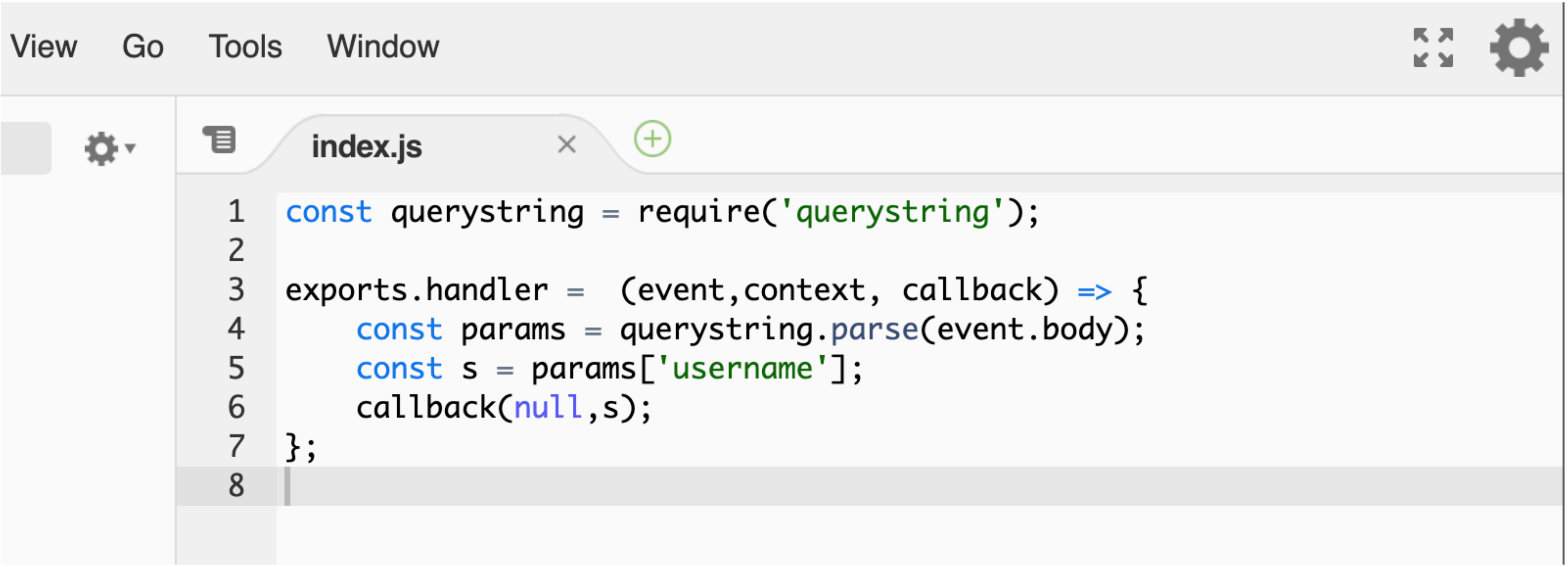
Lambda - POST

Create a Lambda Function

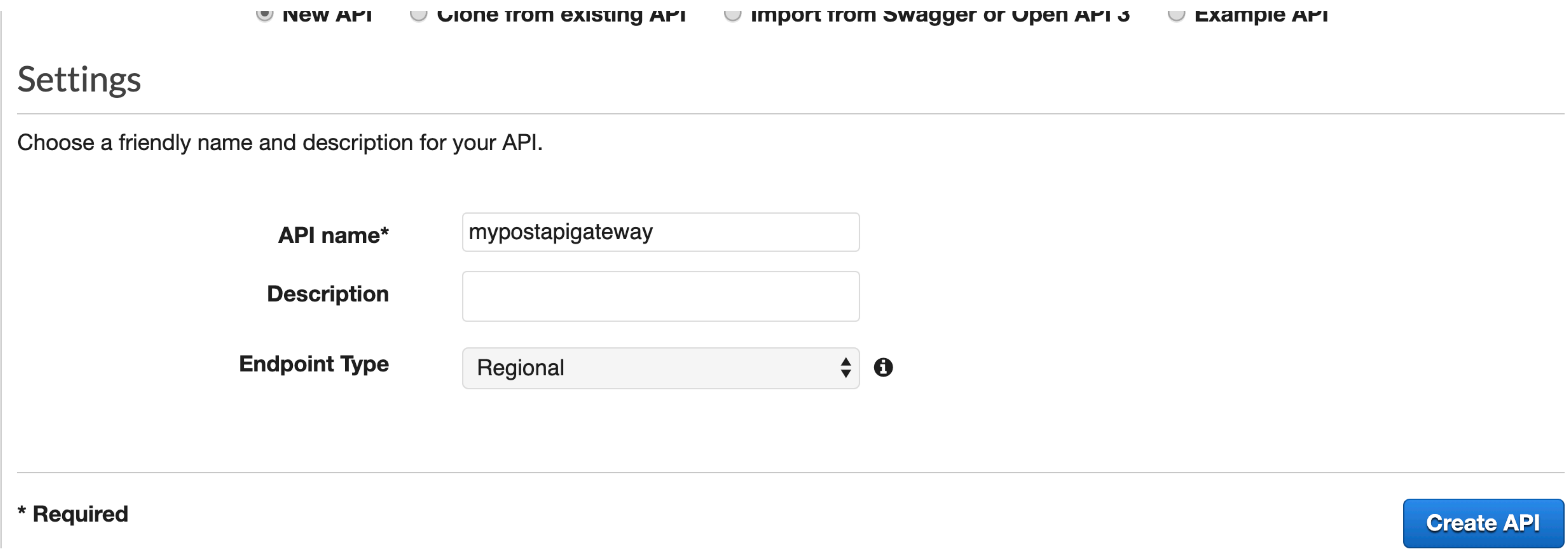


```javascript
const querystring = require('querystring');

exports.handler =  (event,context, callback) => {
    const params = querystring.parse(event.body);
    const s = params['username'];
    callback(null,s);
};

```

Configuring API Gateway



● New API ○ Clone from existing API ○ Import from Swagger or Open API 3 ○ Example API

