

AWS Lambda with a simple HTTP proxy



Setting up API Gateway

Example code from - <https://github.com/Aishwarya26I/vue-emojis>

Navigate to your Lambda function

The screenshot displays the AWS Lambda console interface for a function named 'vueEmojiTest'. At the top, there are tabs for 'Configuration' (selected) and 'Monitoring'. To the right of the tabs are buttons for 'Throttle', 'Qualifiers', 'Actions', a dropdown for 'Select a test event', and 'Test' and 'Save' buttons. Below the tabs, the 'Designer' section is visible. On the left side of the Designer, there is a key icon and a button labeled '+ Add trigger'. A red box labeled 'Click Here' with an arrow points to the '+ Add trigger' button. To the right of the key icon, there is a box for the function 'vueEmojiTest' with a 'Saved' status and a 'Layers' section showing '(0)'. At the bottom right, there is a dashed box containing the text 'Resources that the function's role has access to appear here'.

vueEmojiTest

Throttle Qualifiers Actions Select a test event Test Save

Configuration Monitoring

▼ Designer

Click Here

+ Add trigger

vueEmojiTest
Saved

Layers (0)


Resources that the function's role has access to appear here

Add trigger

Lambda > Add trigger

Add trigger

Trigger configuration

 API Gateway
api application-services aws serverless

We'll set up an API Gateway endpoint with a [proxy integration type](#) (learn more about the [input](#) and [output](#) format). Any method (GET, POST, etc.) will trigger your integration. To set up more advanced method mappings or subpath routes, visit the [Amazon API Gateway console](#).

API
Pick an existing API, or create a new one.

Lambda will add the necessary permissions for Amazon API Gateway to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel Add

Select API Gateway as the Trigger

1. Click on Amazon API Gateway Console for advanced settings.

API Gateway console

The screenshot displays the AWS API Gateway console interface. At the top, the AWS logo is on the left, followed by 'Services' and 'Resource Groups' dropdown menus. On the right, there's a notification bell, a user profile 'vocstartsoft/user257930=aish...', a region selector 'N. Virginia', and a 'Support' link. Below the top bar, the 'Amazon API Gateway' header is on the left, and 'APIs' is on the right. A 'Show all hints' link and a help icon are also present. The main content area is titled 'APIs' and lists several APIs: 'analysisFunction-API', 'checkUrlTestcases-API', 'cloud9-infBreastCancer-infBr...', 'cloud9-infCheck-infCheck-B0...', 'cloud9-infDiabetes-infDiabete...', 'cloud9-infHousing-infHousing...', 'cloud9-lambdaTestSklearn-la...', and 'cloud9-pythonTests'. A red circle highlights the '+ Create API' button, and a red box labeled 'Create API' points to it. Below the list, three API cards are shown, each with details like 'Created on', 'Created by', 'Protocol', 'Endpoint Configuration', 'Endpoint Type', and 'Regional' status, along with a 'Configure Tags' button.

aws Services Resource Groups

Amazon API Gateway APIs

APIs

analysisFunction-API

checkUrlTestcases-API

cloud9-infBreastCancer-infBr...

cloud9-infCheck-infCheck-B0...

cloud9-infDiabetes-infDiabete...

cloud9-infHousing-infHousing...

cloud9-lambdaTestSklearn-la...

cloud9-pythonTests

+ Create API

Create API

analysisFunction-API

Created on 6/18/2019

Created by AWS Lambda

Protocol: HTTP

Endpoint Configuration

Endpoint Type ⓘ

Regional

Configure Tags

checkUrlTestcases-API

Created on 4/30/2019

Created by AWS Lambda

Protocol: HTTP

Endpoint Configuration

Endpoint Type ⓘ

Regional

Configure Tags

cloud9-infBreastCancer-infBreastCa

Created on 6/6/2019

Created by AWS Lambda

Protocol: HTTP

Endpoint Configuration

Endpoint Type ⓘ

Regional

Configure Tags

API Gateway console

APIs > Create [Show all hints](#) ?

Choose the protocol

Select whether you would like to create a REST API or a WebSocket API.

☒ REST ☐ WebSocket

Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

☒ New API ☐ Clone from existing API ☐ Import from Swagger or Open API 3 ☐ Example API

Settings

Choose a friendly name and description for your API.

