

# 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.

1. First of all we go the api gateway and create an api to-do-application.

Successfully created REST API 'To Do Application (98r4241thf)'

API Gateway > APIs

APIs (3/3)

Find APIs

	Name ▲	Description ▼	ID ▼	Protocol ▼	API endpoint type	Created
<input type="radio"/>	<a href="#">api test</a>		m66dmztqxl	REST	Regional	2024-02-23
<input type="radio"/>	<a href="#">RestApiTest</a>	Test	ooy3nq7f8l	REST	Regional	2024-01-23
<input type="radio"/>	<a href="#">To Do Application</a>		98r4241thf	REST	Regional	2024-02-23

2. Create a new resource with following methods and using proxy resources on.

Successfully created REST API 'To Do Application (98r4241thf)'

[API Gateway](#) > [APIs](#) > [Resources - To Do Application \(98r4241thf\)](#) > [Create resource](#)

## Create resource

### Resource details

☒ Proxy resource [Info](#)  
Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

Resource path

Resource name

☒ CORS (Cross Origin Resource Sharing) [Info](#)

Create an OPTIONS method that allows all origins, all methods, and several common headers.

[Cancel](#) [Create resource](#)

3. After that we have to create a new table from dynamo db from the left control panel of the aws menus.

[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.

4. We can see the following table after the table is created.

Creating the ToDoAppTable table. It will be available for use shortly.

DynamoDB > Tables

Tables (3) Info

Find tables by table name

Any tag key

Any tag value

< 1 >

<input type="checkbox"/>	Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mode	Total size	Table class
<input type="checkbox"/>	<a href="#">serverless_table</a>	Active	date (S)	name (S)	0	Off	Provisioned (1)	Provisioned (1)	0 bytes	Standard
<input type="checkbox"/>	<a href="#">Tabletest_1</a>	Active	Partition Key (S)	Sort Key (S)	0	Off	Provisioned (1)	Provisioned (1)	0 bytes	Standard
<input type="checkbox"/>	<a href="#">ToDoAppTable</a>	Active	todoId (S)	-	0	Off	Provisioned (5)	Provisioned (5)	0 bytes	Standard

5. Then create a new lambda function for the to do app.

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

Lambda > Functions > ToDoApplicationFunction

ToDoApplicationFunction

Throttle

Copy ARN

Actions

Function overview Info

Export to Application Composer

Download

Diagram

Template

ToDoApplicationFunction

Layers (0)

+ Add trigger

+ Add destination

Description

-

Last modified

8 seconds ago

Function ARN

arn:aws:lambda:us-east-1:979147213970:function:ToDoApplicationFunction

Function URL

Info

-

Code

Test

Monitor

Configuration

Aliases

Versions

Code source Info

Upload from

- Then we have to create the 4 methods GET, PUT, POST, DELETE and it should be inside our todoApplication.


## 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 ▼

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

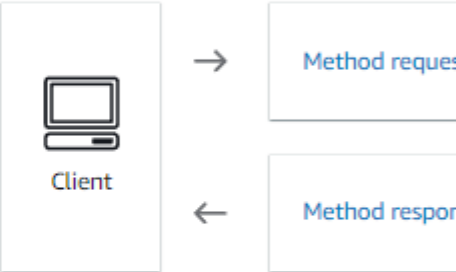
# Resources

Create resource

- /
- /to-do-app-resource
  - DELETE
  - GET
  - OPTIONS
  - POST
  - PUT

## /to-do-app-resource - GET - Met

ARN  
arn:aws:execute-api:us-east-1:9791472



Method request | Integration request

### Method request settings

Authorization  
NONE

Request validator  
None

Request paths (0)

## 7. 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

## 8. Create a new stage.

Successfully created deployment for To Do Application. This deployment is active for todoappstage.

API Gateway > APIs > To Do Application (98r4241thf) > Stages

### Stages

Stage actions ▼ Create stage

todoappstage
<div><div><div><b>Stage details</b> info</div><div><div><div>Stage name</div><div>todoappstage</div></div><div><div>Cache cluster info</div><div>⊖ Inactive</div></div><div><div>Default method-level caching</div><div>⊖ Inactive</div></div></div><div><div><div>Rate info</div><div>-</div></div><div><div>Burst info</div><div>-</div></div></div><div><div><div>Web ACL</div><div>-</div></div><div><div>Client certificate</div><div>-</div></div></div></div><div><div>Invoke URL</div><div>https://98r4241thf.execute-api.us-east-1.amazonaws.com/todoappstage</div></div><div><div>Active deployment</div><div>ofzmg3 on February 23, 2024, 18:13 (UTC+05:45)</div></div></div> <div><div><b>Logs and tracing</b> info</div><div>Edit</div></div>

