Creating an HTTP API resource using Amazon API Gateway

The Amazon API Gateway is a tool to help developers create links between back-end systems to their respective applications.

This tutorial covers how to create an HTTP API resource for an existing Lambda function for a web application.

Prerequisites

- An Amazon AWS account
- A Lambda Function and its respective Function ARN.
- The expected HTTP method type for the Lambda Function.

Creating the API resource

This process will walk through each step of the API creation wizard.

Navigating to Amazon API Gateway

1. Open a web browser and navigate to the Amazon API Gateway main page.
2. Click the **Create API** button.
3. Under **Choose an API type**, select **Build** for the **HTTP API** box.

Create an API

1. In the **Create and configure integrations** box, click the **Add integration** button.
2. In the new dropdown option, select **Lambda**.
3. Set the **AWS Region** to the region where the Lambda function resides.
4. In the **Lambda function** textbox, enter the function ARN or search for the function name and click that option.

5. In the **Version** dropdown, select the specified version.

   > To learn more about version differences, review Working with AWS Lambda proxy integrations for HTTP APIs.

6. Add any additional integrations with the **Add integration** button.
7. Provide a name for this API in the **API name** textbox.
8. Click **Next**.

Configure routes

1. Select the required method from the **Method** dropdown box.
2. Rename the **Resource path** as desired.

   > **Note:** This will be part of the URL for this API.

3. Add any additional routes as necessary.

Configure stages

1. By default, HTTP APIs have the **$default** stage name.
2. Click **Add stage** to add a stage for this API.
3. Leave the **Auto-deploy** in its default position.
4. Click **Next**.

Review and create

1. Review the details entered for this API for any errors.
2. When ready, click **Create**.

Configuring access to the new resource

After creating the resource, you'll need to confirm the URL path to this resource and configure the **Cross-origin resource sharing (CORS)** settings to access this new resource.

Getting the resource URL

1. On the left-hand sidebar, click **APIs**.
2. In the new APIs box, click the API name you want.
3. In the **Stages for (your API name)**, copy the **Invoke URL**.
4. In the left-hand sidebar, navigate to **Develop > Routes**.
5. Under the **Routes for (your API name)**, append the route name to the **Invoke URL**.

   > **Example:**
   > `https://123456.execute-api.us-west-1.amazonaws.com/helloWorld`

Configuring CORS

1. On the left-hand sidebar, navigate to **Develop > CORS**.
2. In the new menu, under **Configure CORS**, click the **Configure** button.
3. In the **Access-Control-Allow-Origin**, enter the URL of the website accessing this resource.

   > **Note:** Entering **an asterisk (*)** in this box allows any website to access this resource.

4. Click **Save**.

   > To learn more about CORS, review Configuring CORS for an HTTP API.

At this point, your web application can send an HTTP request to this new API resource and execute your Lambda function.