Lab 3: Connect your Lambda function to an API Gateway

Create a new Lambda function named Student#API matching your student number

Select the API Gateway trigger in your Lambda function,

Create a new API, Set the security to "Open" click on Additional settings then name it to match your student#. Name the deployment stage "lab3", and click Add :

Configure triggers

We'll set up an API Gateway endpoint with a proxy integration type (learn more about the input and output format). Any me
set up more advanced method mappings or subpath routes, visit the Amazon API Gateway console.

API
Pick an existing API, or create a new one.

Create a new API ▼

Security
Configure the security mechanism for your API endpoint.

Open ▼

Warning: Your API endpoint will be publicly available and can be invoked by all users.

▼ **Additional settings**

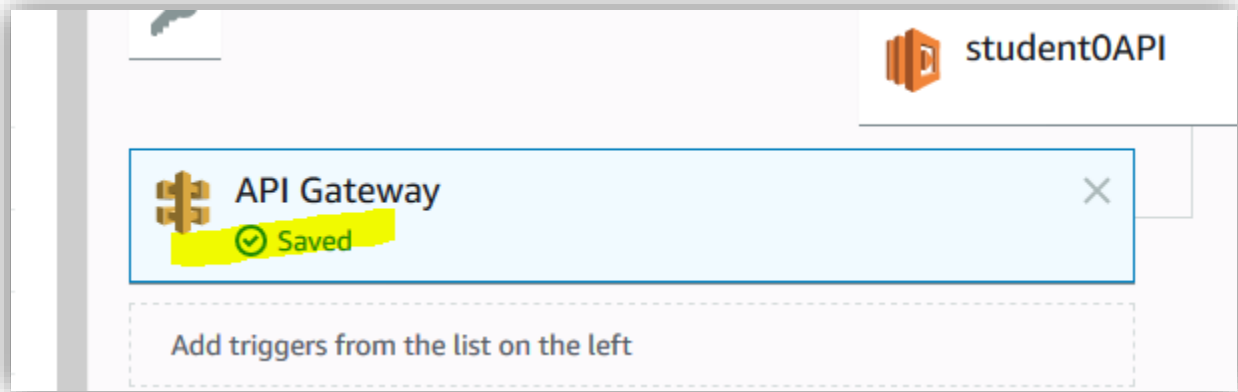**API name**
Enter a name to uniquely identify your API.

student0-API

Deployment stage
The name of your API's deployment stage.

lab3

CORS
To enable cross-origin resource sharing (CORS) for a proxy integration, you must add Access-Control-Allow-Origin: <domain name> to the out

Your API Gateway should be automatically configured once you click save in the upper right hand corner:


Click on the API Gateway trigger that is attached to the function in the designer, it will have a link matching what you named the gateway. Clicking it will take you to the API Gateway management console (Be sure to click on the name of the API, not the link to the API endpoint):
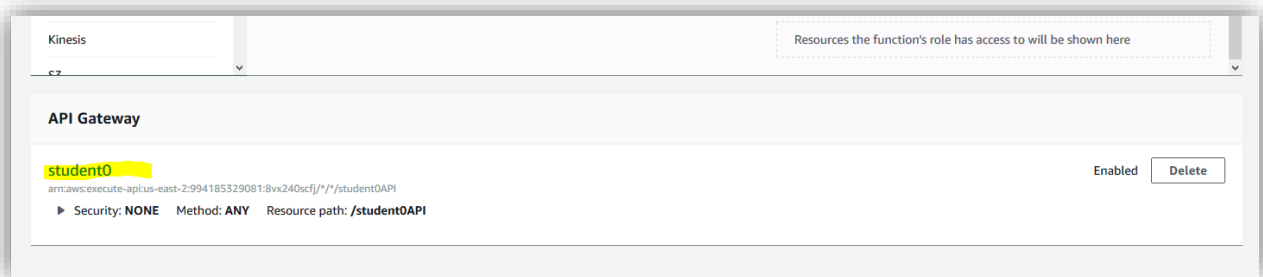
In the API management console, you can see the flow and mappings for the requests, and the functions:
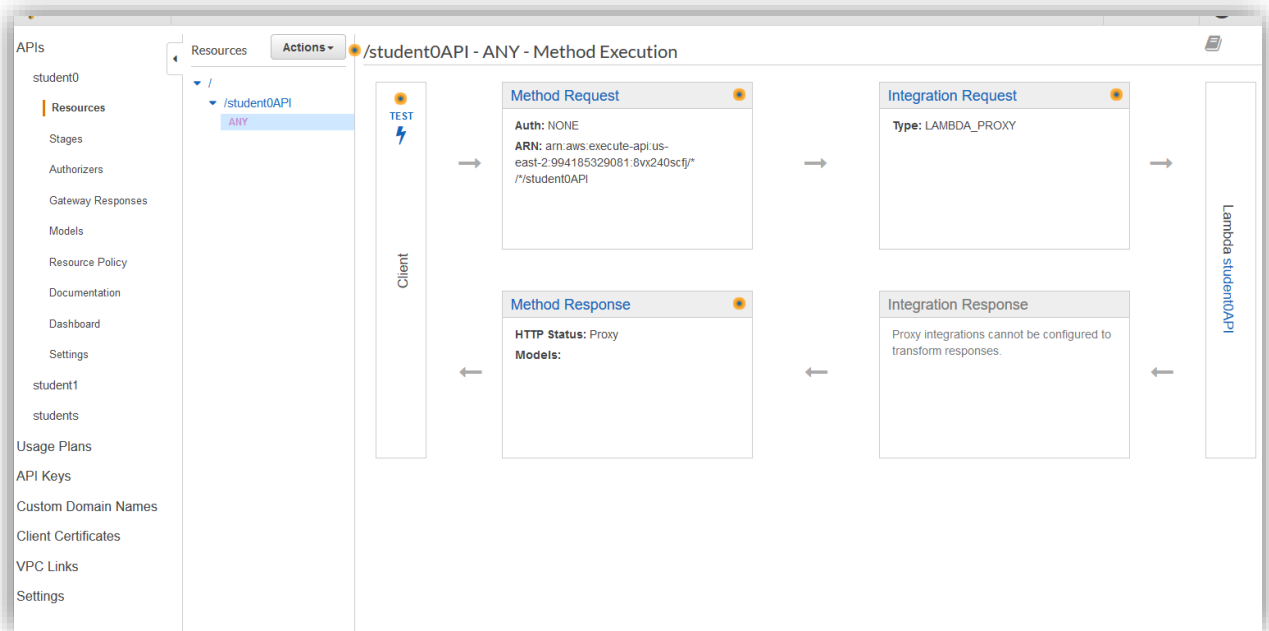
To be able to access your API, you will need to find the invocation URL. Click on Stages, then "lab3". You will see the invocation URL at the top. Click on it:
