Hewlett Packard Enterprise

Cloud Service Automation

Propel Integration

Software version: 4.60

Document release date: January 2016

Software release date: January 2016

Contents

Introduction

Propel is a self-service catalog web application that exposes services provided by back end systems for consumption. Propel can be integrated with HPE Cloud Service Automation (CSA), so that the CSA offerings, catalogs, and marketplace layers are replaced by corresponding micro-services running in Propel.

This document describes basic concepts of the CSA-Propel integration, describes the use cases provided by the integrated solution, and provides technical details such as the adapter layer and CSA public APIs. The solution limitations are documented at the end of this document.

Basic concepts

The following concepts are enumerated and then described in detail later in this document:

- CSA takes care of resource provider definition, service design creation, fulfillment execution, and service instance and subscription management.
- Propel exposes service offerings based on CSA service designs. Propel manages catalogs and organizations, and publishes the service offerings in the Propel Shop for service consumers.
- Service instance and subscription is held in CSA but is also mirrored in Propel, so that service consumers can view and manage their purchased services.
- CSA and Propel communicate using a Service Exchange (SX) middle layer with an adapter designed specifically to interact with CSA.

CSA role

In the integrated solution, CSA plays the role of fulfillment engine. It exposes a set of APIs that provide capabilities to realize a service design and manage the running service. Service designs must be created and published in CSA and resource providers that are capable of providing the required resources must be configured.

Because catalogs are not used in the Propel integration, resource environments do not play a role. They usually link providers and catalogs together, so they are not used. Resource provider selection is done automatically using implicit logic, or it can be driven by user selected values coming in through the option model.

When a service design is fulfilled, a service instance and service subscription are created in CSA. Both entities are linked to the service design directly. There is no service offering in CSA. Both the instance and the subscription can be managed with lifecycle and public actions from the CSA Management Console.

Propel role

Propel provides a customer-facing user interface to administer organizations and to publish services for each organization. The organization administrator can publish service offerings that are fulfilled by a back end system such as CSA. Back end systems are called *suppliers* in Propel. The service offerings are loaded from suppliers using a catalog aggregation process and exposed to consumers in the Propel Shop. Realized services can be managed using subscriptions.

Service Exchange middle layer

When Propel communicates with a back end system, it uses Service Exchange (SX) as a translation layer. There is specific adapter implemented in SX for each type of back end system. The adapter knows how to translate messages going back and forth and what API should be invoked on the back end system when a specific event or operation is initiated from Propel.

Environment configuration

An environment with CSA 4.60 and Propel 2.10 is required. CSA operates independently and does not have information about Propel. CSA exposes APIs that other systems use to interact with CSA. The other systems use these APIs to fulfil services and manage each fulfilled service during its lifetime. Propel, however, is the initiator and must know where the back end system (such as CSA) is running, how to connect to it, and how to communicate with it.

Propel uses SX with a specific adapter implementation to communicate with CSA. The adapter is shipped with CSA and must be installed for SX in Propel.

To install the CSA adapter for SX into Propel, do the following steps:

1. Copy **<CSA_HOME>\sxadapter\sx-adapter-csa-ng.jar** to **<PROPEL_HOME>\sx\WEB-INF\lib**.
2. Restart the SX service.
3. Upload **<CSA_HOME>\sxadapter\content-csa-ng-r2f.jar** using the Propel administration UI:
    a. Log in to **https://<propel-hostname>:9000/org/Provider** as the super admin (admin/propel).
    b. Go to Content Management and click **Upload New Content Pack**.
    c. Browse to locate the JAR file and import it. There will be new record in the content packs list.
4. Upload Operations Orchestration content **<CSA_HOME>\sxadapter\oo-csa-ng-r2f-cp.jar** to the OO Central instance used by Propel:
    a. Log in to **https://<propel-hostname>:8443/oo** as the admin (admin/propel)
    b. Go to **Content Management** > **Content Packs**.
    c. Click **Deploy New Content** and confirm in the dialog.

End-to-end scenario

This section describes a common end-to-end scenario starting with creating a service design through publishing offerings to service management. Users with different roles perform different actions during this process. The process and the actions are elaborated in detail in the following text.

Service design

The service designer is responsible for designing service blueprints in CSA. Currently, we support two different approaches: topology designs and sequence designs. Service design creation methodology is different for each of these two types, but both types of designs can be exposed in the Propel Shop.

Creating a service design is not described in detail in this paper. See other documents to learn more about how to design a service in CSA. Remember that resource providers must be defined in CSA as well.