API name*

Description

Endpoint Type ⓘ

* Required

[Create API](#)

1. Fill out your API name
<Lambda-Function-Name>-API
Example -
vueEmojiTest-API
2. Click on Create API

API Gateway console

vueEmojiTest-API (8sd0tcuxvh) > Resources > / (3w353vkwI5) [Show all hints](#) ?

Resources **Actions** / Methods

/

No methods defined for the resource.

Your API has been created. Let us setup a proxy

API Gateway console

vueEmojiTest-API (8sd0tcuxvh) > Resources > / (3w353vkw15)

[Show all hints](#)



Resources

Actions ▾



/Methods



RESOURCE ACTIONS

Create Method

Create Resource

Enable CORS

Edit Resource Documentation

API ACTIONS

Deploy API

Import API


Edit API Documentation


Delete API


No methods defined for the resource.



1. Click on Actions
2. Create Resource

API Gateway console

vueEmojiTest-API (8sd0tcuxvh) > Resources > / (3w353vkwI5) > Create [Show all hints](#) 

Resources **Actions**  **New Child Resource**


Use this page to create a new child resource for your resource. 

Configure as  proxy resource 

Resource Name*

Resource Path*

You can add path parameters using brackets. For example, the resource path **{username}** represents a path parameter called 'username'.
Configuring `/ {proxy+}` as a proxy resource catches all requests to its sub-resources. For example, it works for a GET request to `/foo`. To handle requests to `/`, add a new ANY method on the `/` resource.

Enable API Gateway CORS ☐ 



* Required

[Cancel](#) [Create Resource](#)

1. Click on configure as proxy resource
2. Create resource

API Gateway console

vueEmojiTest-API (8sd0tcuxvh) > Resources > /{proxy+} (vs0b2y) > ANY [Show all hints](#) ?

Resources **Actions**  /{proxy+} - ANY - Setup 

▼ /
▼ /{proxy+}
 ANY

API Gateway will configure your ANY method as a proxy integration. Proxy integrations can communicate with HTTP endpoints or Lambda functions. API Gateway sends the entire request to HTTP endpoints, including resource path, headers, query string parameters, and body. For Lambda integrations, API Gateway applies a default mapping to send all of the request information and responses follow a default interface. To learn more read our [documentation](#)

Integration type ☒ Lambda Function Proxy ⓘ
☐ HTTP Proxy ⓘ
☐ VPC Link ⓘ

Lambda Region us-east-1 ▼

Lambda Function
vueEmojiTest

Use Default Timeout ☐ ⓘ

Custom Timeout 400

Save

1. Enter the name of your Lambda Function
2. Set custom timeout to 400ms
3. Save

API Gateway console

Add Permission to Lambda Function

You are about to give API Gateway permission to invoke your Lambda function:
arn:aws:lambda:us-east-1:216730220204:function:vueEmojiTest

