

2. Creating a Serverless API****

Objective: Develop a serverless API using AWS Lambda and API Gateway.

Approach:

- **Define API:** Design a simple RESTful API (e.g., for a todo list application).
- **Lambda Functions:** Create Lambda functions for each API method (GET, POST, PUT, DELETE).
- **API Gateway Setup:** Use API Gateway to set up the API endpoints, connecting each endpoint to the corresponding Lambda function.
- **Testing:** Test the API using tools like Postman or AWS API Gateway test functionality.

Goal: Gain hands-on experience in building and deploying a serverless API, understanding the integration between Lambda and API Gateway.

First create a Lambda function

[Lambda](#) > [Functions](#) > Create function

Create function [Info](#)

Choose one of the following options to create your function.

☒ **Author from scratch**
Start with a simple Hello World example.

☐ **Use a blueprint**
Build a Lambda application from sample code and configuration presets for common use cases.

☐ **Container image**
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
☒ x86_64
☐ arm64

Permissions [Info](#)

Successfully created the function **task2serverless**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

[Lambda](#) > [Functions](#) > task2serverless

task2serverless

[Throttle](#) [Copy ARN](#) [Actions](#) ▼

▼ **Function overview** [Info](#)

Diagram

Template

task2serverless

Layers (0)

+ Add trigger

+ Add destination

Export to Application Composer

Download ▼

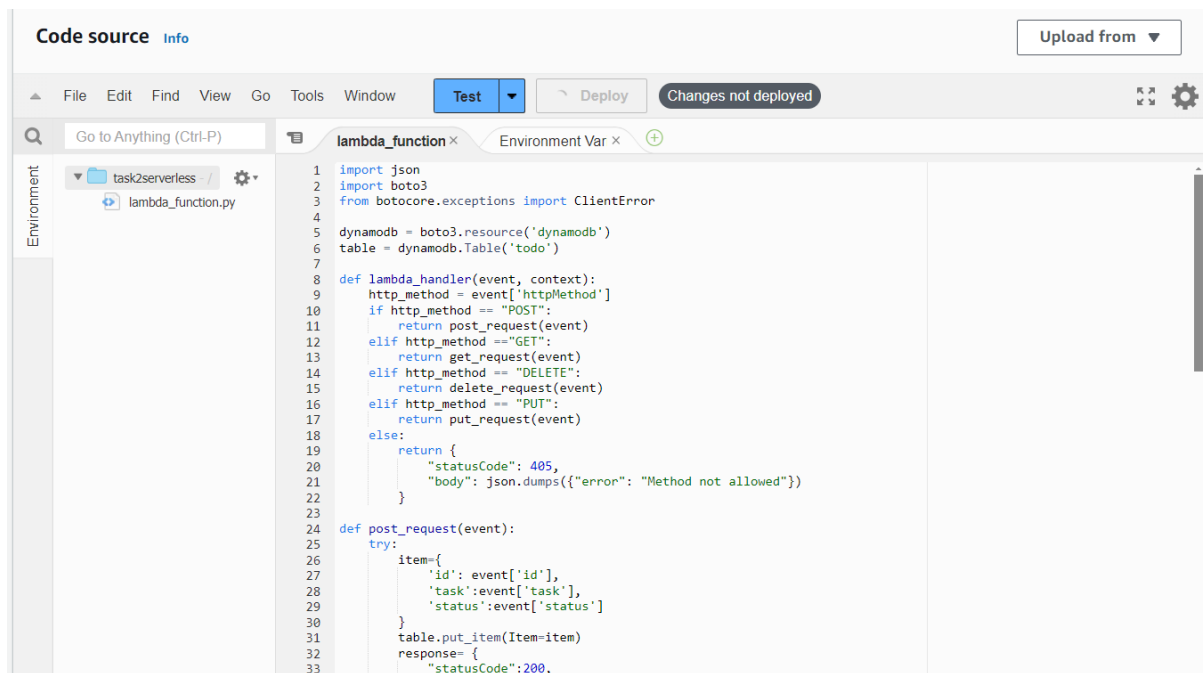
Description
-

Last modified
9 seconds ago

Function ARN
 arn:aws:lambda:us-east-1:695125708392:function:task2serverless

Function URL [Info](#)
-

Now we will replace the lambda code the code that we created for todo list .



