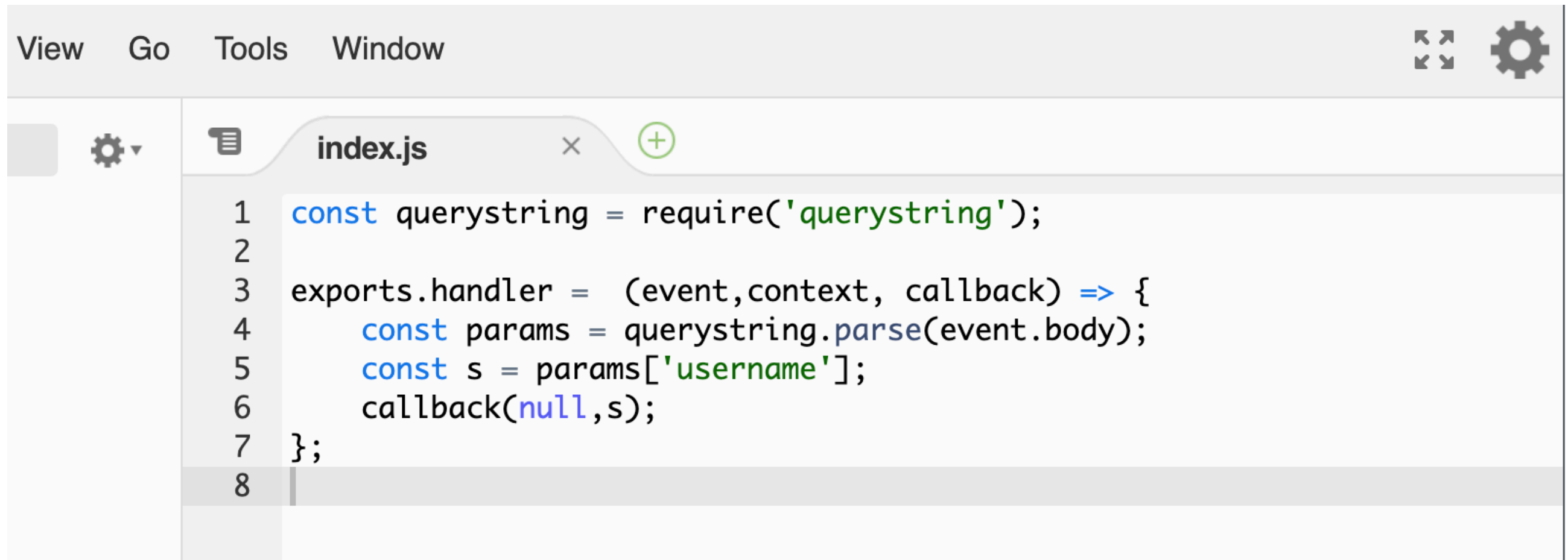


Lambda - POST

Create a Lambda Function



The image shows a screenshot of a code editor interface. At the top, there is a menu bar with 'View', 'Go', 'Tools', and 'Window'. On the right side of the menu bar are icons for a window and a gear. Below the menu bar, there is a tab labeled 'index.js' with a close button and a plus sign. The code editor displays the following JavaScript code:

```
1  const querystring = require('querystring');
2
3  exports.handler = (event, context, callback) => {
4      const params = querystring.parse(event.body);
5      const s = params['username'];
6      callback(null, s);
7  };
8
```