Cancel

OK

HTTP Proxy

VPC Link

Lambda Region

us-east-1

Lambda Function

vueEmojiTest

Use Default Timeout

☐

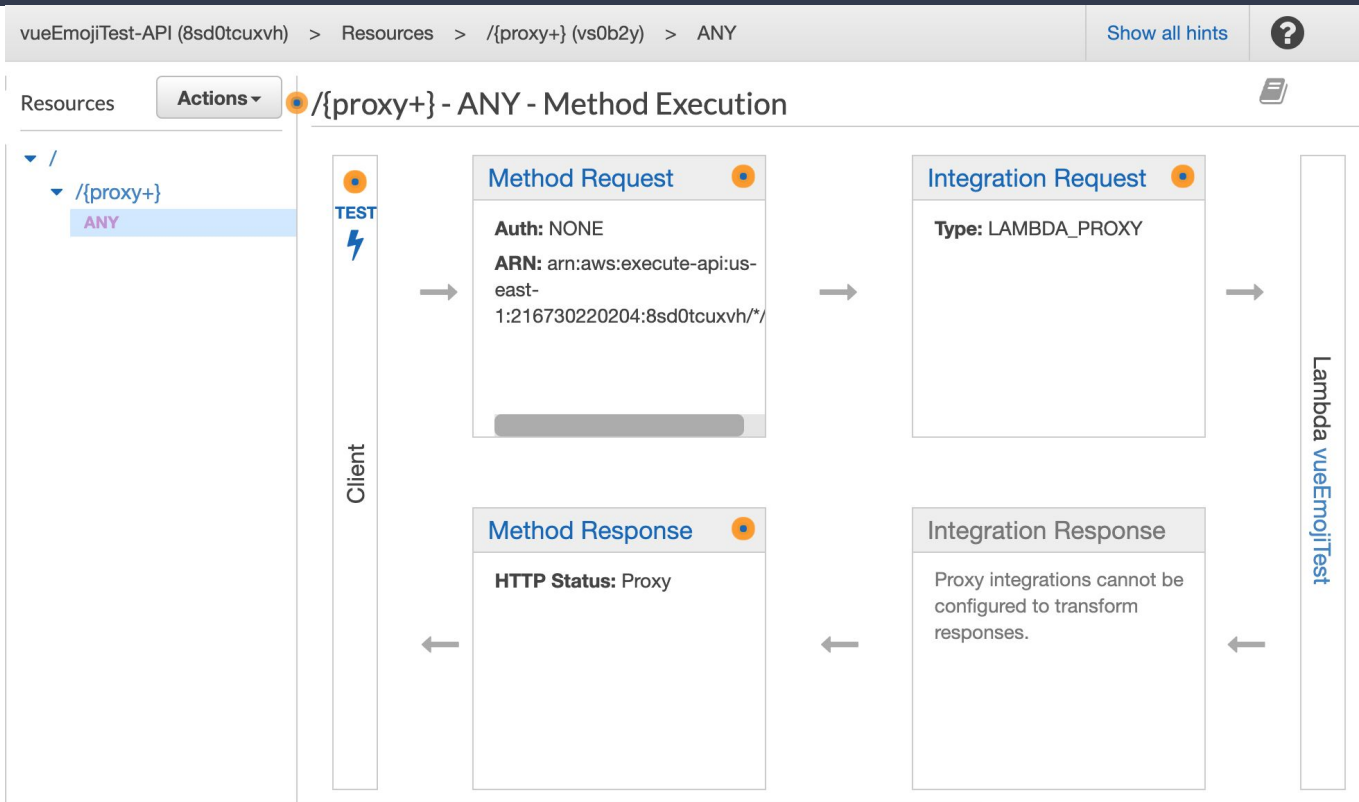
Custom Timeout

400

Save

Confirm

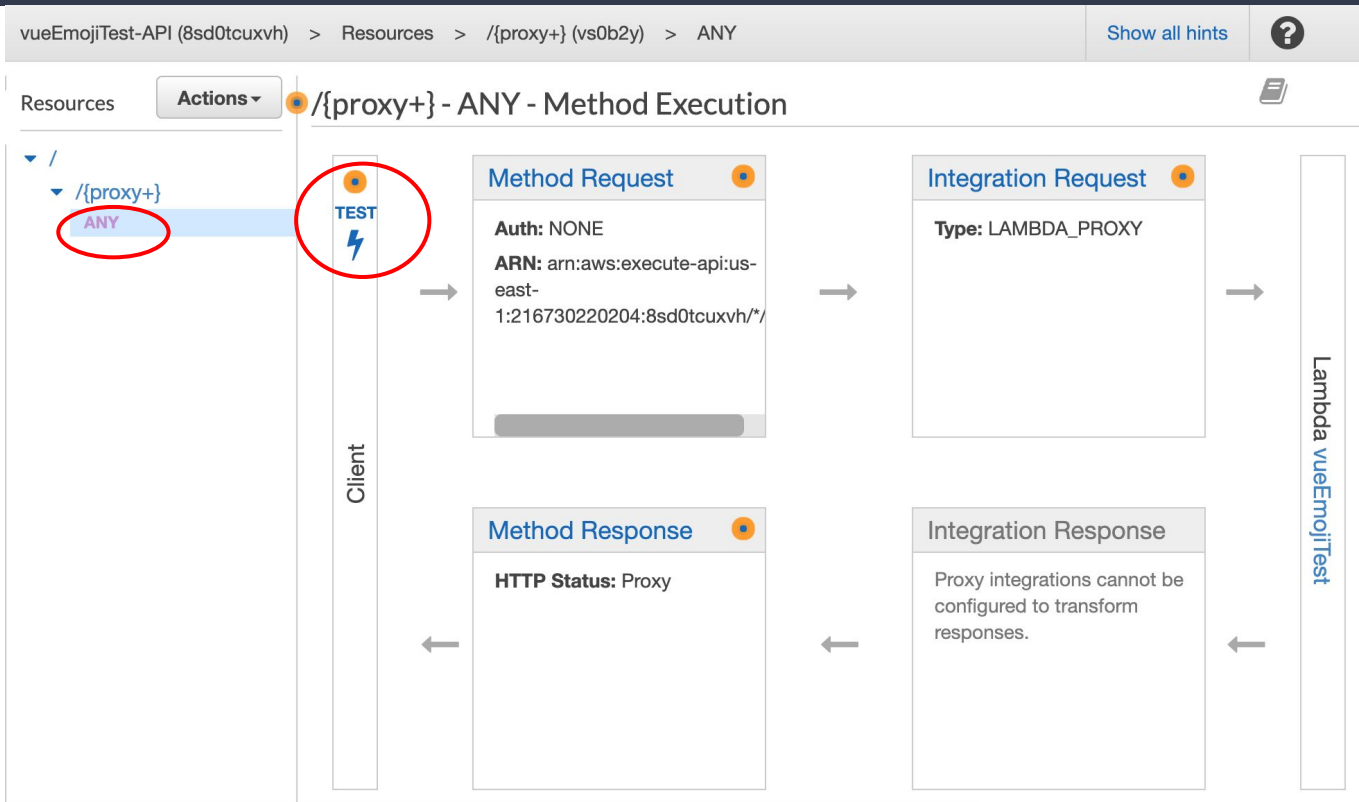
API Gateway console



Your proxy has been set up.

1. Let us test our API
2. Then deploy

API Gateway console



1. Click on ANY under {proxy+}
2. Click on test to test our proxy

Resources

Actions ▾

← Method Execution /{proxy+} - ANY - Method Test

▼ /
▼ /{proxy+}
ANY

Make a test call to your method with the provided input

Method

GET

Path

{proxy}

/main.js

Query Strings

{proxy}

param1=value1¶m2=value2

Headers

{proxy}

Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json.

Stage Variables

No [stage variables](#) exist for this method.

Client Certificate

No client certificates have been generated.

Request Body

Request Body is not supported for GET methods.

⚡ Test

1. Select the GET method
2. Select the route /main.js
3. Click on test



Resources

Actions

Method Execution /{proxy+} - ANY - Method Test

/

/{proxy+}

ANY

Make a test call to your method with the provided input

Method

GET

Path

{proxy}

/main.js

Query Strings

{proxy}

param1=value1&param2=value2

Headers

{proxy}

Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg.
Accept:application/json.

Stage Variables

No stage variables exist for this method.

Client Certificate

No client certificates have been generated.

Request Body

Request Body is not supported for GET methods.

Test

