



nextwork.org

APIs with Lambda + API Gateway



saqibh49@gmail.com

The screenshot shows the AWS API Gateway 'Create method' interface. At the top, the method type is set to 'GET'. Under 'Integration type', 'Lambda function' is selected, indicated by a blue border around its box. Below this, 'Lambda proxy integration' is selected, with the note 'Send the request to your Lambda function as a structured event.' The 'Response transfer mode' is set to 'Buffered'. In the 'Lambda function' section, the ARN 'arnaws:lambda:us-east-2:510482603806:function:RetrieveUserData' is entered. The interface includes standard AWS navigation and search bars at the top.

saqibh49@gmail.com

NextWork Student

nextwork.org

Introducing Today's Project!

In this project, I will demonstrate how to build a serverless Lambda function, configure an API, connect Lambda with an AP Gateway, and write JSON documentation for my API. I'm doing this project to learn about the pieces that go into building, maintaining and running a web app.

Tools and concepts

Services I used were Lamnbda and API Gateway. Key concepts I learnt include Lambda functions, APIs, HTTP requests, API Gateway, API resources, API stages, and creating documentation

Project reflection

This project took me approximately 45 minutes. The most challenging part was understanding how all the pieces connect — Lambda, API Gateway, r sources, methods, and stages all have to be set up in the right order for everything to work. It was most rewarding to see the documentation laid out - felt like a real engineer for a second.

placeholder

I chose to do this project today because I want to learn everything about cloud engineering to have the best shot at landing a role someday.



saqibh49@gmail.com

NextWork Student

nextwork.org

Something that would make learning with NextWork even better is if some of the projects had little code areas where I could practice writing code in a controlled environment instead of always just copying and pasting, or switching tabs 100 times if I do decide to type it out myself.

saqibh49@gmail.com

NextWork Student

nextwork.org

Lambda functions

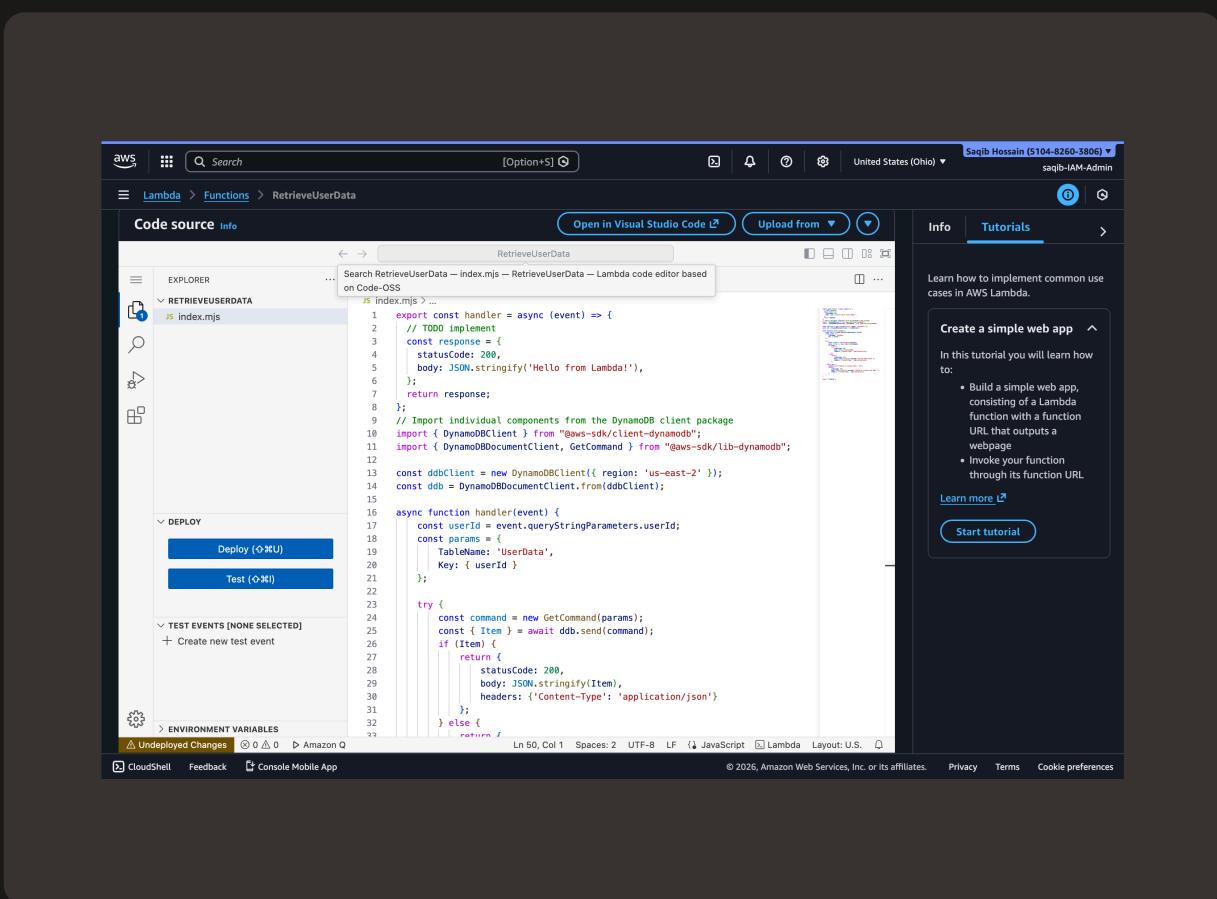
AWS Lambda is a serverless compute service that lets you run code without having to set up or manage servers. You just write your function, upload it, and Lambda runs it for you whenever it's triggered. It's useful because you only pay for the time your code is actually running, and AWS handles all the scaling and infrastructure behind the scenes. I'm using Lambda in this project to handle the backend logic for my API. When a request comes in through API Gateway, Lambda runs the function that processes that request and sends back a response — so it's basically the brain behind my API.