The screenshot shows an IDE window titled "Code source Info" with a menu bar (File, Edit, Find, View, Go, Tools, Window) and buttons for "Test", "Deploy", and "Changes not deployed". The left sidebar shows the "Environment" with a folder "task2serverless -/" and a file "lambda_function.py". The main editor displays the code for "lambda_function.py":

```
1 import json
2 import boto3
3 from botocore.exceptions import ClientError
4
5 dynamodb = boto3.resource('dynamodb')
6 table = dynamodb.Table('todo')
7
8 def lambda_handler(event, context):
9     http_method = event['httpMethod']
10    if http_method == "POST":
11        return post_request(event)
12    elif http_method == "GET":
13        return get_request(event)
14    elif http_method == "DELETE":
15        return delete_request(event)
16    elif http_method == "PUT":
17        return put_request(event)
18    else:
19        return {
20            "statusCode": 405,
21            "body": json.dumps({"error": "Method not allowed"})
22        }
23
24 def post_request(event):
25     try:
26         item={
27             'id': event['id'],
28             'task':event['task'],
29             'status':event['status']
30         }
31         table.put_item(Item=item)
32         response= {
33             "statusCode":200,
```

This code will be triggered whenever the api endpoints has been used,So now lets make a trigger using API gateway.

Trigger configuration [Info](#)



API Gateway

aws api application-services backend HTTP REST serverless

Add an API to your Lambda function to create an HTTP endpoint that invokes your function. API Gateway supports two types of RESTful APIs: HTTP APIs and REST APIs. [Learn more](#)

Intent

Use an existing api or have us create one for you.

- ☒ Create a new API
- ☐ Use existing API

API type

☐ HTTP API

Build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support.

☒ REST API

Develop a REST API where you gain complete control over the request and response along with API management capabilities.

Security

Configure the security mechanism for your API endpoint.

IAM

▼ Additional settings

API name

Choose a name for your API. API names don't need to be unique.

So after configuring API Gateway in our lambda , we can go to the API gateway trigger

Code Test Monitor **Configuration** Aliases Versions

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

Triggers (1) [Info](#) [Refresh](#) [Fix errors](#) [Edit](#) [Delete](#) [Add trigger](#)

☐ Trigger

API Gateway: task2serverless-API

arn:aws:execute-api:us-east-1:695125708392:1mtvlunr70/*/*/task2serverless

API endpoint: <https://1mtvlunr70.execute-api.us-east-1.amazonaws.com/default/task2serverless>

[Details](#)

Then click on the link .It will redirect you to the new page AS api resources

Resources

API actions ▼

Deploy API

Create resource

⌵ /

⌵ /task2serverless

ANY

Resource details

Delete

Update documentation

Enable CORS

Path
/task2serverless

Resource ID
2m2b1u

Methods (1)

Delete

Create method

Method type ▲

Integration type ▼

Authorization ▼

API key

○

ANY

Lambda

IAM

Not rec

Then we create methods like get post put delete .

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



arn:aws:lambda:us-east-1:695125708392:function:task



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.



Default timeout

The default timeout is 29 seconds.

Cancel

Create method

Create method

Method details

Method type

POST

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 arn:aws:lambda:us-east-1:695125708392:function:task: X

- i** 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.

☒ **Default timeout**

The default timeout is 29 seconds.

Cancel

Create method

Create method

Method details

Method type

PUT

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



arn:aws:lambda:us-east-1:695125708392:function:task



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.



Default timeout

The default timeout is 29 seconds.

Cancel

Create method

Create method

Method details

Method type

DELETE

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

arn:aws:lambda:us-east-1:695125708392:function:task

i 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.

☒ Default timeout

The default timeout is 29 seconds.

Cancel

Create method

[API Gateway](#) > [APIs](#) > [Resources - task2serverless-API \(1mtvlunr70\)](#) > **Enable CORS**

Enable CORS

CORS settings [Info](#)

To allow requests from scripts running in the browser, configure cross-origin resource sharing (CORS) for your API.

Gateway responses

API Gateway will configure CORS for the selected gateway responses.

- ☐ Default 4XX
- ☐ Default 5XX

Methods

- ☒ DELETE
- ☒ GET
- ☒ OPTIONS
- ☒ POST
- ☒ PUT

Access-Control-Allow-Methods

DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT

Access-Control-Allow-Headers

API Gateway will configure CORS for the selected gateway responses.

Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token

Access-Control-Allow-Origin

Enter an origin that can access the resource. Use a wildcard "*" to allow any origin to access the resource.

*

▼ Additional settings

Access-Control-Expose-Headers

Enter a comma-separated list of headers the browser can access.

Access-Control-Max-Age

Enter how long, in seconds, that a browser should cache the response to the preflight request.