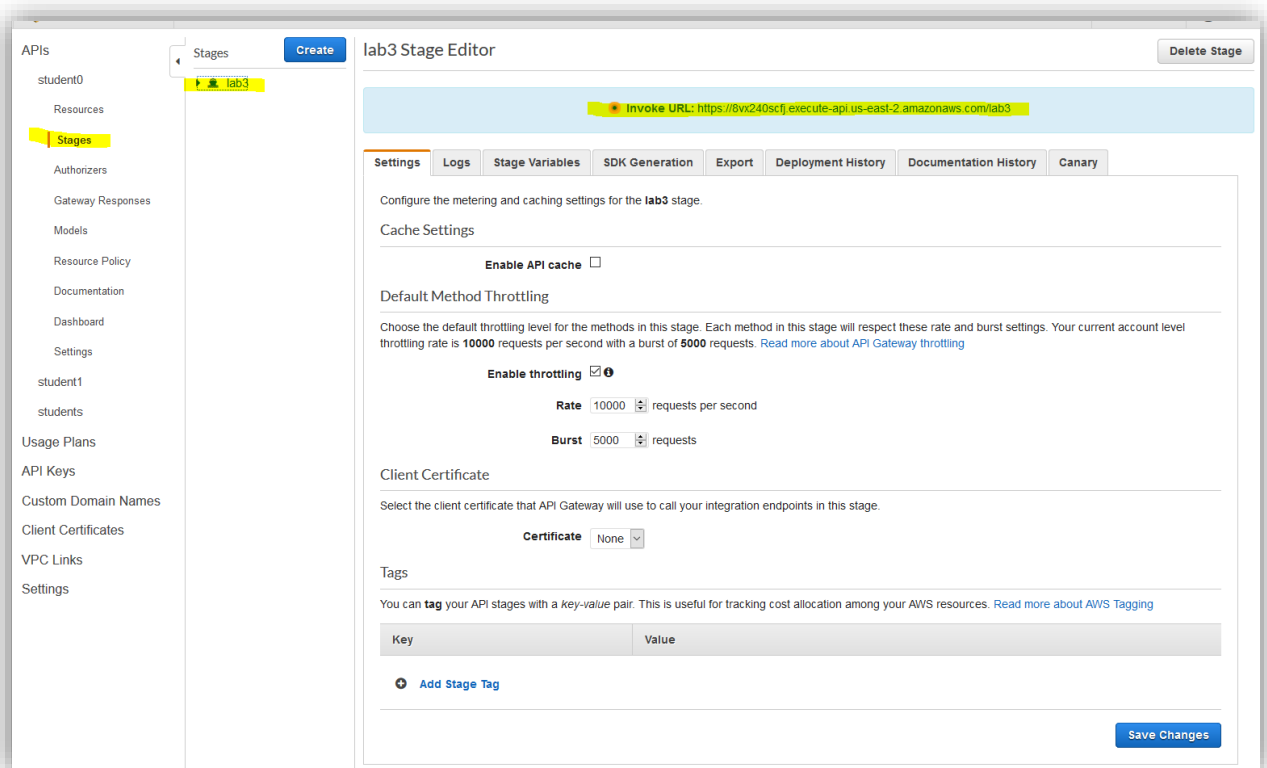
Going to the root URL of an API gateway will give you a not very helpful error that you are missing an auth token. In fact the issue is that you haven't gone to an actual API endpoint:
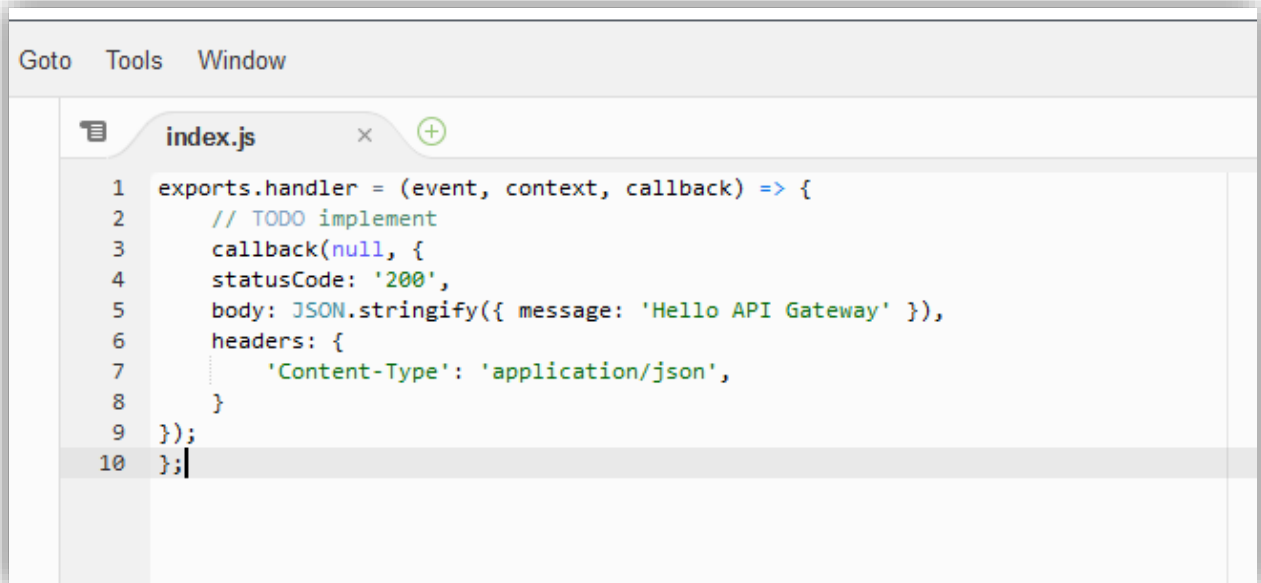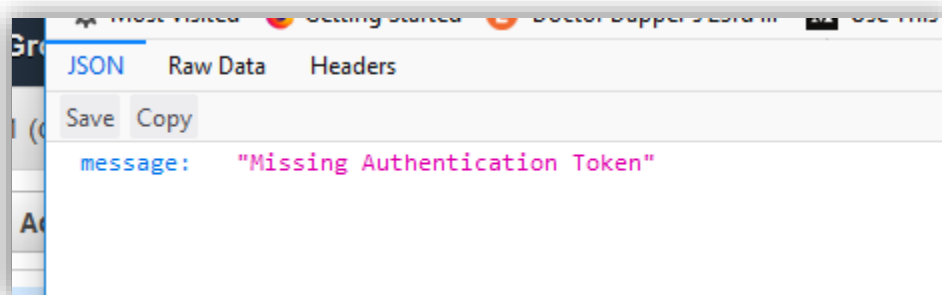
Add /student#API or whatever you named your function to the end of the URL and you should be presented with a new error message:
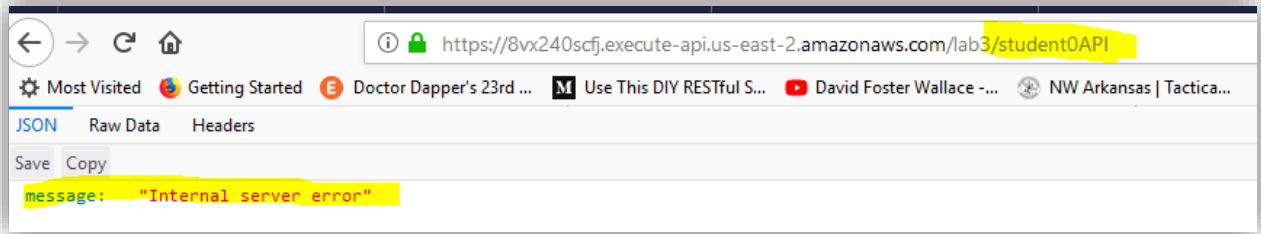
