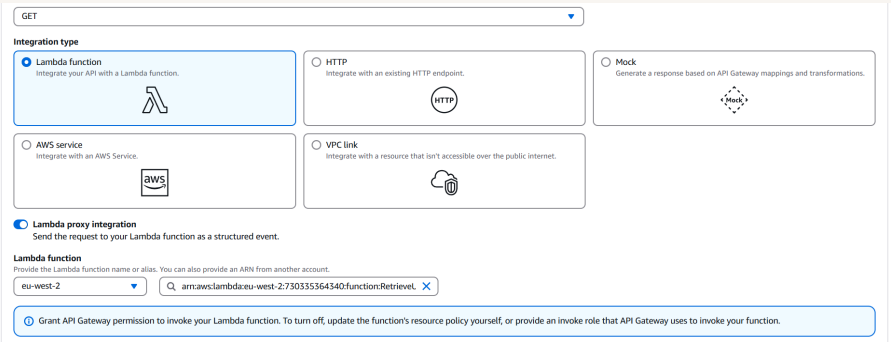
nextwork.org

APIs with Lambda + API Gateway

MA Marzena Pugo



Introducing Today's Project!

In this project I will show how to set up an API using serverless Lamba function and Amazon API Gateway. I am doing this project to learn how APIs work and also to set up a tier in my three-tier architecture.

Tools and concepts

Services I used were Amazon API Gateway and AWS Lambda. Key concepts I learnt include Lambda functions, API resources and methods, Lambda proxy integration, writting API documentation, API stages, invoke URLs for an API.

Project reflection

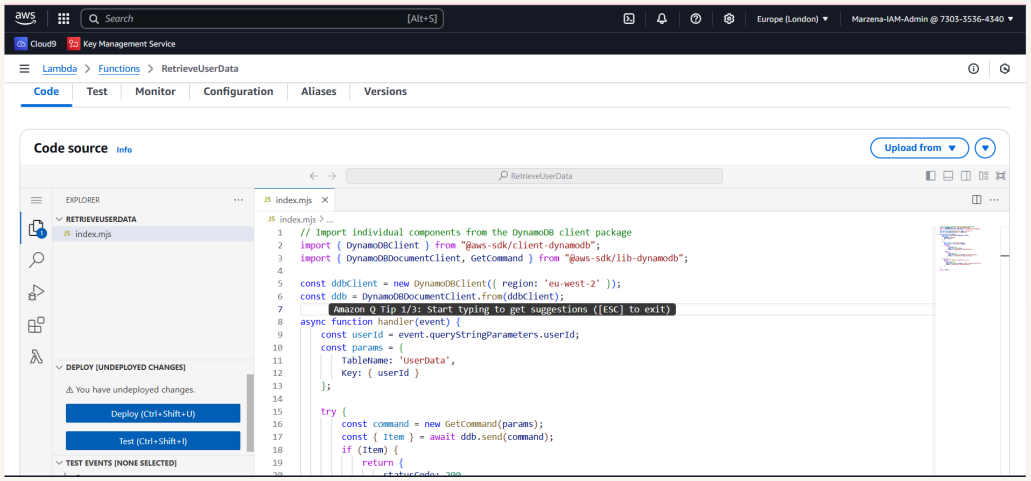
This project took me approximately 1.5 hours. The most challenging part was understanding and writing the Lambda function code. It was most rewarding to to find out how you structure an API.

Today I set up an API using serverless Lamba function and Amazon API Gateway, I created resources and methods with my API, I deployed my API to a live stage. I learnt lots about API.

Lambda functions

AWS Lambda is a service that lets you run code without needing to manage any computers/servers - Lambda will manage them for you.

The code I added to my function will grab a userID from a triggered event (i.e. submitting a userID trough a form/field in a wbsite and queries for piece of data in dynamo DB that matches the UserID. The code also returns the an message (no userID)

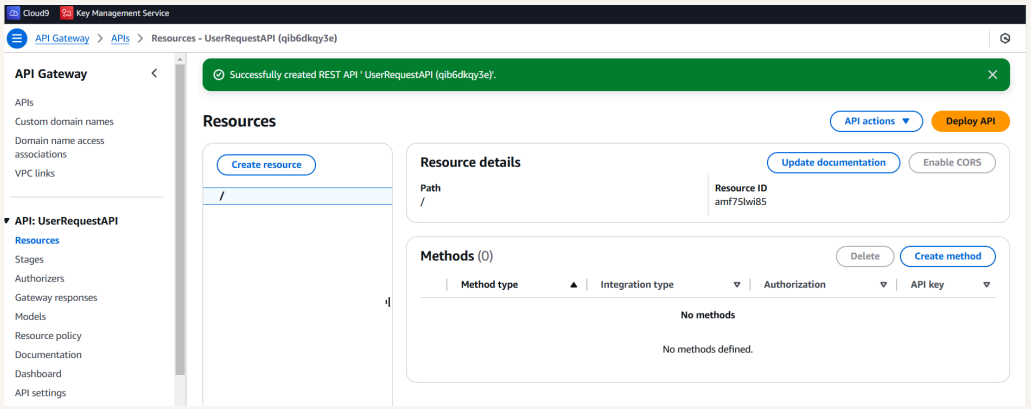


API Gateway

APIs allow different software components to communicate with each other. There are different types of APIs, like REST, HTTP, and WebSocket. My API is a REST API which means uses HTTP methods and it is compatible with most programing languages.

