Red Hat OpenShift API Management 1

Adding and configuring APIs

Adding and configuring APIs in Red Hat OpenShift API Management.

Last Updated: 2024-04-11

Red Hat OpenShift API Management 1 Adding and configuring APIs

Adding and configuring APIs in Red Hat OpenShift API Management.

Legal Notice

Abstract

This guide provides information about how you can add and configure APIs in Red Hat OpenShift API Management.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

CHAPTER 1. ADDING AND CONFIGURING APIS IN RED HAT OPENSHIFT API MANAGEMENT

As a 3scale API provider, you can add and configure your APIs.

Prerequisites

- You have accessed OpenShift API Management.

1.1. PRODUCTS AND BACKENDS FOR 3SCALE APIS

Red Hat 3scale API Management separates your APIs into two main groups:

- Backends: Internal APIs bundled in a product. Backends grant API providers the freedom to map their internal API organization structure to 3scale.
- Products: Customer-facing APIs. Products facilitate the creation of strong and simplified offerings for API consumers.

A product can contain multiple backends, and a backend can be used in multiple products. In other words, to integrate and manage your API in 3scale you need to create both:

- A backend containing at least the URL of your API. The backend can optionally be configured with mapping rules, methods and metrics to facilitate reusability.
- A product for which you define the application plans, and configure APIcast.

1.2. CREATING NEW PRODUCTS TO TEST API CALLS

As a 3scale API provider, create products to test API calls through these public APIs. A product is a customer-facing API that packages one or more backends.

You can create a new product by following one of these options:

- Define the product manually.
- Import the product from OpenShift.

Here you will find details about the manual definition. If you want to import a product from OpenShift, see Service Discovery.

Prerequisites

- You need a 3scale account.

Procedure

1. Go to the Dashboard. Under the APIs section, click **Create Product** in the *Products* card.
2. Provide the following details:
   a. **Name**: Product identifier.

b. **System name**: Identifier used for internal purposes. A product **system_name** is used to generate proxy endpoints and domain names. By default, **system_name** is part of a label whose pattern can be as one of the alternatives below:

- For APIcast staging: **%{system_name}-%{tenant_name}-apicast-staging**
- For APIcast production: **%{system_name}-%{tenant_name}-apicast-production**
- When an auto-generated URL label exceeds 63 characters, the system shortens the label as follows: **<truncated-label>-<unique-hash>**
  - **<truncated-label>** is the first 54 or 55 characters of the original URL.
  - **<unique-hash>** is the first 7 characters of a unique SHA-1 hash calculated from the original label.
    For example, this is the URL before truncation:

    https://my-very-long-system-name-also-very-long-tenant-name-apicast-staging.3scale.net

    This is the URL after the truncation:

    https://my-very-long-system-name-also-very-long-tenant-name-api-72588d2.3scale.net

c. **Description**: Optional field containing more details about the product.

3. Click **Create Product**.

After these steps, you have a product that represents the public facing API. The next steps are creating backends and adding them to the product .

1.3. CREATING BACKENDS FOR YOUR PRODUCTS

A backend is an internal API that is bundled to a product.

Prerequisites

- You need a 3scale account.

Procedure

1. Go to the Dashboard.
2. Under the APIs section, click **Create Backend** in the *Backends* card.
3. Provide the following details:
   - *Name*: Backend identifier.
   - *System name*: Identifier used for internal purposes.
   - *Description*: Optional field containing more details about the backend.
   - *Private endpoint*: Base URL of the private API.
4. Click **Create Backend**.

After performing the steps for *Creating backends for your API* , you have an internal API. Add more backends as needed.

Next steps

- Add this backend to a product .

1.4. ADDING BACKENDS TO YOUR PRODUCTS

By the end of this procedure you will have added a backend to a product. Repeat the procedure to add more backends as you require them.

Prerequisites

- A product. To create one, see Creating new products .
- One or more backends. To create one, see Creating backends for your products .

Procedure

1. Navigate to **[Your_product_name] > Integration > Backends**
2. Click **Add Backend**.
3. From the drop down list, select an existing backend.
4. Specify the routing path in the *Path* textbox. For more details, see The backend path of a specific product.
5. Click **Add to Product**.

After these steps, your product will have a backend. You can follow this procedure again to add more backends to a product or to multiple products.

1.5. THE BACKEND PATH OF A SPECIFIC PRODUCT

When you add a backend to a product, you define the path that the backend uses to communicate with the specified product. This path is not part of the backend.

From the procedure described in Adding backends to your products , APIcast is the API gateway that uses the path of the backend you indicated in step 4. APIcast redirects the traffic to the backend with the specified path matching the prefix of the requested endpoint path.

When defining the path for a backend:

- You can indicate / as the path of one of the backends.
- Paths must be unique inside the product. This means that you cannot add two backends with the same path inside the same product. Neither can you add the same backend twice to the same product.
- You can give the same backend the same path in different products.

This is how the backend path works:

- Each backend added to a product is mounted in the specified path.
- Before redirecting the traffic, the path is removed from the public URL of the request to the product.

Example

Consider this configuration to add a backend to a product:

- Backend: Inventory
- Resource path: **/list**
- Product: Petstore
- Backend path: **/supplies**

These are the URLs used by your configuration:

- Public URL: **<public-api-domain>/supplies/list**
- Private URL: **<private-api-domain>/list**

This is the action flow when a request is sent:

1. The application sends a request.
2. The request is sent to the public URL: **<public-api-domain>/supplies/list**
3. The backend path is removed before redirecting to the private URL: **<private-api-domain>/list**
4. The request is redirected to the **Inventory** backend.

