



nextwork.org

APIs with Lambda + API Gateway

ME

Melvin J Bonner

Create method

Method details

Method type: GET

Integration type:

- Lambda function: Integrate your API with a Lambda function. 
- HTTP: Integrate with an existing HTTP endpoint. 
- Mock: Generate a response based on API Gateway mappings and transformations. 

- AWS service: Integrate with an AWS Service. 
- VPC link: Integrate with a resource that isn't accessible over the public internet. 

Lambda proxy integration: Send the request to your Lambda function as a structured event.

Lambda function
Provide the Lambda function name or alias. You can also provide an ARN from another account.
us-east-1  Q am.awslambdas.us-east-1:123456789012:function:retrieveUserData 

Grant API Gateway permission to invoke your Lambda function
When you save your changes, API Gateway updates your Lambda function's resource-based policy to allow this API to invoke it.

Integration timeout Info
By default, you can enter an integration timeout of 50 - 29,000 milliseconds. You can use Service Quotas to raise the integration timeout to greater than 29,000 ms
29000

Introducing Today's Project!

In this project, I will demonstrate how code can translate user actions into application functionality. I'm doing this project to learn how to build an API without having to manage a traditional server.

Tools and concepts

Services I used were Lambda and API Gateway. Key concepts I learned include Lambda functions, API Resource, and API Method.

Project reflection

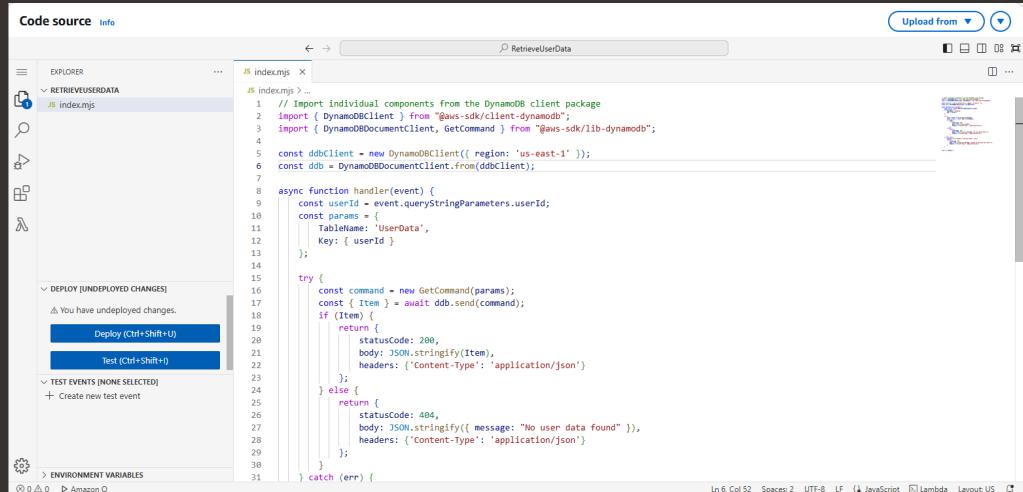
This project took me approximately 75 minutes. The most challenging part was writing the documentation for my API. It was the most rewarding to download the documentation.

I did this project to write and run code that translates user actions to application functionality. This project met my goals.

Lambda functions

AWS Lambda is a service that lets you run code without needing to manage a server. I'm using Lambda in this project to run my code.

The code I added to my function will retrieve data from a DynamoDB table.



The screenshot shows the AWS Lambda code editor interface. The left sidebar displays the project structure with a file named 'index.js' selected. The main area contains the following JavaScript code:

```
JS index.js X
JS index.js > ...
1 // Import individual components from the DynamoDB client package
2 import { DynamoDBClient } from "@aws-sdk/client-dynamodb";
3 import { DynamoDBDocumentClient, GetCommand } from "@aws-sdk/lib-dynamodb";
4
5 const ddBClient = new DynamoDBClient({ region: 'us-east-1' });
6 const ddb = DynamoDBDocumentClient.from(ddBClient);
7
8 async function handler(event) {
9     const userId = event.queryStringParameters.userId;
10    const params = {
11        TableName: 'UserData',
12        Key: { userId }
13    };
14
15    try {
16        const command = new GetCommand(params);
17        const { Item } = await ddb.send(command);
18        if (Item) {
19            return {
20                statusCode: 200,
21                body: JSON.stringify(Item),
22                headers: { 'Content-type': 'application/json' }
23            };
24        } else {
25            return {
26                statusCode: 404,
27                body: JSON.stringify({ message: "No user data found" }),
28                headers: { 'Content-Type': 'application/json' }
29            };
30        }
31    } catch (err) {
```