Amazon API Gateway is an AWS service that manages incoming traffic, directing them to the correct services, and makes sure only authorized requests. I'm using API Gateway in this project to connect my web app users with my Lambda fuction.

When a user makes a request click on a "Get User Data" button. API Gateway's role is to receive the traffic, determine whether or not it's authorized and pass it through to Lambda if it is. API Gateway is also a traffic manager for requests.

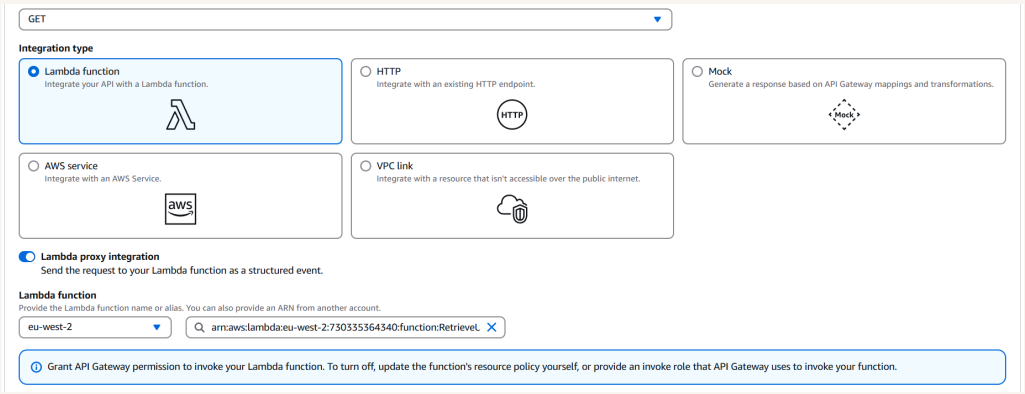

Cloud9
Key Management Service
API Gateway > APIs > Resources - UserRequestAPI (qib6dkqy3e)
API Gateway
APIs
Custom domain names
Domain name access associations
VPC links
API: UserRequestAPI
Resources
Stages
Authorizers
Gateway responses
Models
Resource policy
Documentation
Dashboard
API settings
Successfully created REST API ' UserRequestAPI (qib6dkqy3e)'.
Resources
API actions
Deploy API
Create resource
/
Resource details
Update documentation
Enable CORS
Path
/
Resource ID
amf75lwi85
Methods (0)
Delete
Create method
Method type
Integration type
Authorization
API key
No methods
No methods defined.

API Resources and Methods

API resources 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 based on standard HTTP methods, which are different commands that let you interact with data over the internet. For example: GET to retrieve, POST to add, PUT to update, and DELETE to remove.

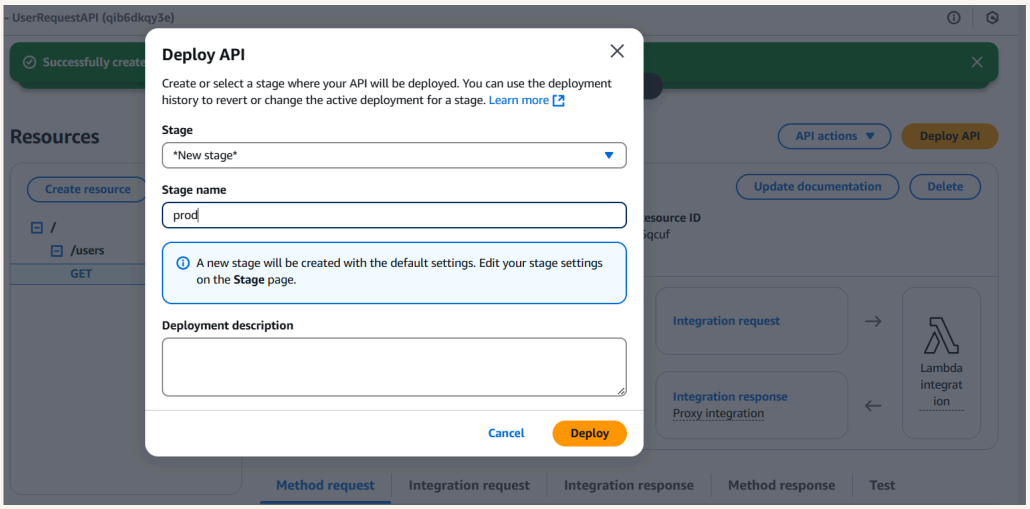
I created a GET method, when the GET method for the /users resource is called, API Gateway will pass that request to the Lambda function you set up. When the function runs, Lambda will retrieve user data in a DynamoDB table.