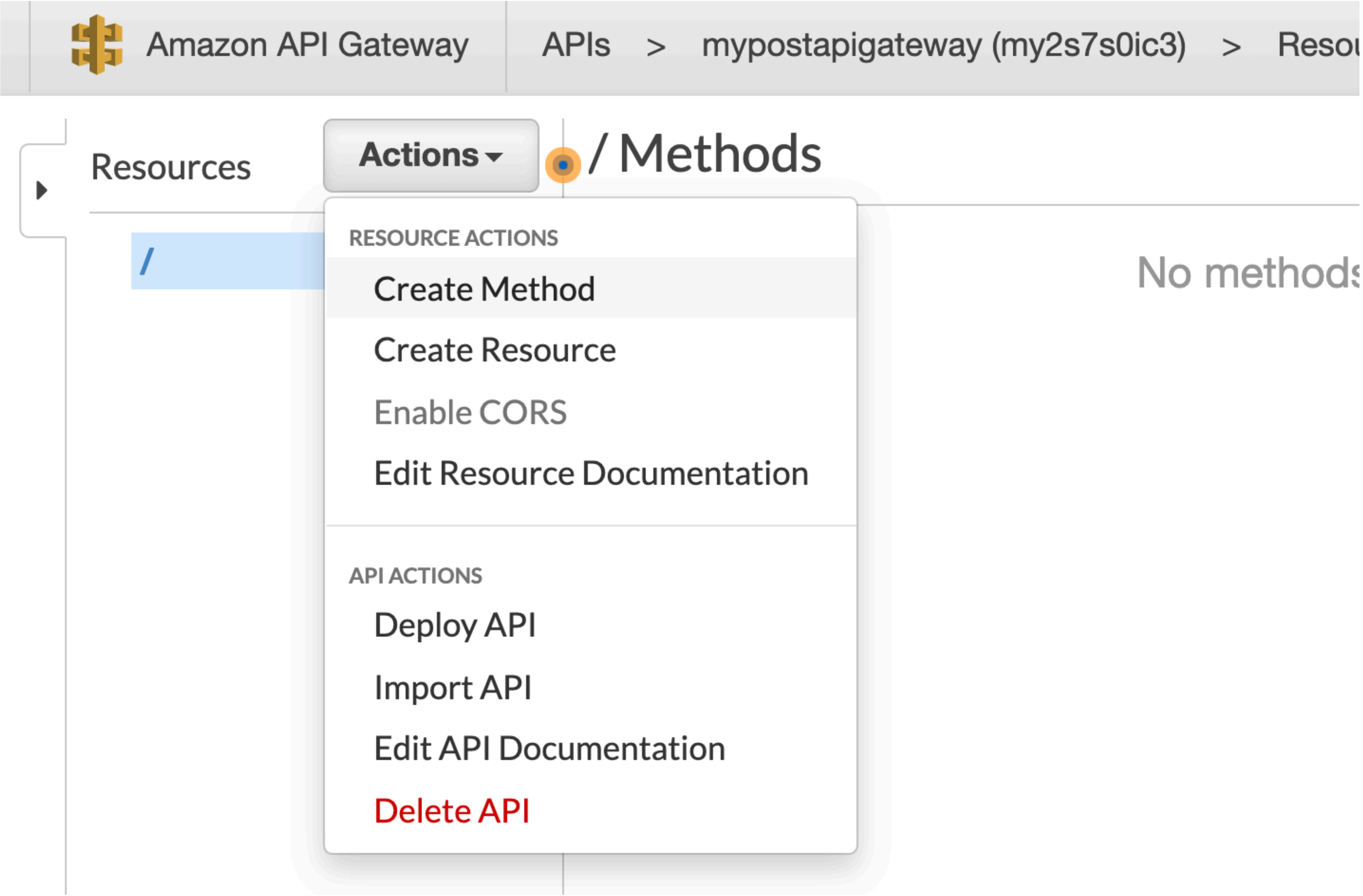
Settings

Choose a friendly name and description for your API.

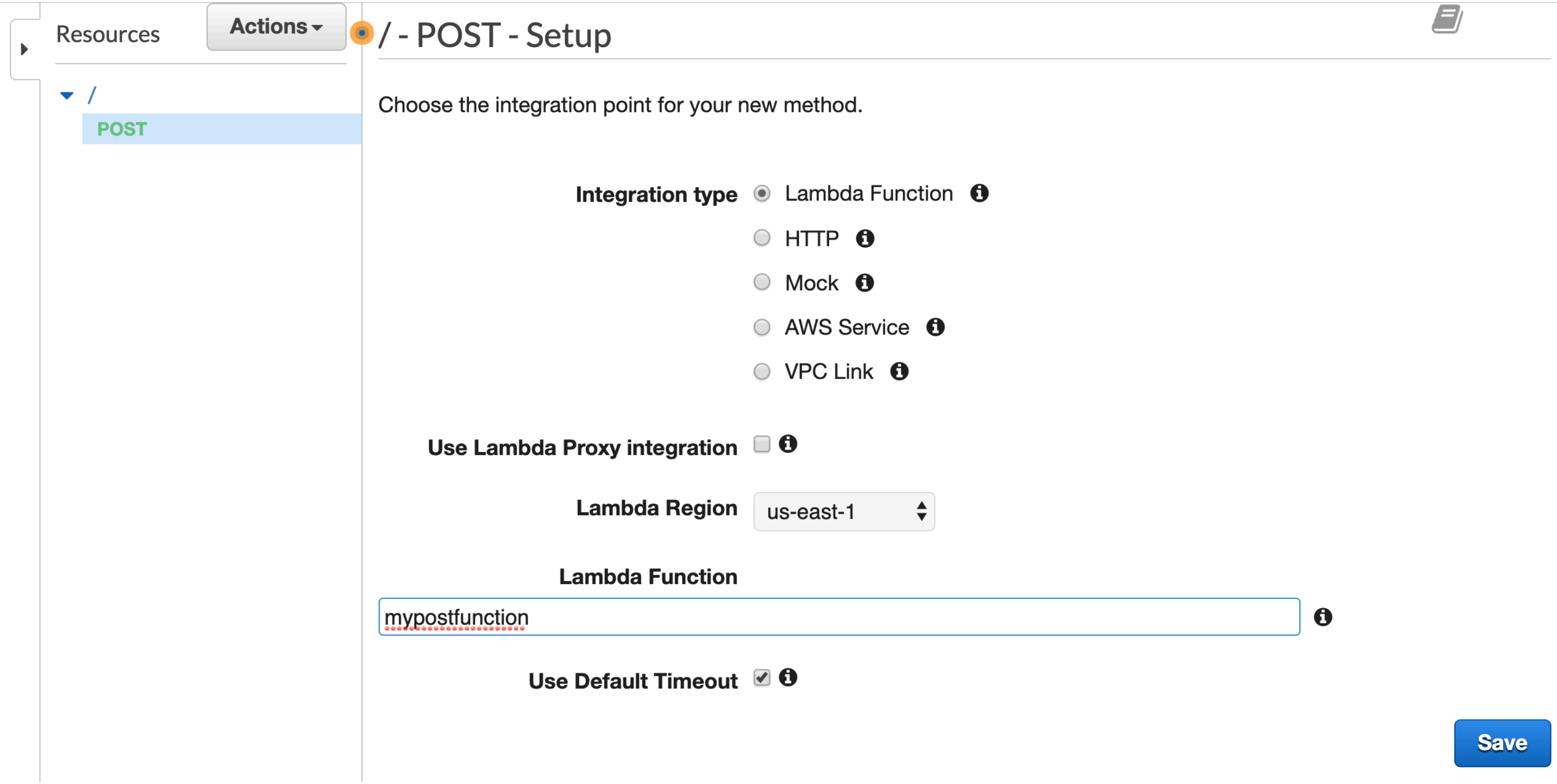
API name*	mypostapigateway
Description	
Endpoint Type	Regional

* Required

Lambda function accessible by an HTTP POST request



Lambda function accessible by an HTTP POST request



/ - POST - Setup

Choose the integration point for your new method.

