

Create Lambda Function and Edit Existing Role to Lab Role

Permissions [Info](#)

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions

☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

LabRole

[View the LabRole role](#) on the IAM console.

► Advanced settings

Cancel Create function

Create Rest API

REST API Private

Create a REST API that is only accessible from within a VPC.

Works with the following:
Lambda, HTTP, AWS Services

Import Build

Create Resources

Resources

API actions Deploy API

Create resource

Path	Resource ID
/	3z14dhw2g8

Resource details

Update documentation Enable CORS

Methods (0)

Delete Create method

Method type	Integration type	Authorization	API key
No methods			

Create Methods like POST, GET, PUT, DELETE

00200

API Gateway > APIs > Resources - to-do-api (c7aluouk25) > 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

☐ VPC link

Methods (2)					Delete	Create method
	Method type ▲	Integration type ▼	Authorization ▼	API key ▼		
<input type="radio"/>	OPTIONS	Mock	None	Not required		
<input type="radio"/>	POST	Lambda	None	Not required		

Methods (3)					Delete	Create method
	Method type ▲	Integration type ▼	Authorization ▼	API key ▼		
<input type="radio"/>	OPTIONS	Mock	None	Not required		
<input type="radio"/>	POST	Lambda	None	Not required		
<input type="radio"/>	PUT	Lambda	None	Not required		

Methods (4)

Delete

Create method

	Method type ▲	Integration type ▼	Authorization ▼	API key ▼
<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

Resources

API actions ▼

Deploy API

Create resource

/

/my-resource

DELETE

GET

OPTIONS

POST

PUT

Resource details

Delete

Update documentation

Enable CORS

Path

Resource ID

/my-resource

v17yaf

Methods (5)

Delete

Create method

	Method type ▲	Integration type ▼	Authorization ▼	API key ▼
<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

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

to-do-stage

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

Deployment description

Cancel

Deploy

© 2024, Amazon Web Services, Inc. or its affiliates

Create Table

Share your feedback on Amazon DynamoDB

Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

Share feedback

DynamoDB > Tables > Create 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.

to-do-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.

name

String

1 to 255 characters and case sensitive.

CloudShell

Feedback

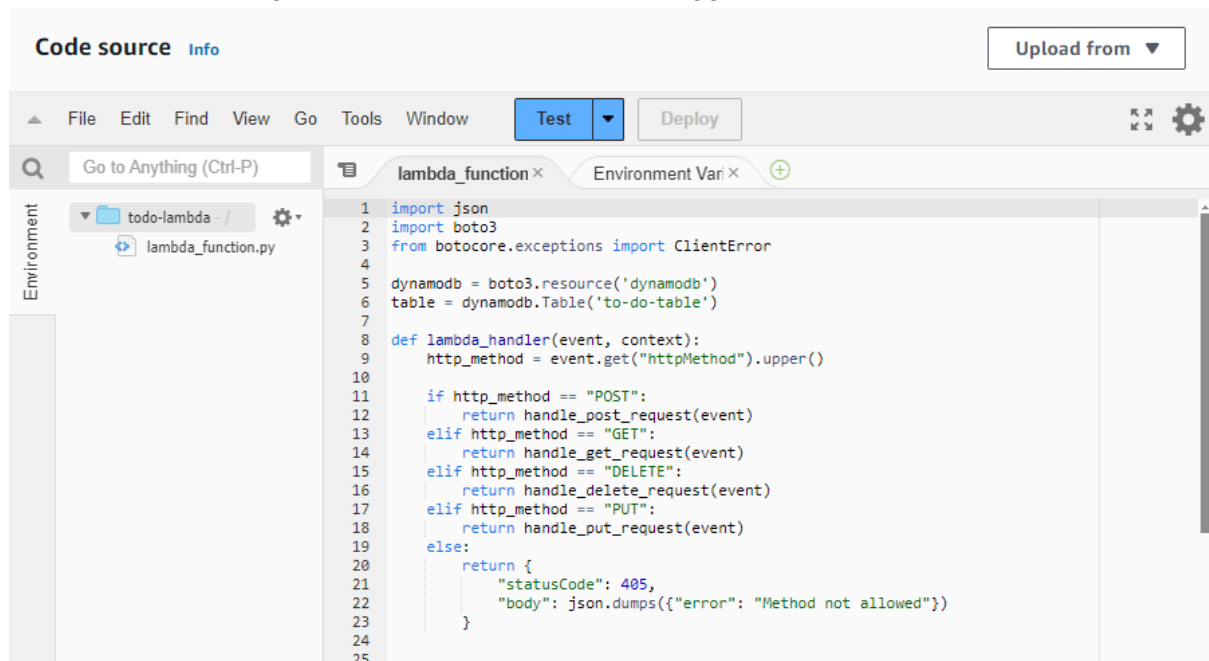
© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

