Enabling CORS for APIs

Cross-Origin Resource Sharing (CORS) is a mechanism that allows accessing restricted resources (i.e., fonts, images, scripts, videos and iframes) from domains outside the domain from which the requesting resource originated. By default, web browsers apply the same-origin policy to avoid interactions between different origins. CORS defines a way in which a browser and a server can interact to determine whether or not it is safe to allow the cross-origin requests.

In API Manager, you can enable Cross-Origin Resource Sharing per API or as a global configuration that is applied across all APIs.

- Enabling CORS Globally
- Enabling CORS Per API

Enabling CORS Globally

Follow the steps below to enable CORS response headers globally. Once this configuration is enabled, it will be applied across all the APIs served by the API Gateway.

1. Open the `<API-M_HOME>/repository/conf/api-manager.xml` file.
2. Locate the following configuration and set the `<Enabled>` attribute to `true` with the required CORS headers in the response. Once this configuration is applied in the API Gateway, it will affect all the API calls served by the Gateway.
<!-- Configuration to enable/disable sending CORS headers in the Gateway response and define the Access-Control-Allow-Origin header value. -->
<CORSConfiguration>
  <!-- Configuration to enable/disable sending CORS headers from the Gateway -->
  <Enabled>true</Enabled>
  <!-- The value of the Access-Control-Allow-Origin header. Default values are API Store addresses, which is needed for swagger to function. -->
  <Access-Control-Allow-Origin>*</Access-Control-Allow-Origin>
  <!-- Configure Access-Control-Allow-Methods -->
  <Access-Control-Allow-Methods>GET,PUT,POST,DELETE,PATCH,OPTIONS</Access-Control-Allow-Methods>
  <!-- Configure Access-Control-Allow-Headers -->
  <Access-Control-Allow-Headers>authorization,Access-Control-Allow-Origin,Content-Type,SOAPAction</Access-Control-Allow-Headers>
  <!-- Specifying this header to true means that the server allows cookies (or other user credentials) to be included on cross-origin requests. It is false by default and if you set it to true then make sure that the Access-Control-Allow-Origin header does not contain the wildcard (*) -->
  <Access-Control-Allow-Credentials>false</Access-Control-Allow-Credentials>
</CORSConfiguration>

CORS configuration is enabled by default. Access control can be done by changing the parameters mentioned above in the api-manager.xml file.

Enabling CORS Per API

It is required to enable CORS globally before you enable CORS Per API. Therefore if you haven't done it yet, follow the steps in Enabling CORS Globally before starting the below steps.

1. Sign in to API Publisher and choose to design a new API.
2. Click **Start Creating**.
3. Give the information in the table below and click **Add** to add the resource.

<table>
<thead>
<tr>
<th>Field</th>
<th>Sample value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>WeatherAPI</td>
</tr>
<tr>
<td>Context</td>
<td>/weather</td>
</tr>
<tr>
<td>Version</td>
<td>v1.0.0</td>
</tr>
<tr>
<td>Resources</td>
<td>URL Pattern: <code>current/{country}/{zipcode}</code></td>
</tr>
<tr>
<td></td>
<td>Request types: GET method to return the current weather conditions of a zip code that belongs to a particular country</td>
</tr>
</tbody>
</table>
4. Once done, click **Next: Implement** >

5. In the **Implementation** tab, provide the following endpoint details.

<table>
<thead>
<tr>
<th>Field</th>
<th>Sample value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint type</td>
<td>HTTP/REST endpoint</td>
</tr>
<tr>
<td>Production endpoint</td>
<td>You can find the Yahoo weather API's endpoint from <a href="https://developer.yahoo.com/weather/">https://developer.yahoo.com/weather/</a>. Copy the part before the '?' sign to get this URL: <a href="https://query.yahooapis.com/v1/public/yql">https://query.yahooapis.com/v1/public/yql</a></td>
</tr>
</tbody>
</table>

6. Select the **Enable API based CORS Configuration** check box to enable CORS for the API.
7. Once you enable CORS, you will be able to see the CORS response header configuration section. Listed below are the CORS specific response headers supported by the API Gateway and how to configure them.

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
<th>Sample values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access-Control-Allow-Origin</td>
<td>Determines whether a resource can be shared with the resource of a given origin. The API Gateway validates the origin request header value against the list of origins defined under the Access Control Allow Origins configuration (this can be All Allow Origins or a specific value like localhost). If the host is in the allowed origin list, it will be set as the Access-Control-Allow-Origin response header in the response.</td>
<td>All Allow Origins(*), localhost</td>
</tr>
<tr>
<td>Access-Control-Allow-Headers</td>
<td>Determines, as part of the response to a preflight request (a request that checks to see if the CORS protocol is understood), which header field names can be used during the actual request. The gateway will set the header values defined under Access Control Allow Headers configurations.</td>
<td>authorization, Access-Control-Allow-Origin, Content-type, SOAPAction</td>
</tr>
<tr>
<td>Access-Control-Allow-Methods</td>
<td>This header specifies the method(s) allowed when accessing the resource in response to a preflight request. Required methods can be defined under the Access Control Allow Method configuration.</td>
<td>GET, PUT, POST, DELETE, PATCH, OPTIONS</td>
</tr>
<tr>
<td>Access-Control-Allow-Credentials</td>
<td>Determines whether or not the response to the request can be exposed to the page. It can be exposed when the header value is true. The header value can be set to true/false by enabling/disabling the Access Control Allow Credentials configuration.</td>
<td>true, false</td>
</tr>
</tbody>
</table>

8. Once the CORS configurations are done, click **Next: Manage >**.

   CORS configuration

   - **Enable API based CORS Configuration**: 
   - **Access Control Allow Origins**: All Allow Origins
   - **Access Control Allow Headers**: `Authorization`, `Access-Control-Allow-Origin`, `Content-Type`, `NGSAction`
   - **Access Control Allow Methods**: `GET`, `PUT`, `POST`, `DELETE`, `PATCH`, `OPTIONS`
   - **Access Control Allow Credentials**: [ ]

9. Select the **Unlimited** subscription tier and click **Save and Publish** to create and publish the API to the API Store.
You have successfully enabled CORS for a specific API.

1. 