



APIs with Lambda + API Gateway



Nchindo Boris

The screenshot shows the AWS API Gateway 'Create method' interface. The 'Method type' dropdown is set to 'GET'. The 'Integration type' section is expanded, showing five options: 'Lambda function' (selected), 'HTTP', 'Mock', 'AWS service', and 'VPC link'. Below these, 'Lambda proxy integration' is listed. The 'Lambda function' field contains the ARN: arn:aws:lambda:eu-north-1:550744777562:function:Retrieval. A note below the field states: 'When you save your changes, API Gateway updates your Lambda function's resource-based policy to allow this API to invoke it.'



Introducing Today's Project!

In this project, I will;

- Develop a serverless Lambda function.
- Configure an API with API Gateway.
- Connect Lambda with API Gateway

Tools and concepts

Services I used were AWS Lambda and API Gateway. Key concepts I learnt include Lambda functions and connecting to APIs that I configured.

Project reflection

This project took me approximately 35 minutes to complete. It was not a challenging project. It was most rewarding to see that the API set up was all welldone.

I did this project today as it is one of the essential projects and goals in getting a cloud career



Lambda functions

AWS Lambda is a service that lets you run code without needing to manage any computers/servers. I'm using Lambda in this project to Create and configure a new Lambda function with its settings. Proceed to add some code to retrieve user data.

The code I added to my function will sets up a Lambda function that retrieves data from a DynamoDB table still to be created

The screenshot shows the AWS Lambda function configuration interface. The top navigation bar includes the account ID (5507-4477-7562), region (Europe Stockholm), and user (NCHINDO-IAM-Admin). The main page displays the 'Info' tab for the 'RetrieveUserData' function. A success message states: "Successfully created the function RetrieveUserData. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." On the left, the 'EXPLORER' panel shows the file 'index.mjs' with the following code:

```
// Import individual components from the DynamoDB client package
import { DynamoDBClient } from "@aws-sdk/client-dynamodb";
import { DynamoDBDocumentClient, GetCommand } from "@aws-sdk/lib-dynamodb";

const ddBClient = new DynamoDBClient({ region: 'eu-north-1' });
const ddb = DynamoDBDocumentClient.from(ddBClient);

async function handler(event) {
  const userId = event.queryStringParameters.userId;
  const params = {
    TableName: 'UserData',
    Key: { userId }
  };

  try {
    const command = new GetCommand(params);
    const { Item } = await ddb.send(command);
    if (Item) {
      return {
        statusCode: 200,
        body: JSON.stringify(Item),
        headers: { 'Content-Type': 'application/json' }
      };
    } else {
      return {
        statusCode: 404,
        body: JSON.stringify({ error: 'User not found' }),
        headers: { 'Content-Type': 'application/json' }
      };
    }
  } catch (err) {
    console.error(err);
    return {
      statusCode: 500,
      body: JSON.stringify({ error: 'Internal server error' }),
      headers: { 'Content-Type': 'application/json' }
    };
  }
}
```

The right side of the interface features a 'Tutorials' sidebar with a section titled 'Create a simple web app'. It includes a brief description and a 'Start tutorial' button. The bottom of the screen shows the Windows taskbar with various pinned icons.



API Gateway

APIs are ways for different software systems to talk to each other. It's like a messenger that carries requests and responses between systems. There are different types of APIs, like REST, HTTP, and WebSocket. My API is REST

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 provide security for our Lambda function which should not be exposed. API Gateway brings in authentication and authorization features, and advanced API management capabilities

When a user makes a request, the API Gateway receives requests and then forwards them to Lambda functions for processing. Lambda processes the request, then sends the response through the API Gateway back to the user



The screenshot shows the AWS API Gateway Resources page. A success message at the top right says "Successfully created REST API 'UserRequestAPI (fr9gjfl35f)'". The main area displays the "Resources" section for the API. It shows a single resource path "/". The "Resource details" panel shows the Resource ID as "elw5mn681h". The "Methods (0)" panel indicates "No methods defined." The left sidebar lists various API gateway features like APIs, Stages, Authorizers, and Documentation. The bottom of the screen shows the Windows taskbar with icons for CloudShell, Feedback, and several pinned browser tabs.