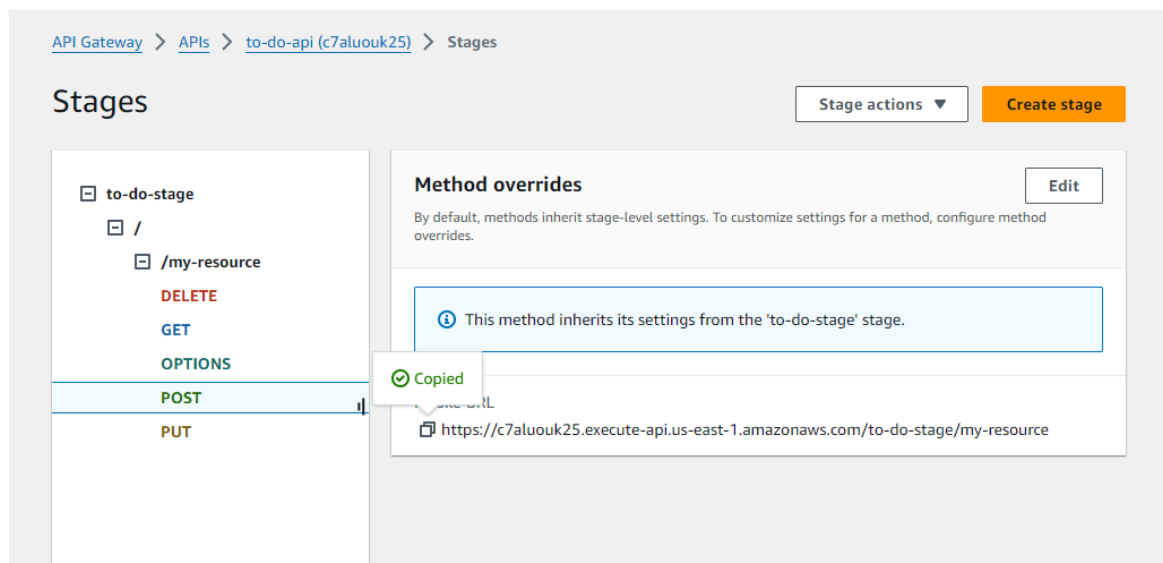
Cookie preferences

And add the following code to the lambda and add trigger

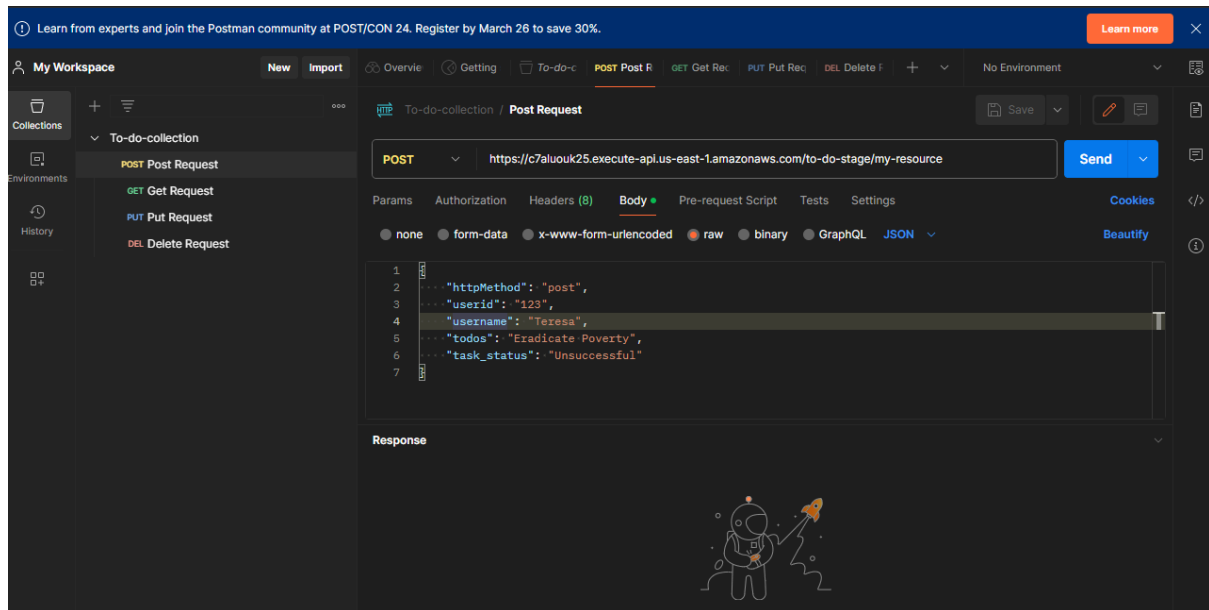


The screenshot shows the AWS Lambda console's code editor. The top bar includes 'Code source' and 'Info' tabs, and an 'Upload from' button. Below is a menu bar with 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test', and 'Deploy'. A search bar on the left says 'Go to Anything (Ctrl-P)'. The left sidebar shows the 'Environment' tab with a folder 'todo-lambda - /' and a file 'lambda_function.py'. The main editor area shows the code for 'lambda_function.py' with line numbers 1 through 25. The code imports 'json' and 'boto3', sets up a DynamoDB client and table, and defines a 'lambda_handler' function that routes HTTP requests to different handlers based on the 'httpMethod'.