Integration type ● Lambda Function ○ HTTP ○ Mock ○ AWS Service ○ VPC Link

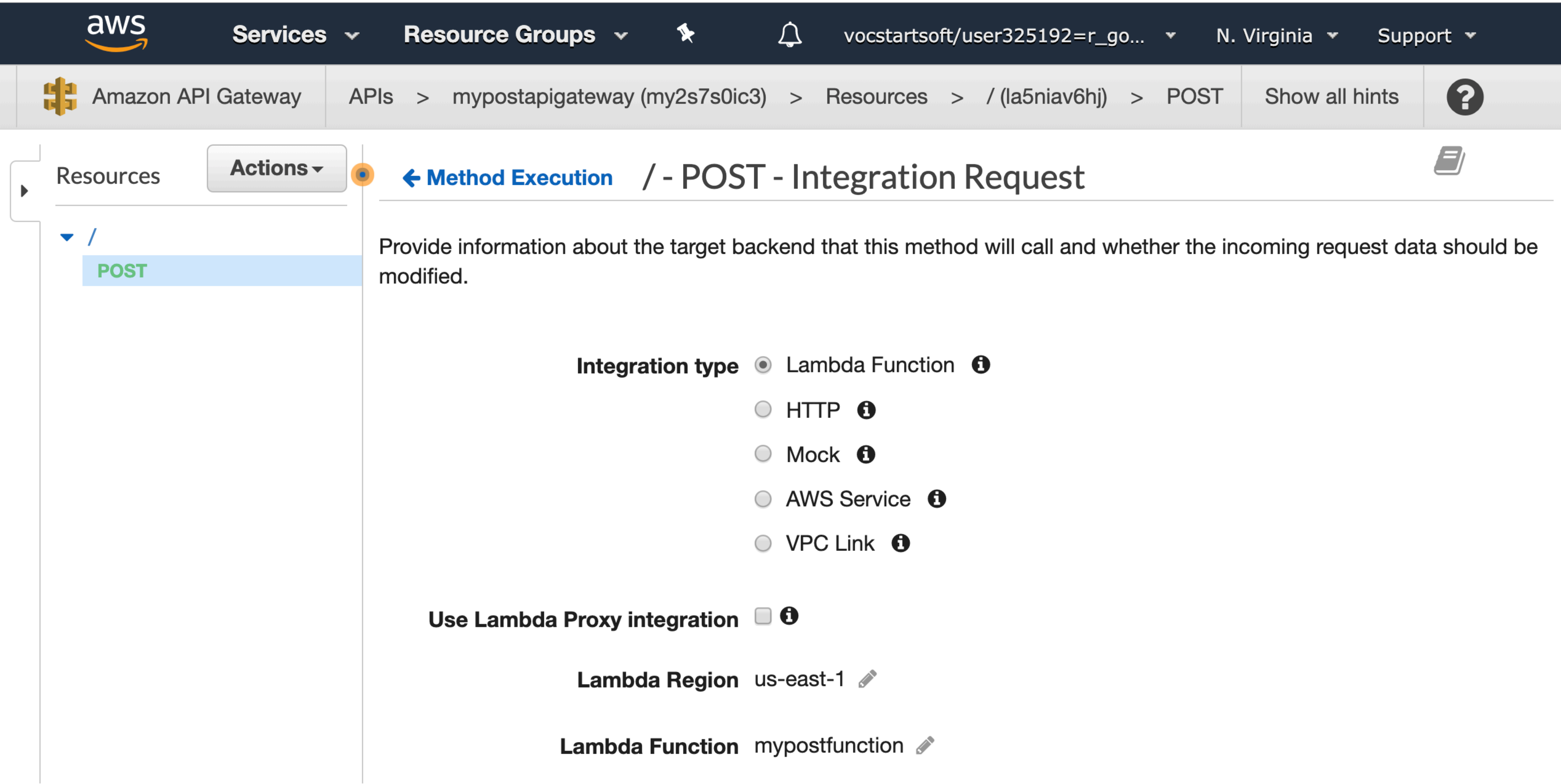
Use Lambda Proxy integration □

Lambda Region us-east-1

Lambda Function mypostfunction

Use Default Timeout ☑

Integration Request section



/ - POST - Integration Request

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

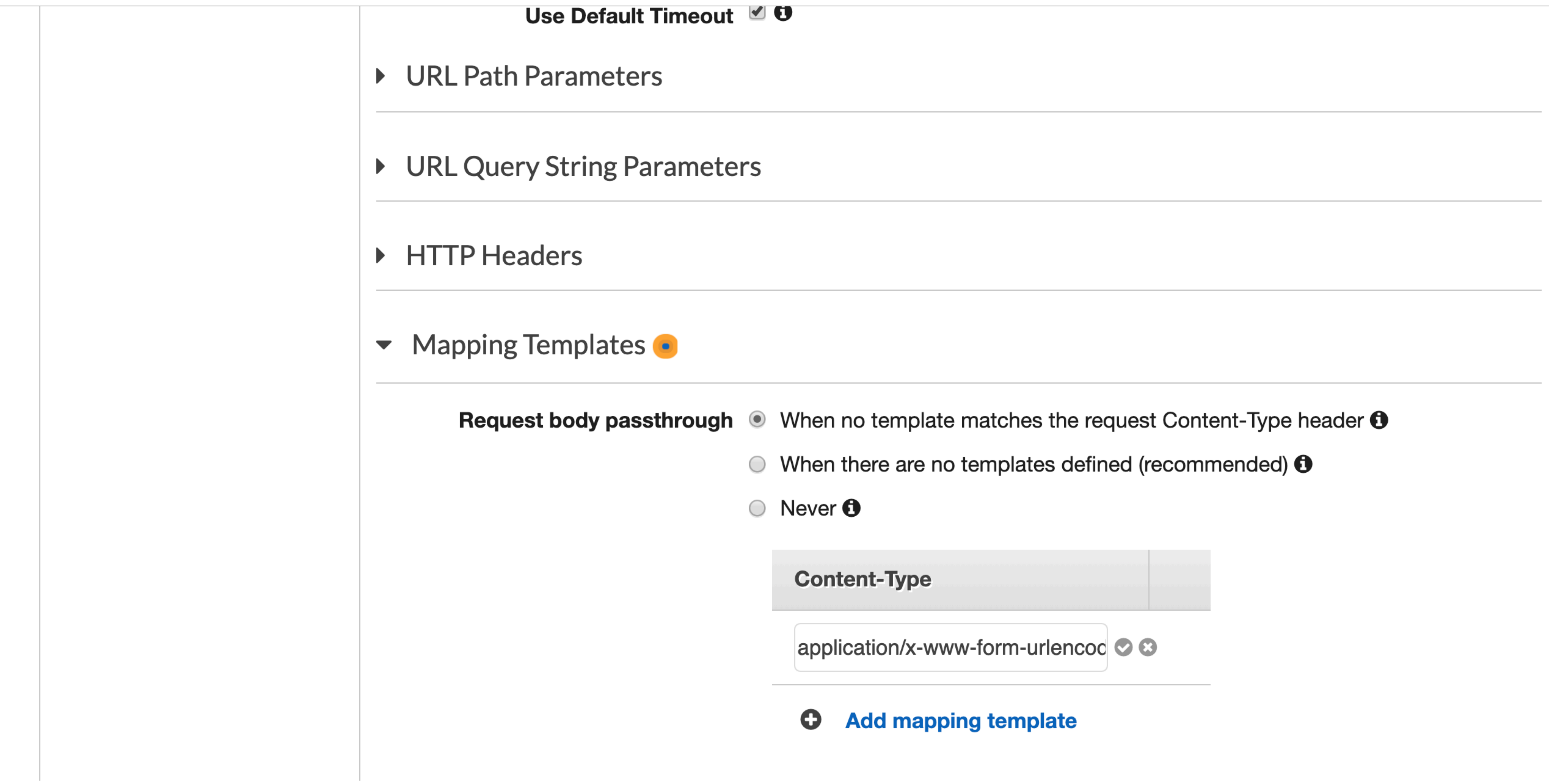
Integration type ● Lambda Function ○ HTTP ○ Mock ○ AWS Service ○ VPC Link

