Assignment 12 – API Gateway

No late and **no email** submission. Even if you have done it, if there no submission, it is 0.

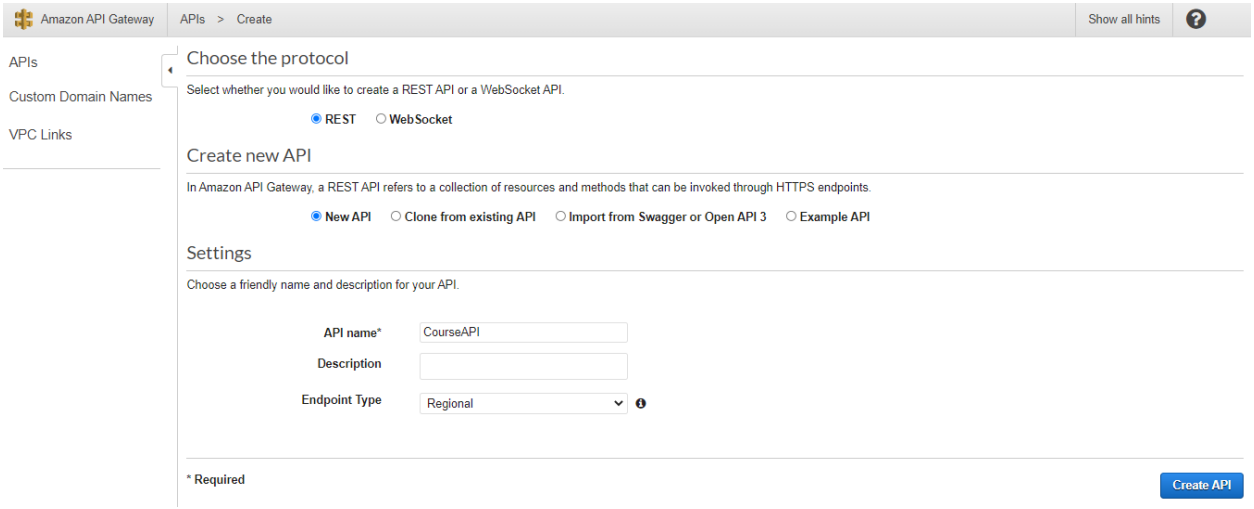
Forums – If you have trouble understanding the concept, do some research. Still, confused? write how much progress you have made so far then ask a question in the Forums section. **Plus 2** scores on the exam if you asked a good question or answered correctly.

Bonus

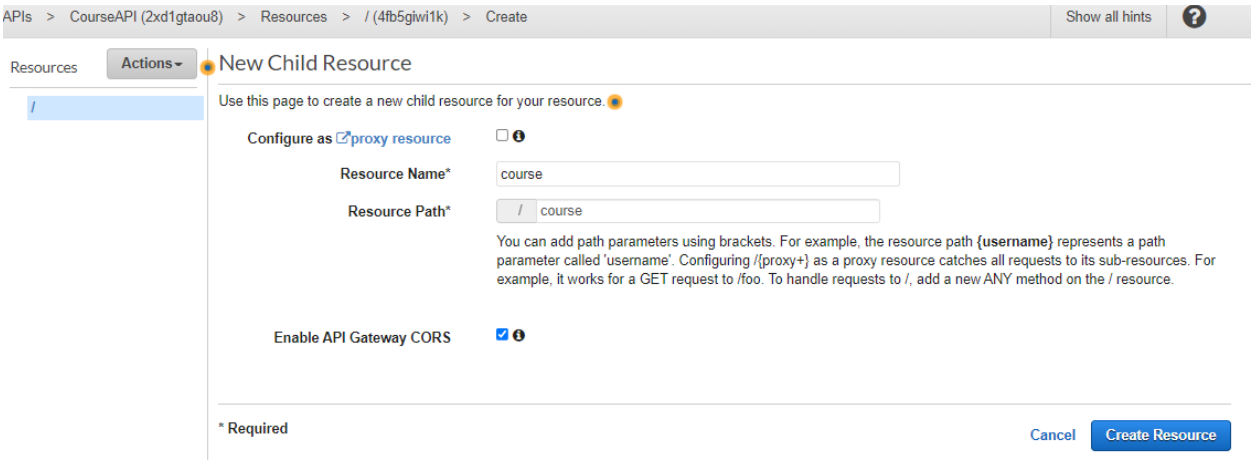
- Store the event payload in S3.

