "Package reject",
"description" : "Allows to view and reject application packages" } ],
"stages" : [ {
"@self" : "/csa/api/codar/app-lifecycle/stages/84905d24-d92f-4916-b312-78af206d31a7",
```

```
 "@type" : "urn:x-hp:2012:software:cloud:data_model:lifeCycleStage",
"name" : "STAGING", "image" : "images/applications/stage.png", "color" : "#E188CA" }
] } ] }
}
```

Create a custom role

Creates a role other than the out-of-box roles that exist in Codar.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/role	POST		201 - Created 400 - bad request, 500 - Server Exception

Example

The following URL was sent:

```
https://localhost:8444/csa/api/codar/role
```

The following JSON was sent:

```
{
 "name": "System Integration QA",
  "description": "Responsible to test the sub-system integration",
  "icon": "test icon",
  "permissions" : ["APPLICATION_PACKAGE_DEPLOY", "APPLICATION_PACKAGE_PROMOTE",
"APPLICATION_PACKAGE_REJECT"]

}
```

The following is the response JSON:

```
{
  "@self": "/csa/api/codar/role/8a81848d4febf322014feec17886001a",
  "@type": "urn:x-hp:2012:software:cloud:data_model:role",
  "name": "SYSTEM_INTEGRATION_QA",
  "displayName": "System Integration QA",
  "description": "Responsible to test the sub-system integration",
  "icon": "test icon",
  "readOnly": false,
  "permissions": [
```

```
  {
   "name": "APPLICATION_PACKAGE_DEPLOY",
   "displayName": "Package deploy",
   "description": "Allows to view and deploy and undeploy application packages"
  },
  {
   "name": "APPLICATION_PACKAGE_PROMOTE",
   "displayName": "Package promote",
   "description": "Allows to view and promote application packages"
  },
  {
   "name": "APPLICATION_PACKAGE_REJECT",
   "displayName": "Package reject",
   "description": "Allows to view and reject application packages"
  }
 ]
}
```

Update a custom role

Updates a custom role for pipeline management.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/role	PUT	<*Role ID*>	200 - OK 400 - bad request 500 - Server Exception

Example

The following URL was sent:

```
https://<codar server>:8444/csa/api/codar/role/8a81848d4febf322014feec17886001a
```

The following JSON was sent to update a custom role:

```
{
  "name": "System Integration QA",
  "description": "Responsible to test the sub-system integration and perform
regression",
  "icon": "system test icon",
  "permissions" : ["APPLICATION_PACKAGE_DEPLOY", "APPLICATION_PACKAGE_PROMOTE",
"APPLICATION_PACKAGE_REJECT"]
}
```

The following JSON was returned:

```
{
  "@self": "/csa/api/codar/role/8a81848d4febf322014feec17886001a",
  "@type": "urn:x-hp:2012:software:cloud:data_model:role",
  "name": "SYSTEM_INTEGRATION_QA",
  "displayName": "System Integration QA",
  "description": "Responsible to test the sub-system integration and perform
regression",
  "icon": "system test icon",
  "readOnly": false,
  "permissions": [
    {
      "name": "APPLICATION_PACKAGE_DEPLOY",
      "displayName": "Package deploy",
      "description": "Allows to view and deploy and undeploy application packages"
    },
    {
      "name": "APPLICATION_PACKAGE_PROMOTE",
      "displayName": "Package promote",
      "description": "Allows to view and promote application packages"
    },
    {
      "name": "APPLICATION_PACKAGE_REJECT",
      "displayName": "Package reject",
      "description": "Allows to view and reject application packages"
    }
  ]
}
```


Delete a custom role

Deletes a custom role for pipeline management.

URI	Method	Parameters	Returns
http://[host]:[port]/csa/api/codar/role	DELETE	<*Role ID*>	200 - No Content 400 - bad request 500 - Server Exception

Example

The following URL was sent:

```
https://<codar server>:8444/csa/api/codar/role/8a81848d4febf322014feec17886001a
```

Codar CLI set up

Codar provides a command line interface (CLI) for accessing all Codar REST end points. You can perform all Codar specific functionality from your local machine using the CLI.

Use the `--help` option to view a list of available commands, and to read documentation about the functionality and options available for each command.

Set up your local Windows machine to use Codar CLI

The Codar CLI is provided in an executable .jar file. To access the CLI remotely from your Windows system, perform the following steps to copy the CLI files from the Codar installation on to your local Windows system:

1. Create a working directory on your local Windows system.
2. Make sure java is installed and the JAVA_HOME environment variable is set.
3. Copy the following files from the `Tools\CodarCLI` directory of the Codar installation (C:\Program Files\Hewlett-Packard\Codar\Tools\CodarCLI) to the working directory:

   a. `codar-clis.jar` – Standalone java executable .jar file with all CLI specific java classes and required dependencies
   b. `codarexec.bat` – Batch file to use to execute the above .jar file by starting a JVM process
   c. `env.bat` – Configuration for CLI
   d. `log4j.properties` – Logging information

   Note: All of the above files must be in the same working directory.

4. Copy the `cliforcodar.properties` file to the user home directory. Edit the URL and credentials used to access Codar.

   Note: The password in the `clidorcodar.properties` file is stored in plain text. For security reasons, HPE advises that you enter the credentials when prompted and do not store the credentials in a clear text file. If you decide to use the `clidorcodar.properties` file to store the API user credentials, make sure the file is stored in the home directory with limited permissions specific to the user. Consider encrypting the file in storage using your own preferred tool.

5. Run the `codarexec.bat` batch file to see the list of available commands.

Set up your local Linux machine to use Codar CLI

The Codar CLI is provided in an executable .jar file. To access the CLI remotely from your Linux system, perform the following steps as a root user to copy the CLI files from the Codar installation on to your local Linux system:

1. Create a working directory on your local Linux system.
2. Make sure java is installed and the JAVA_HOME environment variable is set.

3. Copy the following files from the `Tools\CodarCLI` directory of the Codar installation (\Hewlett-Packard\Codar\Tools\CodarCLI) to the working directory:

   a. `codar-clis.jar` – Standalone java executable .jar file with all CLI specific java classes and required dependencies
   b. `codarexec.sh` – Batch file to use to execute the above .jar file by starting a JVM process
   c. `log4j.properties` – Logging information

   Note: The password in the `clidorcodar.properties` file is stored in plain text. For security reasons, HPE advises that you enter the credentials when prompted and do not store the credentials in a clear text file. If you decide to use the `clidorcodar.properties` file to store the API user credentials, make sure the file is stored in the home directory with limited permissions specific to the user. Consider encrypting the file in storage using your own preferred tool.

4. Copy the `cliforcodar.properties` file to the user home directory. Edit the URL and credentials used to access Codar.

   Note: The password in the `clidorcodar.properties` file is always displayed in plain text (it is not encrypted.) For security reasons, the properties file should always be kept in the user home directory. For example, /home/user1.

5. Run the `codarexec.sh` file to see the list of available commands.

   Note: Verify this configuration using the root user. If you are not a root user, then the necessary permission must be provided for all files for that user

If you are not a root user, perform the following steps to access the CLI remotely from your Linux system:

1. Open a shell prompt.
2. If needed, perform the following steps to create a new root user:

   a. Type the `su-` command and enter the root password.
   b. Type the `useradd` command followed by a space and the user name for the new user. For example, `useradd newuser1`. This creates a directory in the new user home directory. For example, home/newuser1.

3. Copy the following files from the `Tools\CodarCLI` directory of the Codar installation (\Hewlett-Packard\Codar\Tools\CodarCLI) to the home directory:

   a. `codar-clis.jar`
   b. `codarexec.sh`
   c. `log4j.properties`
   d. `cliforcodar.properties`

4. Run the `codarexec.sh` file to see the list of available commands.

Codar command line interface commands

The following command line interface (CLI) commands provide access to Codar REST end points. You can perform all Codar specific functionality from your local machine using the CLI.

Use the `codarexec.bat` file to run a CLI command. To run a CLI command, type `codarexec.bat` followed by the command name and any required parameters.

Use the `--help` option to view a list of available commands, and to read documentation about the functionality and options available for each command.

See "Codar CLI set up" on page 110 for more information on how to set up your system to run the Codar CLI commands.

Application design commands

Command	See also
design list	"List application designs" below
design export	"Export an application design" on the next page
design import	"Import an application design" on page 114
design update	"Update an application design" on page 115
design delete	"Delete an application design" on page 116

List application designs

Lists all available application designs that the user is eligible to access.

Command	Alias	Syntax
design list	dsn list	design list

Options

Option	Description
`-s, --server`	The URL of the HP Codar instance. For example, https://localhost:8444/csa.

Option	Description
`-u, --user`	The HP Codar username.
`-p, --password`	The password for the HP Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password.” If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis>codarexec.bat dsn list
ID                                      version    displayName

954d5bb5-ebd2-457b-a538-58f3a7d0a37f    1.0.0      vCenter

8bcb57e7-190c-42ac-964b-20b34ff43c7f    1.0.0      Database service
```

Export an application design

Exports an application design in JSON format into the specified file.

