Create a Prototyped API with an Inline Script

In this tutorial, you create a prototyped API with an inline script, deploy it as a prototype, and invoke it using the API Console integrated in the API Store. You create APIs with inline scripts typically for testing purposes. An API prototype is created for the purpose of early promotion and testing. You can deploy a new API or a new version of an existing API as a prototype. It gives subscribers an early implementation of the API that they can try out without a subscription or monetization, and provide feedback to improve. After a period of time, publishers can make changes that the users request and publish the API.


2. Select the option to design a new REST API and click Start Creating.

3. Give the information in the table below. To add resources, click the Add button. Since the URL Pattern used here is a variable, it is denoted within curly braces.

<table>
<thead>
<tr>
<th>Field</th>
<th>Sample value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Location_API</td>
</tr>
<tr>
<td>Context</td>
<td>/location</td>
</tr>
<tr>
<td>Version</td>
<td>1.0.0</td>
</tr>
<tr>
<td>Resources</td>
<td>URL pattern</td>
</tr>
<tr>
<td>Request types</td>
<td>GET</td>
</tr>
</tbody>
</table>

4. After the resource is added, expand its GET method and note that a parameter by the name town is added under the resource. You use it to pass the payload to the backend. Once done, click Next: Implement >.
5. In the Prototyped API section under the Implement tab, select the implementation method as Inline.

To specify multiple parameters in the API resource, separate the parameters with a forward slash.

```
{param1}/{param2}
```

6. Expand the GET method and give the following as the script. It reads the payload that the user sends with the API request and returns it as a JSON value. The value `mc` is the message context.

```
mc.setProperty('CONTENT_TYPE', 'application/json');
var town = mc.getProperty('uri.var.town');
mc.setPayloadJSON('{ "Town" : "+town+"'});
```
7. Click **Deploy as a Prototype**.
8. Go to the API Store and note that the newly deployed API is listed there.

**Tip**: You can invoke prototyped APIs without signing in to the API Store or subscribing to the API. The purpose of a prototype is advertising and giving an early implementation for users to test.
9. Click the API to open it and go to its API Console tab.

10. Expand the GET method, give any value for the town (say London) and invoke the API.

    GET /[town]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>town</td>
<td>London</td>
<td></td>
</tr>
</tbody>
</table>

Response Messages

<table>
<thead>
<tr>
<th>HTTP Status Code</th>
<th>Reason</th>
<th>Response Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Try it out!

11. Note the payload you gave as a JSON output in the response.

You have created an API with inline script, deployed it as a prototype and invoked it through the integrated API Console.

An API can also be prototyped by moving the API to the prototyped state in the API lifecycle. For more information, see the Deploy and Test as a Prototype tutorial.

Related Tutorials

- Create and Publish an API
- Create a WebSocket API
- Create and Publish an API from Swagger definition