These are the list of Methods that we created, now we must deploy The API in a new Stage.

API Gateway > APIs > Resources - task2serverless-API (1mtvlunr70)

Resources API actions ▼ Deploy API

Create resource

/task2serverless

ANY
DELETE
GET
OPTIONS
POST
PUT

Resource details Delete Update documentation Enable CORS

Path
/task2serverless

Resource ID
ZmZb1u

Methods (6) Delete Create method


	Method type ▲	Integration type ▼	Authorization ▼	API key ▼
<input type="radio"/>	ANY	Lambda	IAM	Not required
<input type="radio"/>	DELETE	Lambda	None	Not required
<input type="radio"/>	GET	Lambda	None	Not required
<input type="radio"/>	OPTIONS	Mock	None	Not required
<input type="radio"/>	POST	Lambda	None	Not required
<input type="radio"/>	PUT	Lambda	None	Not required

Deploy API

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Stage
New stage ▼

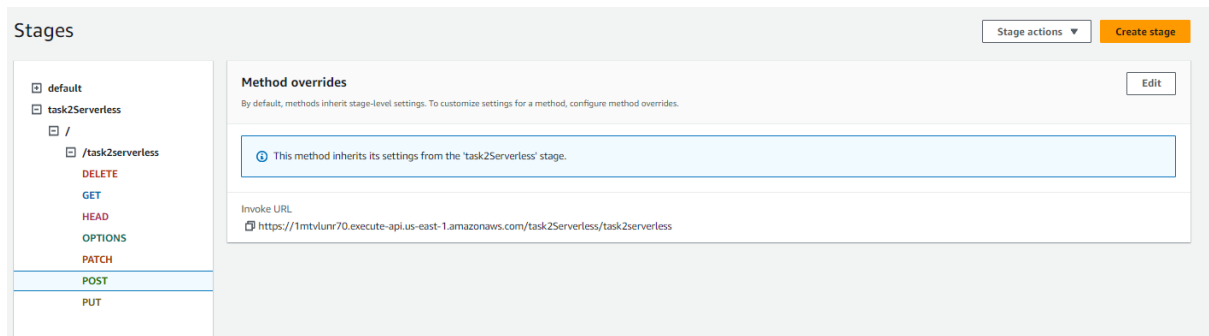
Stage name
task2Serverless

 A new stage will be created with the default settings. Edit your stage settings on the **Stage** page.