Hands-on lab

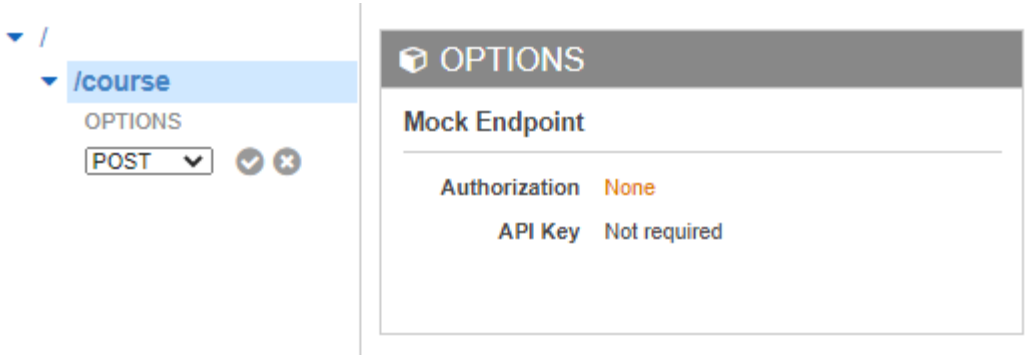
1. AWS Academy doesn't support API Gateway. Create your own AWS account. And setup the billing alarm for you. Create the lambda and dynamodb table. Copy and paste the code in lambda from AWS academy account to your account.
2. Create a "**CourseAPI**" API on API Gateway in front of the "**CourseLambda**"
    a. Search on the top bar and go to the **API Gateway** on AWS Console.
    b. **REST API** (Not REST API private!!)-> click on the orange **Build** button.
    c. On the popup, press **OK.**
    d. In **Create new API**, select **New API** radio button.
    e. In **Settings, API name** is **CourseAPI**. Hit **Create API.**



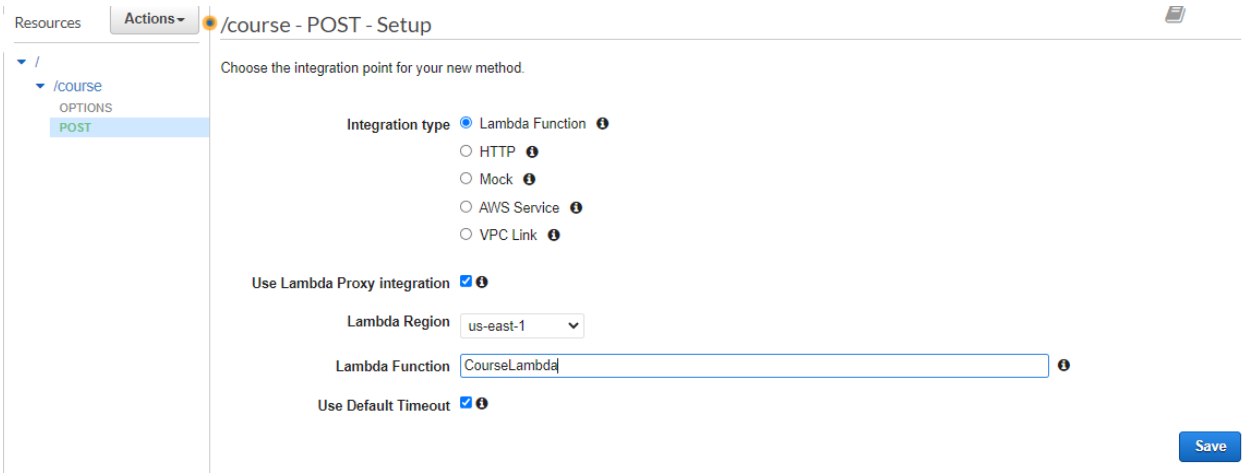
    f. Click on **Actions** dropdown and hit **Create Resource.**
    g. Resource Name is **course.** check **Enable API Gateway CORS.** Hit **Create Resource**.