Configuring API Gateway

☒ 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

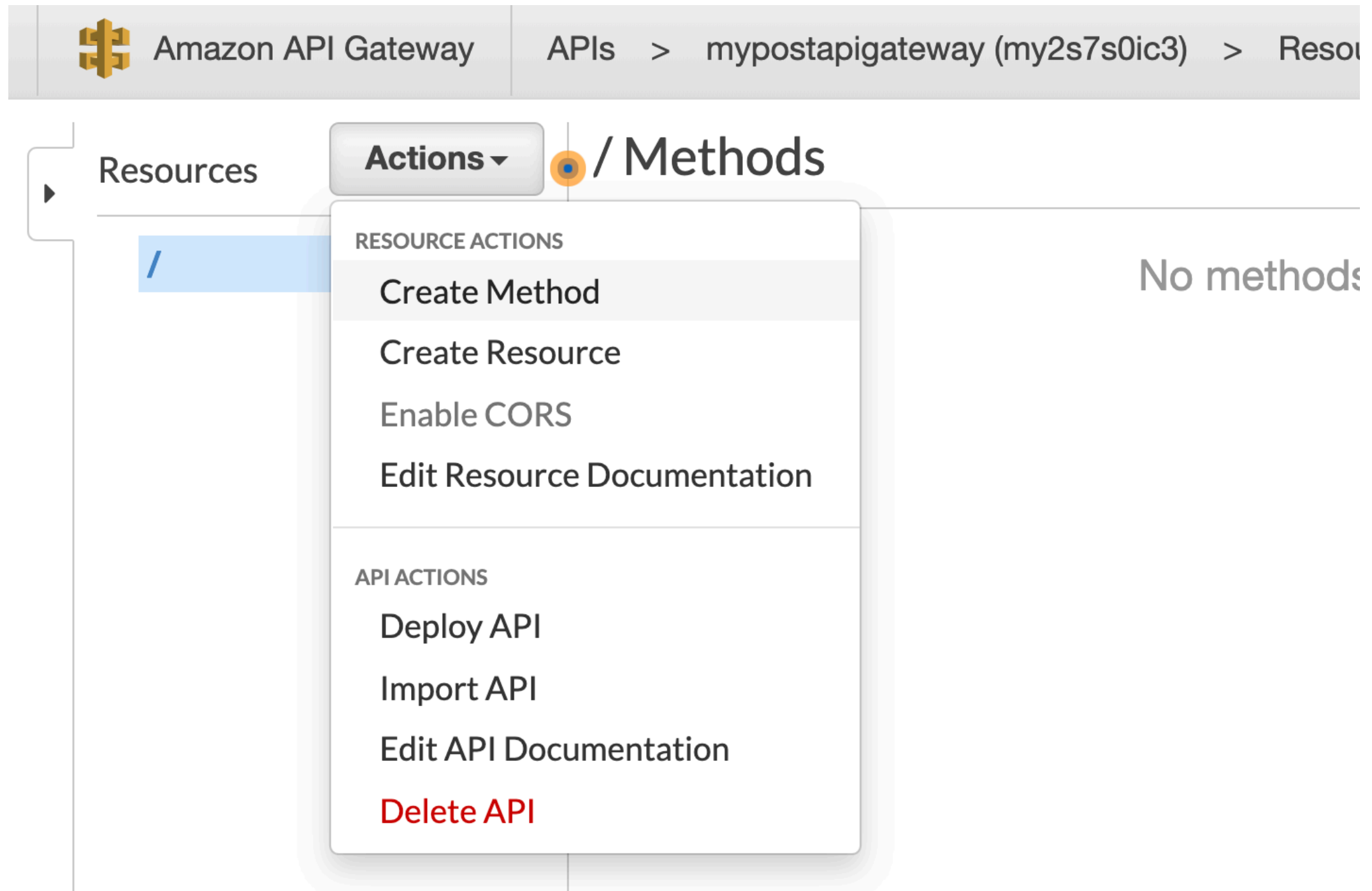
Endpoint Type



* Required

Create API

Lambda function accessible by an HTTP POST request



The screenshot displays the Amazon API Gateway console interface. At the top, the breadcrumb navigation shows 'Amazon API Gateway' followed by 'APIs > mypostapigateway (my2s7s0ic3) > Resources'. On the left, a sidebar contains a 'Resources' section with a blue highlight on the root resource '/'. The main content area is titled 'Actions / Methods'. The 'Actions' dropdown menu is open, revealing two categories of actions: 'RESOURCE ACTIONS' and 'API ACTIONS'. Under 'RESOURCE ACTIONS', the options are 'Create Method' (highlighted), 'Create Resource', 'Enable CORS', and 'Edit Resource Documentation'. Under 'API ACTIONS', the options are 'Deploy API', 'Import API', 'Edit API Documentation', and 'Delete API' (in red text). To the right of the dropdown, the text 'No methods' is visible.

