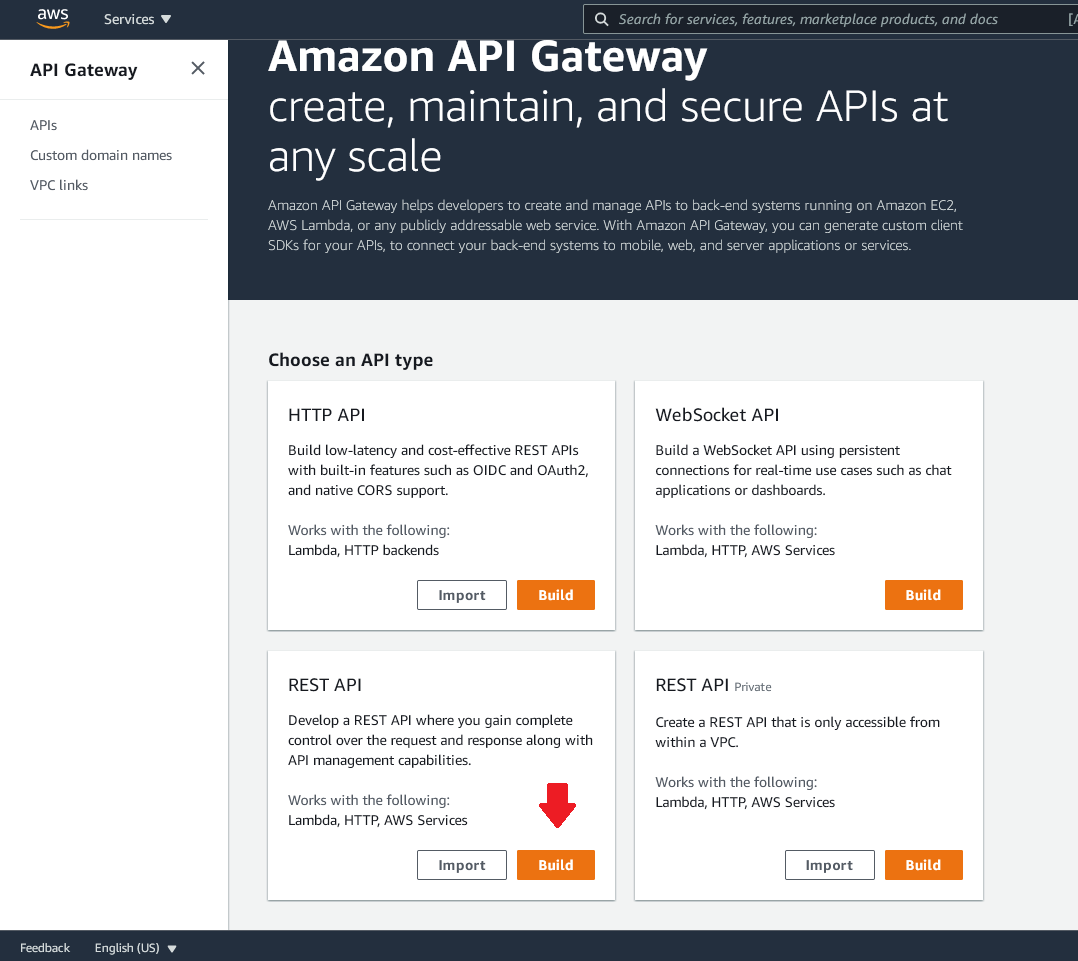
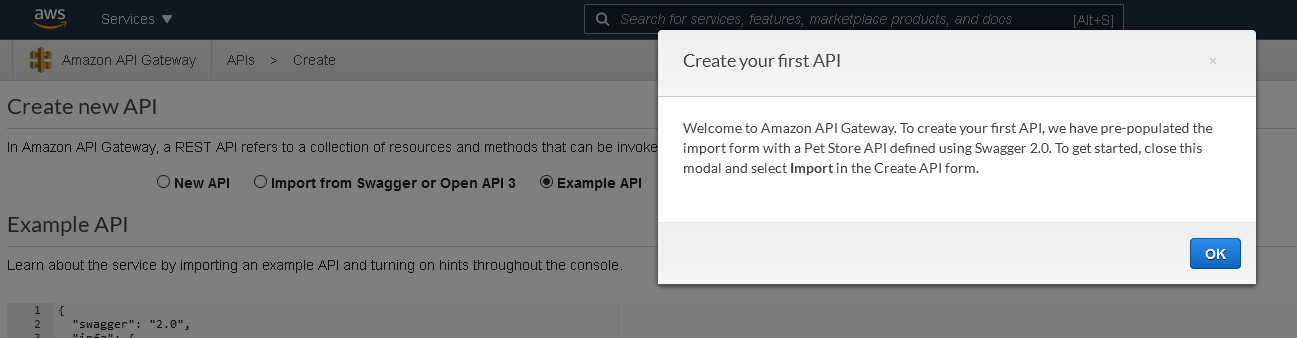
# **CS 470 Module Four Assignment Two Guide**

