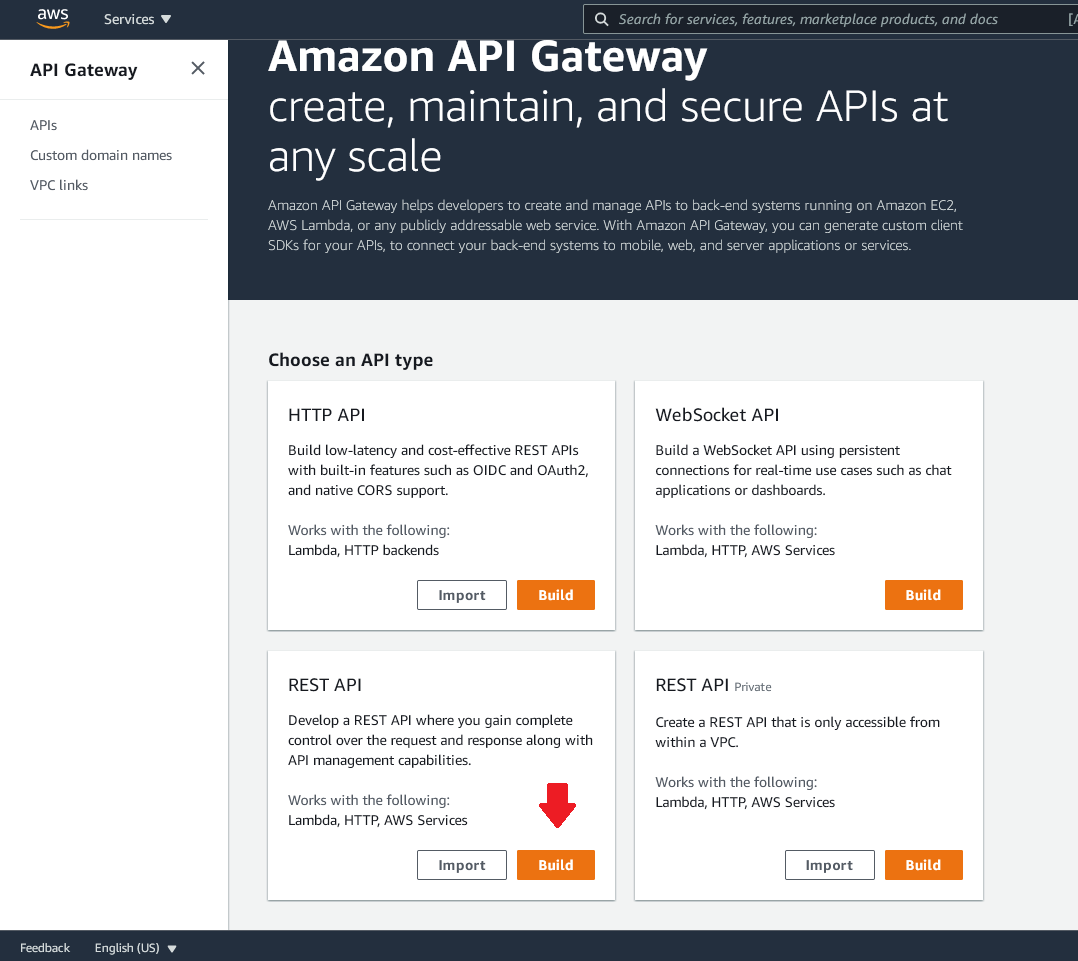
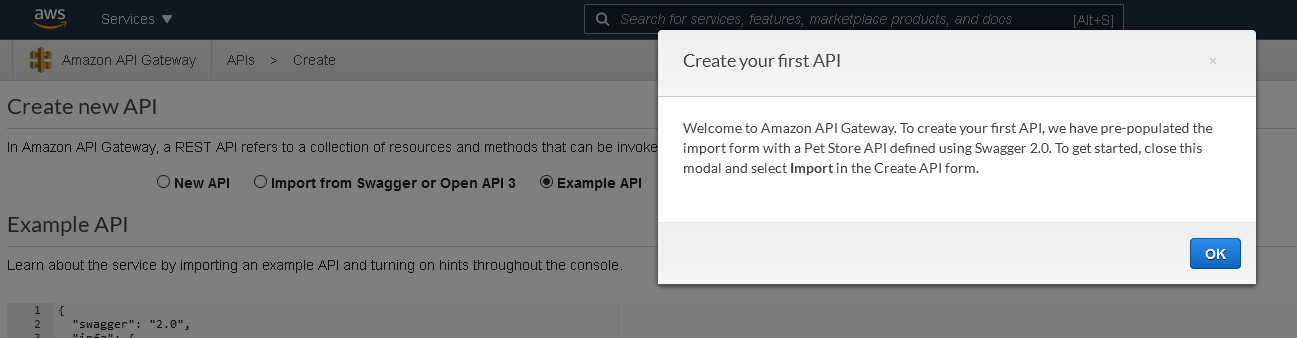
# CS 470 Module Four Assignment Two Guide