Amazon API Gateway

APIs > mypostapigateway (my2s7s0ic3) > Resources

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

Lambda function accessible by an HTTP POST request

Resources

Actions ▾

▼ /
POST

/ - POST - Setup

Choose the integration point for your new method.

Integration type

☒ Lambda Function ⓘ
☐ HTTP ⓘ
☐ Mock ⓘ
☐ AWS Service ⓘ
☐ VPC Link ⓘ

Use Lambda Proxy integration

☐ ⓘ

Lambda Region

us-east-1 ▾

Lambda Function




ⓘ



Use Default Timeout

☒ ⓘ

Save

Integration Request section

 **Services** ▾ **Resource Groups** ▾   vocstartsoft/user325192=r_go... ▾ N. Virginia ▾ Support ▾

 Amazon API Gateway APIs > mypostapigateway (my2s7s0ic3) > Resources > / (la5niav6hj) > POST Show all hints 


Resources **Actions** ▾


▼ /
POST

Provide information about the target backend that this method will call and whether the incoming request data should be modified.

Integration type ☒ Lambda Function ⓘ
☐ HTTP ⓘ
☐ Mock ⓘ
☐ AWS Service ⓘ
☐ VPC Link ⓘ

Use Lambda Proxy integration ☐ ⓘ

Lambda Region us-east-1 

Lambda Function mypostfunction 


Click Mapping Templates from Integration Request

Use Default Timeout ☒ ⓘ

▶ URL Path Parameters

▶ URL Query String Parameters

▶ HTTP Headers

▼ Mapping Templates 

Request body passthrough ☒ When no template matches the request Content-Type header ⓘ
☐ When there are no templates defined (recommended) ⓘ
☐ Never ⓘ

Content-Type

application/x-www-form-urlencoded ☒ ☐

+

 Add mapping template

Enter Content-Type as “application/x-www-form-urlencoded”

Click Mapping Templates from Integration Request

Change passthrough behavior

⚠ Your current passthrough behavior will pass all request payloads directly to the endpoint without transformation, unless there is a match for the incoming Content-Type. Do you want to secure this integration to only allow requests that match one of your defined Content-Types?

No, use current settings

Yes, secure this integration

☐ When there are no templates defined (recommended) ⓘ

☐ Never ⓘ

Content-Type

application/x-www-form-urlencoded

+ Add mapping template

Click “application/x-www-form-urlencoded”