## Part One – Creating an API Using API Gateway

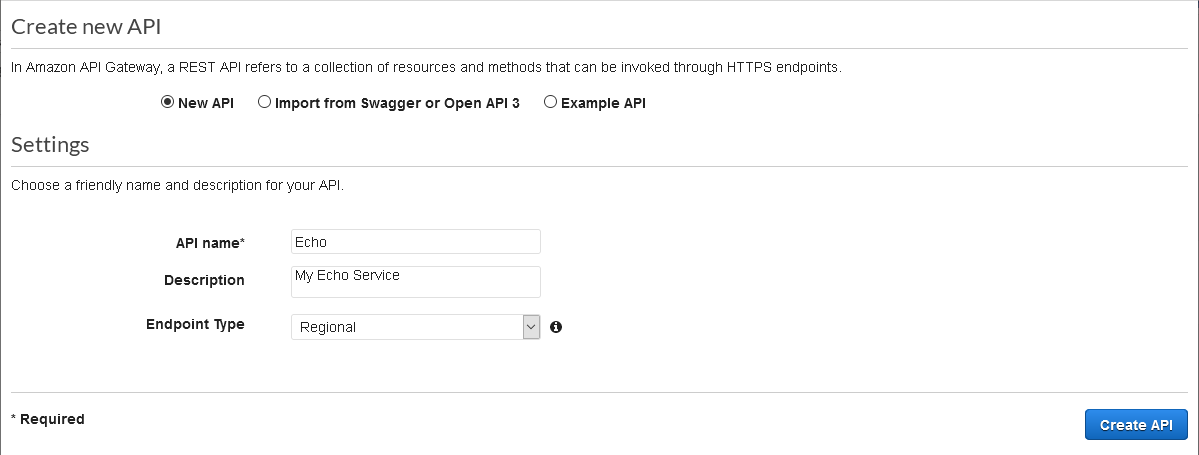
1. Navigate to the **API Gateway** in the AWS Console. It is located under the **Networking & Content Delivery** section. You can also search for it.
2. Select **REST API** by clicking the **Build** button in the **REST API** box.



1. Dismiss the dialog box shown below, and set Create New APIto **New API**.



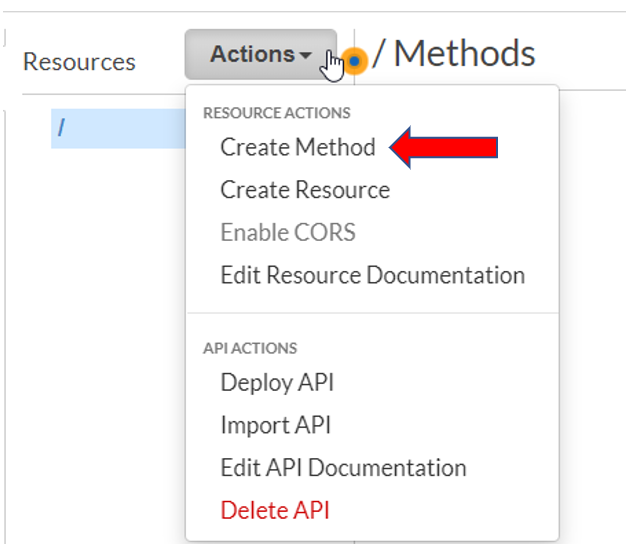
1. Set the API Name to “Echo” and give it a description of “My Echo Service”.
2. Leave the Endpoint Type as **Regional**.
3. Now click the blue **Create API** button in the lower-right corner.



