

## **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.**

- Create 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 sample policy. Read example.

☐ **Use a blueprint**  
Build a custom 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.  

Python 3.12

**Architecture** info  
Choose the instruction set architecture you want for your function code.  

☒ x86\_64  
☐ arm64

**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

**Logging role**  
Choose the logging role that you've created to be used with this Lambda function. The role must have permissions to upload logs to Amazon CloudWatch Logs.  

LabRole

[View the Lambda role](#) in the IAM console.

---

**Advanced settings**

[Cancel](#) [Create function](#)

- Create DynamoDB table

[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.  
  
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.

- Create REST api

### Create REST API

**API details**

☒ **New API**  
Create a new REST API.

☐ **Clone existing API**  
Create a copy of an API in this AWS account.

☐ **Import API**  
Import an API from an OpenAPI definition.

☐ **Example API**  
Learn about API Gateway with an example API.

API name  
todo-api

Description - optional  
api for todo app

API endpoint type  
Regional

Cancel Create API

- Create Resource and create PUT, GET, POST, DELETE 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.

☒ **Lambda proxy integration**  
Send the request to a Lambda function.

Lambda function  
us-east-1

arn:aws:lambda:us-east-1:469425480758:function:ToDoApp

arn:aws:lambda:us-east-1:469425480758:function:RedshiftEventSubscription

arn:aws:lambda:us-east-1:469425480758:function:ToDoApp

arn:aws:lambda:us-east-1:469425480758:function:ModLabRole

arn:aws:lambda:us-east-1:469425480758:function:RoleCreationFunction

arn:aws:lambda:us-east-1:469425480758:function:RedshiftOverwatch

arn:aws:lambda:us-east-1:469425480758:function:MainMonitoringFunction

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

- Method list

API Gateway > APIs > Resources - toDo-api (tjxw2hvr51)

### Resources

Create resource

- /
- /toDoresource
  - DELETE**
  - GET
  - OPTIONS
  - POST
  - PUT

**/toDoresource - DELETE - Method execution**

Update documentation Delete

ARN: `arn:aws:execute-api:us-east-1:469425480758:tjxw2hvr51:/DELETE/toDoresource` Resource ID: `0991h`

Client → Method request → Integration request → Lambda integration on ...

← Method response ← Integration response

Method request Integration request Integration response Method response Test

**Method request settings** Edit

|                         |  |
|-------------------------|--|
| Authorization: NONE     | API key required: False                                |
| Request validator: None | SDK operation name: Generated based on method and path |

Request paths (0) < 1 >

| Name             | Caching |
|------------------|---------|
| No request paths |         |

- Enable CORS

API Gateway > APIs > Resources - toDo-api (tjxw2hvr51) > 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

**Access-Control-Allow-Methods**

☒ DELETE

☒ GET

☒ OPTIONS

☒ 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

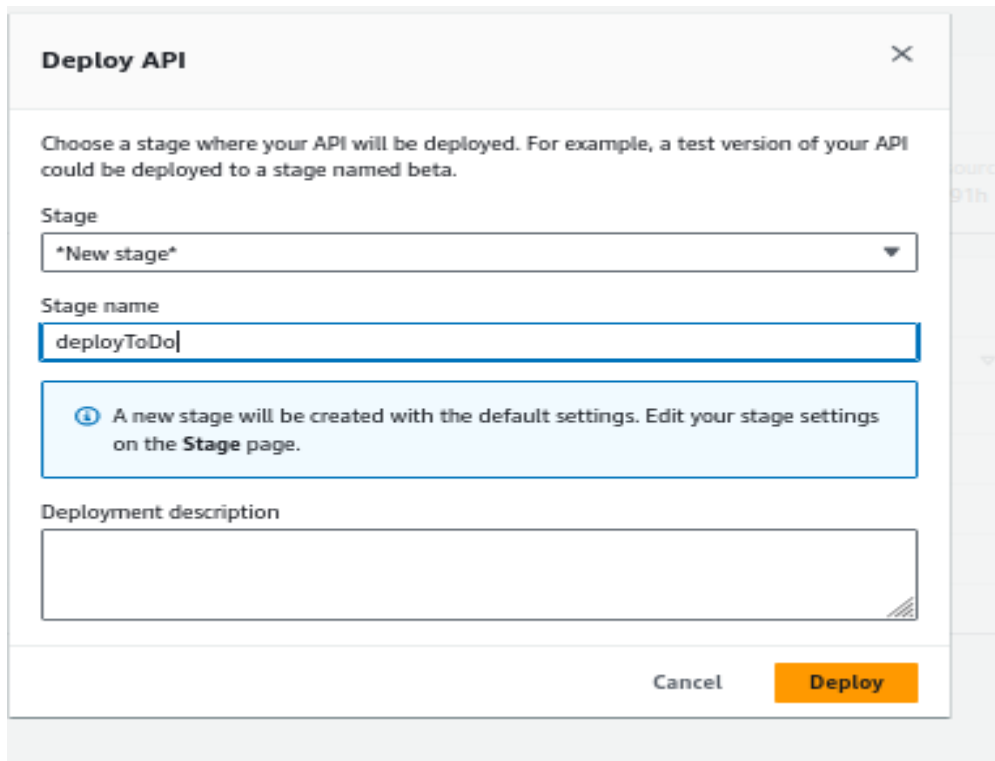
**Access-Control-Allow-Origin**  
Enter an origin that can access the resource. Use a wildcard "\*" to allow any origin to access the resource.