Command	Alias	Syntax
design export	dsn export	dsn export –did <designID> -out <output-file>

Options

Option	Description
`-did, --<designID>`	Application design ID. Use the "List application designs" command to obtain the design ID.
`-out, --<output-file>`	The output file to which the application design is exported. This command overwrites the file if it already exists.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The HP Codar username.
`-p, --password`	The password for the HP Codar user.

Option	Description
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis>codarexec.bat dsn export -did c0929227-1fbb-4f9f-b8e4-e587938c42d7  -out
dsn.json

Design 'PetClinic Application' (c0929227-1fbb-4f9f-b8e4-e587938c42d7)  was exported
```

Import an application design

Imports a specified application design in JSON format.

Command	Alias	Syntax
design import	dsn import	dsn import –in <input-file>

Options

Option	Description
`-in, --<input-file>`	Input file containing an application design in JSON format. **Note:** If you are importing the exported design, the version must be changed.
`-u, --user`	The HP Codar username.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-p, --password`	The password for the HP Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis>codarexec.bat dsn import -in dsn.json

Design was added with id 'e639d171-23f6-4b08-b624-c08e523502ff'
```

Update an application design

Updates the application design for the specified design ID using the JSON provided in the input file.

Command	Alias	Syntax
design update	dsn update	design update -did <designID> –in <input-file>

Options

Option	Description
`-did, --<designid>`	Application design ID. Use the "List application designs" command to obtain the design ID.
`-in, --<input-file>`	Input file containing the modifiable properties of the application design.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The HP Codar username.
`-p, --password`	The password for the HP Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis>codarexec.bat dsn update –did e639d171-23f6-4b08-b624-c08e523502ff -in
update.json

Design 'App_Service' (e639d171-23f6-4b08-b624-c08e523502ff) was updated
```

Delete an application design

Deletes the application design associated with the specified design ID.

Command	Alias	Syntax
design delete	dsn delete	design delete -did <designID>

Options

Option	Description
`-did, --<designID>`	Application design ID. Use the "List application designs" command to obtain the design ID.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The HP Codar username.
`-p, --password`	The password for the HP Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis>codarexec.bat dsn delete -did c0929227-1fbb-4f9f-b8e4-e587938c42d7

Deleting Design 'c0929227-1fbb-4f9f-b8e4-e587938c42d7' was successful
```

Package Commands

Command	Description
package list	"List application packages" below
package get	"Get application package properties" on the next page
package create	"Create an application package" on page 119
package update	"Update an application package" on page 120
package deploy	"Deploy a package" on page 121
package redeploy	"Redeploy a package" on page 123
package promote	"Promote a package" on page 125
package reject	"Reject a package" on page 125
package activedeployments	"List active deployments" on page 126
package listdeployments	"List deployments" on page 127
package listeligibledesigns	"List eligible designs" on page 128

List application packages

Lists all available application packages for the specified application design.

Command	Alias	Syntax
package list	pkg list	package list -did <designID> [-stage <stage>] [-start <start>] [-page <page>]

Options

Option	Description
`-did, --<designID>`	Application design ID. Use the "List application designs" command to obtain the design ID.
`-stage, --<lifecycleStage>`	Name of the life cycle stage.
`-start , --<startIndex>`	Specifies the offset of the first entry to be included in the page.
`-page, --<pageSize>`	Specifies the page size.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.

Option	Description
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis>codarexec.bat pkg list –did ff319835-99d0-4205-a89a-6e5d328e5355
name            ID                                    stage          state

Package 5    3c83e586-06de-4ccd-b9a8-c0ffe192d52b    DEVELOPMENT    ACTIVE
Package 3    cdf95956-85ce-46f1-bf74-ed201650aa61    DEVELOPMENT    ACTIVE
Package 2    faa53cdc-5cb2-479f-938b-6eb6552f5ff5    DEVELOPMENT    ACTIVE
Package 1    ca3214ae-6ce9-4dd7-8beb-1713b7542b66    TESTING        ACTIVE
```

Get application package properties

Lists the properties of the application package with the specified ID.

Command	Alias	Syntax
package get	pkg get	package –pid <packageID> [-out <output-file>]

Options

Option	Description
`-pid, --<packageID>`	Application package ID. Use the "List application packages" command to obtain the package ID.
`-out, --<output-file>`	Output file into which the JSON of the application package is exported.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.

Option	Description
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis>codarexec.bat  pkg get -pid  3c83e586-06de-4ccd-b9a8-c0ffe192d52b –out
getdetail.json
name                 value                displayName          type

vCenter Server
customizationSpec    useVmName_Linux      customizationSpec    String
vmNamePrefix         Demo                 vmNamePrefix         String
vmTemplateReference  ubuntu1204-hemant    vmTemplateReference  String
```

Create an application package

Creates an application package.

Command	Alias	Syntax
package create	pkg create	package create -name <name> -desc <description> -did <designID> [-in <input-file>]

Options

Option	Description
`-did, --<designID>`	Application design ID. Use the "List application designs" command to obtain the design ID.
`-name, --<name>`	Name of the package.
`-desc, --<description>`	Package description.
`-in, --<input-file>`	Input file containing properties of the application package. If there is already a package for the design, use the "List application designs" command to obtain the JSON from the

Option	Description
	existing package and change the value of the attributes.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

When creating a package for the first time, use the `name` and `description` parameters. Properties from the design are associated to the package by default. The properties of the package can be specified as an input file in JSON format. For sample JSON input format use the "List application packages" command to obtain the properties of any existing package and modify the values of the package appropriately.

Note: Only the property’s value can be modified.

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
  C:\CODAR\clis> codarexec.bat pkg create -did cb335b9b-aa57-4150-a6c5-a54f80401fde  -
name package1 -desc packagedescription –in packageproperties.json

Package 'package1' was added with id 'e44cd095-aa35-468a-81bb-a1c0a36ae959'
```

Update an application package

Updates the package name, description, and properties. Note that only the property’s value attribute can be changed through CLI.

Command	Alias	Syntax
package update	pkg update	package update -pid <packageID> [-name <name>] [-desc <description>] [-in <input-file>]

Options

Option	Description
`-pid, --<packageID>`	Application package ID. Use the "List application packages" command to obtain the package ID.
`-name, --<name>`	Name of the package.
`-desc,--<description>`	Package description.
`-in, --<input-file>`	Input file containing properties of the application package. If there is already a package for the design use"List application packages" command to obtain the JSON from the existing package and change the value of the attributes.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis> codarexec.bat pkg update –pid e44cd095-aa35-468a-81bb-a1c0a36ae959  -name
package2 -desc desc –in packageproperties.json

Package 'package2' ( e44cd095-aa35-468a-81bb-a1c0a36ae959 ) was updated
```

Deploy a package

Deploys a package. The package can belong to either a partial or complete design.

Command	Alias	Syntax
package deploy	pkg deploy	package deploy -pid <packageID> –name <name> [–envid <environmentID>] [-did <infrastructureDesign>] [-in <input-file>]

Options

Option	Description
`-pid, --<packageID>`	Application package ID. Use the "List application packages" command to obtain the package ID.
`-name, --<name>`	Name of the deployment.
`–envid <environmentID>`	Environment ID. Use the "List eligible designs" command to obtain the environment ID.
`-did <infrastructureDesign>`	Application design ID obtained from "List eligible designs" command for partial designs. If not provided, the package deploy command takes the design ID associated with the package. The design ID can be obtain from the "List application designs" command.
`-in, --<input-file>`	Input file containing the modifiable properties in JSON format.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

You do not need to specify the `-did <infrastructureDesign>` parameter for a complete design.

For partial designs,use the `-did <infrastructureDesign>` parameter to specify an service design that has the required infrastructure capabilities.

If more than one microservices meets the requirements of the partial design requirements then you must also include the designToRequirementsMap parameter in the JSON input file.

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis> codarexec.bat pkg deploy -pid e65e26cd-f14c-4029-937b-d6e56507be11 –name
deploy_name –did  954d5bb5-ebd2-457b-a538-58f3a7d0a37f –in deploy.txt –envid
8a8187834c6ff639014c79cb097d03a3
```

```
Deployment deploy_name  for the package "e65e26cd-f14c-4029-937b-d6e56507be11" is
initiated with service instance id 8a8187834c6ff639014cd61c228311c4
```

The following is a sample input file in JSON format for deploying a partial application design with open requirements. Note that this JSON file contains the details of modifiable properties that should be applied during deployment. The JSON also contains the microservice service designs that satisfy the specific requirements of the partial application design.