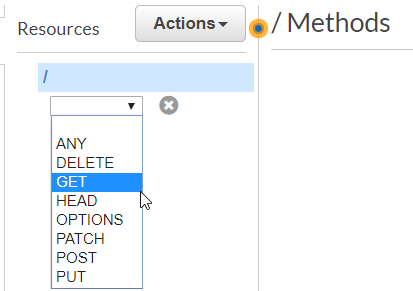
1. You will now see a mostly blank screen with the “/Methods” title.
2. Believe it or not, you have now created an API. It will not do anything yet, but that is for the next part.

## Part Two – Adding a Method to Your API

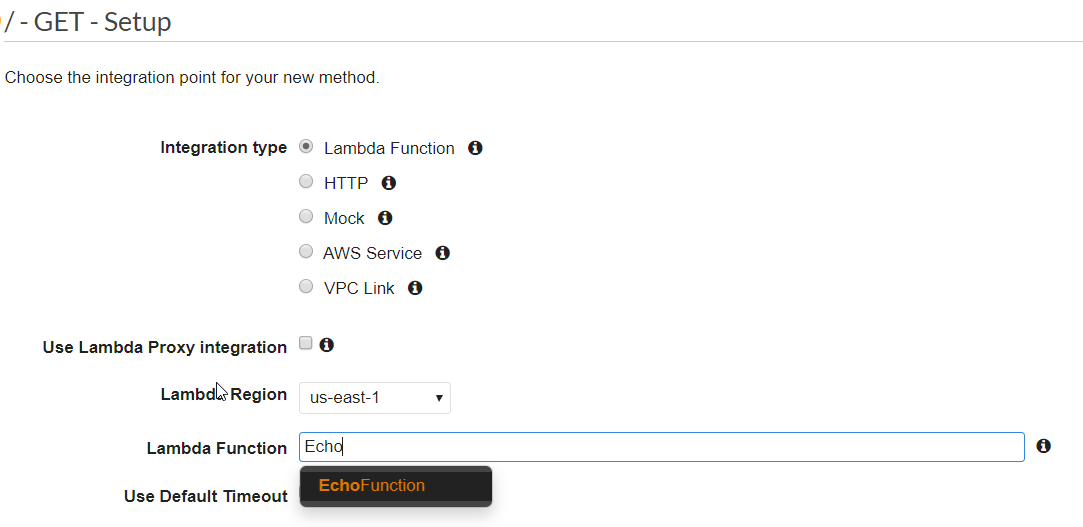
1. Select the **Actions** drop-down menu and click **Create Method**.

****

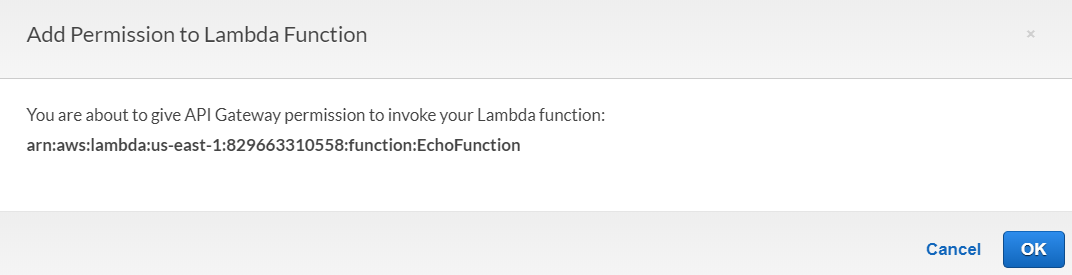
1. A new blank drop-down box will be created under the forward slash (/). This is where you select the REST methods to process for your API. In this case, you will select GET to handle HTTP GET requests. The other options are direct mappings to HTTP methods. However, ANY is used to match all of the requests.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