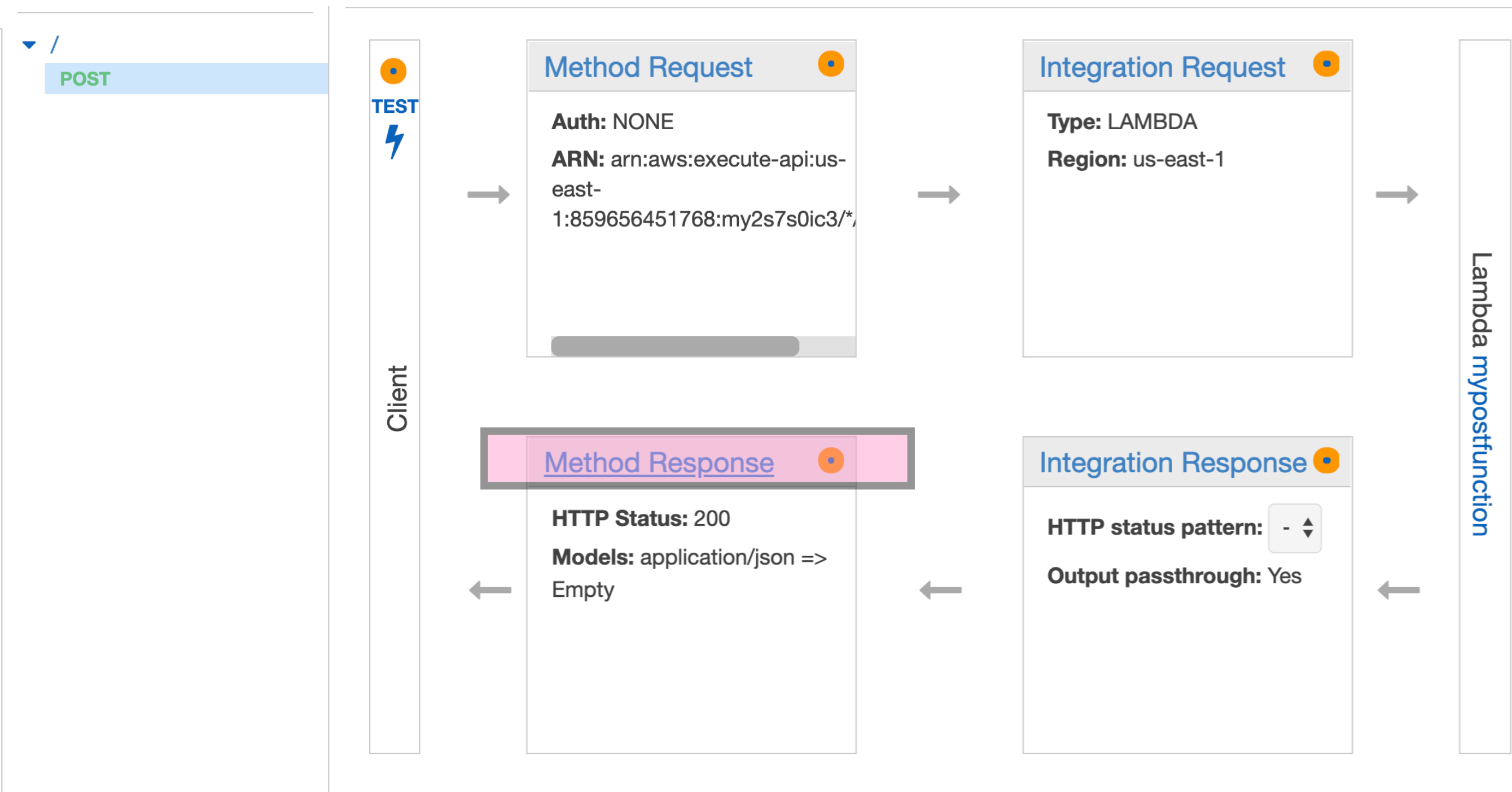
application/x-www-form

Generate template:

```
1 {  
2   "body": "$input.body"  
3 }
```

Enter above JSON content

Add Content-Type header to the response



Click Method Response

POST - Method Response

The screenshot shows the AWS API Gateway console interface. At the top, the navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a user profile 'vocstartsoft/user32519'. Below this, the breadcrumb trail reads: 'Amazon API Gateway' > 'APIs' > 'mypostapigateway (my2s7s0ic3)' > 'Resources' > '/ (la5nia)'. The main content area is divided into two panes. The left pane, titled 'Resources', shows a tree view with a dropdown arrow next to '/', and a 'POST' method listed below it. The right pane, titled 'Actions', contains a link 'Method Execution' with a left-pointing arrow, followed by the title '/ - POST - Method Response'. Below the title is a descriptive text: 'Provide information about this method's response types, their headers'. A table with one row is visible, with the header 'HTTP Status' and the value '200'. A pink rectangular box highlights a play button icon in the first column of this table. At the bottom of the right pane, there is a button with a plus icon and the text 'Add Response'.