API Resources and Methods

An API is made up of resources, which are individual endpoints within your API that handle different parts of its functionality. For example, an API for a messaging app might have separate resources for retrieving messages and for retrieving user profiles.

Each resource consists of methods, which are commands that define the actions you can perform on a resource letting you to interact with data over the internet. For example: GET to retrieve, POST to add, PUT to update, and DELETE to remove data.

I created a GET method for the /users resource



The screenshot shows the 'Create method' dialog box in the AWS API Gateway console. The 'Method type' dropdown is set to 'GET'. The 'Integration type' section is expanded, showing five options:

- Lambda function**: Selected. Description: "Integrate your API with a Lambda function." Icon: Lambda symbol.
- HTTP**: Description: "Integrate with an existing HTTP endpoint." Icon: HTTP symbol.
- Mock**: Description: "Generate a response based on API Gateway mappings and transformations." Icon: Mock symbol.
- AWS service**: Description: "Integrate with an AWS Service." Icon: AWS symbol.
- VPC link**: Description: "Integrate with a resource that isn't accessible over the public internet." Icon: VPC link symbol.

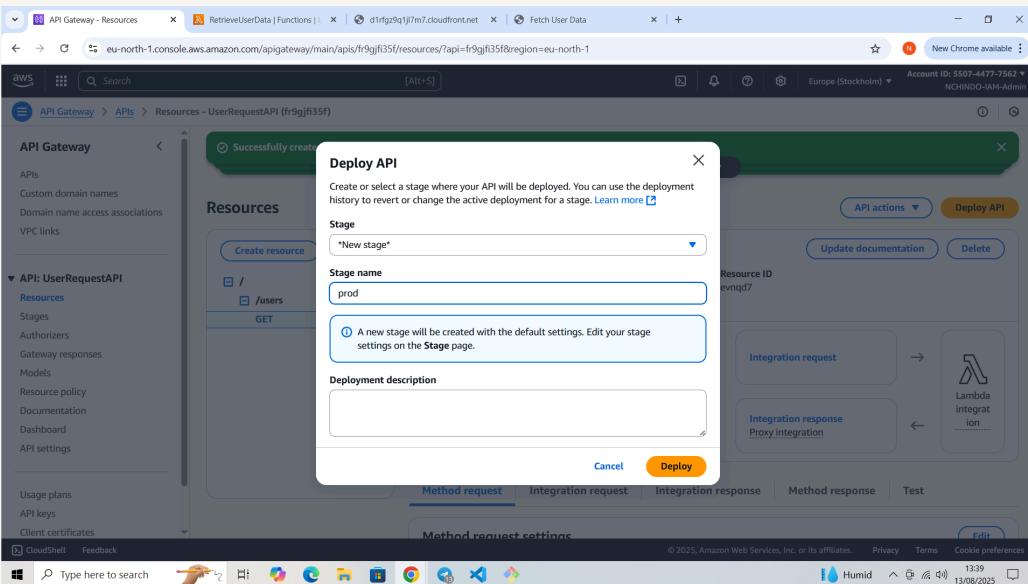
Below the integration type section, there is a note about **Lambda proxy integration**: "Send the request to your Lambda function as a structured event." A 'Lambda function' input field contains the ARN: `arn:aws:lambda:eu-north-1:55074477562:function:Retrieve`. A note below it says: "Grant API Gateway permission to invoke your Lambda function".
At the bottom of the dialog, there are links for CloudShell, Feedback, and a search bar. The status bar at the bottom right shows: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 13:34 13/08/2025.



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 the production stage.

To visit my API, I Copied the Invoke URL, accessed the URL in a new tab on my browser. The API displayed an error because I haven't set up my DynamoDB table yet.





nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