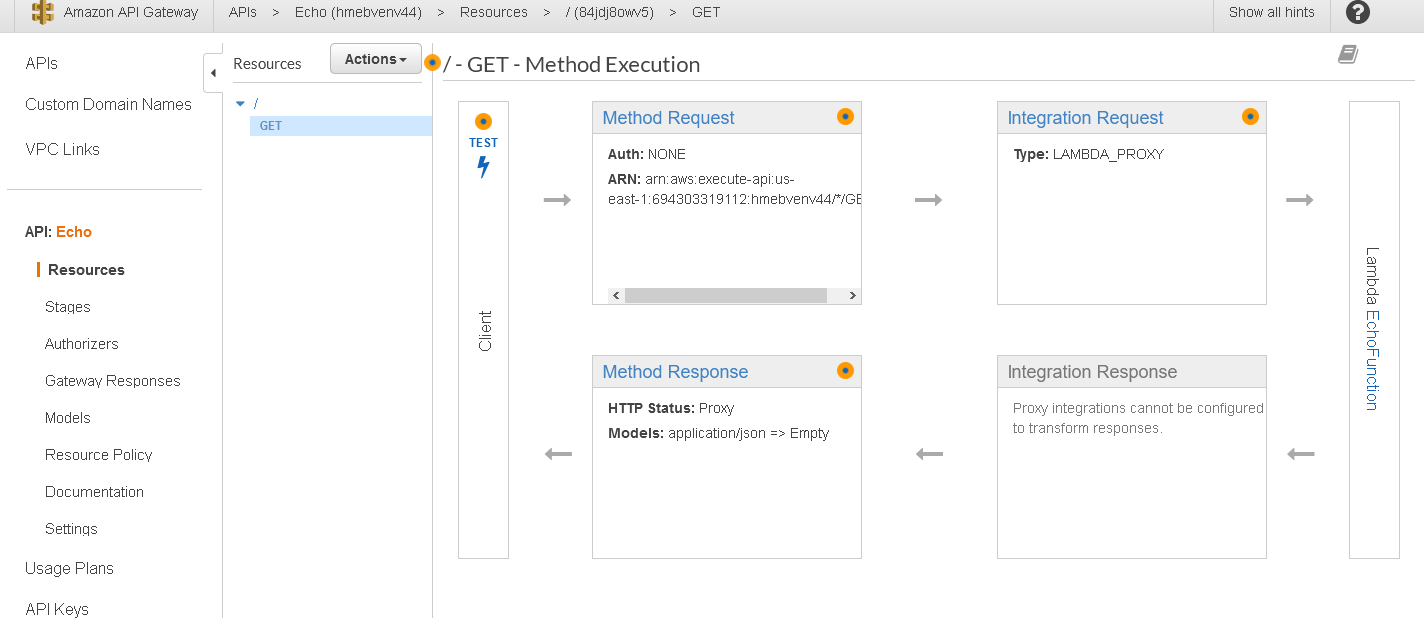
1. Click the check box next to the drop-down menu that just appeared.
2. Now we need to set up our method. Make sure the integration type is **Lambda Function**.
3. Make sure the **Use Lambda Proxy Integration** check box is checked.
4. Leave the region as “us-east-1”.
5. Type “EchoFunction” into the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



1. Make sure **Use Default Timeout** is checked.
2. Click the blue **Save** 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.



