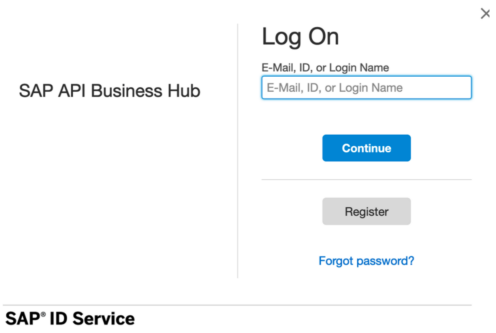
Exercise 1: Using 3scale to manage APIs from SAP’s Business Hub

This scenario will show how to configure 3scale to centralize API management from SAP Business Hub and enrich those APIs with all the capabilities 3scale can offer. To demonstrate this, you are going to use the Analytical reporting APIs from SAP Ariba, but this same procedure can apply to any other API from SAP Business Hub.

A user in https://api.sap.com is required to complete this configuration. A free account can be created, using the Log On button on the top right and using the Register option when prompted for your login details.

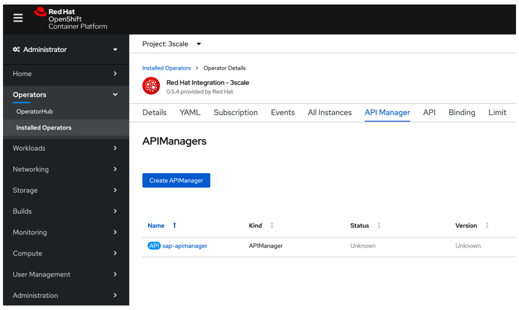


SAP API Business Hub

Log On

E-Mail, ID, or Login Name

Once you have created a user for SAP Business Hub, you can now login into **Red Hat 3scale API Management** which has been deployed already in your environment using the Red Hat Integration - 3scale Operator in the 3scale Namespace.



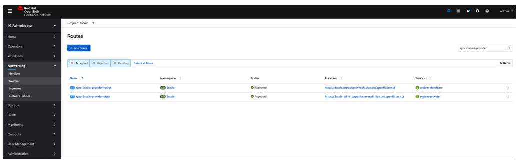
Red Hat Integration - 3scale

0.5.4 provided by Red Hat

APIManagers

Name	Kind	Status	Version
sap-apimanager	APIManager	Unknown	Unknown

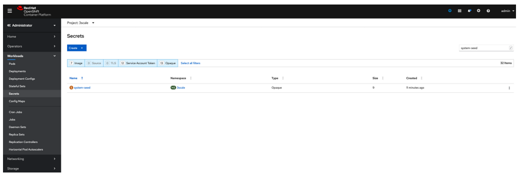
You can find all the exposed Routes for 3scale under Networking - Routes in the 3scale Project. If you filter by zync-3scale-provider both the 3scale API Management and Developer Portal URLs info will be shown. You will receive this information as well in the email you will get with all the environment information.



Routes

Name	Namespace	Status	Location	Service
zync-3scale-provider-np9gt	3scale	Accepted	https://3scale.apps.cluster-maki.blue.osp.opentlc.com	system-developer
zync-3scale-provider-xkpjx	3scale	Accepted	https://3scale-admin.apps.cluster-maki.blue.osp.opentlc.com	system-provider

Credentials for 3scale API Management are provided in the email you will get with all the environment information, but you can check those anytime in the system-seed secret in your 3scale Namespace under Workloads - Secrets.

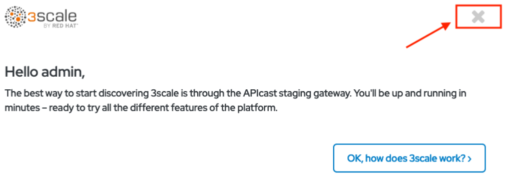


Secrets

system-seed

Name	Namespace	Type	Size	Created
system-seed	3scale	Opaque	9	11 minutes ago

The first time you open the 3scale API Management portal you will be asked for these credentials. Enter the information required and you will be presented with the assistant to configure your APIs. We are going to skip this part using the **X** symbol on the top right and configure all the required information manually afterwards.



Hello admin,

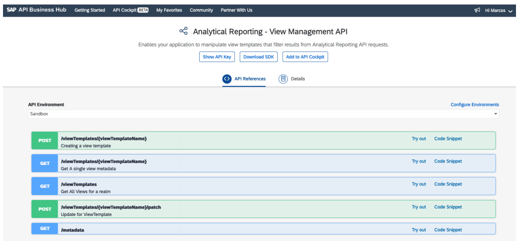
The best way to start discovering 3scale is through the APIcast staging gateway. You'll be up and running in minutes – ready to try all the different features of the platform.