aws Services Resource Groups vocstartsoft/user32519

Amazon API Gateway APIs > mypostapigateway (my2s7s0ic3) > Resources > / (la5nia)

Resources Actions


Method Execution / - POST - Method Response

Provide information about this method's response types, their headers

HTTP Status
200


+ Add Response

POST - Method Response

 Amazon API Gateway

APIs > mypostapigateway (my2s7s0ic3) > Resources > / (la5niav6hj) > POST

Show all hints




Resources


Actions ▾

▼ /




POST



 Method Execution

 / - POST - Method Response



Provide information about this method's response types, their headers and content types.

HTTP Status				
▼ 200	 			
<div>Response Headers for 200</div> <table><thead><tr><th>Name</th><th></th></tr></thead><tbody><tr><td colspan="2">No headers</td></tr></tbody></table> <div> Add Header</div>	Name		No headers	
Name				
No headers				

 Response Body for 200 | Content type | Models | | |------------------|--------|---| | application/json | Empty |   | Add Response Model |

POST - Method Response

aws

Services

Resource Groups

vocstartsoft/user325192=r_go...

N. Virginia

Support

Amazon API Gateway

APIs > mypostapigateway (my2s7s0ic3) > Resources > / (la5niav6hj) > POST

Show all hints

Resources

Actions

Method Execution / - POST - Method Response

POST

Provide information about this method's response types, their headers and content types.

HTTP Status
200

Response Headers for 200

Name
Content-Type

Response Body for 200

Content type	Models
application/json	Empty

Remove application/json from under Response Body

POST - Method Response

Resources

Actions ▾

▼ /

POST

← Method Execution

/ - POST - Method Response

Provide information about this method's response types, their headers and content types.