## Part One: Creating an API Using API Gateway

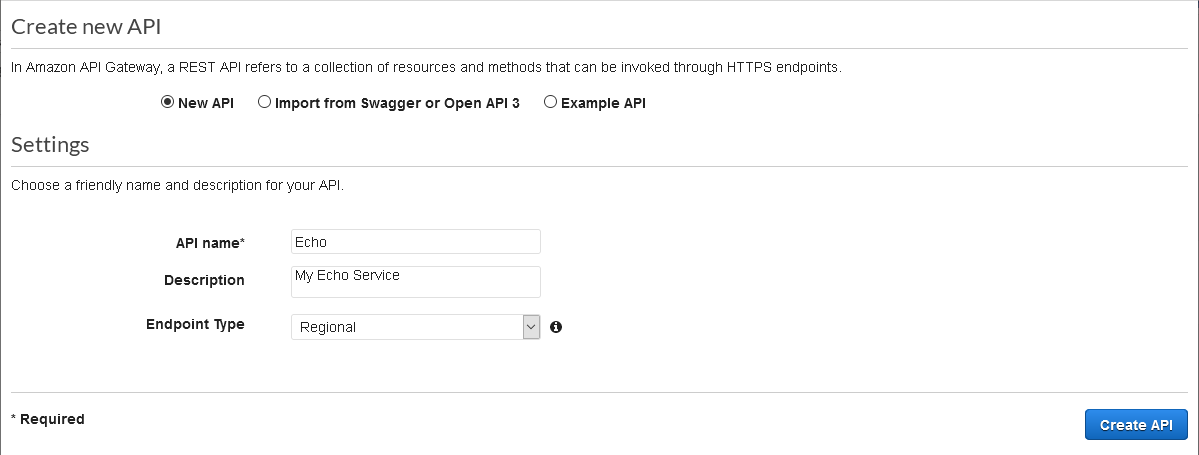
1. Navigate to the **API Gateway** in the AWS console. It is located under the **Networking & Content Delivery** section. You can also search for the gateway.
2. Select **REST API** by clicking the **Build** button in the **REST API** box.



1. Dismiss the dialog box shown below and select **New API** in the **Create new API** section.



1. Set the API name to “Echo” and give it the description “My Echo Service”.
2. Leave the Endpoint Type as **Regional**.
3. Click the blue **Create API** button in the lower-right corner.