```
{
       "modifiableProperties":[{"id":"1d894fc5-f9fc-f27d-5051-ba6df53c6fec",
               "itemType":"NODE",
               "propertyName":"vmTemplateReference",
               "propertyType":"string",
               "propertyValue":"csata-rhelsa915"},
               {"id":"1d894fc5-f9fc-f27d-5051-ba6df53c6fec",
               "itemType":"NODE",
               "propertyName":"customizationSpec",
               "propertyType":"string",
               "propertyValue":"Linux"}]
                   "designToRequirementsMap": [{
                    "designId": "0949335a-e46e-4f2b-beb6-d9d7a2db396d",
                    "requirementIds": ["Server__VERSION__1__GROUPID__
com.hp.csa.type0001"]
                    },
                   {
                   "designId": "8bcb57e7-190c-42ac-964b-20b34ff43c7f",
                   "requirementIds": ["Server__VERSION__1__GROUPID__
com.hp.csa.type0002"]
                   }
                  ]
}
```

Redeploy a package

Redeploys a package on a given active deployment instance for the user. The instance should be deployed in the same life cycle stage.

Command	Alias	Syntax
package redeploy	pkg redeploy	package redeploy –pid <packageID> –name <name> -depid <deploymentID> [-in input-file>]

Options

Option	Description
-pid, --<packageID>	Application package ID. Use the "List application packages" command to obtain the package ID.

Option	Description
`-name, --<name>`	Name of the deployment.
`-depid, --<deploymentID>`	Application deployment ID. Use the "List deployments" command to obtain the application deployment ID. The deployment must be in an ACTIVE or FAILED state.
`-in, --<input-file>`	Input file containing the modifiable properties in JSON format.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

The following JSON input was provided:

```
{
  "name":"redeployName",
  "serviceInstanceId":"8a81841f4b733315014b76c191b70099",
  "redeployProperties":[{"id":"VcenterServerType__VERSION__04.20.0000__GROUPID__
com.hp.csa.type0001",
    "itemType":"NODE",
    "propertyName":"cpuCount",
    "propertyType":"string",
    "propertyValue":"4"},
    {"id":"VcenterServerType__VERSION__04.20.0000__GROUPID__com.hp.csa.type0001",
    "itemType":"NODE",
    "propertyName":"memorySize",
    "propertyType":"string",
    "propertyValue":"1024"}]
}
```

```
C:\CODAR\clis> codarexec.bat pkg redeploy -name redeploypkg -pid b3364704-ff76-44a2-
88bd-90064a21b9b9 -depid 8a8187834c6ff639014cd662e3961483
```

```
Redeployment for the package “b3364704-ff76-44a2-88bd-90064a21b9b9” with service
instance id “8a8187834c6ff639014cd662e3961483” is initiated.
```

Promote a package

Promotes a package to the next life cycle stage.

Command	Alias	Syntax
package promote	pkg promote	package promote -pid <packageID>

Options

Option	Description
`-pid, --<packageID>`	Application package ID. Use the "List application packages" command to obtain the package ID.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis>  codarexec.bat pkg promote -pid   e44cd095-aa35-468a-81bb-a1c0a36ae959

Package 'package2' (e44cd095-aa35-468a-81bb-a1c0a36ae959) promote to TESTING was
successful
```

Reject a package

Rejects an application package. After a package is rejected, no operations can be performed on it and it cannot be brought back to active state.

Command	Alias	Syntax
package reject	pkg reject	package reject -pid <packageID>

Options

Option	Description
`-pid, --<packageID>`	Application package ID. Use the "List application packages" command to obtain the package ID.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis>  codarexec.bat pkg reject -pid   e44cd095-aa35-468a-81bb-a1c0a36ae959

Package 'package2' ( e44cd095-aa35-468a-81bb-a1c0a36ae959) reject was successful
```

List active deployments

Lists the active deployments for a given package belonging to the user for the same life cycle stage.

Command	Alias	Syntax
package activedeployments	pkg activedeployments	package activedeployments –pid <packageID>

Options

Option	Description
`-pid, --<packageID>`	Application package ID. Use the "List application packages"

Option	Description
	command to obtain the package ID.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis> codarexec.bat package activedeployments -pid  e65e26cd-f14c-4029-937b-d6e56507be11

 id                                      displayName
 8a8187834c6ff639014cd61c228311c4        deploy_name
```

List deployments

Lists all the deployments of the specified package.

Command	Alias	Syntax
package listdeployments	pkg listdeployments	package listdeployments –pid <packageID>

Options

Option	Description
`-pid, --<packageID>`	Application package ID. Use the "List application packages" command to obtain the package ID.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.

Option	Description
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis> codarexec.bat package listdeployments -pid  e65e26cd-f14c-4029-937b-d6e56507be11

createdBy       deploymentStage    id                                  status
admin           DEVELOPMENT        8a8187834c6ff639014cd61c228311c4    ACTIVE
```

List eligible designs

Lists the eligible candidate designs which meet the requirements of the specified package. These designs can be chosen for deployment. The designs listed are grouped based on the requirements that they satisfy.

Command	Alias	Syntax
package listeligibledesigns	pkg listeligibledesigns	package listeligibledesigns –did <designID> –pid <packageID>

Options

Option	Description
`-did, --<designID>`	Application design ID. Use the "List application designs" command to obtain the design ID.
`-pid, --<packageID>`	Application package ID. Use the "List application packages" command to obtain the package ID.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.

Option	Description
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis>codarexec.bat pkg listeligibledesigns -pid f26ab723-0ca1-4f59-ac3f-
01aba98c2194 -did c0929227-1fbb-4f9f-b8e4-e587938c42d7

id                                          artifactId
displayNameWithVersion

ALL SERVERS
2c34a633-f108-4a11-bba0-df1bc977d5ad     6c77c2bf-43d6-4b30-9f00-8659ee10f098
copy of Tomcat and Mysql service (1.0.0)
2e779d6a-8cd9-4a77-91a6-7697947ab16b     8a8187834c56d1f0014c6aaeb56b0917
Tomcat and Mysql service (1.0.0)

Application Server (ApplicationServer__VERSION__1__GROUPID__com.hp.csa.type0001)
0949335a-e46e-4f2b-beb6-d9d7a2db396d     53a8f2c4-aa55-4303-b209-e42df5a59407
Tomcat Service (1.0.0)
5aa495a7-d700-4f2a-81fc-597cb435b825     913b8a24-3899-46a1-bb9d-e7532a88ae10
copy of Tomcat Service (1.0.0)

Database Server (DatabaseServer__VERSION__1__GROUPID__com.hp.csa.type0001)
8bcb57e7-190c-42ac-964b-20b34ff43c7f     1243a3c8-c3ac-4d79-90ad-7fa2ecd5ca86
Database service (1.0.0)
8d337d10-04b0-4299-999e-e927446ed835     c99dc0b4-1c00-4349-9cc2-112f130f527e
copy of Database service (1.0.0)
```

Deployment commands

Command	Description
deployment cancel	"Cancel a deployment" on the next page
deployment delete	"Delete a deployment" on the next page

Cancel a deployment

Undeploys the specified application package deployment. Only deployments in ACTIVE or FAILED states can be canceled.

Command	Alias	Syntax
deployment cancel	dep cancel	deployment cancel -depid <deploymentID>

Options

Option	Description
`-depid, --<deploymentID>`	Application deployment ID. Use the "List deployments" command to obtain the application deployment ID. The deployment must be in an ACTIVE or FAILED state.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis> codarexec.bat dep cancel -depid 8a8187834c6ff639014cd662e3961483

Deployment ‘8a8187834c6ff639014cd61c228311c4’ (deploy_name) Cancel  was successful
```

Delete a deployment

Deletes the given application package deployment. Only deployments in CANCELLED state can be deleted.

Command	Alias	Syntax
deployment delete	dep delete	deployment delete -depid <deploymentID>

Options

Option	Description
`-depid, --<deploymentID>`	Application deployment ID. Use the "List deployments" command to obtain the application deployment ID. The deployment should already be cancelled.
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis> codarexec.bat dep delete -depid 8a8187834c6ff639014cd662e3961483

Deleting Deployment '8a8187834c6ff639014cd662e3961483' was successful
```

Environment commands

Command	Description
environment list	"List existing environments" below

List existing environments

Lists all existing resource environments.

Command	Alias	Syntax
environment list	env list	environment list

Options

Option	Description
`-s, --server`	The URL of the Codar instance. For example, https://localhost:8444/csa.
`-u, --user`	The Codar username.
`-p, --password`	The password for the Codar user.
`-c, --config`	A file that contains default general option values in the format “<name>=<value>“, where <name> is “server”, “user”, or “password”. If not specified, the `cliforcodar.properties` file in the home directory of the current user is used if it exists.

Usage guidelines

This command must be pre-defined in a configuration file or defined in the command line. The `--server(-s)` option defaults to `localhost` value(s) if it is not pre-defined in a configuration file.

Example

```
C:\CODAR\clis> codarexec.bat env list

ID                                  name
8a8187834c6ff639014c79cb097d03a3    Env1
8a8187834c6ff639014c79cb646b03a8    Env2
```

Appendix A: API return examples

The following sections provide full output examples for Codar RESTful API calls:

Application design API examples

- "Export an application design example" on the next page
- "Import an application design example" on page 139

Composition API examples

- "List the candidate topologies that can fulfill the specified partial topology example" on page 151

Container API examples

Package API examples

Export an application design example

Exports an application design by providing the application design ID. The application design is returned in JSON format.