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

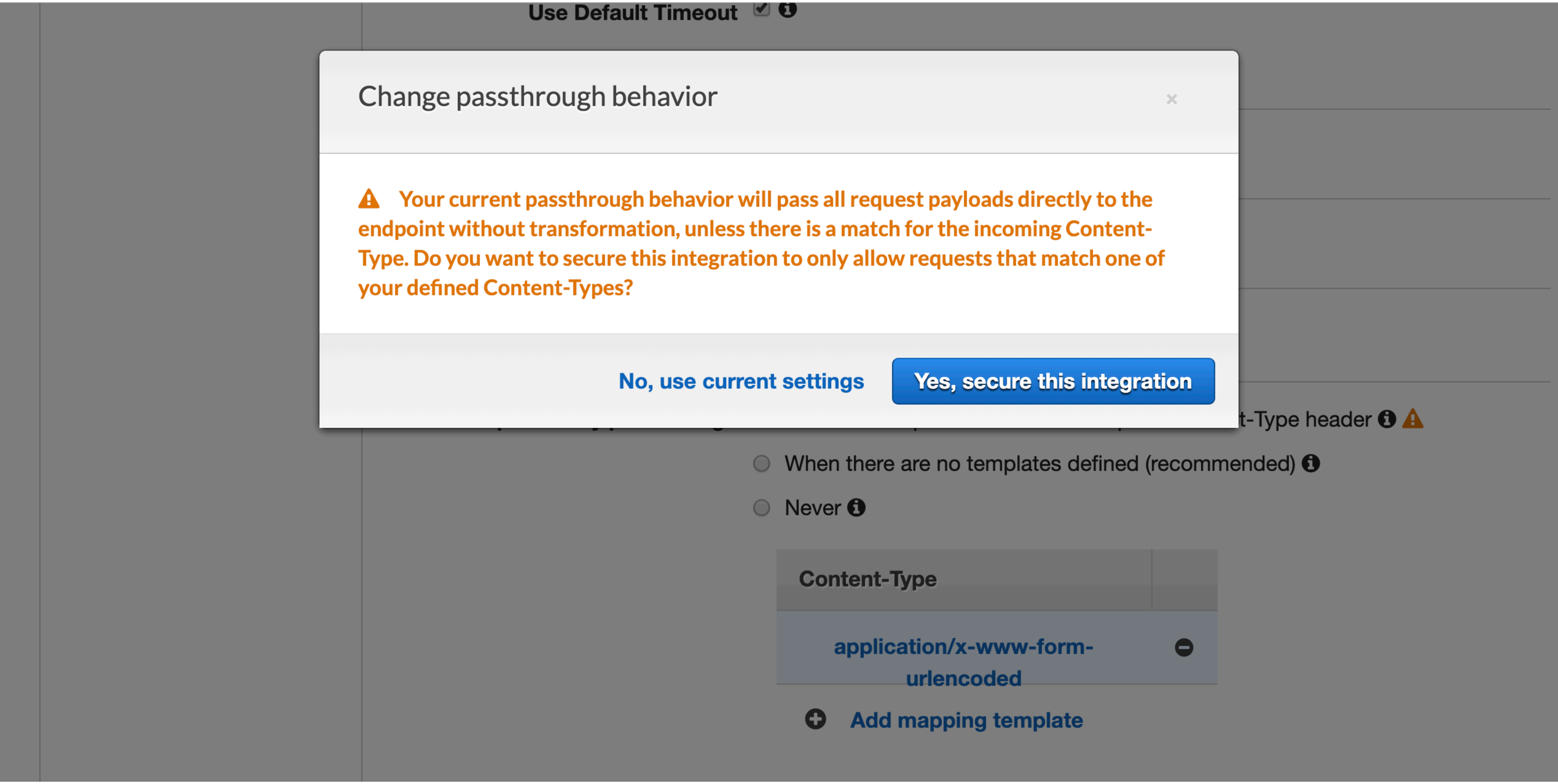
Mapping Templates

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-urlencoc

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

Click Mapping Templates from Integration Request



Use Default Timeout ☑

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?