Before a service design is exposed to Propel, it should be tested first (currently this is only possible for topology designs) and subscriber options should be defined. When the design is ready for the consumption layer, the service designer persona can publish the design. Note that this is exactly same process for both exposing the service through the CSA Marketplace Portal and the Propel Shop. Moreover, a service can be published in both catalogs at the same time. A service offering can be created in CSA and exposed through the CSA Marketplace Portal and the service design can serve as a blueprint for a Propel service offering published in the Propel Shop (more on that later).

Only a published service design can be exposed through Propel, so do not forget to publish the service design when it is ready for consumption. Propel can be set up to load all service designs published in CSA or to load only published designs with a specific tag. Use these service design tags to mark only those designs that are intended to be exposed through Propel.

Subscriber options

Subscriber options allow you to customize an exposed service. They are defined in the service design by the service designer. When a service design is published and loaded into Propel, subscriber options are translated into service options that are available for consumers in the Propel Shop.

CSA comes with a rich structure of subscriber options, including editable properties and drop-down lists with static and dynamic values provided by a script (JSP or JavaScript). Moreover, a dynamic list of values for a property can be populated based on the value of another property, or based on the context of the user who orders the service in the CSA Marketplace Portal. Service options in Propel reflect the subscriber options in CSA. All types of properties are supported in the Propel Shop, with the exception of dynamic property values based on user context. Dynamic properties in Propel can still depend on another property value but cannot depend on the user, organization, etc.

Propel currently does not support service option modification for existing service subscriptions like CSA does. Therefore, marking an option set as modifiable during subscription modification in CSA has no effect in Propel.

Connecting Propel to CSA

Propel must be configured to connect to CSA. Note that every organization defined in Propel is a stand-alone tenant. The administrator for a specific organization configures the connection to CSA and this configuration only affects this organization.

The configuration is split into two steps: a supplier definition and an aggregation configuration.

Supplier

There are several types of suppliers supported by Propel. They are called Backend System Types and they are related to specific adapters installed into SX. The CSA adapter has to be installed into SX according to the preceding Environment configuration section on page 3. After the adapter is installed, a new CSANG supplier can be configured in Propel.

Note that there are CSA and CSANG supplier types available side by side. Always use CSANG adapter. The old CSA adapter is not recommended and the integration using the old adapter works differently from what is described in this document.

To configure the CSA as the supplier, follow these steps:

1. Login to Propel as the administrator of an organization. In case of seeded Consumer organization, just go to **https://<propel-hostname>:9000/org/CONSUMER** and log in using orgadmin/propel credentials.
2. Go to Suppliers and add a new CSANG supplier. Fill all the required parameters:
    a. Propel needs access to CSA designs, dynamic property values, requests, subscriptions, and instances. Propel uses an integration account created in CSA. The CSA admin account can be used as the integration account, if desired.
    b. CSA APIs require an IDM token. Propel must reach the CSA IDM to acquire the token for the integration user. Propel uses an IDM integration user to achieve that. You can use the idmTransportUser account that comes with CSA IDM.

Aggregation

The aggregation is set up on top of the supplier that was configured in the previous section. The aggregation definition can be created on the Catalog Connect page and consists of selecting a supplier and filling an optional query filter. In case of CSANG supplier, all published service designs are loaded from CSA and a catalog item is created for each. The query filter can restrict the set of designs to aggregate by specifying a service design tag, and only the service designs associated with that tag will be aggregated to Propel.

The service design aggregation can be either manual or automatic. Manual aggregation is triggered on demand by user interaction. Automatic aggregation spawns a process that checks CSA every 30 seconds for changes and automatically propagates the changes to Propel.

Note that there is a known issue in Propel 2.10 related to aggregation. If the aggregation is configured in advance and no service design has been published, the aggregation fails with the following message: *(1) Invalid aggregation configuration. Please change parameters and try again.* A single published service design is enough to avoid this problem.

Catalog items are created in Propel when the aggregation process (manual or automatic) is executed. The catalog items refer to service designs published in CSA.

Publishing offerings

Service offerings are catalog items created in Propel as a result of the aggregation process. The administrator of the organization can further tune the offerings. Service options that are generated based on the underlying service design subscriber options can be edited. You can set default values, hide, or reorder options. Pricing for the offering can be defined same as other aspects such as access control, approvals, and attachments.

The administrator of the organization can publish service offerings in different Propel catalogs within the organization, and each published copy of the offering can be set up differently, such as different default values, pricing, access control, approvals, and attachments. After the offering is published in a catalog, consumers with access to that catalog and offering can order it in the Propel Shop.

Shopping for services

The Propel Shop is the place where consumer users go to shop for services. Consumers can order services in the Propel Shop based on access rights. The service order and checkout process is straightforward. A consumer user can customize the service using available options and either check out instantly or put the service into a cart and continue shopping.