After skipping the assistant, you will be presented to the 3scale API Management **Dashboard**. You will create a new **PRODUCT** using the NEW PRODUCT link on the bottom section, adding the following values when prompted:

- Name: SAP Business Hub
- System name: sap_business_hub
- Description: SAP Business Hub

[New product]

Once the new Product has been created, you need to create a **BACKEND** that will be used for this Product. This Backend will be used to define our API configuration, including the Private Base URL, Methods and Mapping Rules. At this point, you will go back to Analytical Reporting - View Management API definition in SAP Business Hub to understand the existing API References and Specifications for that particular API. As you can see there are 4 different methods available for this API.



Analytical Reporting - View Management API

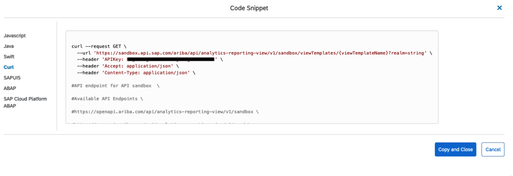
Enables your application to manipulate view templates that filter results from Analytical Reporting API requests.

API Environment

Sandbox

POST	/viewTemplates/{viewTemplateName} Creating a view template
GET	/viewTemplates/{viewTemplateName} Get A single view metadata
GET	/viewTemplates Get All Views for a realm
POST	/viewTemplates/{viewTemplateName}/patch Update for ViewTemplate
GET	/metadata

If you click on the Code Snippet link for any of these methods, you can see the URL used to call this method (you will get the information from the Sandbox System URL, for production purposes another one from the available list will be used, but not for this demo) as well as values that can be parametrized or a custom header that needs to be added to this request which requires authentication.



```
curl --request GET \
  --url 'https://sandbox.api.sap.com/ariba/api/analytics-reporting-view/v1/sandbox/viewTemplates/{viewTemplateName}?realm=string' \
  --header 'APIKey: [redacted]' \
  --header 'Accept: application/json' \
  --header 'Content-Type: application/json' \

#API endpoint for API sandbox  \

#Available API Endpoints \

#https://openapi.ariba.com/api/analytics-reporting-view/v1/sandbox \
```

Once you have captured all this information, you can go back to the 3scale API Management's dashboard, where you will create a new **BACKEND** adding the following values when prompted:

- Name: Analytical Reporting
- System Name: analytical_reporting
- Description: SAP Bussines Hub Analytical Reporting
- Private Base URL: https://sandbox.api.sap.com:443/ariba/api/analytics-reporting-view/v1/sandbox

[New backend]

You have successfully created a Product and a Backend. You need to link those now, adding the Backend you created to the Product, and specify a Public Path for this one. To do this, browse to the top menu and select the SAP Business Hub Product you have created. From the left menu, click on Integration - Backends. The list of Backends associated to this Product will be empty. Click on Add Backend link on the right and select the Analytical Reporting Backend from the dropdown menu. In the Path field you will add /**reporting** as the path to match the Backend for this Product.

[Link Backend]

Now the Backend has been added to the Product, you need to configure a new Policy for the Product. You are going to use this Policy to add a custom header to every request is made to SAP Business Hub. While examining the API References and Specifications for the Analytical Reporting - View Management API, on the Code Snippet for each method you could see an authentication header required to authenticate every request. This information is the API Key for the user who is making the request. Taking as an example what a curl command should look like for one of the methods exposed, this looked like this:

```
curl --request GET \
  --url 'https://sandbox.api.sap.com/ariba/api/analytics-reporting-view/v1/sandbox/viewTemplates?realm=string' \
  --header 'APIKey: XXXXXXXXXXXXXXXXXXXXXXXXXXX' \
  --header 'Accept: application/json' \
  --header 'Content-Type: application/json'
```

You need to use that APIKey information now and add it to the new Policy you are going to create for your SAP Business Hub Product in 3scale. To do this, browse to the top menu and select the SAP Business Hub Product you have created. From the left menu, click on Integration - Policies. Select Add Policy link, and from the list of policies you will be presented, select Header Modification. This will be added automatically to the Product policies but is not configured yet. To do this, click on the Header Modification Policy from the Policy Chain list, and use the + button from the **REQUEST** section. Add the following information for this Policy:

- op: Create the header when not set, add the value when setting
- header: APIKey

- value_type: Evaluate 'value' as plain text
- value: XXXXXXXXXXXXXXXXXXXXXXXXXXXX (this value must contain the real value for your API Key in SAP Business Hub)

[3Scale policy]

The Product has now the required Policy configured to authenticate a request to SAP Business HUB’s APIs. The next thing to do is to create an Application Plan to enable developers and users to consume your API. An Application Plan will be used to establish the rules (limits, pricing, features) for using your API. By default, your API will require every request to be authenticated using an API Key. You are going to create later an Account containing one Application linked to this Application Plan that will contain the required API Key for authentication. This way the metrics for who is consuming you API can be measured and apply things as billing or limits peruse.