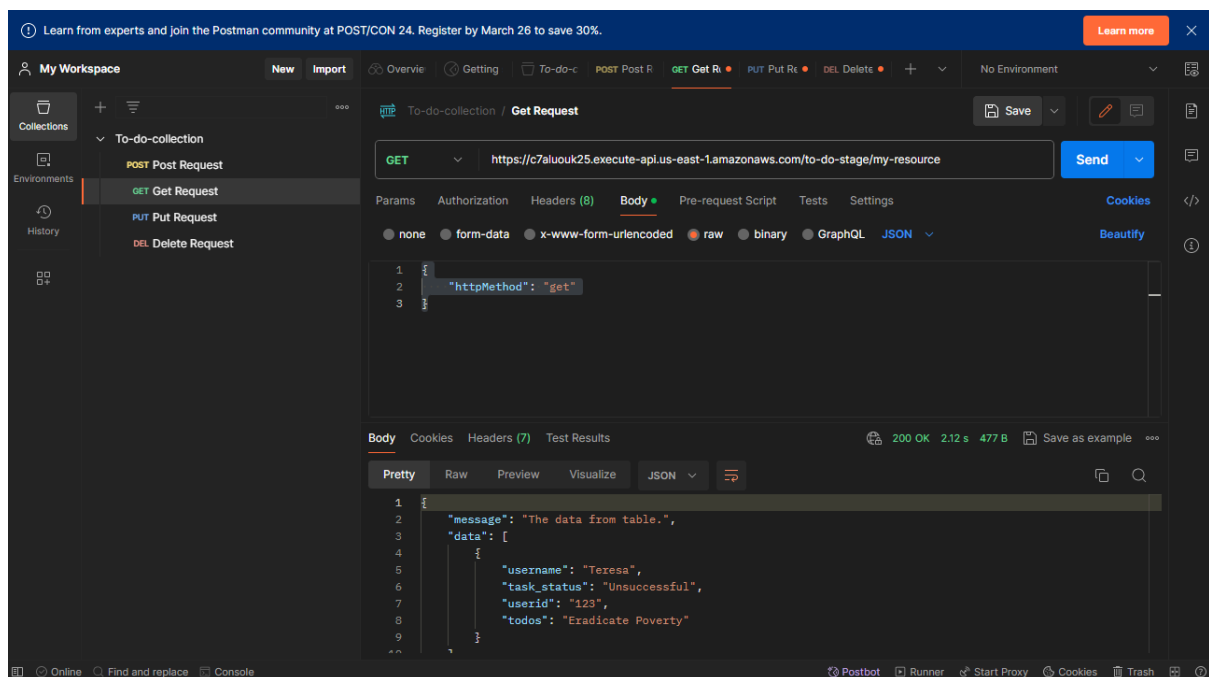
```
1 import json
2 import boto3
3 from botocore.exceptions import ClientError
4
5 dynamodb = boto3.resource('dynamodb')
6 table = dynamodb.Table('to-do-table')
7
8 def lambda_handler(event, context):
9     http_method = event.get("httpMethod").upper()
10
11     if http_method == "POST":
12         return handle_post_request(event)
13     elif http_method == "GET":
14         return handle_get_request(event)
15     elif http_method == "DELETE":
16         return handle_delete_request(event)
17     elif http_method == "PUT":
18         return handle_put_request(event)
19     else:
20         return {
21             "statusCode": 405,
22             "body": json.dumps({"error": "Method not allowed"})
23         }
24
25
```