\*

Additional settings

Cancel Save

- Deploy API by creating new stage



**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\*

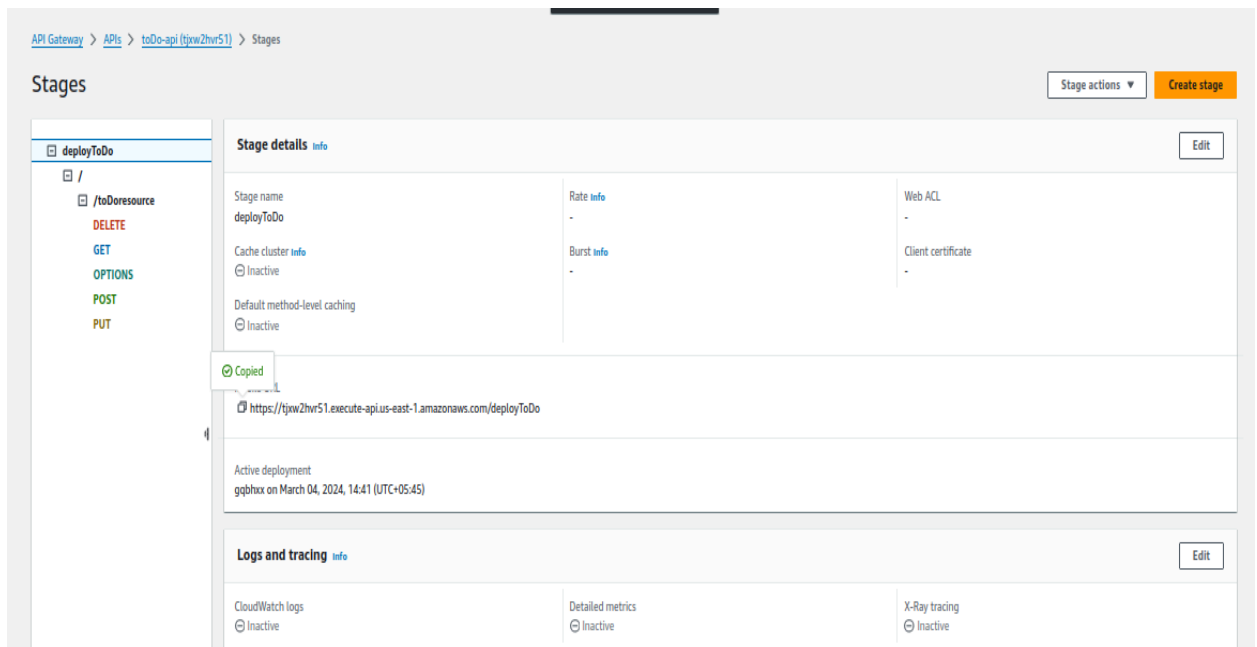
Stage name  
deployToDo

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

Deployment description

Cancel Deploy

- From resource copy api Invoke url for particular method



API Gateway > APIs > toDo-api (tjwz2hr51) > Stages

Stages

Stage actions Create stage

deployToDo

/

/toDoresource

DELETE

GET

OPTIONS

POST

PUT

**Stage details** info

Stage name  
deployToDo

Cache cluster info  
Inactive

Default method-level caching  
Inactive

Rate info  
-

Burst info  
-

Web ACL  
-

Client certificate  
-

Copied

https://tjwz2hr51.execute-api.us-east-1.amazonaws.com/deployToDo

Active deployment  
gqbhxx on March 04, 2024, 14:41 (UTC+05:45)

**Logs and tracing** info


CloudWatch logs  
Inactive

Detailed metrics  
Inactive

X-Ray tracing  
Inactive

- Add trigger

### 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

**Existing API**  
Attach an existing API.

**REST APIs**

- ToDo-api
- tjxw2hvr51

Invoke your Lambda function from this trigger.

Cancel Add

- Test POST method using Postman

OverviewGetting startedNew CollectionPOST New Request+No Environment

New Collection / New Request

Save

Send

POSThttps://g4q8lr1cpa.execute-api.us-east-1.amazonaws.com/ToDoDeployment/ToDoRes

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies</>

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSONBeautify

```
1 {
2   "httpMethod": "post",
3   "id": "AB2024",
4   "name": "Amrit",
5   "todo": "labs",
6   "todo_status": "partial"
7 }
```

BodyCookiesHeaders (8)Test Results200 OK2.34 s432 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "statusCode": 200,
3   "body": "\nSuccessfully saved to database!\n"
4 }
```

- Table after post

✓ Completed. Read capacity units consumed: 0.5

Items returned (1)

↻

Actions ▼

Create item

< 1 > ⚙️

|                          |                        |       |      |             |
|--------------------------|------------------------|-------|------|-------------|