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-design/e57673db-0ee7-4061-8f9e-ffdcabc07b4c
```

The following JSON was returned:


```
{
  "@self" : "/csa/api/topology-model/topology/e57673db-0ee7-4061-8f9e-ffdcabc07b4c",
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
  "groupId" : "com.hp.csa",
  "artifactId" : "3eb835ae-750f-41c9-8861-de70a285ca40",
  "version" : "1.0.0",
  "displayName" : "Test_Design-1",
  "description" : "my design",
  "resources" : [ {
    "id" : "VcenterServer0001",
    "name" : "vCenter Server 1",
    "component" : {
      "@self" : "/csa/api/topology-model/component-type/b83f4e21-d8a9-4ff3-a76e-
3a69775d1d86"
    },
    "properties" : [ {
      "propertyKey" : "requestedMemorySize",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
```

```
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "vmTemplateReference",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "cpuCount",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "ipAddressList",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "closeSession",
      "propertyValue" : {
        "value" : "true",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "privateKey",
      "propertyValue" : {
        "value" : "",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "memorySize",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
```

```
    }, {
      "propertyKey" : "requestedCpuCount",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "customizationSpec",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "hostname",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "macAddress",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "password",
      "propertyValue" : {
        "value" : "null",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "response",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "_modifiable_properties",
      "propertyValue" : {
```

```
        "value" : "",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "username",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "vmID",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "ipAddress",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "vmNamePrefix",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    } ],
    "outgoingRelations" : [ ],
    "annotations" : {
      "description" : "",
      "displayName" : "vCenter Server 1",
      "y" : "-400",
      "x" : "-400"
      }
    }, {
      "id" : "VcenterNetworkInterface0001",
      "name" : "vCenter Network Interface 1",
      "component" : {
        "@self" : "/csa/api/topology-model/component-type/05c69cad-ad1d-4285-befe-
```

```
4963c7f86ae1"
      },
      "properties" : [ {
        "propertyKey" : "response",
        "propertyValue" : {
          "value" : null,
          "type" : "string",
          "confidential" : false,
          "consumerVisible" : true
      }
    }, {
      "propertyKey" : "_modifiable_properties",
      "propertyValue" : {
        "value" : "",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "networkInterfaceType",
      "propertyValue" : {
        "value" : "E1000",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "networkInterfaceId",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "portGroupName",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "vmID",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
```

```
      "propertyKey" : "macAddress",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "ipAddresses",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    } ],
    "outgoingRelations" : [ {
      "relationType" : "association",
      "id" : "rel0001",
      "properties" : [ {
        "propertyKey" : "_modifiable_properties",
        "propertyValue" : {
          "value" : "",
          "type" : "string",
          "confidential" : false,
          "consumerVisible" : true
      }
    }, {
      "propertyKey" : "_relation_type",
      "propertyValue" : {
        "value" : {
          "first" : "Association",
          "second" : "ASSOCIATION"
        },
        "type" : "pair",
        "confidential" : false,
        "consumerVisible" : true
      }
    } ],
    "relationDescriptorName" : "VcenterNetworkInterfaceTypeToVcenterServerType",
    "resourceReference" : {
      "name" : "vCenter Server 1"
    }
  } ],
  "annotations" : {
    "description" : "",
    "displayName" : "vCenter Network Interface 1",
    "y" : "-400",
    "x" : "-200"
  }
} ],
```

```
"revision" : 0,
"properties" : [ {
  "propertyKey" : "metamodelId",
  "propertyValue" : {
    "value" : "test",
    "type" : "string",
    "confidential" : false,
    "consumerVisible" : true
  }
}, {
  "propertyKey" : "is_express",
  "propertyValue" : {
    "value" : null,
    "type" : "boolean",
    "confidential" : false,
    "consumerVisible" : true
  }
} ],
"tagIds" : [ ],
"state" : "UNLOCKED",
"profiles" : [ ],
"annotations" : {
  "providerType" : "independent",
  "serviceBlueprintId" : "8a818cf045d640b00145d64a7c240017"
 }
}
```

Import an application design example

Imports an application design by providing the application design in JSON format as input. If the design already exists, as error is returned.

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-design/import
```

The following JSON was sent:

```
{
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
  "groupId" : "com.hp.csa",
  "artifactId" : "3eb835ae-750f-41c9-8861-de70a285ca43",
  "version" : "1.0.0",
  "displayName" : "Test_Design-2",
  "description" : "my design 02",
  "resources" : [ {
    "id" : "VcenterServer0001",
    "name" : "vCenter Server 1",
    "component" : {
      "@self" : "/csa/api/topology-model/component-type/b83f4e21-d8a9-4ff3-a76e-
```

```
3a69775d1d86"
    },
    "properties" : [ {
      "propertyKey" : "requestedMemorySize",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "vmTemplateReference",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "cpuCount",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "ipAddressList",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "closeSession",
      "propertyValue" : {
        "value" : "true",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "privateKey",
      "propertyValue" : {
        "value" : "",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
```

```
      "propertyKey" : "memorySize",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "requestedCpuCount",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "customizationSpec",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "hostname",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "macAddress",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "password",
      "propertyValue" : {
        "value" : "null",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "response",
      "propertyValue" : {
        "value" : null,
```

```
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "_modifiable_properties",
    "propertyValue" : {
      "value" : "",
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "username",
    "propertyValue" : {
      "value" : "test",
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "vmID",
    "propertyValue" : {
      "value" : null,
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "ipAddress",
    "propertyValue" : {
      "value" : null,
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "vmNamePrefix",
    "propertyValue" : {
      "value" : "test",
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  } ],
  "outgoingRelations" : [ ],
  "annotations" : {
    "description" : "",
    "displayName" : "vCenter Server 1",
    "y" : "-400",
    "x" : "-400"
```

```
    }
  }, {
    "id" : "VcenterNetworkInterface0001",
    "name" : "vCenter Network Interface 1",
    "component" : {
      "@self" : "/csa/api/topology-model/component-type/05c69cad-ad1d-4285-befe-
4963c7f86ae1"
    },
    "properties" : [ {
      "propertyKey" : "response",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "_modifiable_properties",
      "propertyValue" : {
        "value" : "",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "networkInterfaceType",
      "propertyValue" : {
        "value" : "E1000",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "networkInterfaceId",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "portGroupName",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "vmID",
      "propertyValue" : {
```

```
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "macAddress",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "ipAddresses",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    } ],
    "outgoingRelations" : [ {
      "relationType" : "association",
      "id" : "rel0001",
      "properties" : [ {
        "propertyKey" : "_modifiable_properties",
        "propertyValue" : {
          "value" : "",
          "type" : "string",
          "confidential" : false,
          "consumerVisible" : true
        }
      }, {
        "propertyKey" : "_relation_type",
        "propertyValue" : {
          "value" : {
            "first" : "Association",
            "second" : "ASSOCIATION"
          },
          "type" : "pair",
          "confidential" : false,
          "consumerVisible" : true
        }
      } ],
      "relationDescriptorName" : "VcenterNetworkInterfaceTypeToVcenterServerType",
      "resourceReference" : {
        "name" : "vCenter Server 1"
      }
    } ],
    "annotations" : {
```

```
      "description" : "",
      "displayName" : "vCenter Network Interface 1",
      "y" : "-400",
      "x" : "-200"
    }
  } ],
  "revision" : 0,
  "properties" : [ {
    "propertyKey" : "metamodelId",
    "propertyValue" : {
      "value" : "test",
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "is_express",
    "propertyValue" : {
      "value" : null,
       "type" : "boolean",
       "confidential" : false,
       "consumerVisible" : true
   }
 } ],
 "tagIds" : [ ],
 "state" : "UNLOCKED",
 "profiles" : [ ],
}
```