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

## Part Three – Testing Your API to Lambda Method

1. Click the **Test** link above the lightning bolt.
2. Click the blue **Test** button.
3. The right side of the page will show you the **Response Body, Response Headers**, and **API Gateway Logs**. It should look something like this:

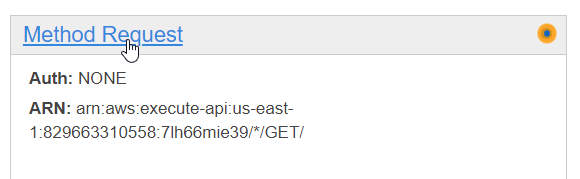
Graphical user interface, text, application, email

Description automatically generated

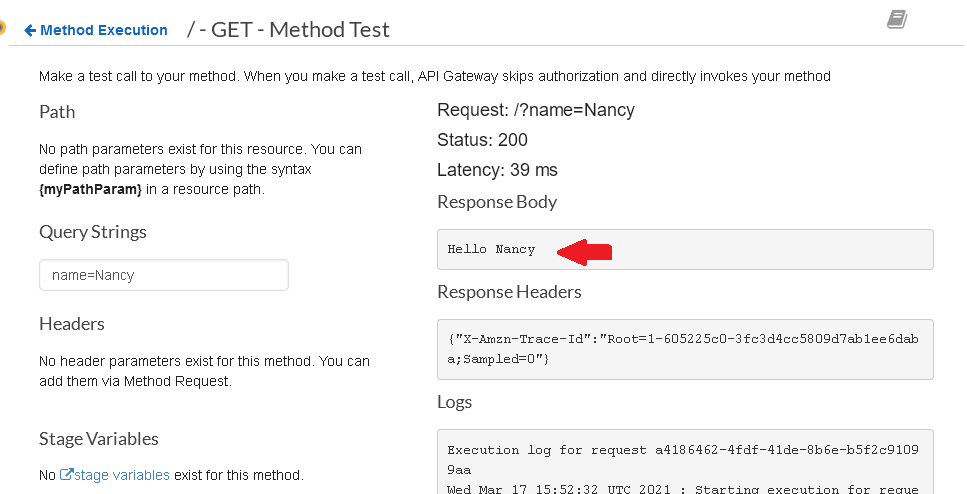
1. It worked! But, not really. We didn’t pass it a name to reply with, so it says “Hello unknown”. Let’s fix that.
2. Select the **Method Execution** link with the arrow pointing left, located in the upper right.

"Method Execution" link with arrow pointing left

1. Select the **Method Request** box by clicking on its name.



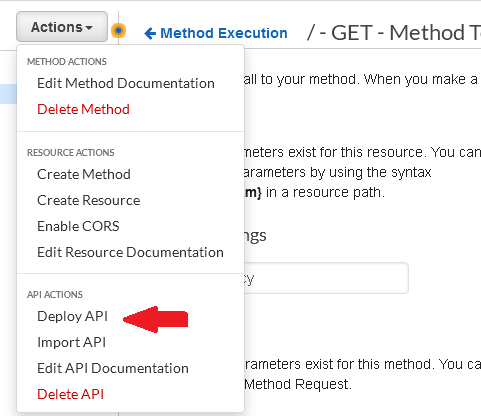
1. Open the **URL Query String Parameters** section.
2. Click the **Plus (+) Add Query** string.
3. Enter “name” (without quotes) in the input box and click the grey check mark on the right side.
4. Select the **Method Execution** link in the upper right.
5. Click **Test**.
6. Now you can enter a query string for your test. Type “name=Nancy” (without quotes).
7. Click the **Test** button.