The code defines a Lambda function named 'handler' that takes an event object with a 'userId' query parameter. It uses the AWS SDK for JavaScript to interact with a 'UserData' table in DynamoDB. If a user is found, it returns their data in JSON format. If no user is found, it returns a 404 error message.

API Gateway

APIs are a way for different software systems to talk to each other. There are different types of APIs, like REST, HTTP, and WebSocket. My API is REST. REST is suited for maintaining a standard web API.

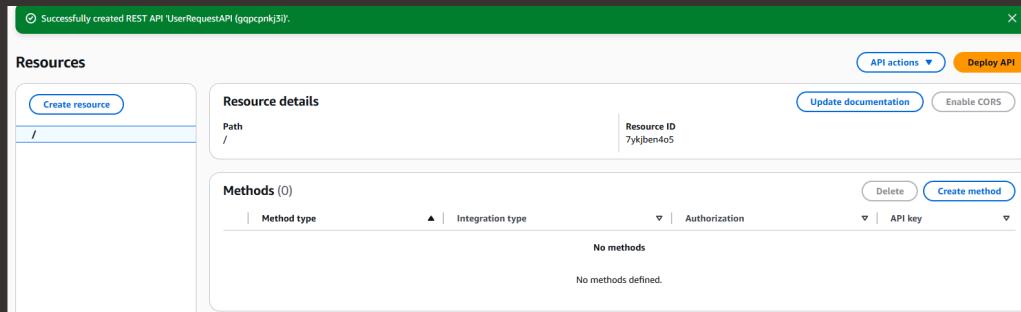
Amazon API Gateway is an AWS service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. I'm using API Gateway in this project to forward the user requests to our Lambda function.

When a user makes a request API Gateway receives the request and forwards them to Lambda.

ME

Melvin J Bonner
NextWork Student

nextwork.org



API Resources and Methods

An API is made up of resources, which are individual endpoints within an API that handle different parts of its functionality.

Each resource consists of methods, which define the actions you can perform on a resource.

I created a GET method. When the method is called, API Gateway will pass the request to the Lambda function.

ME

Melvin J Bonner
NextWork Student

nextwork.org

Create method

Method details

Method type
GET

Integration type

Lambda function
Integrate your API with a Lambda function.


HTTP
Integrate with an existing HTTP endpoint.


Mock
Generate a response based on API Gateway mappings and transformations.


AWS service
Integrate with an AWS Service.


VPC link
Integrate with a resource that isn't accessible over the public internet.


Lambda proxy integration
Send the request to your Lambda function as a structured event.

Lambda function
Provide the Lambda function name or alias. You can also provide an ARN from another account.
us-east-1 ▾ Q arnaws:lambda:us-east-1:812541449234:function:RetrieveUserData X