h. Click on **Actions** dropdown and hit **Create Method.** Select **POST** in the small dropdown under the resource. Click on the small OK icon.



i. Check **Use Lambda Proxy integration**
j. Type the lambda name **CourseLambda** as Lambda Function. Click **Save.**
k. There will be a popup. Read that and hit **OK**.



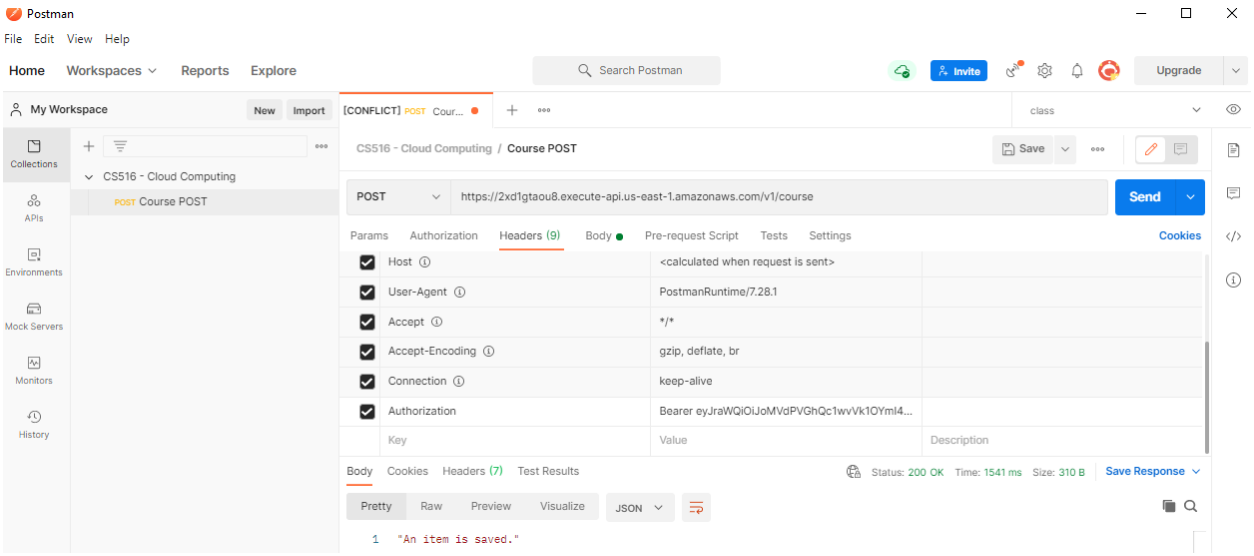
l. Click on the **Actions** dropdown and hit **Deploy API**
m. On the popup, Deployment stage is **[New Stage]**. **Stage name** is v1. Hit **Deploy**.