## 8. Deploy the 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

todoappstage ▼

Deployment description

Cancel Deploy



## 9.Post method test

### ToDoAppTable

#### ▼ Scan or query items

☒ Scan

☐ Query

Select a table or index

Table - ToDoAppTable ▼

Select attribute projection

All attributes

► Filters

Run

Reset

✔ Completed. Read capacity units consumed: 0.5

#### Items returned (1)

<input type="checkbox"/>	todoid (String) ▼	state ▼	task
<input type="checkbox"/>	<a href="#">1</a>	Pending	Serverless lab

10.API TEST

/to-do-app-resource - POST method test results

Request	Latency	Status
/to-do-app-resource	1223	200

Response body

{  
 "statusCode": 200,  
 "message": "New task added"  
}

Response headers

{  
 "Access-Control-Allow-Origin": "",  
 "Content-Type": "application/json",  
 "X-Amzn-Trace-Id": "Root=1-65d89ea1-3f289fbcaa7e226ebc221d8d;Parent=795dc6bb7e09f1bc;Sampled=0;lineage=b62a6d32:0"  
}

Log

Execution log for request 9fde651c-5ac1-4c72-88f5-bfd6fb019c16  
Fri Feb 23 13:33:21 UTC 2024 : Starting execution for request: 9fde651c-5ac1-4c72-88f5-bfd6fb019c16  
Fri Feb 23 13:33:21 UTC 2024 : HTTP Method: POST, Resource Path: /to-do-app-resource  
Fri Feb 23 13:33:21 UTC 2024 : Method request path: {}  
Fri Feb 23 13:33:21 UTC 2024 : Method request query string: {}  
Fri Feb 23 13:33:21 UTC 2024 : Method request headers: {}  
Fri Feb 23 13:33:21 UTC 2024 : Method request body before transformations: {  
 "httpMethod": "POST",  
 "todoid": "1",  
 "task": "Serverless lab",  
 "state": "Pending"  
}  
Fri Feb 23 13:33:21 UTC 2024 : Endpoint request URI: https://lambda.us-east-1.amazonaws.com/2015-03-31/functions/arn:aws:lambda:us-east-1:979147213970:function:toDoApplicationFunction/invocations  
Fri Feb 23 13:33:21 UTC 2024 : Endpoint request headers: {X-Amz-Date=20240223T133321Z, x-amzn-apigateway-api-id=98r4241thf, Accept=application/json, User-Agent=AmazonAPIGateway\_98r4241thf, Host=lambda.us-east-1.amazonaws.com, X-Amz-Content-Sha256=b488fe751a232b6238eb812e6184528f2d3beebb661a33b96f61f98e11c2029f, X-Amzn-Trace-Id=Root=1-65d89ea1-3f289fbcaa7e226ebc221d8d, x-amzn-lambda-  
...

## Test method

Make a test call to your method. When you make a test call, API Gateway skips authorization and directly invokes your method.

### Query strings

`param1=value1&param2=value2`

### Headers

Enter a header name and value separated by a colon (:). Use a new line for each header.

`header1:value1`  
`header2:value2`

### Client certificate

No client certificates have been generated.

### Request body

```
1 {  
2   "httpMethod": "POST",  
3   "todoId": "1",  
4   "task": "Serverless lab",  
5   "state": "Pending"  
6 }
```

## 11. Get request test

HTTP <https://98r4241thf.execute-api.us-east-1.amazonaws.com/todoappstage/to-do-app-resource> Save

GET <https://98r4241thf.execute-api.us-east-1.amazonaws.com/todoappstage/to-do-app-resource> Send

Body ...

raw JSON Beautify

```
1 {
2   "httpMethod": "GET"
3 }
```

Body ... 200 OK 1581 ms 476 B Save as example ...