To create an Application Plan for your Product, browse to the top menu and select the SAP Business Hub Product you have created. From the dashboard, on the Published Application Plans section, use the Create Application Plan link to create one. Add the following information to this Application Plan, leaving the rest of the fields with the defaults values:

- Name: Free Unlimited
- System name: free_unlimited

Once created, the Application Plan will be in hidden state. To publish it, click on the Publish link for that Application Plan.

[3Scale application plan]

The next thing you need to do is to create a new Account. This Account will be the top-level information for the API consumers and can contain multiple objects like Applications, Users or Subscriptions. To create an Account, select Dashboard from the top menu and click on the ACCOUNT link on the top. An Account called 'Developer' already exists in the system, but you are going to create a new one using the Create link from the right providing the following information:

- Username: account_admin
- Email: admin@myorg.com
- Password: (use any random password)
- Organization/Group Name: MyOrg

Once the Account is created, you will see different links to objects linked to the Account like Application, Users or Service Subscriptions on the top. You need to create an Application for this Account now, which will generate the API Key required for authentication. Click on the Application link from the menu on the top, and then use the Create Application link on the right. Add the following information to create the Application:

- Application Plan: SAP Business Hub - Free Unlimited
- Service Plan: Default
- Name: SAP APIs Unlimited

- Description: SAP APIs Free Unlimited

Once the Application has been created, you will be able to see the API Key required to authenticate a request for this Application for this Account.

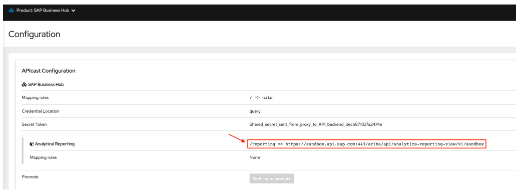
[3Scale new account]

At this point, you should be able to use your **SAP Business Hub** Product you have created in 3scale. There is only one step missing, as the APIcast has not been promoted yet to Staging or Production. This capability will allow you to make changes in your Product, promote those changes to your Stage environment and once validated promote it to Production.

Let’s promote your APIcast configuration to Stage, validate you can use any of the methods exposed by the Analytical Reporting - View Management API and promote this to Production if everything is working as expected. To do this, browse to the top menu and select the SAP Business Hub Product. From the left menu go to Integration - Configuration and click on the Promote to Staging APIcast link.

[3Scale promotion]

Once the APIcast has been promoted to Stage you can test it. From the same window where you have promoted to Stage, check the Staging APIcast section and get the URL from Example curl for testing. There are different ways to check this. You can use your browser, or you can use curl or a tool like Postman, which is the option selected for this demo. Check the Analytical Reporting backend configuration on the APIcast Configuration section form the same window you did the promotion. You will see the initial configuration done for the Analytical Reporting Backend you did when configuring the Backend. The Public Path configured for this one, in particular, is /reporting, and this is the path we need to add to the Stagging URL to test our API.



Configuration

APIcast Configuration

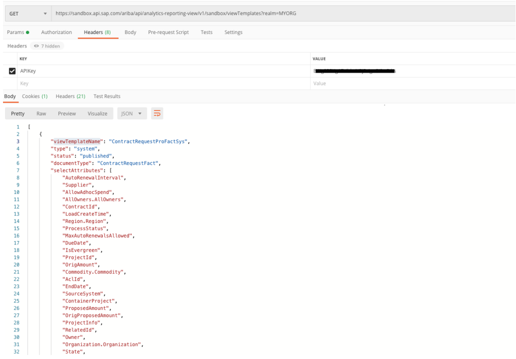
SAP Business Hub

Mapping rules	/ => hits
Credential Location	query
Secret Token	Shared_secret_sent_from_proxy_to_API_backend_3ecb87102fa2474a
Analytical Reporting	/reporting => https://sandbox.api.sap.com:443/ariba/api/analytics-reporting-view/v1/sandbox
Mapping rules	None
Promote	Nothing to promote

Let’s test one of the methods supported for this API in SAP Business Hub. You can check the Swagger Specification here. You can use the GET /viewTemplates method as an example or any other of the available methods for this API. When checking this method, we can see the request should look like the following when using the SAP Bussiness Hub Sandbox directly:

```
curl --request GET \
  --url 'https://sandbox.api.sap.com/ariba/api/analytics-reporting-view/v1/sandbox/viewTemplates?realm=string' \
  --header 'APIKey: XXXXXXXXXXXXXXXXXX' \
  --header 'Accept: application/json' \
  --header 'Content-Type: application/json'
```