Propel allows users to order multiples of each customized service with the same options selected. Propel sends multiple requests for different items in the cart but identical items are wrapped into a single request with a quantity value. CSA cannot handle this type of bulk order. CSA always fulfills only one deployment for each incoming request. So, it understands different items in the cart well but it does not recognize multiples of the same item.

Propel sends one or more order requests to CSA. Each request contains information about the service, options, requester, and organization. CSA accepts the request, then creates the organization and user if they do not exist already, to mirror the requester and initiates the service fulfillment. The resulting service instance and subscription belongs to the same user and organization in CSA as in Propel.

Managing services

The information about the service instance and subscription is automatically synchronized from CSA back to Propel, so that the consumer user can view the subscription and instance details in Propel. The service subscription can be terminated or modified from the Propel Subscriptions view. Public actions can be invoked on the service instance as well. Each modification or action invocation posts a request to CSA, where the request is fulfilled and the result is synchronized back to Propel.

The subscription modification capabilities are limited in Propel. Changing the configuration options on a running service is not possible in Propel. The modifiable parameters of the subscription are the display name, description, and termination date.

Operations

Consumer users purchase and manage services using the Propel Shop. However, if something goes wrong, an operator can access the low-level information about the service fulfillment using the CSA Management Console. The operator can drill-down into the Operations Orchestration flow execution details and figure out the problem.

The CSA Management Console displays service subscriptions and instances that were initiated from Propel in similar fashion as it does subscriptions and instances initiated from the CSA Marketplace Portal. The difference is that the Propel subscriptions are based on service designs and the information about corresponding service offerings stays in Propel only and is not available in the CSA Management Console.

Technical details

Technical details for CSA – Propel interaction are described in this chapter. CSA exposes a number of public HTTP APIs, providing capabilities to load service designs, post requests for fulfillment, and periodically check for changes in the related data.

Service design API

- GET https://[host]:[port]/csa/api/service/design
- POST https://[host]:[port]/csa/api/service/design/filter
- GET https://[host]:[port]/csa/api/service/design/[designID]
- GET https://[host]:[port]/csa/api/option-model/[optionModeID]

These endpoints list service designs, search for designs using a filter, get the design detail, and get the design option model. All endpoints require basic authentication header. User with sufficient privileges must be used. The usage, parameter and payload description is available in the swagger documentation published on the running CSA instance at https://[host]:[port]/csa/apidocs.jsp

Propel uses these endpoints so list designs and to aggregate them in form of offerings that can be published to the Propel Shop.

Request API

- POST https://[host]:[port]/csa/api/consumption/v2/request

The request API is a versatile interface for service fulfillment and management. It is used by Propel to realize and manage services purchased by customers in the Propel Shop.

An X-Auth-Token is required to call the request endpoint. The X-Auth-Token can be retrieved from IDM service using an IDM integration user (aka idmTransportUser). Get the token for a user with sufficient privileges to post request into CSA.

The following requests are supported: order, action execution, modification, and cancelation. Each request has its own specific body. All those requests are posted on the same endpoint:

- POST https://[host]:[port]/csa/api/consumption/v2/request/?onBehalf=[username]&onBehalfOrg=[orgname]

The request parameters are described in the following table.

[username]	User name of the request owner. In case the user does not exist in the system, it is automatically created.
[orgname]	Name of the organization the user described above belongs into. If the organization does not exist, it is automatically created. Organizations created in this way have *(Propel)* suffix in their display name.

The result of the request creation is the request ID that can be used to get the request detail containing information about related subscription and instance.

- GET https://[host]:[port]/csa/api/consumption/v2/request/[requestID]

Order request

The order requests initiates a design/offering fulfillment resulting in an active subscription and running service instance. Both subscription and instance are owned by the request owner.

Order Request Body

```
{
"action": "ORDER",
"subscriptionName": "POR00057-PR00058 Fullfilment demo design 1.0.0",
"contextId": "8fe9cd96521604720152164876730 10f",
"subscriptionDescription": "",
"startDate": "2016-01-06T09:57:47.000Z",
"endDate": "2017-01-06T09:57:47.000Z",
"fields": {
  "field_8fe9cd965216047201521649l8d6011b": true,
  "field_8fe9cd96521604720152l650b47d013c": "1"
          }
}
```

contextId	UUID of an artifact that is the subject of the fulfillment. The artifact can be either a service design or a service offering.
fields	Input values for option model properties. It is a flat list of pairs [*ID: value*] representing selected options and property values.
field_[id]	ID of the field. A field can be an option or property in option model. To get the filed IDs, load the option model details using an option-model API described in Service Design API section. Selected options have value *true*. Properties have the actual set value.