Pretty Raw Preview Visualize JSON

```
1 {
2   "statusCode": 200,
3   "message": "Success",
4   "data": [
5     {
6       "todoid": "1",
7       "task": "Serverless lab",
8       "state": "Pending"
9     }
10  ]
11 }
```

## 12. Put test

### Request body

1	
2	{
3	"httpMethod": "PUT",
4	"todoid": "2",
5	"update_key": "task",
6	"update_val": "state"
7	}

Test

/to-do-app-resource - PUT method test results		
Request	Latency	Status
/to-do-app-resource	1152	200
Response body		
{ "statusCode": 200, "message": "Task is updated successfully" }		
Response headers		
{ "Access-Control-Allow-Origin": "", "Content-Type": "application/json", "X-Amzn-Trace-Id": "Root=1-65d8a1a0-d8780f334680500082e0108f;Parent=12c17aa0a4e445c0;Sampled=0;lineage=b62a6d32:0" }		
Log		
Execution log for request d8a5b691-9a89-4d03-b376-fd7320b19e80		
Fri Feb 23 13:46:08 UTC 2024 : Starting execution for request: d8a5b691-9a89-4d03-b376-fd7320b19e80		
Fri Feb 23 13:46:08 UTC 2024 : HTTP Method: PUT, Resource Path: /to-do-app-resource		
Fri Feb 23 13:46:08 UTC 2024 : Method request path: {}		
Fri Feb 23 13:46:08 UTC 2024 : Method request query string: {}		
Fri Feb 23 13:46:08 UTC 2024 : Method request headers: {}		
Fri Feb 23 13:46:08 UTC 2024 : Method request body before transformations:		
{ "httpMethod": "PUT", "todoid": "2", "update_key": "task", "update_val": "state" }		
Fri Feb 23 13:46:08 UTC 2024 : Endpoint request URI: https://lambda.us-east-1.amazonaws.com/2015-03-31/functions/arn:aws:lambda:us-east-1:979147213970:function:todoApplicationFunction/invocations		
Fri Feb 23 13:46:08 UTC 2024 : Endpoint request headers: {X-Amz-Date=20240223T134608Z, x-amzn-apigateway-api-id=98r4241thf, Accept=application/json, User-		

Items returned (2)				<input type="button" value="Refresh"/> <input type="button" value="Actions"/> <input type="button" value="Create item"/>
<input type="checkbox"/>	todooid ( <i>String</i> )	state	task	
<input type="checkbox"/>	<a href="#">2</a>		state	
<input type="checkbox"/>	<a href="#">1</a>	Pending	Serverless lab	

## 14. Delete method test

/

/to-do-app-resource

DELETE

GET

OPTIONS

POST

PUT

Client certificate

No client certificates have been generated.

Request body

1

2

3

4

5

{

"httpMethod": "DELETE",

"todoId": "1"

}

Test

<div><div></div><div>/to-do-app-resource - DELETE method test results</div></div>		
Request	Latency	Status
/to-do-app-resource	511	200
Response body		
{ "statusCode": 200, "message": "Data is deleted successfully" }		
Response headers		
{ "Access-Control-Allow-Origin": "", "Content-Type": "application/json", "X-Amzn-Trace-Id": "Root=1-65d8a225-ce7a9d24a2e529c631914d77;Parent=049184920db7cb70;Sampled=0;lineage=b62a6d32:0" }		
Log		
Execution log for request 113e18e0-92f4-4b33-ae6a-c6135dd05f37		
Fri Feb 23 13:48:21 UTC 2024 : Starting execution for request: 113e18e0-92f4-4b33-ae6a-c6135dd05f37		
Fri Feb 23 13:48:21 UTC 2024 : HTTP Method: DELETE, Resource Path: /to-do-app-resource		
Fri Feb 23 13:48:21 UTC 2024 : Method request path: {}		
Fri Feb 23 13:48:21 UTC 2024 : Method request query string: {}		
Fri Feb 23 13:48:21 UTC 2024 : Method request headers: {}		
Fri Feb 23 13:48:21 UTC 2024 : Method request body before transformations:		
{ "httpMethod": "DELETE", "todoId": "1" }		
Fri Feb 23 13:48:21 UTC 2024 : Endpoint request URI: https://lambda.us-east-1.amazonaws.com/2015-03-31/functions/arn:aws:lambda:us-east-1:979147213970:function:toDoApplicationFunction/invocations		
Fri Feb 23 13:48:21 UTC 2024 : Endpoint request headers: {X-Amz-Date=20240223T134821Z, x-amzn-apigateway-api-id=98r4241thf, Accept=application/json, User-Agent=AmazonAPIGateway_98r4241thf, Host=lambda.us-east-1.amazonaws.com, X-Amz-Content-Sha256=a7693e16dc478fc2f0e0132c43532f370b495bb0cdc51223d756d163f6c984d2, X-Amzn-Trace-Id=Root=1-65d8a225-ce7a9d24a2e529c631914d77, x-amzn-lambda-		

15. Data deleted successfully.

Items returned (1)			
<input type="checkbox"/>	todoId (String)	▼	task
<input type="checkbox"/>	<u>2</u>		state