Request: /main.js

Status: 200

Latency: 63 ms

Response Body

```
var app = new Vue({
  el: "#app",
  data: {
    problems: [1, 2, 3, 4],
    currentProblem: 2,
    directions: {
      1: "Use the 🤔 emoji in place of",
      2: "Use the 😊 emoji in place of",
      3: "Use the 🍪 emoji in place of",
      4: "Use the 🙏 emoji to replace",
    },
    givens: {
      1: "I am crying laughing.",
      2: "I am so sad.",
      3: "I love cookies.",
      4: "Hopefully"
    },
    solutions: {
      1: "I am 🤔 laughing.",
      2: "I am so 😊.",
      3: "I 🍪 cookies.",
      4: "🙏"
    }
  },
  methods: {
    isComplete: function(problem) {
      return this.givens[problem] === this.solutions[problem];
    }
  }
});
```

Response Headers

```
{"X-Amzn-Trace-Id":"Root=1-5d807826-9f80c78eedf173b76c770c68;Sampled=0","Content-Type":"text/javascript;charset=UTF-8"}
```

1. You can view your test results
2. Confirm with expected behaviour
3. Here the response body contains the contents of main.js. It is as expected
4. Let us deploy our API

API Gateway console

vueEmojiTest-API (8sd0tcuxvh) > Resources > / (3w353vkw15)

[Show all hints](#)



Resources

Actions ▾

/ Methods



RESOURCE ACTIONS

- Create Method
- Create Resource
- Enable CORS
- Edit Resource Documentation

API ACTIONS

- Deploy API
- Import API
- Edit API Documentation
- Delete API

No methods defined for the resource

1. Navigate to the root
2. Click on actions
3. Deploy API

API Gateway console

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.