The following JSON was returned:

```
{
  "@self" : "/csa/api/topology-model/topology/e57673db-0ee7-4061-8f9e-ffdcabc07b4c",
  "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
  "groupId" : "com.hp.csa",
  "artifactId" : "3eb835ae-750f-41c9-8861-de70a285ca40",
  "version" : "1.0.0",
  "displayName" : "Test_Design-1",
  "description" : "my design",
  "resources" : [ {
    "id" : "VcenterServer0001",
    "name" : "vCenter Server 1",
    "component" : {
      "@self" : "/csa/api/topology-model/component-type/b83f4e21-d8a9-4ff3-a76e-
3a69775d1d86"
    },
    "properties" : [ {
      "propertyKey" : "requestedMemorySize",
      "propertyValue" : {
        "value" : null,
```

```
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "vmTemplateReference",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "cpuCount",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "ipAddressList",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "closeSession",
      "propertyValue" : {
        "value" : "true",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "privateKey",
      "propertyValue" : {
        "value" : "",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "memorySize",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
```

```
    }
  }, {
    "propertyKey" : "requestedCpuCount",
    "propertyValue" : {
      "value" : null,
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "customizationSpec",
    "propertyValue" : {
      "value" : "test",
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "hostname",
    "propertyValue" : {
      "value" : null,
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "macAddress",
    "propertyValue" : {
      "value" : null,
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "password",
    "propertyValue" : {
      "value" : "null",
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "response",
    "propertyValue" : {
      "value" : null,
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
  }, {
    "propertyKey" : "_modifiable_properties",
```

```
      "propertyValue" : {
        "value" : "",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "username",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "vmID",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "ipAddress",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "vmNamePrefix",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
     } ],
     "outgoingRelations" : [ ],
     "annotations" : {
       "description" : "",
       "displayName" : "vCenter Server 1",
       "y" : "-400",
       "x" : "-400"
     }
   }, {
     "id" : "VcenterNetworkInterface0001",
     "name" : "vCenter Network Interface 1",
     "component" : {
       "@self" : "/csa/api/topology-model/component-type/05c69cad-ad1d-4285-befe-
```

```
4963c7f86ae1"
      },
      "properties" : [ {
      "propertyKey" : "response",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "_modifiable_properties",
      "propertyValue" : {
        "value" : "",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "networkInterfaceType",
      "propertyValue" : {
        "value" : "E1000",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "networkInterfaceId",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "portGroupName",
      "propertyValue" : {
        "value" : "test",
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "vmID",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
```

```
      "propertyKey" : "macAddress",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    }, {
      "propertyKey" : "ipAddresses",
      "propertyValue" : {
        "value" : null,
        "type" : "string",
        "confidential" : false,
        "consumerVisible" : true
      }
    } ],
    "outgoingRelations" : [ {
      "relationType" : "association",
      "id" : "rel0001",
      "properties" : [ {
        "propertyKey" : "_modifiable_properties",
        "propertyValue" : {
          "value" : "",
          "type" : "string",
          "confidential" : false,
          "consumerVisible" : true
        }
      }, {
        "propertyKey" : "_relation_type",
        "propertyValue" : {
          "value" : {
            "first" : "Association",
            "second" : "ASSOCIATION"
          },
          "type" : "pair",
          "confidential" : false,
          "consumerVisible" : true
        }
      } ],
      "relationDescriptorName" : "VcenterNetworkInterfaceTypeToVcenterServerType",
      "resourceReference" : {
        "name" : "vCenter Server 1"
      }
    } ],
    "annotations" : {
      "description" : "",
      "displayName" : "vCenter Network Interface 1",
      "y" : "-400",
      "x" : "-200"
    }
  } ],
```

```
  "revision" : 0,
  "properties" : [ {
    "propertyKey" : "metamodelId",
    "propertyValue" : {
      "value" : "test",
      "type" : "string",
      "confidential" : false,
      "consumerVisible" : true
    }
   }, {
     "propertyKey" : "is_express",
     "propertyValue" : {
       "value" : null,
       "type" : "boolean",
       "confidential" : false,
       "consumerVisible" : true
     }
 } ],
   "tagIds" : [ ],
   "state" : "UNLOCKED",
   "profiles" : [ ],
   "annotations" : {
     "providerType" : "independent",
     "serviceBlueprintId" : "8a818cf045d640b00145d64a7c240017"
   }
}
```


List the candidate topologies that can fulfill the specified partial topology example

Lists the candidate topologies that can fulfill the specified partial topology.

The following example shows a partial design with two requirements, ApplicationServer and DatabaseServer. The initial members array provides the service designs, which are topologies that can satisfy all requirements. After the initial members array, the requirements member array contains details of microservices per requirement (exclusive of the service designs.)

The following URL was sent:

`https://localhost:8444/csa/api/ui/topology-model/composition/9dc7a8f7-9a11-4d4f-a149-6e5b3aeb7618/candidateTopology`

The following JSON was returned:


```
{
   "@count" : 1,
   "members" : [ {
      "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
      "@self" : "/csa/api/topology-model/topology/9dc7a8f7-9a11-4d4f-a149-
```

```
6e5b3aeb7618",
      "id" : "9dc7a8f7-9a11-4d4f-a149-6e5b3aeb7618",
      "iconUrl" : "/csa/designer/img/noimage-lightgray.svg",
      "displayName" : "Allin1",
      "displayNameWithVersion" : "Allin1 - 1.0.0"
      "description" : "",
      "version" : "1.0.0",
      "artifactId" : "f20b7a8b-0abe-454b-b6d4-cd45764e70c4",
      "groupId" : "com.hp.csa",
      "published" : false,
      "tagIds" : [ ]
 } ],
 "requirements" : {
   "count" : 2,
   "members" : [ {
     "id" : "dc3ec34b-a8d1-e588-fee1-49d3b0c900ad",
     "displayName" : "ApplicationServer",
     "scalingGroupId" : "",
     "scalingGroupName" : "",
     "matchingTopologies" : {
      "count" : 1,
       "members" : [ {
        "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
         "@self" : "/csa/api/topology-model/topology/9dc7a8f7-9a11-4d4f-a149-
6e5b3aeb7618",
         "id" : "9dc7a8f7-9a11-4d4f-a149-6e5b3aeb6728",
         "iconUrl" : "/csa/designer/img/noimage-lightgray.svg",
         "displayName" : "Tomcat_Infra",
         "displayNameWithVersion" : "Allin1 - 1.0.0"
         "description" : "",
         "version" : "1.0.0",
         "artifactId" : "f20b7a8b-0abe-454b-b6d4-cd45764e62c4",
         "groupId" : "com.hp.csa",
         "published" : false,
        "tagIds" : [ ]
       } ]
     }
   }, {
     "id" :"4ea90e2a-a7b0-3185-d348-868f259235c1",
     "displayName" : "DatabaseServer",
     "scalingGroupId" : "7b0e62d9-ca22-4128-a7d2-df45f6ec2a6a",
     "scalingGroupName" : "micro_mixed",
     "matchingTopologies" : {
       "count" : 2,
       "members" : [ {
         "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
         "@self" : "/csa/api/topology-model/topology/69efd92b-4be7-480c-abb9-
e54948f5e2f1",
         "id" : "69efd92b-4be7-480c-abb9-e54948f5e2f1",
         "iconUrl" : "/csa/designer/img/noimage-lightgray.svg",
         "displayName" : "SQL_Infra",
```

```
         "displayNameWithVersion" : "SQL_Infra- 1.0.0"
         "description" : "",
         "version" : "1.0.0",
         "artifactId" : "77b8d440-4a75-4c77-94e9-66ebf136f781",
         "groupId" : "com.hp.csa",
         "published" : false,
         "tagIds" : [ ]
       }, {
         "@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
         "@self" : "/csa/api/topology-model/topology/e535c5cf-28fd-483e-a8d0-
792627f7a7b6",
         "id" : "e535c5cf-28fd-483e-a8d0-792627f7a7b6",
         "iconUrl" : "/csa/designer/img/noimage-lightgray.svg",
         "displayName" : "Orcl_Infra",
         "displayNameWithVersion" : "Orcl_Infra- 1.0.0"
         "description" : "",
         "version" : "1.0.0",
         "artifactId" : "d068b895-dd53-4ba2-82db-946b1a77bb59",
         "groupId" : "com.hp.csa",
         "published" : false,
         "tagIds" : [ ]
       } ]
     }
   }
   } ]
  }
}
```


List a container example

Returns the existing container, versions, and number of packages for each life cycle stage for each version of the specified container ID.

The following URL was sent:

`https://csa-server:8444/csa/api/codar/app-container/8a81855b4cf9cdd6014cfa2981c40004`

The following JSON was returned:


```
{
  "@self": "/csa/api/container/topology/8a81855b4cf9cdd6014cfa2981c40004",
  "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
  "@created": "2015-04-27T09:16:06.470Z",
  "@modified": "2015-04-29T10:41:54.603Z",
  "name": "design1",
  "description": "",
  "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
  "ext":
```

```
{
 "csa_name_key": "design1",
 "csa_critical_system_object": false
},
  "tags": [ {
    "@self": "/csa/api/tag/8a818cf8ara15bc772b0145cb6efjhg99",
    "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
    "name": "Application",
    "description": "Codar Application",
    "icon": "/csa/api/blobstore/management-darkgray.svg?tag=library",
    "color": "",
    "scopes": [ "APPLICATION_ARTIFACT_CONTAINER" ]
} ],
    "members": [ {
       "@self": "/csa/api/service/design/bb8d6c92-ff61-4ada-90be-da82363d841f",
       "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint",
       "@created": "2015-04-29T10:41:54.603Z",
       "description": "this is 3rd version of the design",
       "published": null,
       "version": "4.0.0",
       "members": [ {
         "stage": "DEVELOPMENT",
         "count": 0
     }, {
        "stage": "TESTING",
        "count": 0
    }, {
        "stage": "STAGING",
        "count": 0
    }, {
        "stage": "PRODUCTION",
        "count": 0
     } ]
 }, {
      "@self": "/csa/api/service/design/cfe17c5a-df0d-4063-a538-d6a301085141",
      "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint",
      "@created": "2015-04-29T06:05:48.507Z",
      "description": "",
      "published": null,
      "version": "2.0.0",
      "members":
 [ {
     "stage": "DEVELOPMENT",
     "count": 2
 }, {
     "stage": "TESTING",
     "count": 1
 }, {
     "stage": "STAGING",
```

```
        "count": 0
    }, {
        "stage": "PRODUCTION",
        "count": 0
    } ]
    }, {
         "@self": "/csa/api/service/design/3dbe26ae-f629-4556-b8ce-b27d34a80193",
         "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint",
         "@created": "2015-04-27T09:16:10.970Z",
         "description": "",
         "published": null,
         "version": "1.0.0",
         "members":
     [ {
         "stage": "DEVELOPMENT",
         "count": 2
    }, {
         "stage": "TESTING",
         "count": 0
    }, {
         "stage": "STAGING",
         "count": 0
    }, {
         "stage": "PRODUCTION",
         "count": 0
    } ]
    }, {
         "@self": "/csa/api/service/design/4e47e409-1209-4826-aebc-c535c93a7057",
         "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint",
         "@created": "2015-04-29T08:51:48.117Z",
         "description": "",
         "published": null,
         "version": "3.0.0",
         "members":
    [ {
         "stage": "DEVELOPMENT",
         "count": 0
   }, {
         "stage": "TESTING",
         "count": 0
   }, {
         "stage": "STAGING",
         "count": 0
   }, {
         "stage": "PRODUCTION",
         "count": 0
   } ]
   } ],
         "@total_results": 4,
```

```
         "@items_per_page": 4,
         "@start_index": 0 }
```

List existing topology design containers example

Returns a list of all existing topology design containers.