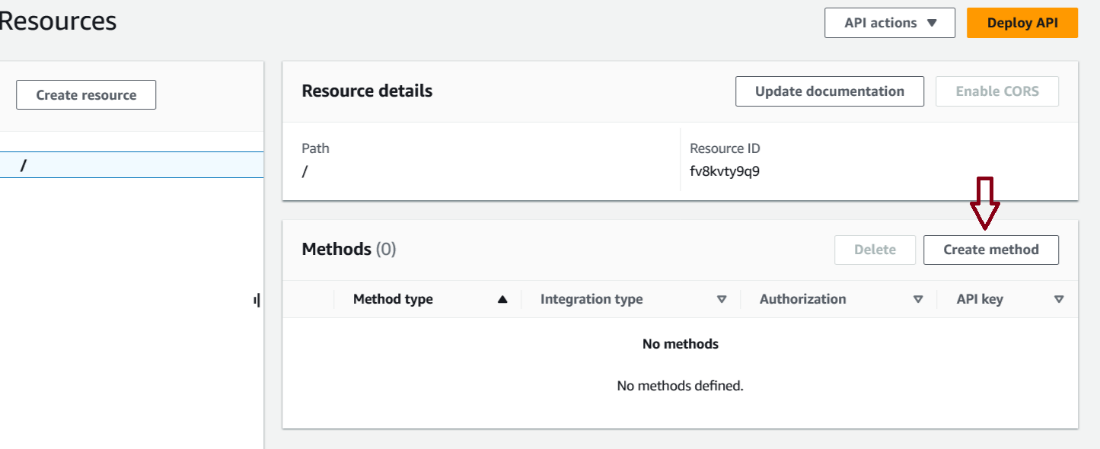


1. You will now see a mostly blank screen titled “/Methods”.

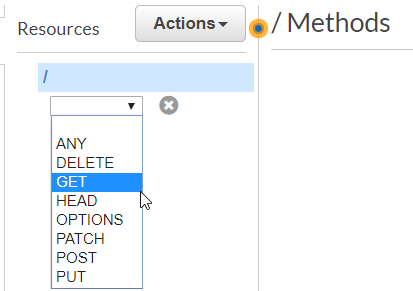
Believe it or not, you have now created an API. The API will not do anything yet, but that is for the next part.

## Part Two: Adding a Method to Your API

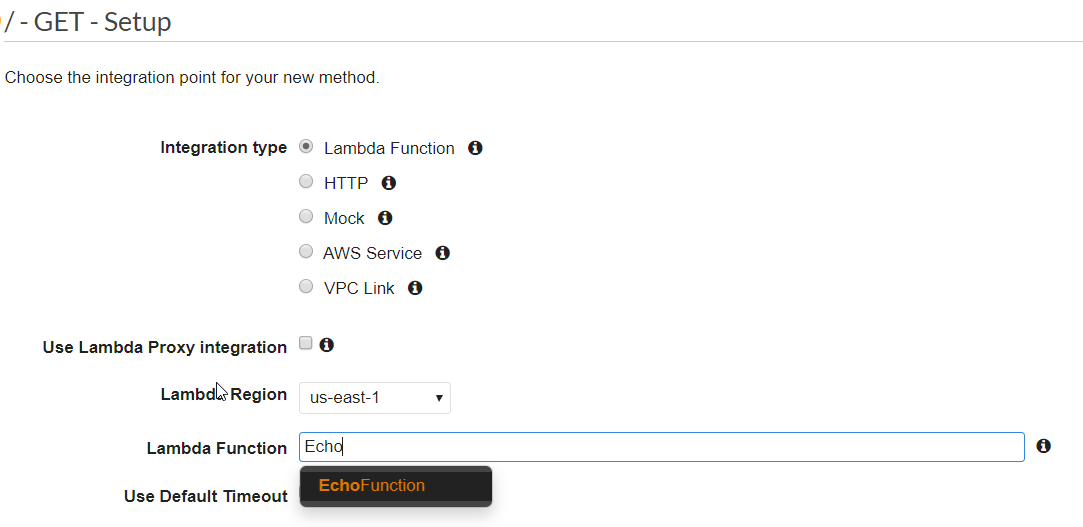
1. Click the **Create method** button toward the right of your screen.