The code I added to my function will retrieve data from a dynamoDB table. It looks for specific user data based on a 'userId' and returns that data. If there's an error e.g. the userId doesn't exist in the database, it returns an error message.

saqibh49@gmail.com

NextWork Student

nextwork.org



saqibh49@gmail.com

NextWork Student

nextwork.org

API Gateway

APIs are basically messengers that let different software applications talk to each other. When an app pulls data from a server, there's an API handling that communication behind the scenes. There are different types of APIs, like REST APIs, HTTP APIs and WebSocket APIs. My API is a REST API, meaning it uses HTTP requests like GET, POST, PUT, and DELETE to send and receive data.

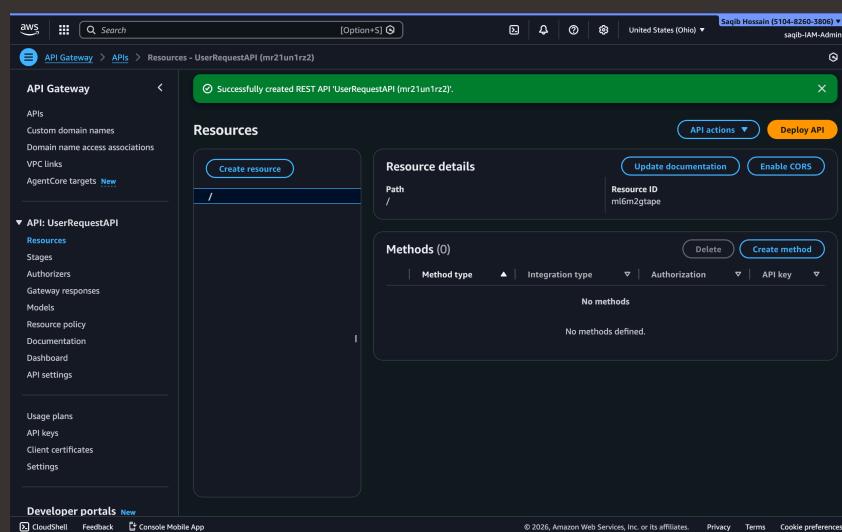
Amazon API Gateway is like the front door to my Lambda function. I'm using API Gateway in this project to send data requested by my API and retrieved by my Lambda function back to the user in a secure way.

When a user makes a request, Lambda retrieves the data from the database, then hands it off to the API Gateway, which then delivers the data to the consumer.

saqibh49@gmail.com

NextWork Student

nextwork.org



saqibh49@gmail.com

NextWork Student

nextwork.org

API Resources and Methods

An API is made up of resources, which are basically specific paths in my API's URL that users can send requests to. Think of it like creating a page on a website — without it, there's nowhere for requests to go.