The following URL was sent:

`https://localhost:8444/csa/api/container/topology/?start-index=1&page-size=3`

The following JSON was returned:


```
{
  "@total_results": 1,
  "@start_index": 0,
  "@items_per_page": 1,
  "@self": "/csa/api/container/topology/",
  "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
  "members": [
    {
      "@self": "/csa/api/container/topology/8a81848d4d47af63014d47b847230003",
      "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
      "@created": "2015-05-12T10:42:48.740Z",
      "@modified": "2015-05-12T10:42:49.390Z",
      "global_id": "8a81848d4d47af63014d47b847230003",
      "name": "Debian Infra",
      "description": "Debian Infra",
      "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
      "ext": {
        "csa_name_key": "e67b408c66d840a788593690a665e0d3",
        "csa_critical_system_object": false
      },
      "container_type": "TOPOLOGY_ARTIFACT_CONTAINER",
      "tags": [
        {
          "@self": "/csa/api/tag/ede6b70286454f929d9566b8016f282e",
          "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
          "name": "Testing",
          "description": "Testing Micro Service",
          "icon": "/csa/api/blobstore/organizationDefault58.png?tag=library",
          "color": "#fbd75b",
          "scopes": [
            "LIFECYCLE_ARTIFACT_CONTAINER"
          ]
        },
        {
          "@self": "/csa/api/tag/d22c213f7e644d1d8362b2830b2c93a9",
```

```
          "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
          "name": "Development",
          "description": "Development Micro Service",
          "icon": "/csa/api/blobstore/organizationDefault58.png?tag=library",
          "color": "#ff887c",
          "scopes": [
            "LIFECYCLE_ARTIFACT_CONTAINER"
          ]
        }
     ],
     "members": [
       {
         "@self": "/csa/api/service/design/9c586c28-0fa0-4814-8b5c-eaf097a788f7",
         "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint",
         "@created": "2015-05-12T10:42:49.390Z",
         "published": false,
         "version": "1.0.0"
       }
    ],
    "@total_results": 1,
    "@items_per_page": 1,
    "@start_index": 0
   }
 ],
 "@modified": "2015-05-13T08:59:44.730Z"
}
```


List a service design container example

Lists the service design container specified by the Container ID parameter.

The following URL was sent:

`https://localhost:8444/csa/api/container/topology/8a81848d4d47af63014d47b847230003`

The following JSON was returned:


```
{
  "@self": "/csa/api/container/topology/8a81848d4d47af63014d47b847230003",
  "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
  "@created": "2015-05-12T10:42:48.740Z",
  "@modified": "2015-05-12T10:42:49.390Z",
  "global_id": "8a81848d4d47af63014d47b847230003",
  "name": "Debian Infra",
  "description": "Debian Infra",
  "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
  "ext": {
    "csa_name_key": "e67b408c66d840a788593690a665e0d3",
    "csa_critical_system_object": false
```

```
},
"container_type": "TOPOLOGY_ARTIFACT_CONTAINER",
"tags": [
   {
     "@self": "/csa/api/tag/775566f589944c489c7caaaf91ce3e8d",
     "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
     "name": "Production",
     "description": "Production Micro Service",
     "icon": "/csa/api/blobstore/organizationDefault58.png?tag=library",
     "color": "#ffffff",
     "scopes": [
       "LIFECYCLE_ARTIFACT_CONTAINER"
     ]
  },
  {
    "@self": "/csa/api/tag/ede6b70286454f929d9566b8016f282e",
    "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
    "name": "Testing",
    "description": "Testing Micro Service",
    "icon": "/csa/api/blobstore/organizationDefault58.png?tag=library",
    "color": "#fbd75b",
    "scopes": [
      "LIFECYCLE_ARTIFACT_CONTAINER"
    ]
  },
  {
    "@self": "/csa/api/tag/0c8ae53917c14e8f98a11e45eef386e0",
    "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
    "name": "Staging",
    "description": "Staging Micro Service",
    "icon": "/csa/api/blobstore/organizationDefault58.png?tag=library",
    "color": "#5484ed",
    "scopes": [
      "LIFECYCLE_ARTIFACT_CONTAINER"
    ]
  },
  {
    "@self": "/csa/api/tag/d22c213f7e644d1d8362b2830b2c93a9",
    "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
    "name": "Development",
    "description": "Development Micro Service",
    "icon": "/csa/api/blobstore/organizationDefault58.png?tag=library",
    "color": "#ff887c",
    "scopes": [
      "LIFECYCLE_ARTIFACT_CONTAINER"
    ]
   }
 ],
 "members": [
```

```
    {
       "@self": "/csa/api/service/design/9c586c28-0fa0-4814-8b5c-eaf097a788f7",
       "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint",
       "@created": "2015-05-12T10:42:49.390Z",
       "description": "Debian Infra",
       "published": false,
       "version": "1.0.0"
    }
  ],
  "@total_results": 1,
  "@items_per_page": 1,
  "@start_index": 0
}
```


List containers matching a filter on tag and type example

Lists all containers for a specified tag and type.

The following URL was sent:

`https://localhost:8444/csa/api/container/filter`

The following input JSON was sent:


```
{
 "tag": {
 "@self": "/csa/api/tag/8a81848d4d6cd612014d6fdee47f0036"
}
 "container_type": "topology_artifact_container"
}
```


The following JSON was returned:


```
{
 "@total_results": 3,
 "@start_index": 0,
 "@items_per_page": 3,
 "@self": "/csa/api/container/topology/",
 "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
 "members": [
  {
   "@self": "/csa/api/container/topology/8a81848d4d6cd612014d6fdfcb590038",
   "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
   "@created": "2015-05-20T05:50:47.130Z",
   "@modified": "2015-05-20T05:50:48.420Z",
   "global_id": "8a81848d4d6cd612014d6fdfcb590038",
   "name": "Developer Design 1",
   "description": "Developer Design 1",
   "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
```

```
   "ext": {
    "csa_name_key": "307c9284-358d-4a5c-b6ee-41e14fa664c8",
    "csa_critical_system_object": false
  },
  "container_type": "TOPOLOGY_ARTIFACT_CONTAINER",
  "tags": [
    {
     "@self": "/csa/api/tag/8a81848d4d6cd612014d6fdee47f0036",
     "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
     "name": "Developer Desings",
     "description": "Developer Desings",
     "icon": "/csa/api/blobstore/other.png?tag=library",
     "color": "#ffffff",
     "scopes": [
      "TOPOLOGY_ARTIFACT_CONTAINER"
    ]
   }
  ],
  "members": [
   {
    "@self": "/csa/api/service/design/111d16e9-6b0b-47ce-bab2-2e2f1b18478c",
    "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint",
    "@created": "2015-05-20T05:50:48.420Z",
    "published": false,
    "version": "1.0.0"
   }
  ],
  "@total_results": 1,
  "@items_per_page": 1,
  "@start_index": 0
 },
 {
  "@self": "/csa/api/container/topology/8a81848d4d6cd612014d7044313d00c1",
  "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
  "@created": "2015-05-20T07:40:26.813Z",
  "@modified": "2015-05-20T07:40:26.813Z",
  "global_id": "8a81848d4d6cd612014d7044313d00c1",
  "name": "Sample Container",
  "description": "Sample Container Description",
  "icon": "/csa/api/blobstore/Arrow_02_48.png?tag=library",
  "ext": {
   "csa_name_key": "fb1e1294-6e5b-4323-a0cc-7414234c10a1",
   "csa_critical_system_object": false
 },
 "container_type": "TOPOLOGY_ARTIFACT_CONTAINER",
 "tags": [
   {
    "@self": "/csa/api/tag/8a81848d4d6cd612014d6fdee47f0036",
    "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
    "name": "Developer Desings",
    "description": "Developer Desings",
```

```
    "icon": "/csa/api/blobstore/other.png?tag=library",
    "color": "#ffffff",
    "scopes": [
     "TOPOLOGY_ARTIFACT_CONTAINER"
   ]
  }
 ]
},
{
 "@self": "/csa/api/container/topology/8a81848d4d6cd612014d6fe02be20051",
 "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
 "@created": "2015-05-20T05:51:11.843Z",
 "@modified": "2015-05-20T05:51:12.950Z",
 "global_id": "8a81848d4d6cd612014d6fe02be20051",
 "name": "Developer Design 2",
 "description": "Developer Design 2",
 "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
 "ext": {
  "csa_name_key": "a1203274-e7e9-4b09-b8f6-1410d6f4cf22",
  "csa_critical_system_object": false
 },
 "container_type": "TOPOLOGY_ARTIFACT_CONTAINER",
 "tags": [
   {
    "@self": "/csa/api/tag/8a81848d4d6cd612014d6fdee47f0036",
    "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
    "name": "Developer Desings",
    "description": "Developer Desings",
    "icon": "/csa/api/blobstore/other.png?tag=library",
    "color": "#ffffff",
    "scopes": [
     "TOPOLOGY_ARTIFACT_CONTAINER"
   ]
  }
 ],
 "members": [
  {
   "@self": "/csa/api/service/design/7a0d026b-5384-4dd9-8ac8-80cb092b4067",
   "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint",
   "@created": "2015-05-20T05:51:12.950Z",
   "published": false,
   "version": "1.0.0"
  }
 ],
 "@total_results": 1,
 "@items_per_page": 1,
 "@start_index": 0
 }
],
"@modified": "2015-05-20T07:40:26.813Z"
}
```

List topology design containers matching a filter on tag and type example

List containers matching a filter on tag and type.