1.6. DEFINING MAPPING RULES

A mapping rule associates a call to an endpoint with designated methods and metrics for tracking and limiting access to your API. You can define mapping rules at the backend and product levels. The advantage of defining mapping rules at the backend level is that you can add a backend to multiple products. To learn more about the metrics or methods for which to collect usage information depending on the requests to your API, both at the product and backend levels, see How APIcast applies mapping rules for capturing usage of 3scale APIs.

Prerequisites

- A backend. To create it, see Creating backends for your products .

Procedure

1. Navigate to the *Backend* you want to define mapping rules for.
2. In the page containing information about the backend, navigate to **Mapping Rules**.
3. Click **Add Mapping Rule**.
4. Specify the following settings:
    - *Verb*: The HTTP request verb (**GET**, **POST**, **DELETE**, or **PUT**).

- *Pattern*: The pattern to match. For example, /**hello**.
- *Metric or method to increment* : The metric or method name.
- *Increment by*: The metric increment number. For example, **1**.
- *Last?*: If this mapping rule should be considered as the last one, to stop processing other mapping rules.
- *Position*: Number that indicates the position of the execution of the mapping rule, to sort the mapping rules.

5. Click **Create Mapping Rule**.

Next steps

After these steps, the mapping rule is added to **Backends** under **[Your_API_backend] > Mapping Rules**. The mapping rule is also available for each product currently using the backend. To have the mapping rule active at the product level, promote the latest configuration under the **Products** tab in **[Your_product_name] > Integration > Configuration**

Example

After you promote the configuration, 3scale activates the backend mapping rules at the product level. The mapping rules follow the backend path specified in the product. For example, suppose you have this configuration:

- Pattern of the mapping rule at the backend: /**thousands**
- Backend is added to a product with path: /**unitprice**

The mapping rule at the product level is: /**unitprice/thousands**.

1.7. CREATING 3SCALE APPLICATION PLANS FOR PRODUCTS

A 3scale application plan defines the rules such as limits, pricing, and features for using your API product. For more information, refer to Application plans and Designating methods and adding metrics for capturing usage details.

Prerequisites

- A product. To create one, see Creating new products.

Procedure

1. Navigate to **[Your_product_name] > Applications > Application Plans**
2. Click **Create Application Plan**.
3. On the **Create Application Plan** page, enter a name and a system name for your new plan. A system name must be unique in your 3scale installation.
4. Click **Create Application Plan**.

1.8. CREATING APPLICATIONS FOR THE DEFAULT ACCOUNT TO TEST API CALLS

As a 3scale user, create applications for the default *Developer* account. An application subscribes to an application plan. As a result of this subscription, 3scale provides the user key required to call an API product.

An application is always associated with an application plan. Applications are stored within developer accounts. In basic 3scale plans only a single application is allowed. In enterprise plans, multiple applications per account are allowed.

Prerequisites

- An application plan. To create one, see Creating 3scale application plans for products .

Procedure

1. Navigate to **Audience > Accounts > Listing**
2. Click the default *Developer* account.
3. Go to the **Application** tab.
4. Click **Create Application**.
5. Choose an application plan.
6. **Service plan** contains the product plan configured by the Account Manager, which will be associated to your application.
7. Specify an application name.
8. In the *Description* field, enter information about this application.
9. Click **Create Application**.

You can see your new application in **Dashboard > Audience > Accounts > Applications > Listing**

1.9. SENDING REQUESTS TO YOUR PRODUCT TO TEST THE INTEGRATION OF A BACKEND

As a 3scale API provider, you can send a command line request to a product to test the integration of a backend based on the first mapping rule added to the product.

Before you can send a test request, you must promote an APIcast configuration that includes the backend that you want to test. A specific APIcast configuration consists of the backends you have added to a product with the corresponding mapping rules, applications, and application plans.

3scale directs requests to the backend of a product according to the path specified in the request call. For each backend of a product, you configured the backend path when you added the backend to the product. In other words, each backend has its own path.

Prerequisites

- One or more backends that you added to a product .
- A mapping rule for each backend in a product.
- An application plan to define the access policies.

- An application that subscribes to the application plan.

Procedure

1. Promote the new APIcast configuration to staging:
   a. Navigate to **[Your_product_name] > Integration > Configuration**
   b. Under *APIcast Configuration*, click **Promote v.[n] to Staging APIcast**
      - **v.[n]** indicates the version number to be promoted.
      - If there are no changes to be promoted, there is a grayed button with the text **Nothing to promote**.
2. Under *Staging APIcast*, promote the APIcast configuration to production by clicking **Promote v.[n] to Production APIcast**.
   - **v.[n]** indicates the version number to be promoted.
   - If there are no changes to be promoted, you will see a grayed button with the text **Nothing to promote**.
3. To test requests to your API product, copy the command provided in *Example curl for testing* and run it in a terminal.
   - The **curl** command example is based on the first mapping rule of the product.
   - After you run the command, you should get an HTML response containing results from the backend you are testing.
   - If you do not get a response, delete the catch-all mapping rule from your product, promote the new APIcast configuration to staging and then to production, and run the example **curl** command.

Next steps

You can confirm the different responses when changing metrics and methods, such as limits and pricing rules. For any of the application plans of a product, modify the methods and metrics when testing requests to your product. For more details, refer to adding methods and metrics.

Every time you modify the product configuration and before making calls to your API, you must promote the updated configuration to the staging and production environments. When there are pending changes to be promoted to the staging environment, there is an exclamation mark in the Admin Portal, next to the **Integration** menu item.