| <input type="checkbox"/> | id (String)            | name  | todo | todo_status |
| <input type="checkbox"/> | <a href="#">AB2024</a> | Amrit | labs | partial     |

- Test GET method

Search Postman

Invite ⚙️ 🔔

Upgrade

Overview

Getting started

New Collection

POST New Request

GET get request

No Environment

New Collection / get request

Save

Send

GET

https://g4q8lr1cpa.execute-api.us-east-1.amazonaws.com/ToDodeployment/ToDoers

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {
2   ... "httpMethod": "get"
3 }
```

Body

Cookies

Headers (8)

Test Results

200 OK 691 ms 482 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

Copy

Search

```
1 {
2   "message": "Data from DynamoDB",
3   "data": [
4     {
5       "todo": "labs",
6       "id": "AB2024",
7       "todo_status": "partial",
8       "name": "Amrit"
9     }
10  ]
11 }
```

- Table Before DELETE request

| Items returned (2)       |                        |        |        |               |       | Actions ▼ | Create item |
|--------------------------|------------------------|--------|--------|---------------|-------|-----------|-------------|
|                          |                        |        |        |               | < 1 > |           |             |
| <input type="checkbox"/> | id (String) ▼          | name ▼ | todo ▼ | todo_status ▲ |       |           |             |
| <input type="checkbox"/> | <a href="#">AB2023</a> | Amrita | AWS    | Completed     |       |           |             |
| <input type="checkbox"/> | <a href="#">AB2024</a> | Amrit  | null   | partial       |       |           |             |

- Test DELETE request

Overview | Getting | GET post f | POST post | GET get re | New Co | **DEL delete** | + | No Environment

New Collection / delete request Save

**DELETE** | <https://g4q8lr1cpa.execute-api.us-east-1.amazonaws.com/ToDoDeployment/ToDoRes> Send

Params | Authorization | Headers (8) | **Body** | Pre-request Script | Tests | Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary ● GraphQL **JSON** Beautify

```
1 {
2   ... "httpMethod": "delete",
3   ... "id": "AB2024"
4 }
```