HTTP Status	
▼	200

Response Headers for 200

Name	
Content-Type	

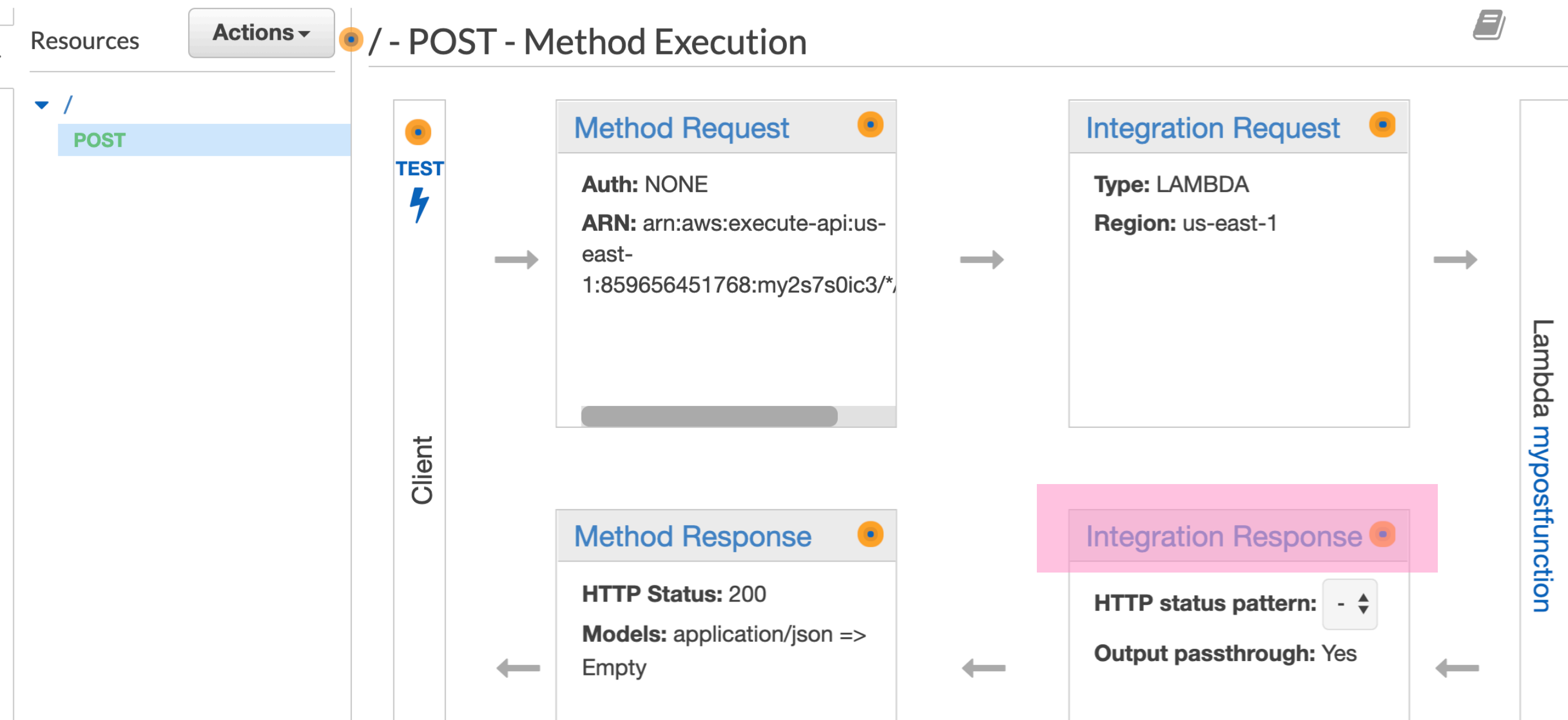
+ Add Header

Response Body for 200

Content type	Models	
	No models	

+ Add Response Model

Translating the response from a JSON string to HTML



POST - Integration Response

Resources

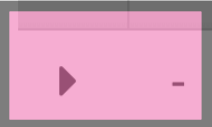

Actions ▾

[← Method Execution](#) / - POST - Integration Response 

▼ /


POST

First, declare response types using [Method Response](#). Then, map the possible responses from the backend to this method's response types.

	Lambda Error Regex	Method response status	Output model	Default mapping	
		200		Yes	

 [Add integration response](#)

POST - Integration Response

	Lambda Error Regex	Method response status	Output model	Default mapping	
▼	-	200		Yes	✕
Map the output from your Lambda function to the headers and output model of the 200 method response.					
	Lambda Error Regex	<input type="text" value="default"/>		i	
	Content handling	<input type="text" value="Passthrough"/>		i	
				Cancel	Save
▼ Header Mappings					
Response header		Mapping value		i	
Content-Type					
▶ Mapping Templates					

POST - Integration Response

▼

-

200

Yes

✕

Map the output from your Lambda function to the headers and output model of the 200 method response.

Lambda Error Regex

default

i

Content handling

Passthrough

⬆️⬆️

i

Cancel

Save

▼ Header Mappings

Response header	Mapping value <div>i</div>	
Content-Type	<div>'text/html'</div>	<div>✓ ✕</div>

▶ Mapping Templates

POST - Integration Response

▼

-

200

Yes

✕

Map the output from your Lambda function to the headers and output model of the 200 method response.

Lambda Error Regex

default

i

Content handling

Passthrough

⬆️⬆️

i

Cancel

Save

▶ Header Mappings

▼ Mapping Templates

Content-Type

application/json

⊖

+

Add mapping template

POST - Integration Response

Cancel

Save

▶ Header Mappings

▼ Mapping Templates

Content-Type

text/html



Add mapping template

text/html

Generate template:



```
1 $input.path('$')
```

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.

Deployment stage

[New Stage]

Stage name*

prod

Stage description

Deployment description

Cancel

Deploy