Action request

The action request initiates a public action execution on an existing service subscription.

Action Request Body

```
{
"action":"sample_xpath_query_ed84cf566f5b43d899f7f6dfd0eb017e",
"contextId":"8fe9cd96521604720152165f1aea0218",
"subscriptionId":"8fe9cd96521604720152165ec7c60199",
"fields": {}
}
```

action	ID of the public action that should be executed.
contextId	UUID of an artifact the action is invoked on. The artifact is the component resource binding in this case.
subscriptionId	ID of the service subscription the action is executed on.

Modification request

The modification request changes an existing service subscription. The subscription end date, selected options, name and description can be changed. Note that Propel lacks the capability to change options selected during the initial order. The CSA API supports the option modification but it is not used by Propel.

Modification Request Body

```
{
"action": "MODIFY_SUBSCRIPTION",
"contextId": "8fe9cd96521604720152165ec7c60199",
"subscriptionName": "POR00057-PR00058 Fullfilment demo design 1.0.0",
"subscriptionDescription": "",
"endDate":"2017-01-06T22:59:55.000Z"
}
```

contextId	UUID of an artifact to modify. The artifact is the service subscription in this case.

Cancelation request

The cancelation request terminates an existing service subscription and cancels the running instance.

Cancelation Request Body

```
{
"action": "CANCEL_SUBSCRIPTION",
"contextId": "8fe9cd96521604720152165ec7c60199"
}
```

contextId	UUID of an artifact to cancel. The artifact is the service subscription in this case.

Checking for changes

There are published endpoints that provide a way to effectively check for changes made to artifacts stored in CSA. This is used for data synchronization between CSA and Propel. An endpoint exists for synchronization of data related to service designs aggregated as offerings into Propel

- GET https://[host]:[port]/csa/api/service/design/published/changes

Since this is an endpoint within the service design API, the same rules apply. Refer to the swagger documentation for more information.

After a request for fulfillment is placed, Propel is checking for updates of the request itself and of the related service subscription and instance. Consumption V2 API exposes the endpoints

- GET https://[host]:[port]/csa/api/consumption/v2/request/changes
- GET https://[host]:[port]/csa/api/consumption/v2/subscription/changes
- GET https://[host]:[port]/csa/api/consumption/v2/instance/changes

The list of changes within a specific time frame can be requested using parameters

from	Optional. Beginning of the time frame, inclusive. Format: yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
to	Optional. End of the time frame, exclusive. Format: yyyy-MM-dd'T'HH:mm:ss.SSS'Z'

Example: …/service/design/published/changes?from=2016-01-08T12%3A56%3A27.389Z

Dynamic properties

Propel is calling an API to fetch values for dynamic properties.

- POST https://[host]:[port]/csa/api/consumption/v2/property/[propertyID]

```
{
"offset": "0",
"limit": "50",
"language": "en_us"
}
```

Known problems and limitations

There are several issues with the CSA – Propel integration that are not addressed in this release.

Invalid aggregation configuration

If the service design aggregation is configured in advance and no service design has been published in CSA, the aggregation setup fails with the following message: *(1) Invalid aggregation configuration. Please change parameters and try again.* A single published service design is enough to work around this problem.

Context token in dynamic property

In CSA, dynamic properties support various tokens as input parameters. The tokens represent other property values or a value from the user contenxt, such as user information, organization information, etc. The context tokens are not supported in Propel. However, dynamic properties can still depend on another property value by using the client property token.

Limited subscription modification

Propel currently does not support service option modification for an existing service subscription like CSA does. Therefore, marking an option set as modifiable during subscription modification in CSA has no effect in Propel.

Changing the service options on a running service is not possible in Propel. The only modifiable parameters of the subscription are the display name, description, and termination date.

No support for group owned subscription or transfer of ownership

There is no way to create a subscription shared by a group of users and the ownership cannot be changed.

Limited shopping cart

Propel allows users to order multiples of each customized service with the same options selected. Propel sends multiple requests for different items in the cart but multiples of the same item are wrapped into a single request with a quantity. CSA cannot handle this type of bulk order. CSA always fulfills only one deployment for each incoming request. So, it understands different items in the cart well but it does not recognize multiples of the same item.

Send documentation feedback

If you have comments about this document, you can send them to clouddocs@hpe.com.

Legal notices

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 the following URL and sign-in or register: https://softwaresupport.hp.com.

Select Manuals from the Dashboard menu to view all available documentation. Use the search and filter functions to find documentation, whitepapers, and other information sources.

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

Support

Visit the Hewlett Packard Enterprise Software Support Online web site at https://softwaresupport.hp.com.