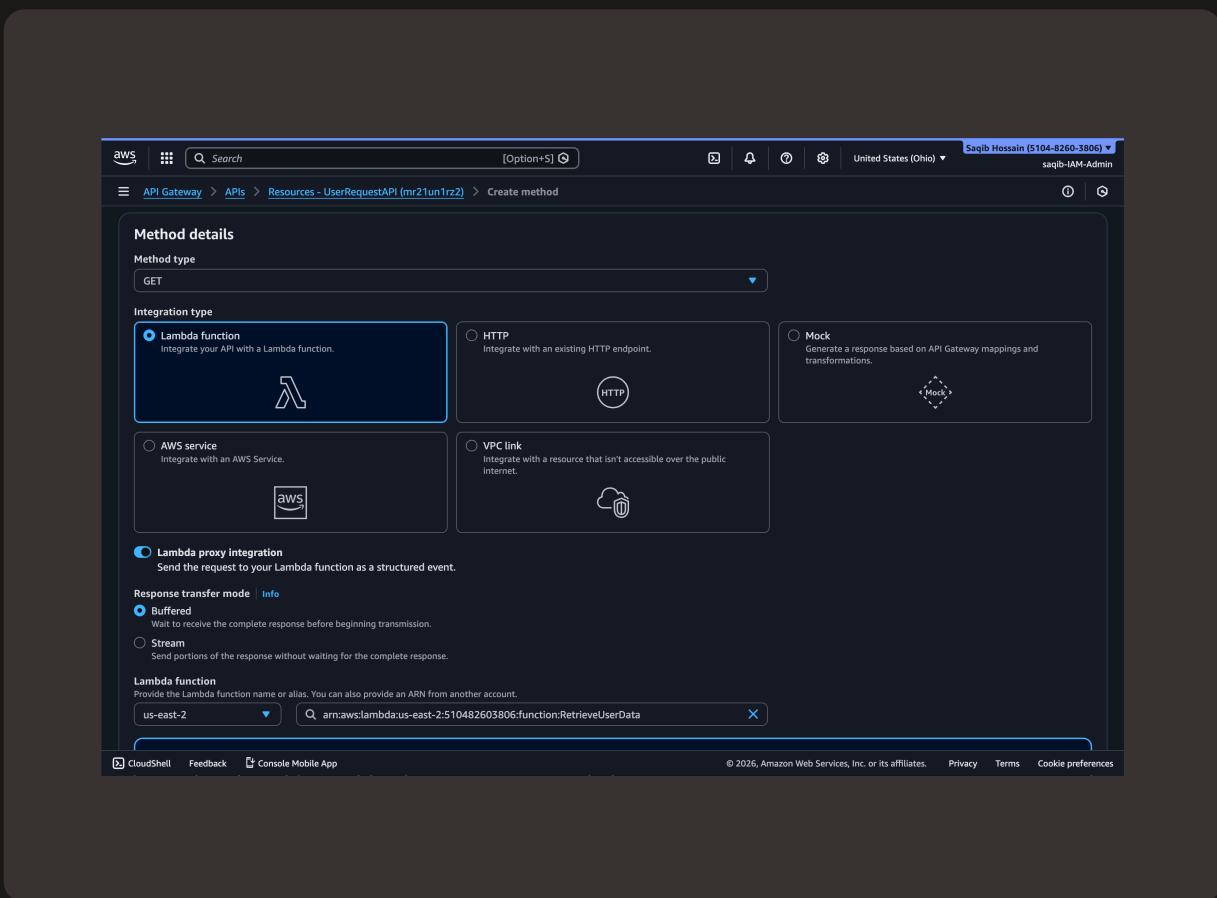
Each resource consists of methods, which are certain HTTP commands that can be done on a resource. For example, adding, updating, deleting and retrieving data.

I created a GET method that is linked to my Lambda function directly.

