

Enable Dynamic Routing

11.3

Define Route Rules to enable dynamic routing in an API Proxy.

Context

A route connects an API Proxy EndPoint to an API Target EndPoint. It governs the path of a request from proxy endpoint to target endpoint and determines which target endpoint to invoke based on the condition defined in Proxy EndPoint definition. Typically, a route includes a URL used to access the API Proxy EndPoint and a URL of the backend service defined in Target EndPoint definition.

In API Management, when you create an API proxy, a default Route Rule is set and it always forwards the request to the default target endpoint defined in Target EndPoint definition. When more than one target endpoint is defined, the Route Rule evaluates the condition set in Proxy EndPoint definition. If the condition evaluates to true, it forwards the request to the named target endpoint.

The following procedure describes how to achieve dynamic routing in API Management. Let's say you want to route an API proxy request to two different target endpoints, a default target endpoint and a new target endpoint based on a condition set in the Proxy EndPoint definition.

For our implementation, let's consider the following two target endpoints:

- Target_Endpoint_1 (default)

`https://services.odata.org/V2/Northwind/Northwind.svc/`

- Target_Endpoint_2

`https://services.odata.org/V2/OData/OData.svc/`

Procedure

Steps for Creating a simple API Proxy

1. Navigate to API Portal.
2. Choose **Develop** tab on the navigation menu.
3. Under **APIs**, choose **Create** to create a simple API proxy.
4. In the **Create API** wizard, choose the **URL** radio button.

Note

You can also choose to create an API by choosing the **API Provider** option. For more information, see [Create an API from the API Management, API portal](#)

5. In the **URL** field, enter the target URL of your backend service. In this case, URL pointing to Target_Endpoint_1 (default).
6. Enter a name and a title for your API proxy. In this case, let's enter the API proxy name as Dynamic_Routing.
7. Scroll down the wizard and enter the base path of your API proxy in the **API Base Path** field. In this case, let's enter the base path as /multitargets.
8. Choose **Create**.
9. Save and deploy your API proxy.

i Note

When the API proxy is created, the default route rule is set. It points to the default target endpoint and no rule is attached to it.

Steps for defining new target endpoint

10. Navigate to the **Develop** tab. From the **APIs** list, choose the API proxy that you deployed.
11. Download the newly deployed API proxy using the **Export** option. For more information, see [Export an API](#)
 A zip file called Dynamic_Routing.zip is downloaded.
12. Unzip the Dynamic_Routing.zip file.
 A parent folder called APIProxy is created. The APIProxy folder consists of various subfolders and files. For more information, see [API Proxy Structure](#).
13. Open the APITargetEndPoint subfolder.
 You see a file named default.xml. The default.xml file contains the URL of the default target endpoint.
14. Create a new XML file named Target_EndPoint_2.xml with the following content. In the Target_EndPoint_2.xml file, you need to enter a name and the URL of the new target endpoint to which the request must be routed dynamically.

i Note

The <isDefault> attribute must be set to false for all the new target endpoints that you define. Whereas, for the default target endpoint the <isDefault> attribute would by default be set to true.

≡ Sample Code

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TargetEndPoint>
  <name>Target_EndPoint_2</name>
  <url>https://services.odata.org/V2/OData/OData.svc</url>
  <provider_id>NONE</provider_id>
```

```

    <isDefault>false</isDefault>
  </properties>
</faultRules>
<preFlow>
  <name>PreFlow</name>
</preFlow>
<postFlow>
  <name>PostFlow</name>
</postFlow>
<conditionalFlows/>
</TargetEndPoint>

```

You see two files named `default.xml` and `Target_EndPoint_2.xml` in the `APITargetEndPoint` subfolder.

Steps for defining conditions using Route Rule

15. Open the `APIProxyEndPoint` subfolder.

You see a file named `default.xml` file.

16. Open the `default.xml` file.

The `default.xml` file contains information about your API proxy such as base path, flows, policies and, the default route rule.

Sample Code

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProxyEndPoint default="true">
  <name>default</name>
  <base_path>/multitargets</base_path>
  <properties/>
  <routeRules>
    <routeRule>
      <name>default</name>
      <targetEndPointName>default</targetEndPointName>
      <sequence>1</sequence>
      <faultRules/>
    </routeRule>
  </routeRules>
  <faultRules/>
  <preFlow>
    <name>PreFlow</name>
  </preFlow>
  <postFlow>
    <name>PostFlow</name>
  </postFlow>
  <conditionalFlows/>

```

```
</ProxyEndPoint>
```

17. Update the value of the <sequence> attribute to 2.

Sample Code

```
<routeRule>
  <name>default</name>
  <targetEndPointName>default</targetEndPointName>
  <sequence>2</sequence>
  <faultRules/>
</routeRule>
```

The resulting default.xml file must reflect the following content.