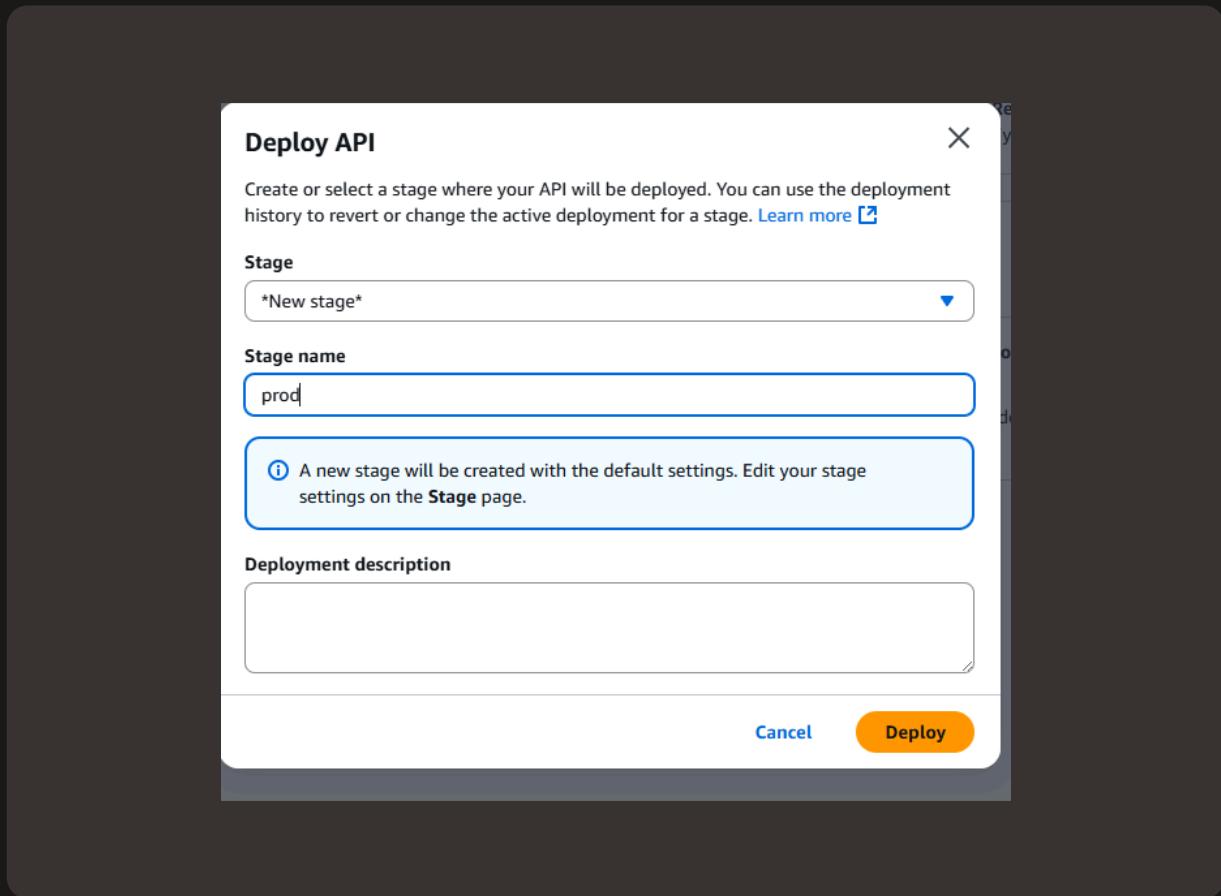
Grant API Gateway permission to invoke your Lambda function
When you save your changes, API Gateway updates your Lambda function's resource-based policy to allow this API to invoke it.

Integration timeout Info
By default, you can enter an integration timeout of 50 - 29,000 milliseconds. You can use Service Quotas to raise the integration timeout to greater than 29,000 ms
29000

API Deployment

When you deploy an API, you deploy it to a specific stage. A stage is a snapshot of your API at a specific point in time. I deployed to a prod stage. Prod stands for production. Production is the live environment where your API is fully working, and there is live traffic and real users using your API.

To visit my API, I copied the invoke URL and tried to access it in a new browser tab. The API displayed an error because I haven't set up a DynamoDB table yet.



API Documentation

For my project's extension, I am writing API documentation because documentation is crucial for developers to understand how to use the API correctly and efficiently. You can do this in JSON for your API.

Once I prepared my documentation, I can publish it to a special file type (either Swagger or OpenAPI). You have to publish your API to a specific stage because it makes sure that your documentation is consistent with the API version deployed to that stage.

My published and downloaded documentation showed me the API's version and title, resources, and method.

ME

Melvin J Bonner
NextWork Student

nextwork.org

```
"/x-amazon-apigateway-documentation" : {  
    "version" : "1",  
    "createdDate" : "2025-06-07T19:18:02Z",  
    "documentationParts" : [ {  
        "location" : {  
            "type" : "API"  
        },  
        "properties" : {  
            "description" : "The UserRequestAPI manages user data retrieval and manipulation. It supports operations to retrieve user details based on unique identifiers.",  
            "baseURL" : "https://gqpcpnkj3i.execute-api.us-east-1.amazonaws.com"  
        }  
    }]
```



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