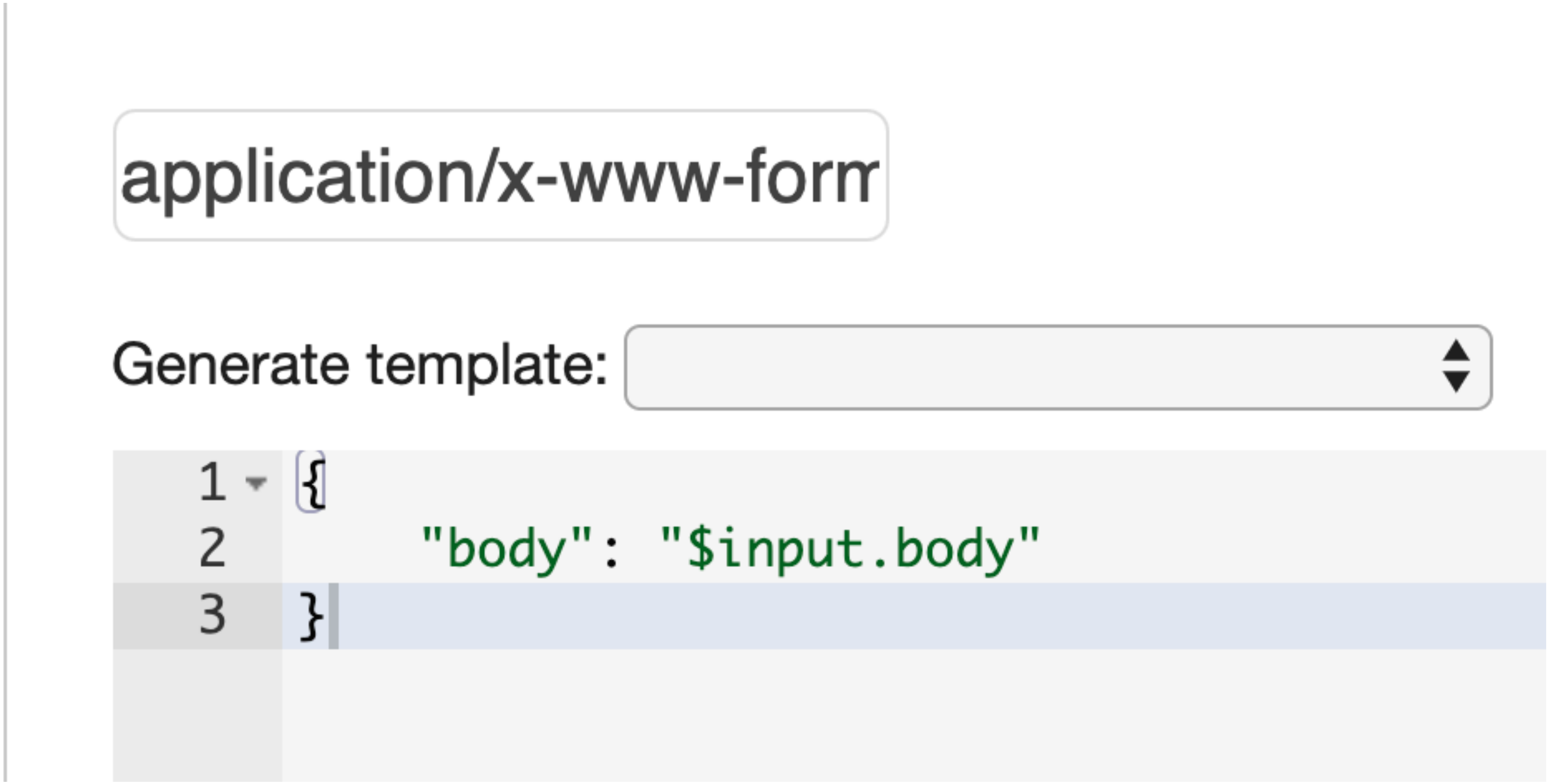
t-Type header ⚠

○ When there are no templates defined (recommended)