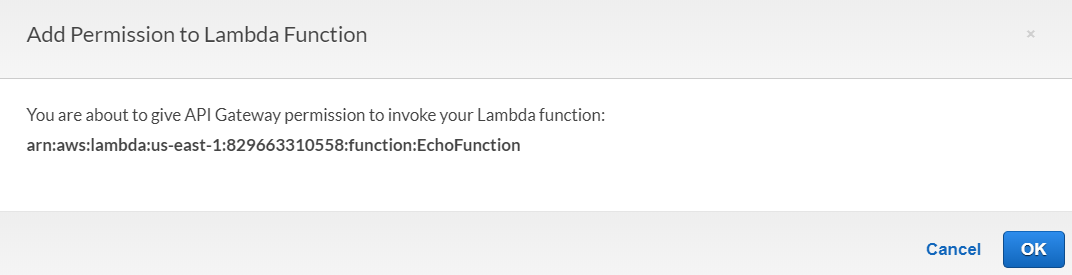
1. A new blank drop-down menu will be created under the forward slash (/). This menu is where you will select the REST methods to process for your API. In this case, you will select GET to handle HTTP GET requests. The other options are direct mappings to HTTP methods. However, ANY is used to match all of the requests. ANY will match any request not explicitly handled by another method definition.



1. Click the check box next to the drop-down menu that just appeared.
2. Now, you must set up your method. Make certain the integration type is **Lambda Function**.
3. Make certain the **Use Lambda Proxy Integration** check box is checked.
4. Leave the region as **us-east-1**.
5. Type “EchoFunction” in the **Lambda Function** box. AWS will suggest it for you as you type.



1. Make certain **Use Default Timeout** is checked.
2. Click the orange **Create method** button in the lower-right corner.
3. You will be asked if you want to add permission to **Lambda Function**.



1. Click **OK**. This is how you tell the API Gateway that the GET method has permission to execute your Lambda.
2. You will now see the fully populated **Method Execution** page.

Congratulations! You have wired up your API to your Lambda.

## Part Three: Testing Your API to Lambda Method

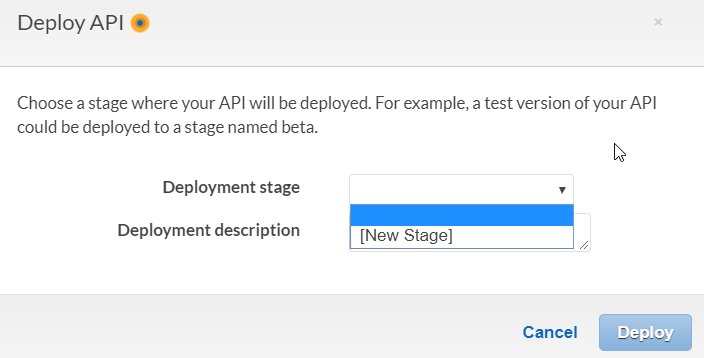
1. Click the **Test** link in the bottom-right section.
2. Click the orange **Test** button.
3. The right side of the page will show you the **Response Body**, **Response Headers**, and **API Gateway Logs**.
4. It worked! But not really. You didn’t pass it a name to reply with, so it says, “Hello unknown”. Let’s fix that.
5. Scroll to the top of the page. In the **Method Execution**, click **Method Request**.
6. Select the **Method Request** box by clicking its name.
7. Click the **Edit** button in the top-right corner.
8. Open the **URL Query** **String Parameters** section.
9. Enter “name” (without quotes) in the input box and click the gray checkmark on the right side.
10. Enter “name” (without quotes) in the input box and click the **Save** button.
11. Select the **Test** link in the upper-right side of the lower section as before.
12. You can now enter a query string for your test. Type “name=Nancy” (without quotes).
13. Click the **Test** button.



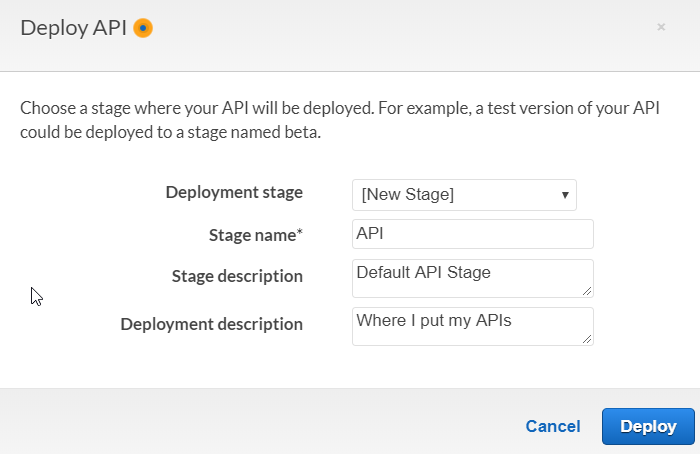
Success! You have tested your API with a parameter calling your Lambda.