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.
eu-west-2
arn:aws:lambda:eu-west-2:730335364340:function:RetrieveL
Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.

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 'prod'. Production is the live environment where your API is fully working.

To visit my API, I copied the Invoke URL for the API that is in production. The API displayed an error because I haven't set up my DynamoDB table yet.

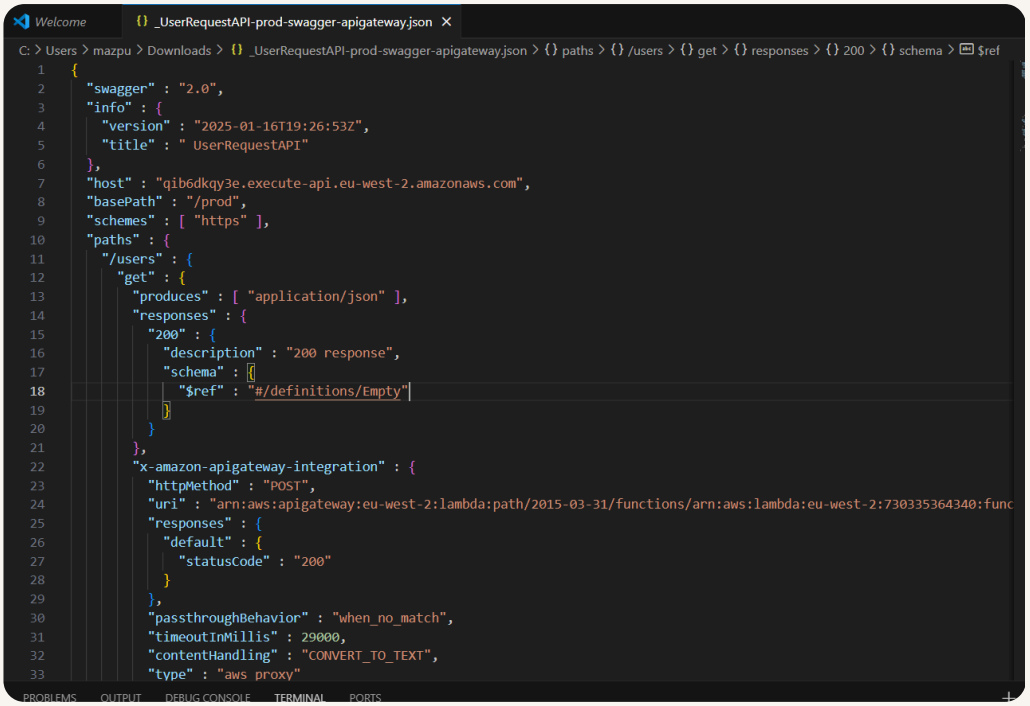


API Documentation

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

Once I prepared my documentation, I can publish it to the prod stage. You have to publish your API to a specific stage because different stages have different versions of the same API.

My published and downloaded documentation showed me my manually written work and some automatically generatated work from my API Gateway. I can upload it into tools like Swagger or Redock to generate nice webpages about my API.


Welcome
_UserRequestAPI-prod-swagger-apigateway.json
C: > Users > mazpu > Downloads > _UserRequestAPI-prod-swagger-apigateway.json > paths > /users > get > responses > 200 > schema > $ref
{
"swagger" : "2.0",
"info" : {
"version" : "2025-01-16T19:26:53Z",
"title" : " UserRequestAPI"
},
"host" : "qib6dkqy3e.execute-api.eu-west-2.amazonaws.com",
"basePath" : "/prod",
"schemes" : ["https"],
"paths" : {
"/users" : {
"get" : {
"produces" : ["application/json"],
"responses" : {
"200" : {
"description" : "200 response",
"schema" : {
"$ref" : "#/definitions/Empty"
}
}
},
"x-amazon-apigateway-integration" : {
"httpMethod" : "POST",
"uri" : "arn:aws:apigateway:eu-west-2:lambda:path/2015-03-31/functions/arn:aws:lambda:eu-west-2:730335364340:func
"responses" : {
"default" : {
"statusCode" : "200"
}
},
"passthroughBehavior" : "when_no_match",
"timeoutInMillis" : 29000,
"contentHandling" : "CONVERT_TO_TEXT",
"type" : "aws_proxy"
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

The place to learn & showcase your skills

Check out nextwork.org for more projects