○ Never

Content-Type	
application/x-www-form-urlencoded	⊖

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



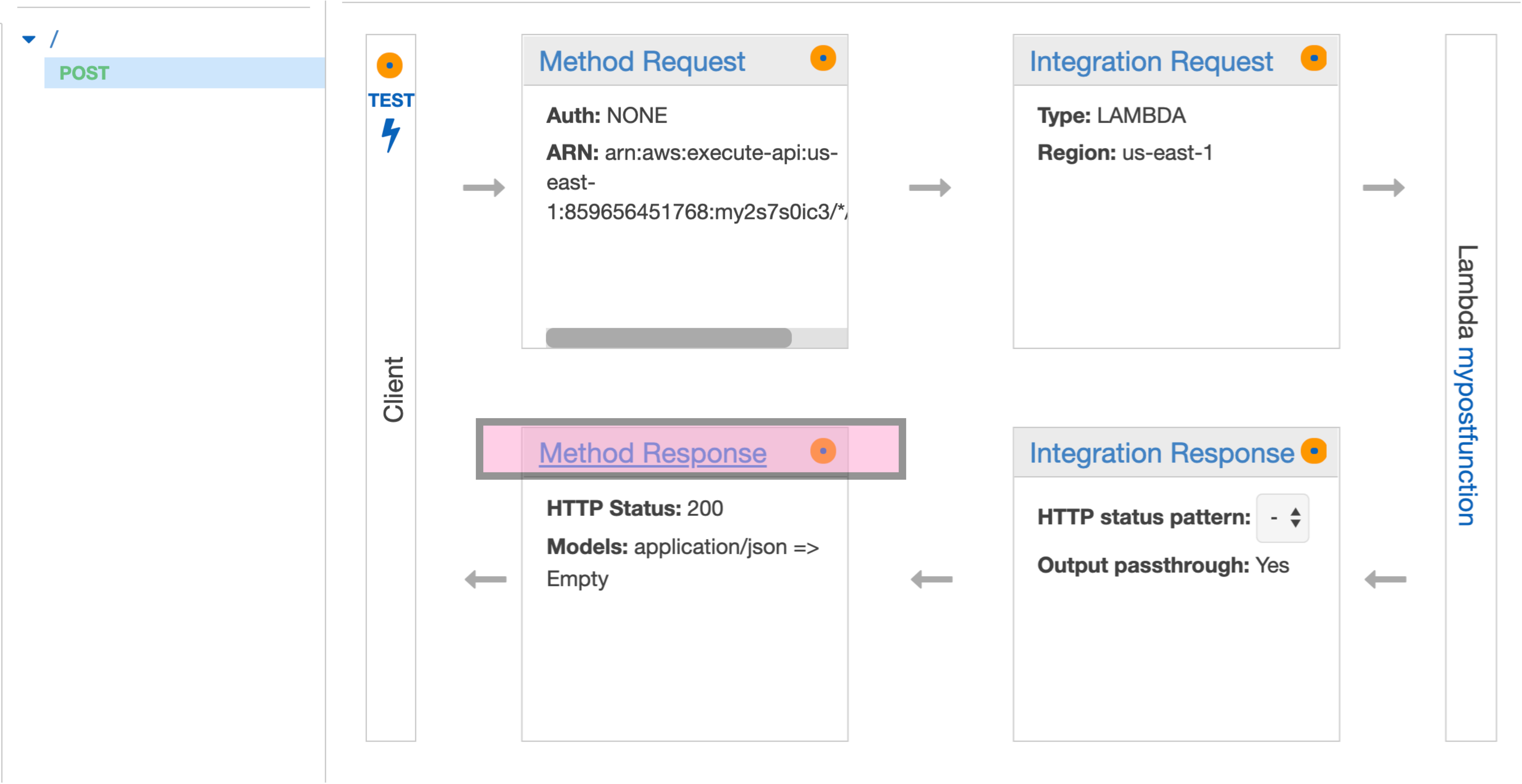
application/x-www-form

Generate template:

```
{
    "body": "$input.body"
}
```

Enter above JSON content

Add Content-Type header to the response



Auth: NONE

ARN: arn:aws:execute-api:us-east-1:859656451768:my2s7s0ic3/*/

Type: LAMBDA

Region: us-east-1

HTTP Status: 200

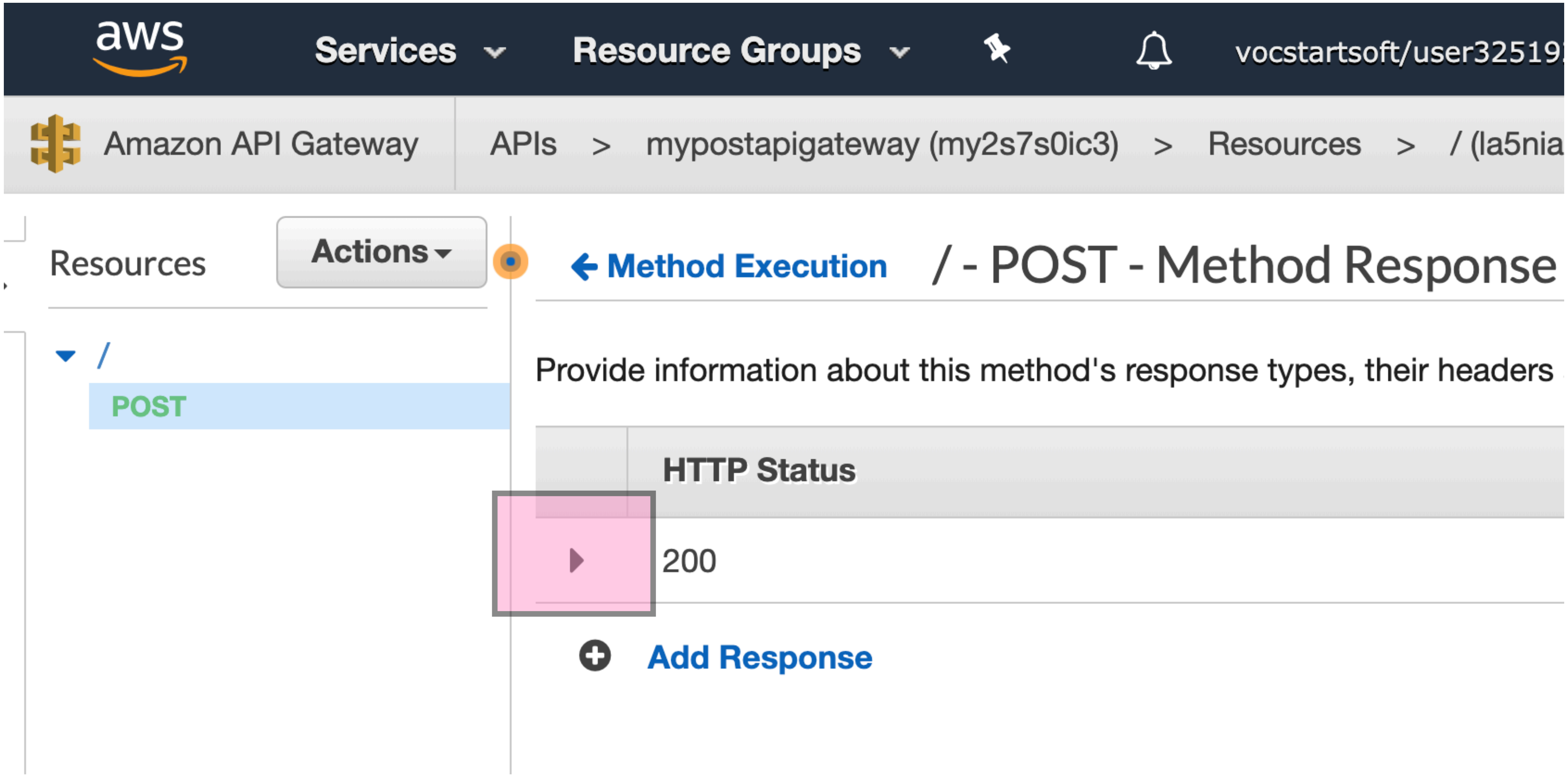
Models: application/json => Empty

HTTP status pattern: -

Output passthrough: Yes

Click Method Response

POST - Method Response

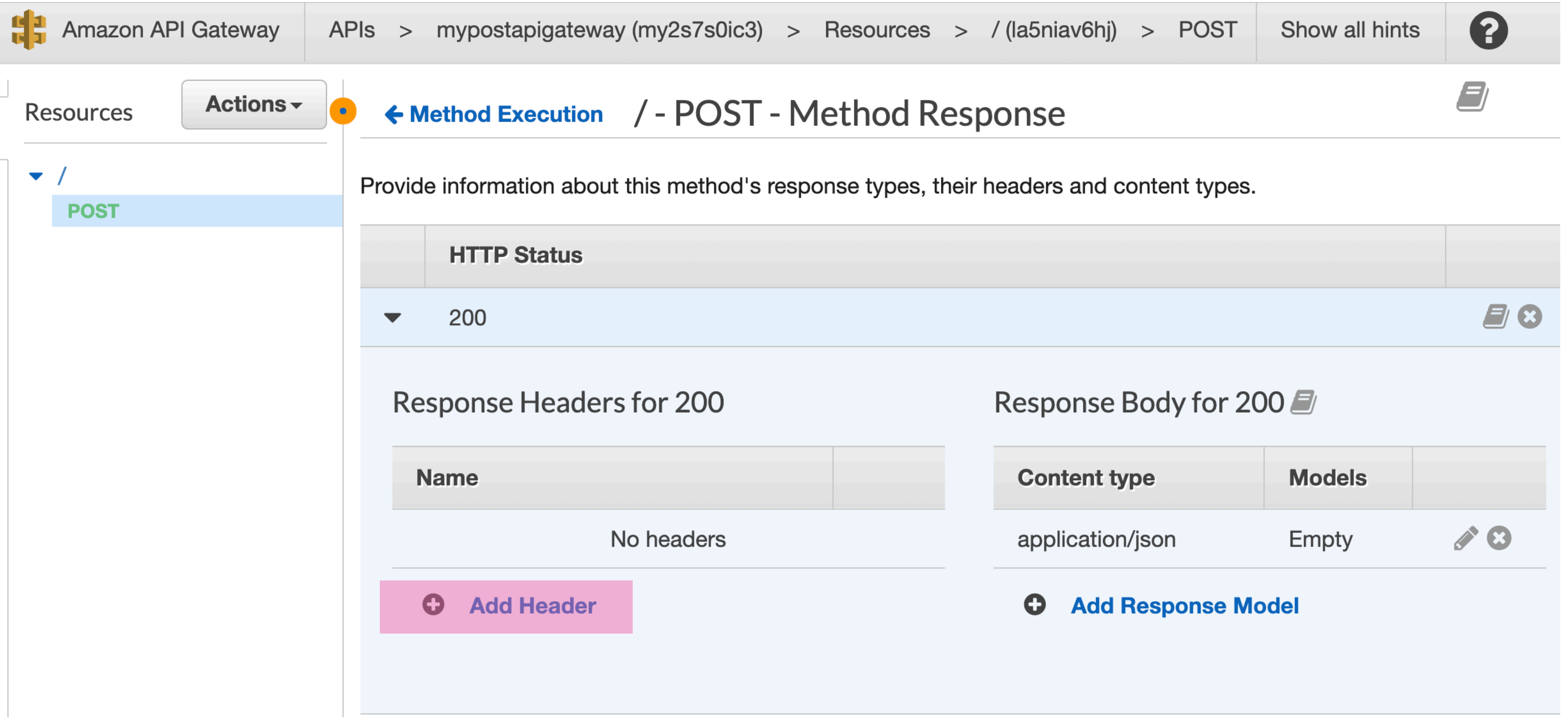