## Part Four: Deploying Your API

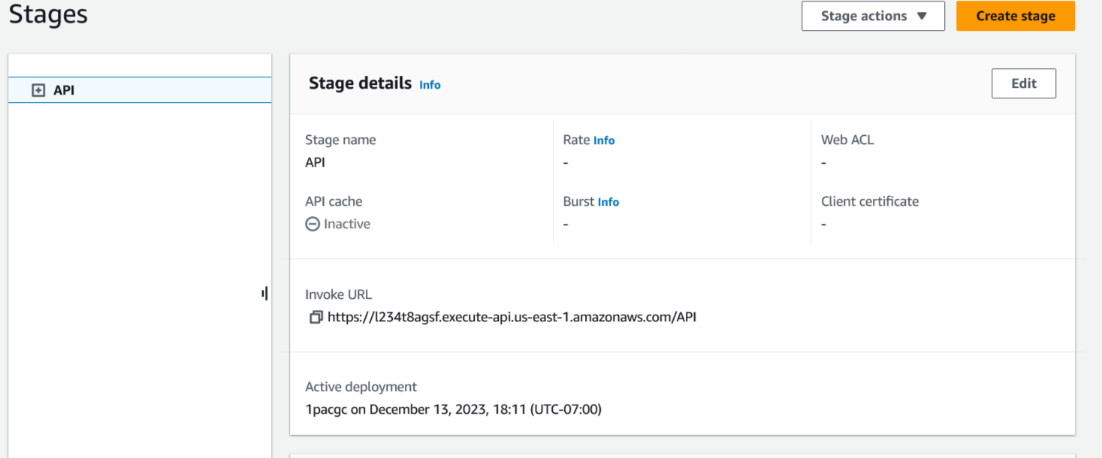
1. You have a shiny new API. But it is not yet callable from outside of AWS. You need to deploy it to allow that. To deploy your API, select the orange **Deploy API** button at the top right of the page.
2. Select **New Stage** in the **Deployment stage** drop-down menu.



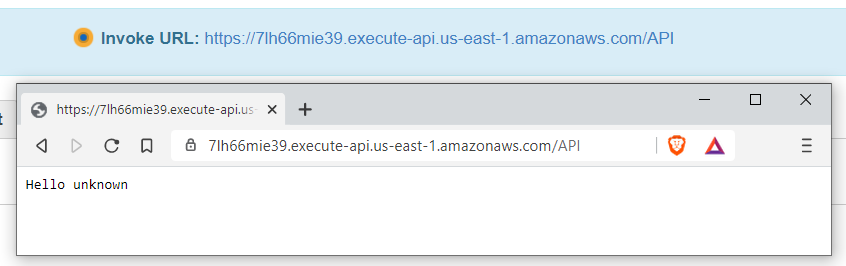
1. Type “API” as your stage name. The two description fields are optional.



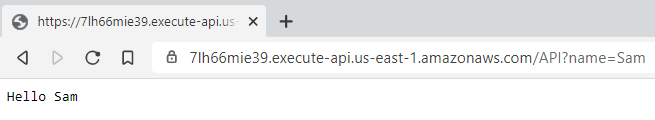
1. Click the **Deploy** button**.**
2. A few seconds later, your API will be deployed, and you will have your shiny new URL.



1. If you navigate to the URL, it will work in your browser. Make certain you are using a browser other than Firefox when you try this.



1. You can put in a query parameter to make the message more personalized.



That’s it. You have created a Lambda, tested it, created an API, connected it to your Lambda, and tested it again. Finally, you deployed your API. Your project will build upon these functions to connect your Angular application to new Lambdas that you will write, which will connect to the database you will learn about in the next module.

Great work!