Your test in Postman should look like this:

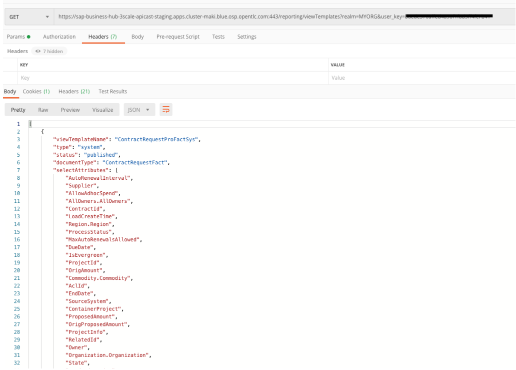


GET https://sandbox.api.sap.com/ariba/api/analytics-reporting-view/v1/sandbox/viewTemplates?realm=MYORG

KEY	VALUE
☑ APIKey	[redacted]
Key	Value

```json
[
    {
        "viewTemplateName": "ContractRequestProFactSys",
        "type": "system",
        "status": "published",
        "documentType": "ContractRequestFact",
        "selectAttributes": [
            "AutoRenewalInterval",
            "Supplier",
            "AllowAdhocSpend",
            "AllOwners.AllOwners",
            "ContractId",
            "LoadCreateTime",
            "Region.Region",
            "ProcessStatus",
            "MaxAutoRenewalsAllowed",
            "DueDate",
            "IsEvergreen",
            "ProjectId",
            "OrigAmount",
            "Commodity.Commodity",
            "AclId",
            "EndDate",
            "SourceSystem",
            "ContainerProject",
            "ProposedAmount",
            "OrigProposedAmount",
            "ProjectInfo",
            "RelatedId",
            "Owner",
            "Organization.Organization",
            "State",
```

You should see that a list of Templates will be retrieved from that call. Let's try to reproduce this now, but using the Backend you have configured in 3scale. Your test in Postman should look like this:

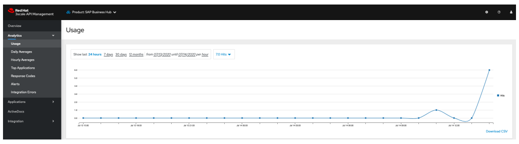


GET https://sap-business-hub-3scale-apicast-staging.apps.cluster-maki.blue.osp.opentlc.com:443/reporting/viewTemplates?realm=MYORG&user_key=[redacted]

KEY	VALUE
Key	Value

```json
[
    {
        "viewTemplateName": "ContractRequestProFactSys",
        "type": "system",
        "status": "published",
        "documentType": "ContractRequestFact",
        "selectAttributes": [
            "AutoRenewalInterval",
            "Supplier",
            "AllowAdhocSpend",
            "AllOwners.AllOwners",
            "ContractId",
            "LoadCreateTime",
            "Region.Region",
            "ProcessStatus",
            "MaxAutoRenewalsAllowed",
            "DueDate",
            "IsEvergreen",
            "ProjectId",
            "OrigAmount",
            "Commodity.Commodity",
            "AclId",
            "EndDate",
            "SourceSystem",
            "ContainerProject",
            "ProposedAmount",
            "OrigProposedAmount",
            "ProjectInfo",
            "RelatedId",
            "Owner",
            "Organization.Organization",
            "State",
```

As you can notice, you don't need to provide the APIKey Header to authenticate with SAP Business Hub anymore, as this is automatically added through the Policy you have configured for your Product in 3scale. Try to reproduce this request a few times in Postman and you should see initial statistics coming up in 3scale.



You can try to do the same with the other methods exposed by the Analytical Reporting - View Management API and check the results are the same when doing the request to 3scale APIcast or doing it directly in SAP Business Hub. If we want users and developers to consume these APIs, they will need to get access to the Swagger Specification. To facilitate this we can import these specifications into 3scale. Go back to Analytical Reporting - View Management API definition in SAP Business Hub and click on Download API Specification button and download it as a **JSON** file. Go back to 3scale API Management portal and select Audience from the top menu. Then select Developer Portal - ActiveDocs and use the Create a new spec link from the right. Add the following information when prompted:

- Name: Analytical Reporting - View Management API
- System Name: AnalyticalReportingView
- Publish?: Ensure is **checked**
- Description: Analytical Reporting - View Management API
- Service: SAP Business Hub

- API JSON Spec: (copy the content from the JSON file you just downloaded)
- Skip swagger validations: Ensure is **checked**

[Swagger]