Body | Cookies | Headers (8) | Test Results 200 OK 660 ms 425 B Save as example

Pretty | Raw | Preview | Visualize | **JSON** 🔍

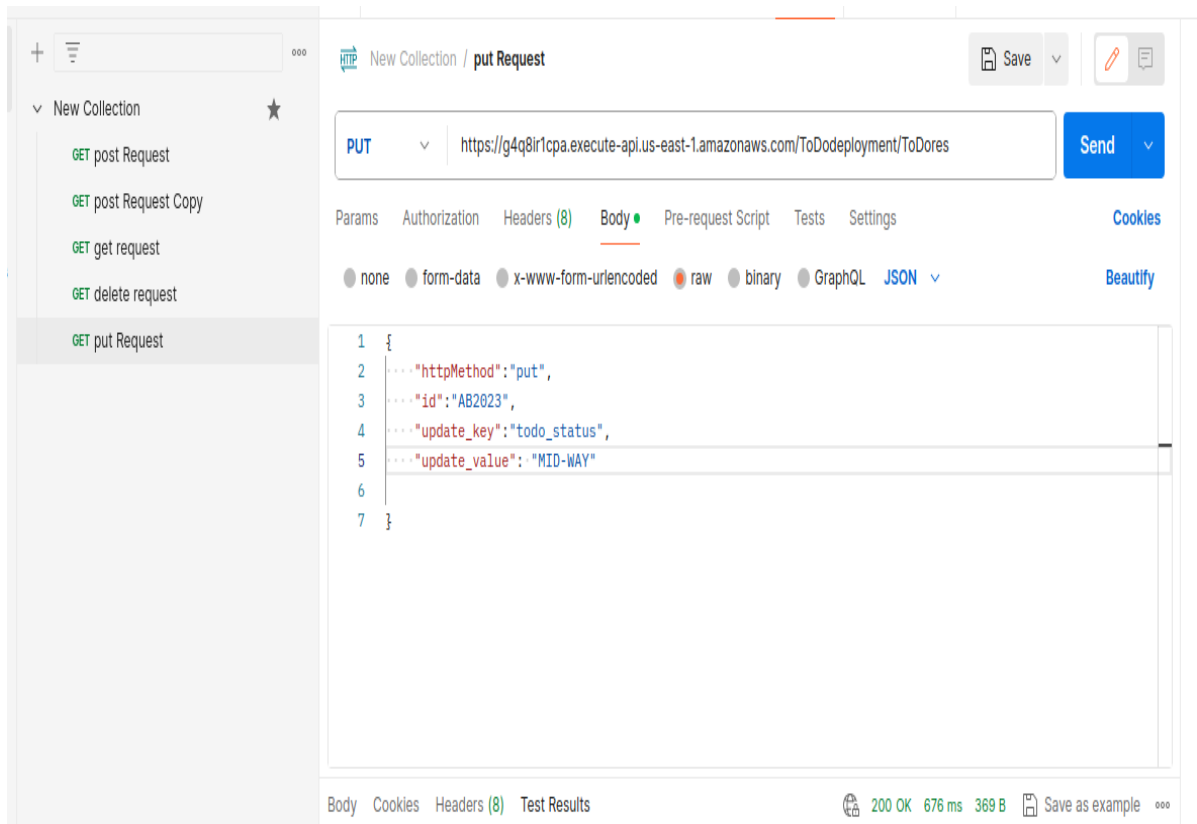
```
1 {
2   "message": "Data with id AB2024 is deleted from DynamoDB"
3 }
```

- Table after DELETE request

| Items returned (1)       |                        |        |        |               |       | Actions ▼ | Create item |
|--------------------------|------------------------|--------|--------|---------------|-------|-----------|-------------|
|                          |                        |        |        |               | < 1 > |           |             |
| <input type="checkbox"/> | id (String) ▼          | name ▼ | todo ▼ | todo_status ▲ |       |           |             |
| <input type="checkbox"/> | <a href="#">AB2023</a> | Amrita | AWS    | Completed     |       |           |             |



- Test PUT method



- After PUT method /Updated table

Items returned (1)

Actions ▾

Create item

< 1 >

| <input type="checkbox"/> | id (String) ▾          | name ▾ | todo ▾  | todo_status ▲ |
|--------------------------|------------------------|--------|---------|---------------|
| <input type="checkbox"/> | <a href="#">AB2023</a> | Amrita | workout | MID-WAY       |