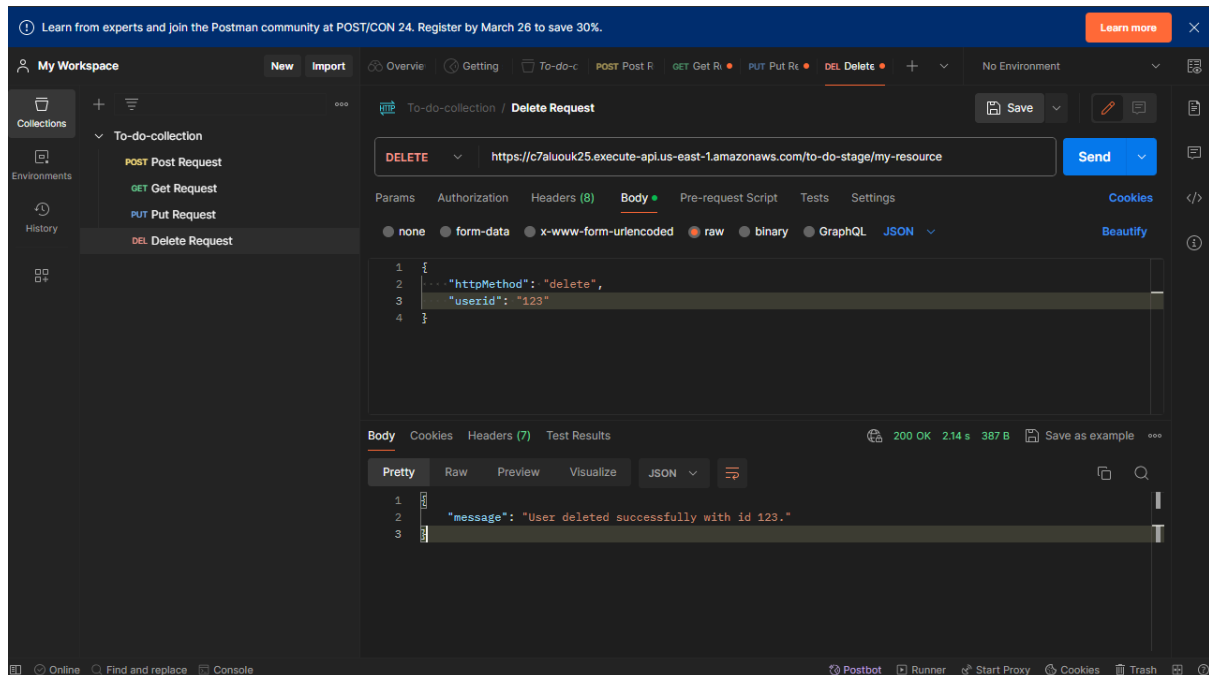
Hitting POST Request from Postman



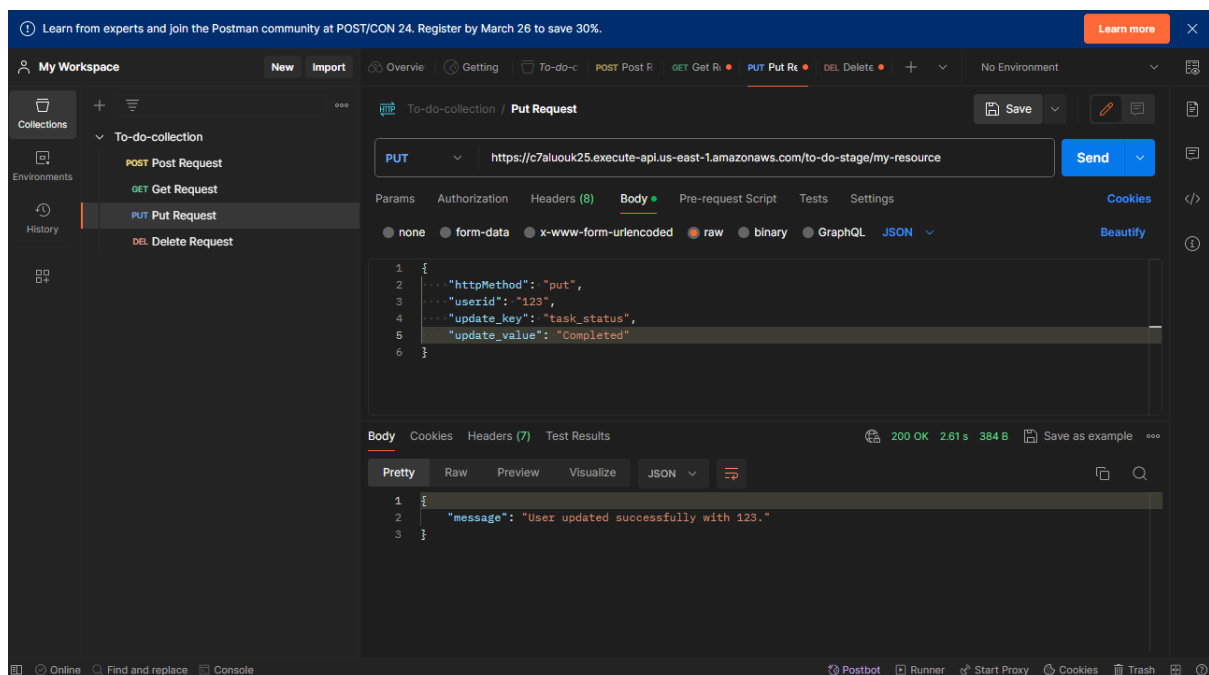
Hitting GET Request from Postman



Hitting DELETE Request



Hitting PUT Request



The manipulated table after:

test-table

to-do-table

Select a table or index

Table - to-do-table

Select attribute projection

All attributes

► Filters

Run

Reset

✔ Completed. Read capacity units consumed: 0.5

✕

Items returned (1)

↺

Actions ▼

Create item

< 1 >

⚙

🔗

<input type="checkbox"/>	userid (String) ▼	task_status ▼	todos ▼	username ▼
<input type="checkbox"/>	123	Completed	Eradicate P...	Teresa