The following URL was sent:

`https://localhost:8444/csa/api/container/topology/8a81848d4d47af63014d47b847230003`

The following JSON request was sent:

```
{
"tag": {
  "@self": "/csa/api/tag/d22c213f7e644d1d8362b2830b2c93a9"
 },
"container_type": "topology_artifact_container"
}
```

The following JSON was returned:


```
{
  "@total_results": 12,
  "@start_index": 0,
  "@items_per_page": 12,
  "@self": "/csa/api/container/topology/",
  "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
  "members": [
    {
      "@self": "/csa/api/container/topology/8a81848d4d47af63014d47b847230003",
      "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint:collection",
      "@created": "2015-05-12T10:42:48.740Z",
      "@modified": "2015-05-12T10:42:49.390Z",
      "global_id": "8a81848d4d47af63014d47b847230003",
      "name": "Debian Infra",
      "description": "Debian Infra",
      "icon": "/csa/api/blobstore/Service_Design.png?tag=library",
      "ext": {
        "csa_name_key": "e67b408c66d840a788593690a665e0d3",
        "csa_critical_system_object": false
      },
      "container_type": "TOPOLOGY_ARTIFACT_CONTAINER",
      "tags": [
        {
          "@self": "/csa/api/tag/d22c213f7e644d1d8362b2830b2c93a9",
          "@type": "urn:x-hp:2012:software:cloud:data_model:tag",
          "name": "Development",
          "description": "Development Micro Service",
          "icon": "/csa/api/blobstore/organizationDefault58.png?tag=library",
```

```
          "color": "#ff887c",
          "scopes": [
            "LIFECYCLE_ARTIFACT_CONTAINER"
          ]
        }
      ],
      "members": [
        {
          "@self": "/csa/api/service/design/9c586c28-0fa0-4814-8b5c-eaf097a788f7",
          "@type": "urn:x-hp:2012:software:cloud:data_model:blueprint",
          "@created": "2015-05-12T10:42:49.390Z",
          "published": false,
          "version": "1.0.0"
        }
      ],
      "@total_results": 1,
      "@items_per_page": 1,
      "@start_index": 0
    }
  ],
  "@modified": "2015-05-13T08:59:44.730Z"
}
```


List packages example

Retrieves a list of all packages for an application design.

`https://localhost:8444/csa/api/codar/app-package/list`

The following JSON was returned:


```
{
  "@self": "/csa/api/package/",
  "@type": "urn:x-hp:2012:software:cloud:data_model:package",
  "redeploy": false,
  "members": [
   {
     "@type": "urn:x-hp:2012:software:cloud:data_model:package",
     "name": "DEVELOPMENT",
     "color": "#D1AF89",
    "image": "images/applications/dev.png",
     "packages": [
     {
      "@self": "/csa/api/package/d207499f-aba3-4e46-aa27-9357e2c6cb8d",
      "@type": "urn:x-hp:2012:software:cloud:data_model:package",
      "stage": "DEVELOPMENT",
      "state": "ACTIVE",
      "deploymentState": "NEW",
      "icon": "images/applications/dev.png",
```

```
      "name": "Package1",
      "deployedInstanceCount": "2",
      "lastUpdated": "2015-03-06T04:34:08.525Z",
      "description": "Package 1 Description",
      "topologyData": null,
      "instanceData": null
     },
     {
      "@self": "/csa/api/package/347d1dca-8a4d-4489-9234-8f1c10423b3f",
      "@type": "urn:x-hp:2012:software:cloud:data_model:package",
      "stage": "DEVELOPMENT",
      "state": "ACTIVE",
      "deploymentState": "NEW",
      "icon": "images/applications/dev.png",
      "name": "Package2",
      "deployedInstanceCount": "0",
      "lastUpdated": "2015-03-06T04:34:13.702Z",
      "description": "Package 2 Description",
      "topologyData": null,
      "instanceData": null
     }
   ]
  },
   {
    "@type": "urn:x-hp:2012:software:cloud:data_model:package",
    "name": "TESTING",
    "color": "#FFC300",
    "image": "images/applications/test.png",
    "packages": [
      {
         "@self": "/csa/api/package/bb2f021d-2256-4479-a5b0-86d0547d034a",
         "@type": "urn:x-hp:2012:software:cloud:data_model:package",
         "stage": "TESTING"
         "state": "REJECTED",
         "icon": "images/applications/test.png",
         "name": "Pkg1",
         "deployedInstanceCount": "1",
         "lastUpdated": "2015-01-30T05:59:11.475Z",
         "description": null,
         "topologyData": null,
         "instanceData": null
       },
    ]
 },
 {
   "@type": "urn:x-hp:2012:software:cloud:data_model:package",
   "name": "STAGING",
   "color": "#E188CA",
   "image": "images/applications/stage.png",
   "packages": []
  },
```

```
{
    "@type": "urn:x-hp:2012:software:cloud:data_model:package",
    "name": "PRODUCTION",
    "color": "#FF7D6A",
    "image": "images/applications/prod.png",
    "packages": []
  }
 ]
}
```

Get package properties example

Retrieves the properties that are parameterized in the package for each component of the application design.

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/07c979d2-afe0-4cc8-a607-
8cbf780ed725/properties
```

The following parameter was sent:

```
packageId
```

The following JSON was returned:

```
{
  "members" : [ {
    "icon" : "/csa/designer//csa/api/blobstore/Tools1.png?tag=library",
    "@self" : "/csa/api/package/component/New_PetDB_855ff95b_fde7_432e_b4dd_
b7f7e8c2ba67_e6b6ca0914a34eec93438670d70e55e4__VERSION__1__GROUPID__
com.hp.csa.type0002",
    "description" : "Creates Pet Clinic Database on the database server",
    "@type" : "urn:x-hp:2012:software:cloud:data_model:package",
    "name" : "PetClinic DB Conf",
    "provider" : null,
    "properties" : [ {
      "modifiable" : false,
      "name" : "mysqlusername",
      "value" : "root",
      "enumeration" : null,
      "displayName" : "mysqlusername",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "port",
      "value" : "22",
```

```
      "enumeration" : null,
      "displayName" : "port",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "servicecommand",
      "value" : "sh /tmp/mysqldb_conf.sh",
      "enumeration" : null,
      "displayName" : "servicecommand",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "remotefilepath",
      "value" : "/tmp/",
      "enumeration" : null,
      "displayName" : "remotefilepath",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "mysqlpassword",
      "value" : "password",
      "enumeration" : null,
      "displayName" : "mysqlpassword",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : true
    }, {
      "modifiable" : false,
      "name" : "configurationurl",
      "value" : "http://10.1.4.233:8085/userContent/mysqldb_conf.sh",
      "enumeration" : null,
      "displayName" : "configurationurl",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    } ],
    "displayName" : "PetClinic DB Conf"
  }, {
```

```
    "icon" : "/csa/designer//csa/api/blobstore/pets.png?tag=library",
    "@self" : "/csa/api/package/component/NewPetClinicApplication_
be9f0cf28800410cb95b57179fa310fd__VERSION__1__GROUPID__com.hp.csa.type0001",
    "description" : "PetClinic Application",
    "@type" : "urn:x-hp:2012:software:cloud:data_model:package",
    "name" : "PetClinic Application",
    "provider" : null,
    "properties" : [ {
      "modifiable" : false,
      "name" : "port",
      "value" : "22",
      "enumeration" : null,
      "displayName" : "port",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "servicecommand",
      "value" : "sh /tmp/petclinic_jdbc_conf.sh",
      "enumeration" : null,
      "displayName" : "servicecommand",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "localfilepath",
      "value" : "petclinic.war",
      "enumeration" : null,
      "displayName" : "localfilepath",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "remotefilepath",
      "value" : "/tmp/",
      "enumeration" : null,
      "displayName" : "remotefilepath",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
```