/ - POST - Method Response

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

HTTP Status
200

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

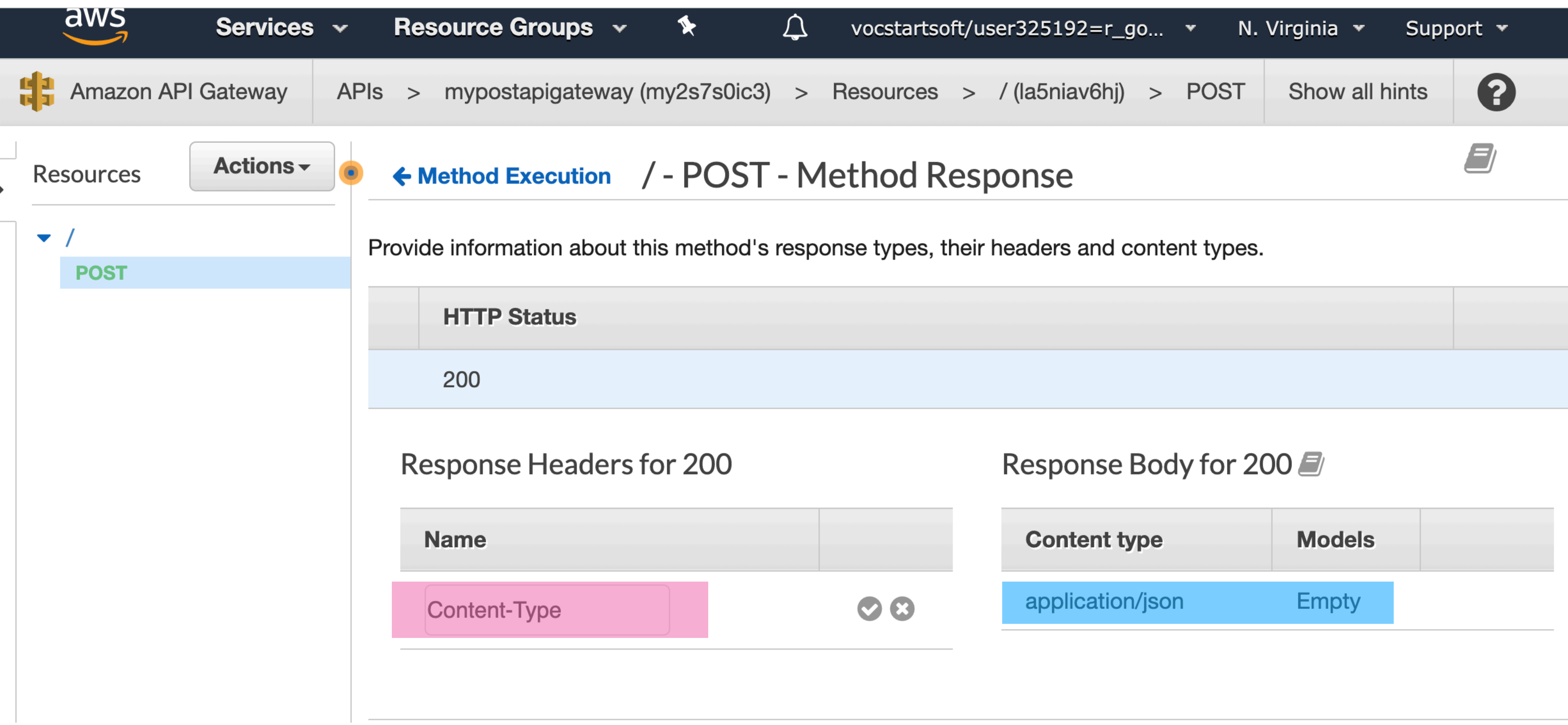
Response Headers for 200

Name	
No headers	

Response Body for 200

Content type	Models	
application/json	Empty	

POST - Method Response



/ - POST - Method Response

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

HTTP Status
200

Response Headers for 200

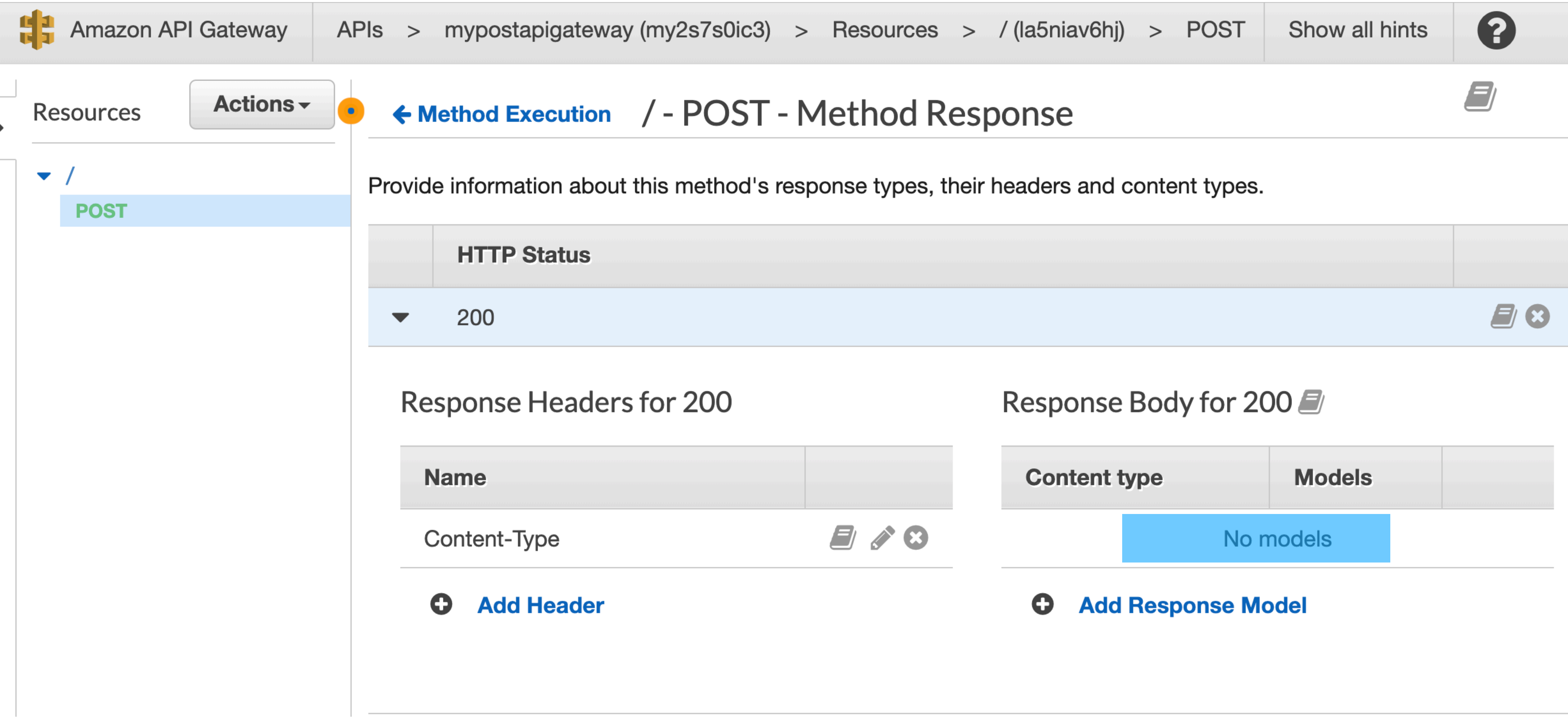
Name
Content-Type

Response Body for 200

Content type	Models
application/json	Empty

Remove `application/json` from under Response Body