saqibh49@gmail.com

NextWork Student

nextwork.org



saqibh49@gmail.com

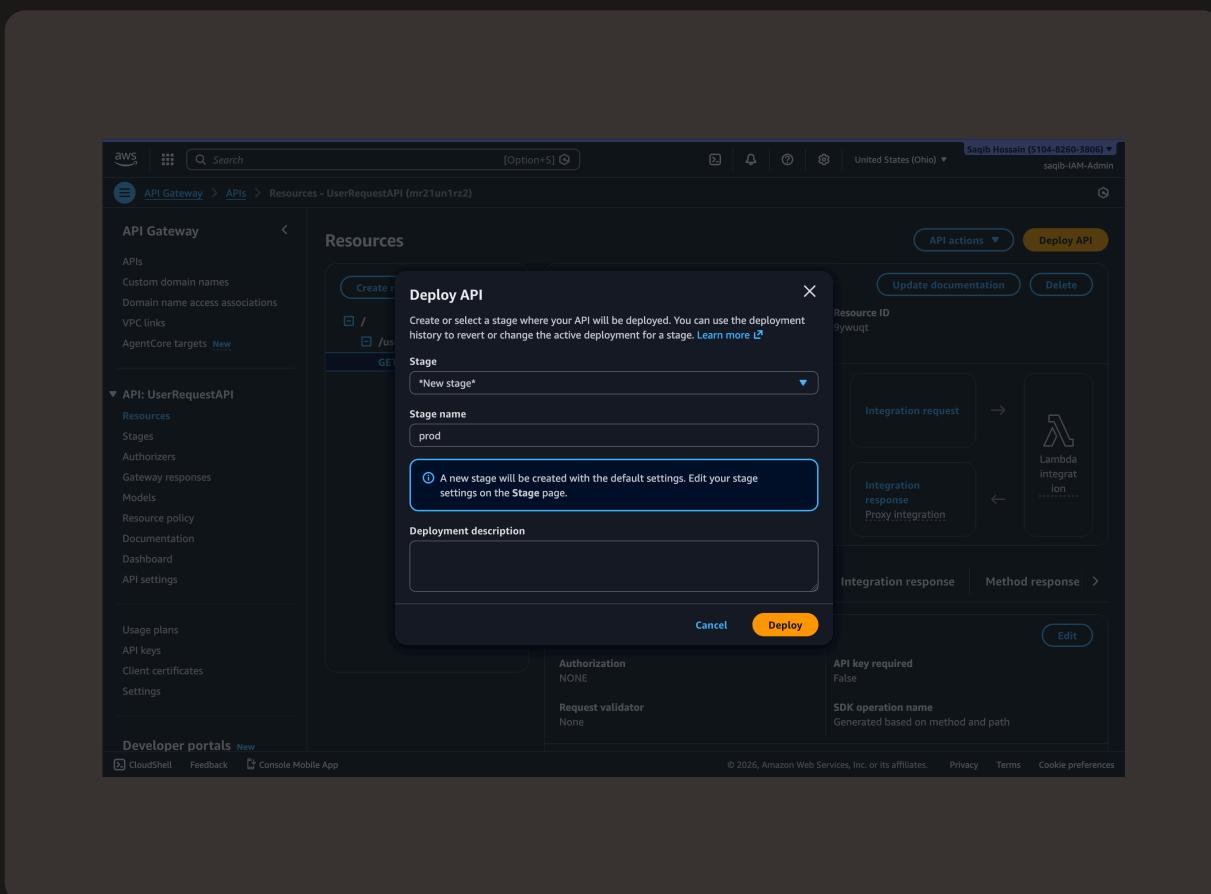
NextWork Student

nextwork.org

API Deployment

When you deploy an API, you deploy it to a specific stage. A stage is a specific version of an API that gives access to different versions to different people, for example, a dev and test stage available only to developers and a prod stage available to the public. I deployed to prod, which is a production environment accessible by anyone.

To visit my API, I went to my invoke URL. The API displayed an error because I haven't created a DynamoDB table for the API to access data from yet.



saqibh49@gmail.com

NextWork Student

nextwork.org

API Documentation

For my project's extension, I am writing API documentation because it's important to document specific info about my API when I'm working in production environments since it helps others understand my API's capabilities as well.

Once I prepared my documentation, I can publish it to a stage, which is basically a version of my API that's live and accessible. Think of it like publishing a draft — until you hit publish, nobody can see it. You have to publish your API to a specific stage because it lets you have different versions running at the same time, like a "dev" stage for testing and a "prod" stage for the real thing. That way, you can make changes and test them without messing up what's already live.

My published and downloaded documentation showed me a JSON file containing all the details of my API. It includes the API name (UserRequestAPI), the base URL and stage (prod), the available path (/users) with its GET method, and how it connects to my Lambda function (RetrieveUserData). It also has the description I wrote earlier

saqibh49@gmail.com

NextWork Student

nextwork.org

```
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
swagger: "2.0"
  info:
    version: "2026-02-16T15:32:56Z"
    title: "UserRequestAPI"
    host: "mr21un1rz2.execute-api.us-east-2.amazonaws.com"
    basePath: "/prod"
    schemes:
      0: "https"
  paths:
    /users:
      get:
        produces:
          0: "application/json"
        responses:
          > 200:
            description: "200 response", schema: { ... }
          "x-amazon-apigateway-integration":
            uri: "arn:aws:apigateway:us-east-2:lambda:path/2015-03-31/functions/arn:aws:lambda:us-east-2:510482603886:function:RetrieveUserData/invocations"
            httpMethod: "POST"
            responses:
              > default:
                statusCode: "200"
              "when_no_match":
                20000
              responseTransferMode: "BUFFERED"
              contentHandling: "CONVERT_TO_TEXT"
              type: "aws_proxy"
        definitions:
          & Empty:
            type: "object"
            title: "Empty Schema"
        <x-amazon-apigateway-documentation>
          version: "1"
          createdDate: "2026-02-16T15:44:48Z"
        documentationParts:
          & 0:
            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://mr21un1rz2.execute-api.us-east-2.amazonaws.com/"
        x-amazon-apigateway-security-policy: "TLS_1_0"
```



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