Deployment stage

[New Stage]

Stage name*

default

Stage description

Deployment description

Cancel

Deploy

1. Select [New Stage] for deployment
2. Call it "default"
3. Click on Deploy

API Gateway console

The screenshot displays the AWS API Gateway console. On the left is a list of APIs, including 'analysisFunction-API', 'checkUrlTestcases-API', 'cloud9-infBreastCancer-infBr...', 'cloud9-infCheck-infCheck-B0...', 'cloud9-infDiabetes-infDiabete...', 'cloud9-infHousing-infHousing...', 'cloud9-lambdaTestSklern-la...', 'cloud9-pythonTests', 'colabExecuter-API', 'diffCodeAnalysis-API', 'final-node-test-API', 'flask-test-API', and 'greetings-API'. The 'default' API is selected. A red-bordered error box at the top states: 'User does not have ListWebACLs and AssociateWebACL permissions for Web Application Firewall (WAF Regional). Stage settings except for WAF can still be changed.' The main content area shows the 'Settings' tab for the 'default' stage. It includes an 'Invoke URL' field with the value 'https://8sd0tcuxvh.execute-api.us-east-1.amazonaws.com/default'. Below this are tabs for 'Settings', 'Logs/Tracing', 'Documentation History', and 'Deployment History'. The 'Settings' tab is active, showing 'Cache Settings' and 'Default Method Throttling'. The 'Default Method Throttling' section indicates a rate of 5000 requests. A red-bordered text box in the center-right of the console contains the following text: 'If you are an AWS Educate user, you might see an error. This is due to restricted access for educate users. Default settings are okay. Please proceed.' At the bottom, there is a 'Rate' input field set to '10000' requests per second.

Amazon A

APIs

analysisFunction-API

checkUrlTestcases-API

cloud9-infBreastCancer-infBr...

cloud9-infCheck-infCheck-B0...

cloud9-infDiabetes-infDiabete...

cloud9-infHousing-infHousing...

cloud9-lambdaTestSklern-la...

cloud9-pythonTests

colabExecuter-API

diffCodeAnalysis-API

final-node-test-API

flask-test-API

greetings-API

default

Invoke URL: <https://8sd0tcuxvh.execute-api.us-east-1.amazonaws.com/default>

Settings Logs/Tracing Documentation History Deployment History

Cache Settings

Default Method Throttling

Choose the default throttle rate and burst settings. 5000 requests. [Read more](#)

Rate 10000 requests per second

If you are an AWS Educate user, you might see an error.

This is due to restricted access for educate users.

Default settings are okay. Please proceed.

Configure Tags

Lambda console

vueEmojiTest

Throttle Qualifiers Actions Select a test event Test Save

vueEmojiTest
✓ Saved

Layers (0)

API Gateway ×

+ Add trigger

API Gateway

vueEmojiTest-API
arn:aws:execute-api:us-east-1:216730220204:8sd0tcuxvh/*/*/*

▶ API endpoint: <https://8sd0tcuxvh.execute-api.us-east-1.amazonaws.com/default/{proxy+}> Authorization: NONE Method: ANY

1. Navigate to your lambda function
2. Click on API Gateway
3. Click on the API to view your contents

Lambda console

vueEmojiTest

Throttle Qualifiers Actions Select a test event Test Save

vueEmojiTest
✓ Saved

Layers (0)

API Gateway ×

+ Add trigger

API Gateway

vueEmojiTest-API
arn:aws:execute-api:us-east-1:216730220204:8sd0tcuxvh/*/*/*

▶ API endpoint: <https://8sd0tcuxvh.execute-api.us-east-1.amazonaws.com/default/{proxy+}> Authorization: NONE Method: ANY

1. Navigate to your lambda function
2. Click on API Gateway
3. Click on the API to view your contents

Lambda API

1. API - <https://8sd0tcuxvh.execute-api.us-east-1.amazonaws.com/default/index.html>

Tests for proxies -

1. <https://8sd0tcuxvh.execute-api.us-east-1.amazonaws.com/default/main.js>

References -

1. Github for example code - <https://github.com/Aishwarya26l/vue-emojis>
2. API Gateway proxy documentation - <https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-create-api-as-simple-proxy-for-http.html>