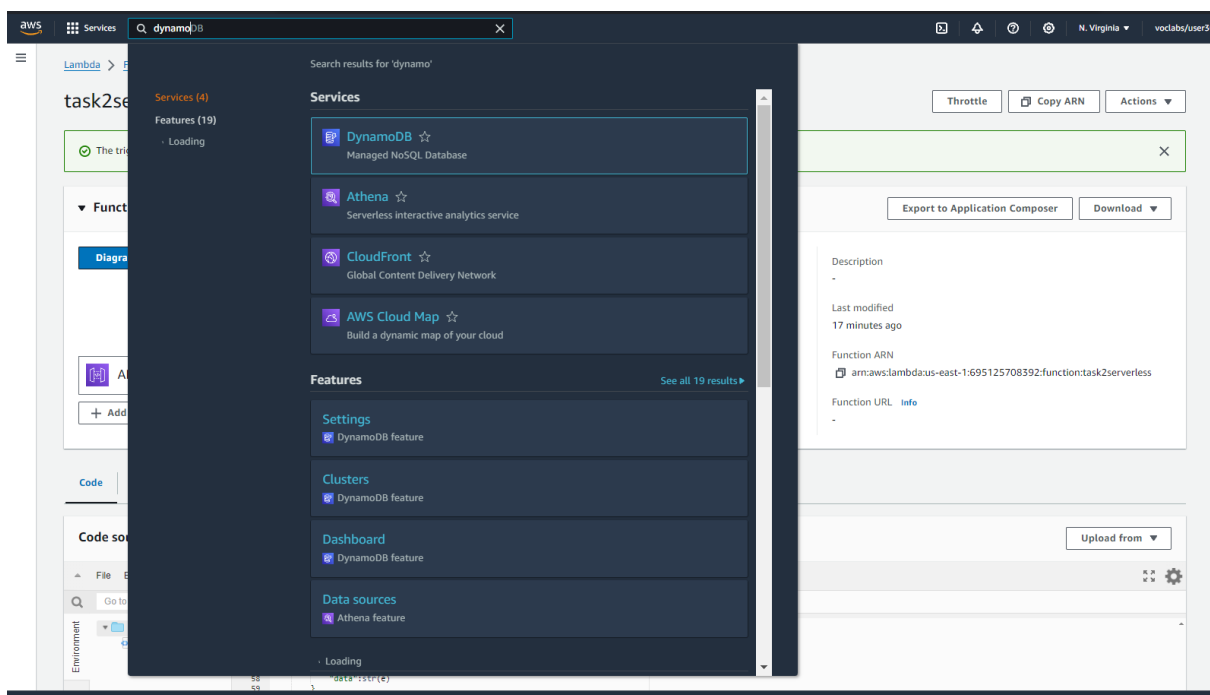
Deployment description
|

Cancel Deploy

As you can see our api has been staged and if you click on Post or any other method it will show you the invoke URL that is a API url which is used to fetch datas.



Now everything is setup, We just need a datatable or database , Using Dynamo DB , create a table



Create table

Table details [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (`_`), hyphens (`-`), and periods (`.`).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

String ▼

1 to 255 characters and case sensitive.

Sort key - *optional*

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

String ▼

1 to 255 characters and case sensitive.

Table settings



Default settings

The fastest way to create your table. You can modify these settings now or after your table has been created.



Customize settings

Use these advanced features to make DynamoDB work better for your needs.

Default table settings

These are the default settings for your new table. You can change some of these settings after creating the table.

DynamoDB > Tables > todo-serverless

Tables (2) ×

Any tag key ▾

Any tag value ▾

Q Find tables by table name

< 1 > ⚙

☐ dynamodb

☒ todo-serverless

todo-serverless

🔄 Actions ▾ Explore table items

Overview | Indexes | Monitor | Global tables | Backups | Exports and streams | Additional settings

📘 Protect your DynamoDB table from accidental writes and deletes

When you turn on point-in-time recovery (PITR), DynamoDB backs up your table data automatically so that you can restore to any given second in the preceding 35 days. Additional charges apply. [Learn more](#)

Edit PITR ×

General information [Info](#)

Partition key id (String)	Sort key -	Capacity mode <u>Provisioned</u>	Table status ✔ Active
Alarms ✔ No active alarms	Point-in-time recovery (PITR) Info ⊖ Off		