2. Test your API with Postman.

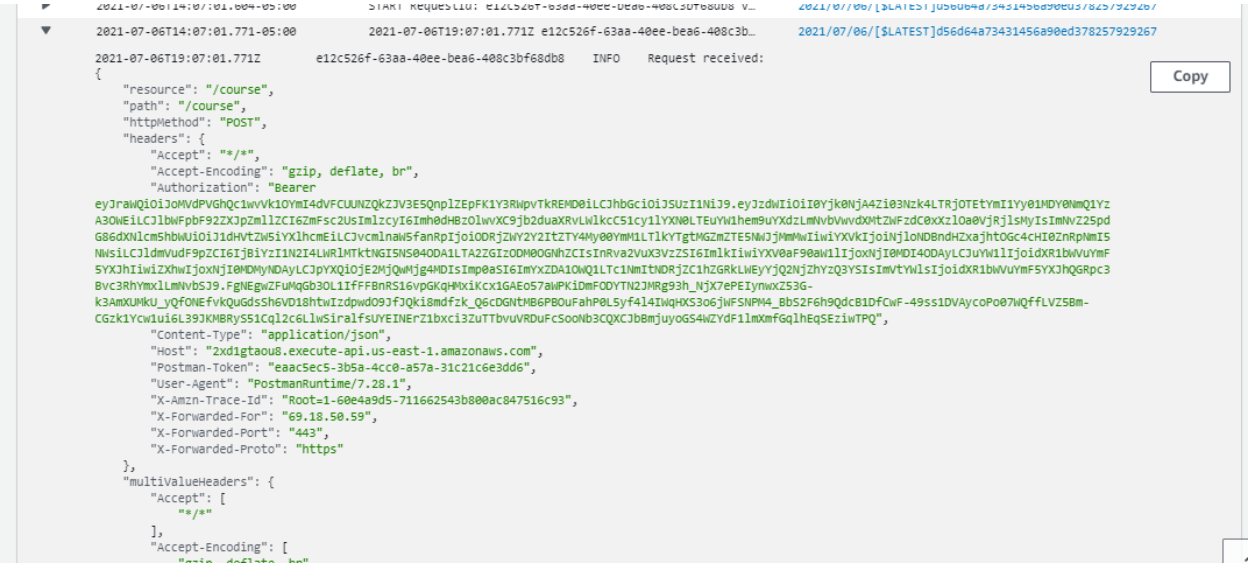
a. Click on **Stages** in left sidebar. Click on **v1**. Grab the **Invoke URL**.
b. Create a new **POST** request in postman. Provide the URL. Append the **course** resource. It will look like this: https://2xd1gtaou8.execute-api.us-east-1.amazonaws.com/v1/**course**
c. The body is below. Feel free to change the value. Body tab -> Select Raw -> Select JSON in the dropdown

```
{
    "courseCode": "CS100",
    "courseName": "My Course",
    "teacherName": "My Teacher",
    "monthYear": "Apr, 2021",
    "students": [
        "Student 1",
        "Student 2"
    ]
}
```

d. You should see the success response below.



3. Update your lambda to store the body we passed instead of hard-coded values.
    a. Go to Lambda -> Configuration -> Permission. In **Resource-based policy**, You will see a new statement. Explain what that is.
    b. Go to Lambda -> Monitor -> View logs in CloudWatch. Click on the orange **Search log group** button. Select **30m** in the top right corner. That will show the latest logs.
    c. We are logging the entire **event** object coming. Like this .log("Request received: " + JSON.stringify(event));. It shows you what was sent from API Gateway to the Lambda.



d. If you scroll all way down, you will find the **body** that the user sent and we want to store that in DB instead of hard-coded values. So you can get the body in the code like this. **const body = JSON.parse(event.body);**

e. Paste the code below and hit **Deploy.**

```
const body = JSON.parse(event.body);

    const saveParams = {
        TableName: tableName,
        Item: {
            "courseCode": {
                S: body.courseCode
            },
            "courseName": {
                S: body.courseName
            },
            "teacherName": {
                S: body.teacherName
            },
            "students": {
                SS: body.students
            },
            "monthYear": {
                S: body.monthYear
            }
        }
    };
```

f. Send the postman request once more, you should be able to see the item you submitted in DynamoDB.

4. Add GET method in the API Gateway and that returns all courses in the DB.
5. Submit the URL of the API.