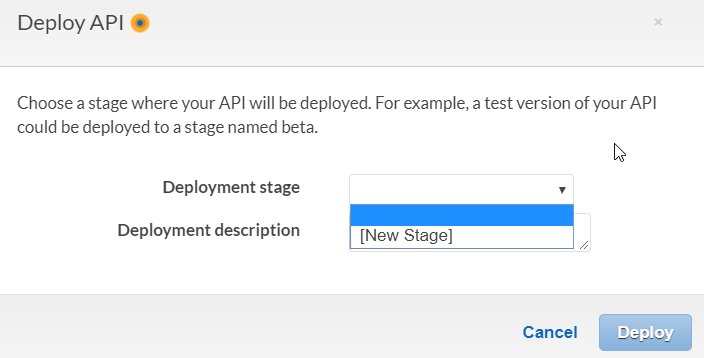
1. Success. You have now tested your API with a parameter calling your Lambda.

## Part Four – Deploying Your API

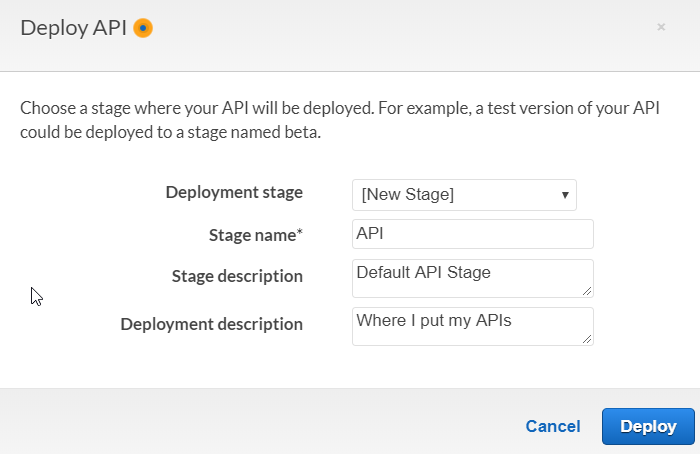
1. You have a shiny new API, but it is not yet callable from outside of AWS. We need to deploy it to allow that. To deploy your API, select **Deploy API** from the **Actions** drop-down menu.



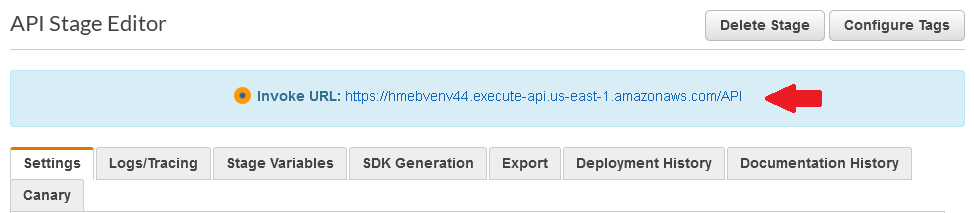
1. Select **New Stage** in the **Deployment stage** drop-down menu.



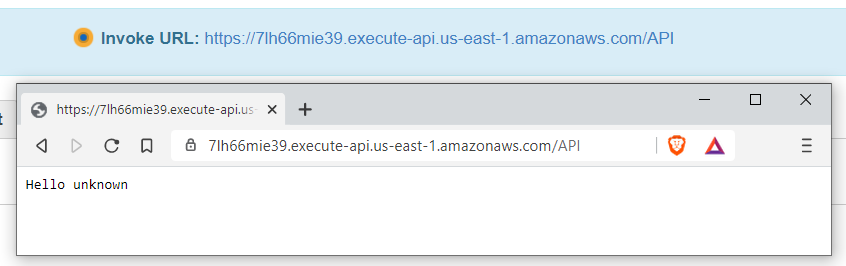
1. Type “API” as your stage name. The two description fields are optional.



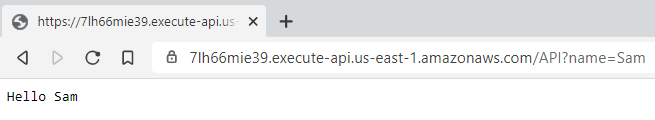
1. Click **Deploy.**
2. A few seconds later, your API will be deployed, and you will have your shiny new URL.



1. If you navigate to the URL, it will work in your browser. (Make sure 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 you will write, which will connect to the database we will learn about in the next module.

Great work!