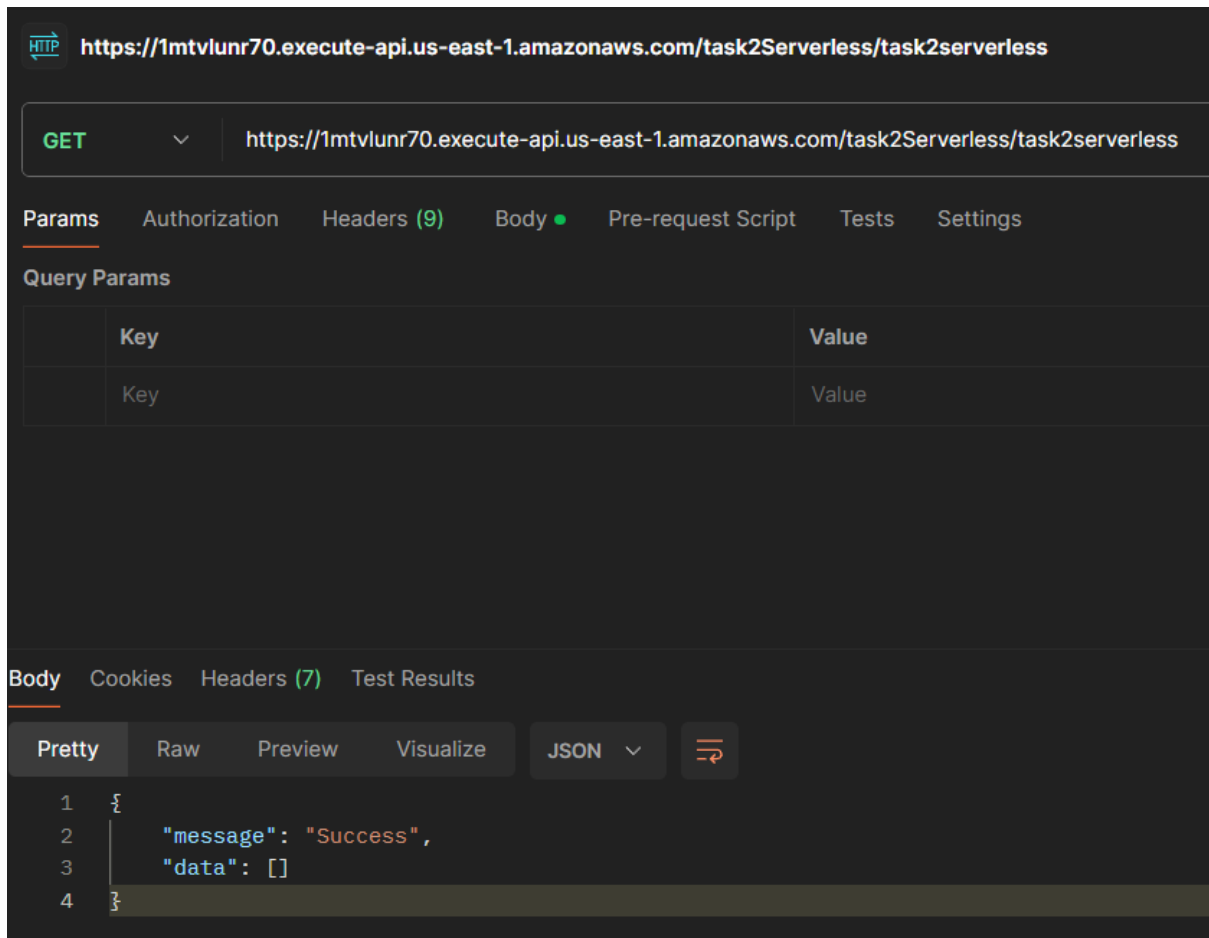
▶ Additional info

Items summary [Get live item count](#)

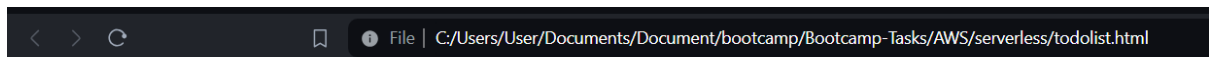
DynamoDB updates the following information approximately every six hours.

Item count 0	Table size 0 bytes	Average item size 0 bytes
-----------------	-----------------------	------------------------------

Implementation



So as we can see API is working . We must implement this api to our frontend Todo List using simple website with HTML.



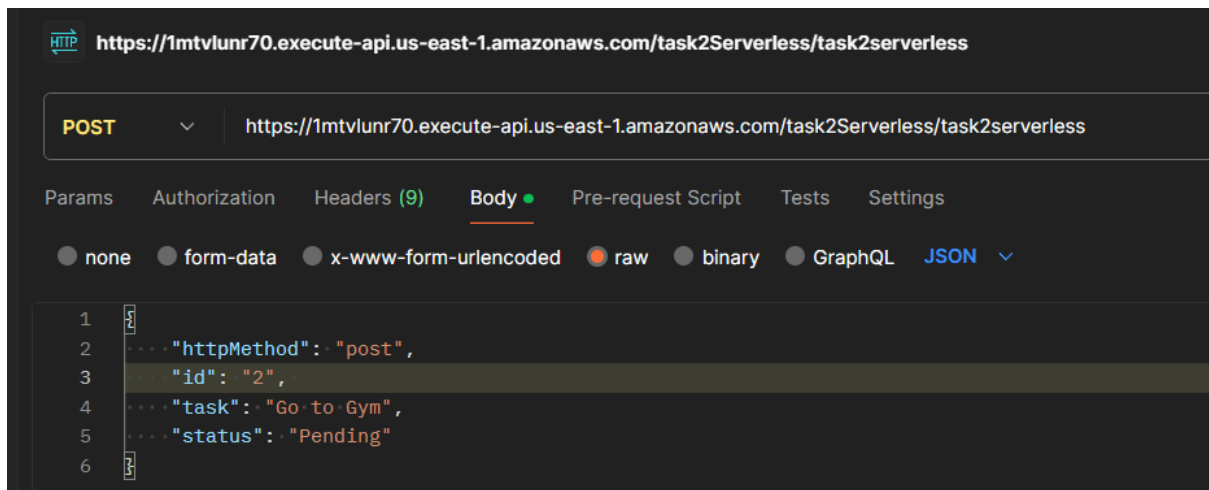
Todo App

Task: Status:

Tasks

Task ID: Update Key: New Value:
 Task ID:

Items returned (2)			
<input type="checkbox"/>	id (String)	status	task
<input type="checkbox"/>	2	Pending	Serverless lab
<input type="checkbox"/>	1	Pending	Basic lab



Testing it in Postman app