Sample Code

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProxyEndPoint default="true">
  <name>default</name>
  <base_path>/multitargets</base_path>
  <properties/>
  <routeRules>
    <routeRule>
      <name>default</name>
      <targetEndPointName>default</targetEndPointName>
      <sequence>2</sequence>
      <faultRules/>
    </routeRule>
  </routeRules>
  <faultRules/>
  <preFlow>
    <name>PreFlow</name>
  </preFlow>
  <postFlow>
    <name>PostFlow</name>
  </postFlow>
```

```

        <conditionalFlows/>
    </ProxyEndPoint>

```

18. Define a new route rule named Target_EndPoint_2 by adding the following content to the default.xml file.

Sample Code

```

<routeRule>
    <name>Target_EndPoint_2</name>
    <targetEndPointName>Target_EndPoint_2</targetEndPointName>
    <sequence>1</sequence>
    <faultRules/>
</routeRule>

```

The resulting default.xml file must reflect the following content.

Sample Code

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProxyEndPoint default="true">
    <name>default</name>
    <base_path>/multitargets</base_path>
    <properties/>
    <routeRules>
        <routeRule>
            <name>Target_EndPoint_2</name>
            <targetEndPointName>Target_EndPoint_2</targetEndPointName>
            <sequence>1</sequence>
            <faultRules/>
        </routeRule>
        <routeRule>
            <name>default</name>
            <targetEndPointName>default</targetEndPointName>
            <sequence>2</sequence>
            <faultRules/>
        </routeRule>
    </routeRules>
    <faultRules/>
    <preFlow>
        <name>PreFlow</name>
    </preFlow>
    <postFlow>
        <name>PostFlow</name>
    </postFlow>
    <conditionalFlows/>
</ProxyEndPoint>

```

19. Define a condition based on which you want to route the request dynamically. In this case, let's add a `proxy.pathsuffix MatchesPath` condition under the `Target_EndPoint_2` Route Rule and set it to the path called `/Categories`.

Sample Code

```
<routeRule>
  <name>Target_EndPoint_2</name>
  <conditions>proxy.pathsuffix MatchesPath "/Categories"</conditions>
  <targetEndPointName>Target_EndPoint_2</targetEndPointName>
  <sequence>1</sequence>
  <faultRules/>
</routeRule>
```

The resulting default.xml file must reflect the following content.

Sample Code

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProxyEndPoint default="true">
  <name>default</name>
  <base_path>/multitargets</base_path>
  <properties/>
  <routeRules>
    <routeRule>
      <name>Target_EndPoint_2</name>
      <conditions>proxy.pathsuffix MatchesPath "/Categories"</conditions>
      <targetEndPointName>Target_EndPoint_2</targetEndPointName>
      <sequence>1</sequence>
      <faultRules/>
    </routeRule>
    <routeRule>
      <name>default</name>
      <targetEndPointName>default</targetEndPointName>
      <sequence>2</sequence>
      <faultRules/>
    </routeRule>
  </routeRules>
  <faultRules/>
  <preFlow>
    <name>PreFlow</name>
```

```

    </preFlow>
    <postFlow>
        <name>PostFlow</name>
    </postFlow>
    <conditionalFlows/>
</ProxyEndPoint>

```

i Note

If you have defined more than one Route Rule in the Proxy EndPoint as shown in the above codeblock, their sequence in the XML configuration is important. The first Route Rule to match gets executed. (Route Rules with no condition always match). In the above codeblock, if the `default` Route Rule appeared first, it would be executed even if the condition of the `Target_EndPoint_2` Route Rule would have matched. Hence, it is always recommended to list your conditional Route Rules before an unconditional Route Rule.

20. Open the `<API_Proxy_Name>.xml` file. In this case, `Dynamic_Routing.xml` file.

21. Add the new target endpoint name that you defined.

≡, Sample Code

```

<targetEndpoints>
    <targetEndPoint>Target_EndPoint_2</targetEndPoint>
    <targetEndPoint>default</targetEndPoint>
</targetEndpoints>

```

Steps for viewing dynamic routing

22. Compress the APIProxy parent folder.

23. Navigate to API Portal and import the compressed APIProxy.zip file. For more information, see [Import an API](#).

24. Choose the imported API proxy.

25. Under **Proxy EndPoint** tab, in the **Route Rules** section, you must see two Route Rules that you defined earlier.

i Note

You can also add route conditions directly in API Portal User Interface instead of adding it manually in the API ProxyEndPoint definition file as shown in step 19.

26. Click on the API Proxy URL.

The request must be routed to the default target endpoint.

27. Append `/Categories` to the API Proxy URL in your browser.

The request must be routed dynamically to the new target endpoint.

To validate the response, copy and paste the actual URL of the backend service with path suffix `/Categories`.

The response obtained must match the response obtained in step 27.

i Note

All the policies that you attach in the TargetEndPoint via the API Portal User Interface are applied only to the default target endpoint. In case, if you need to enforce policies on the non-default target endpoint, then you must import the API Proxy bundle and manually add the policies in the required TargetEndPoint definition file.

Related Information

<https://blogs.sap.com/2019/06/03/building-a-loopback-api-using-sap-cloud-platform-api-management/> 