```
      "name" : "artifacturl",
      "value" : "http://pavan-alm.cdl.local:8652/job/Pavan_
Maven%20Project/27/artifact/Petclinic/target/petclinic.war",
      "enumeration" : null,
      "displayName" : "artifacturl",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "configurationurl",
      "value" : "http://10.1.4.233:8085/userContent/petclinic_jdbc_conf.sh",
      "enumeration" : null,
      "displayName" : "configurationurl",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    } ],
    "displayName" : "PetClinic Application"
  }, {
    "icon" : "/csa/designer//csa/api/blobstore/tomcat.png?tag=library",
    "@self" : "/csa/api/package/component/Tomcat_Server_c2bc9714_19c3_488b_bdd0_
37ee9544a21f_aade4fa90c4a4c8397d9a9bf4d141949__VERSION__1__GROUPID__
com.hp.csa.type0001",
    "description" : "Apache Tomcat Application Server",
    "@type" : "urn:x-hp:2012:software:cloud:data_model:package",
    "name" : "Tomcat Application Server",
    "provider" : null,
    "properties" : [ {
      "modifiable" : false,
      "name" : "remoteFilePath",
      "value" : "/tmp/",
      "enumeration" : null,
      "displayName" : "remoteFilePath",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "configurationUrl",
      "value" : "http://10.1.4.233:8085/userContent/install_tomcat.sh",
      "enumeration" : null,
      "displayName" : "configurationUrl",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
```

```
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "sshPort",
      "value" : "22",
      "enumeration" : null,
      "displayName" : "sshPort",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "installPath",
      "value" : "/opt/tomcat7",
      "enumeration" : null,
      "displayName" : "installPath",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "artifactUrl",
      "value" : "http://10.1.6.81:8082/job/software_
repository/ws/tomcat7/core/apache-tomcat-7.0.56.tar.gz",
      "enumeration" : null,
      "displayName" : "artifactUrl",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "serviceCommand",
      "value" : "sh install_tomcat.sh",
      "enumeration" : null,
      "displayName" : "serviceCommand",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    } ],
    "displayName" : "Tomcat Application Server"
  }, {
    "icon" : "/csa/designer//csa/api/blobstore/mysql.png?tag=library",
    "@self" : "/csa/api/package/component/MySQL_Database_20509e36_558f_4502_ba03_
f76f5d879afb_f21f3034d7ea4431981b737d81ce6299__VERSION__1__GROUPID__
com.hp.csa.type0001",
```

```
    "description" : "MySQL Database Server",
    "@type" : "urn:x-hp:2012:software:cloud:data_model:package",
    "name" : "MySQL Database",
    "provider" : null,
    "properties" : [ {
      "modifiable" : false,
      "name" : "remoteFilePath",
      "value" : "/tmp/",
      "enumeration" : null,
      "displayName" : "remoteFilePath",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "configurationUrl",
      "value" : "http://10.1.4.233:8085/userContent/install_mysql.sh",
      "enumeration" : null,
      "displayName" : "configurationUrl",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "sshPort",
      "value" : "22",
      "enumeration" : null,
      "displayName" : "sshPort",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "artifactUrl",
      "value" : "http://10.1.6.81:8082/job/software_repository/ws/mysql56/mysql-
server_5.6.21-1ubuntu12.04_amd64.deb-bundle.tar",
      "enumeration" : null,
      "displayName" : "artifactUrl",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "serviceCommand",
      "value" : "sh install_mysql.sh",
```

```
      "enumeration" : null,
      "displayName" : "serviceCommand",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    } ],
    "displayName" : "MySQL Database"
  }, {
    "icon" : "/csa/designer/pluginResources/topology/icons/SERVER.svg",
    "@self" : "/csa/api/package/component/VcenterServerType__VERSION__04.20.0000__
GROUPID__com.hp.csa.type0001",
    "description" : "vCenter Server",
    "@type" : "urn:x-hp:2012:software:cloud:data_model:package",
    "name" : "vCenter Server",
    "provider" : "VMWARE_VCENTER",
    "properties" : [ {
      "modifiable" : false,
      "name" : "vmTemplateReference",
      "value" : "ubuntu1204-hemant",
      "enumeration" : null,
      "displayName" : "vmTemplateReference",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "customizationSpec",
      "value" : "useVmName_Linux",
      "enumeration" : null,
      "displayName" : "customizationSpec",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "vmNamePrefix",
      "value" : "app",
      "enumeration" : null,
      "displayName" : "vmNamePrefix",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    } ],
    "displayName" : "vCenter Server"
  }, {
```

```
    "icon" : "/csa/designer/pluginResources/topology/icons/SERVER.svg",
    "@self" : "/csa/api/package/component/VcenterServerType__VERSION__04.20.0000__
GROUPID__com.hp.csa.type0002",
    "description" : "vCenter Server",
    "@type" : "urn:x-hp:2012:software:cloud:data_model:package",
    "name" : "vCenter Server",
    "provider" : "VMWARE_VCENTER",
    "properties" : [ {
      "modifiable" : false,
      "name" : "vmTemplateReference",
      "value" : "ubuntu1204-hemant",
      "enumeration" : null,
      "displayName" : "vmTemplateReference",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : true,
      "name" : "cpuCount",
      "value" : null,
      "enumeration" : null,
      "displayName" : "cpuCount",
      "type" : "BigDecimal",
      "modifiableDuringModification" : true,
      "required" : false,
      "confidential" : false
    }, {
      "modifiable" : true,
      "name" : "memorySize",
      "value" : null,
      "enumeration" : null,
      "displayName" : "memorySize",
      "type" : "BigDecimal",
      "modifiableDuringModification" : false,
      "required" : false,
      "confidential" : false
    }, {
      "modifiable" : false,
      "name" : "customizationSpec",
      "value" : "useVmName_Linux",
      "enumeration" : null,
      "displayName" : "customizationSpec",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    }, {
      "modifiable" : false,
```

```
      "name" : "vmNamePrefix",
      "value" : "db",
      "enumeration" : null,
      "displayName" : "vmNamePrefix",
      "type" : "String",
      "modifiableDuringModification" : false,
      "required" : true,
      "confidential" : false
    } ],
    "displayName" : "vCenter Server"
  } ]
}
```

List candidate designs example

Retrieves the service designs for a given partial design (Topology Composition) of the application design.

The following URL was sent:

```
https://localhost:8444/csa/api/codar/app-package/composition/902442d3-700c-4b57-8abb-8d1a52ae3f3d/candidateTopology
```

The following JSON was sent:

```
{
"count": 5,
"members" : [ {
"@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
"@self" : "/csa/api/topology-model/topology/7843ee06-8a5e-425e-ac0d-424e3a297d52",
"id" : "7843ee06-8a5e-425e-ac0d-424e3a297d52",

"iconUrl" : "/csa/designer/img/noimage-lightgray.svg",
"displayName" : "Infra Design",
"description" : "dfs",
"version" : "1.0.0",
"artifactId" : "40e0dabd-39e0-4e9f-a53b-4a4b9d8bb60c",
"groupId" : "com.hp.csa",
"published" : true,
"tagIds" : [ {
"@self" : "/csa/api/tag/8a818cf8ara15bc772b0145cb6efjhg99"
} ]
}, {
"@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
"@self" : "/csa/api/topology-model/topology/9b073143-8969-43c3-85af-1004084ab05d",
"id" : "9b073143-8969-43c3-85af-1004084ab05d",
"iconUrl" : "/csa/designer/img/noimage-lightgray.svg",

"displayName" : "Exst-srvr",
"description" : "",
```

```
"version" : "1.0.0",
"artifactId" : "4ce4c900-8760-4ede-9431-31737bc175a4",
"groupId" : "com.hp.csa",
"published" : false,
"tagIds" : [ ]
}, {
"@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
"@self" : "/csa/api/topology-model/topology/e7690ecf-b552-4ba1-989f-ae3922c25cda",
"id" : "e7690ecf-b552-4ba1-989f-ae3922c25cda",
"iconUrl" : "/csa/designer/img/noimage-lightgray.svg",
"displayName" : "Complete-TD-1",
"description" : "",
"version" : "1.0.0",
"artifactId" : "6eed3374-5112-4118-9f37-791b191acfeb",
"groupId" : "com.hp.csa",
"published" : false,
"tagIds" : [ ]
}, {
"@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
"@self" : "/csa/api/topology-model/topology/ab0a4b3d-6476-4e24-8fcf-73d4a8122e48",
"id" : "ab0a4b3d-6476-4e24-8fcf-73d4a8122e48",
"iconUrl" : "/csa/designer/img/noimage-lightgray.svg",
"displayName" : "Testing Configure ",
"description" : "",
"version" : "1.0.0",
"artifactId" : "790a9840-2777-422b-8e71-c3f8d87eb997",

"groupId" : "com.hp.csa",
"published" : false,
"tagIds" : [ {
"@self" : "/csa/api/tag/8a818cf8ara15bc772b0145cb6efjhg99"
} ]
}, {
"@type" : "urn:x-hp:2013:software:cloud:topology_model:topology",
"@self" : "/csa/api/topology-model/topology/ac102c58-8cdb-4e65-8083-e1e1c30f464f",

"id" : "ac102c58-8cdb-4e65-8083-e1e1c30f464f",

"iconUrl" : "/csa/designer/img/noimage-lightgray.svg",
"displayName" : "PD-Test_Complete-TD-1",
"description" : "",
"version" : "1.0.0",
"artifactId" : "a4c87694-3a68-43f7-899a-05279a64a983",
"groupId" : "com.hp.csa",
"published" : false,
"tagIds" : [ ]
} ]
}
```

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on API and CLI Reference (Codar 1.60)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to csadocs@hp.com.

We appreciate your feedback!