Lambda functions have to return at minimum a status code, otherwise the API gateway defaults to a gateway error 502. Modify our new lambda function to have the following code:
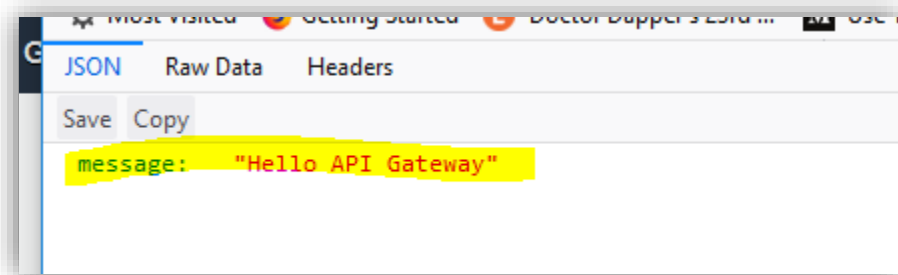
```
exports.handler = (event, context, callback) => {

  const response = {

  statusCode: 200,

  body: JSON.stringify({ message: 'Hello API Gateway' }),

  headers: {

    'Access-Control-Allow-Origin' : '*',

    'Content-Type': 'application/json'

  }

  };

  console.log(response);

  callback(null, response);

};
```

Save the function, and try hitting the API again:

Congratulations, you now have a Lambda service that is accessible from the outside world.