POST - Method Response



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

← 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	

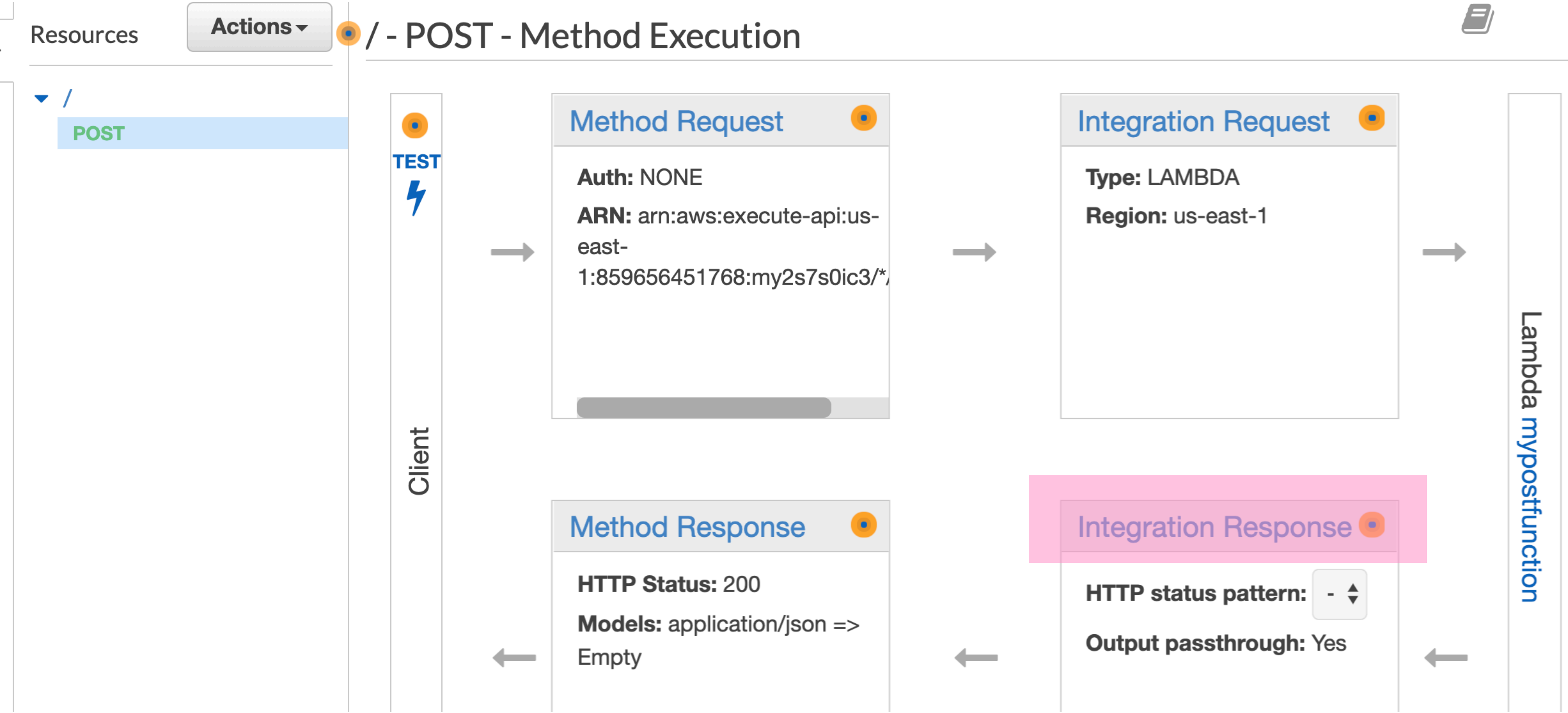
⊕ Add Header

Response Body for 200

Content type	Models	
No models		

⊕ Add Response Model

Translating the response from a JSON string to HTML



Auth: NONE

ARN: arn:aws:execute-api:us-east-1:859656451768:my2s7s0ic3/*/

Type: LAMBDA

Region: us-east-1

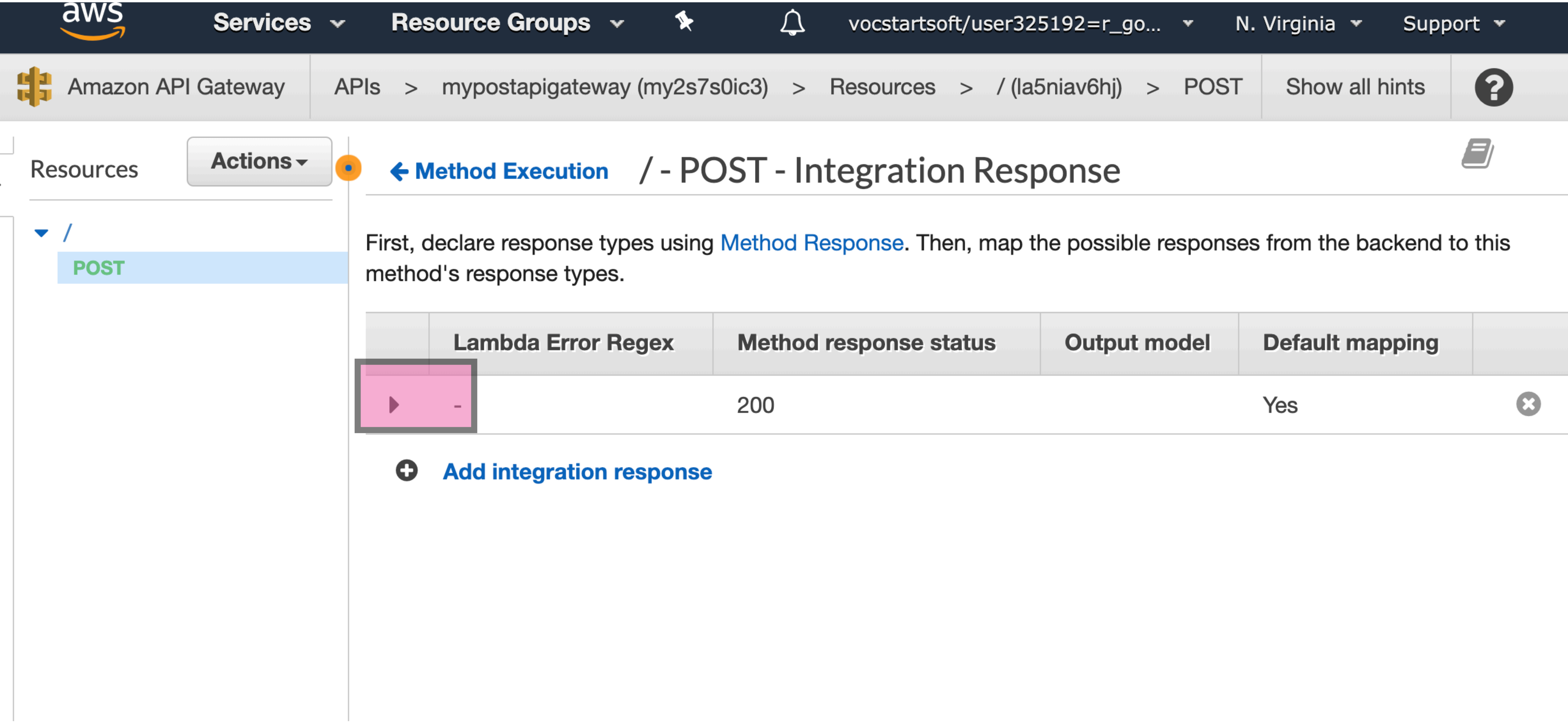
HTTP Status: 200

Models: application/json => Empty

HTTP status pattern: -

Output passthrough: Yes

POST - Integration Response



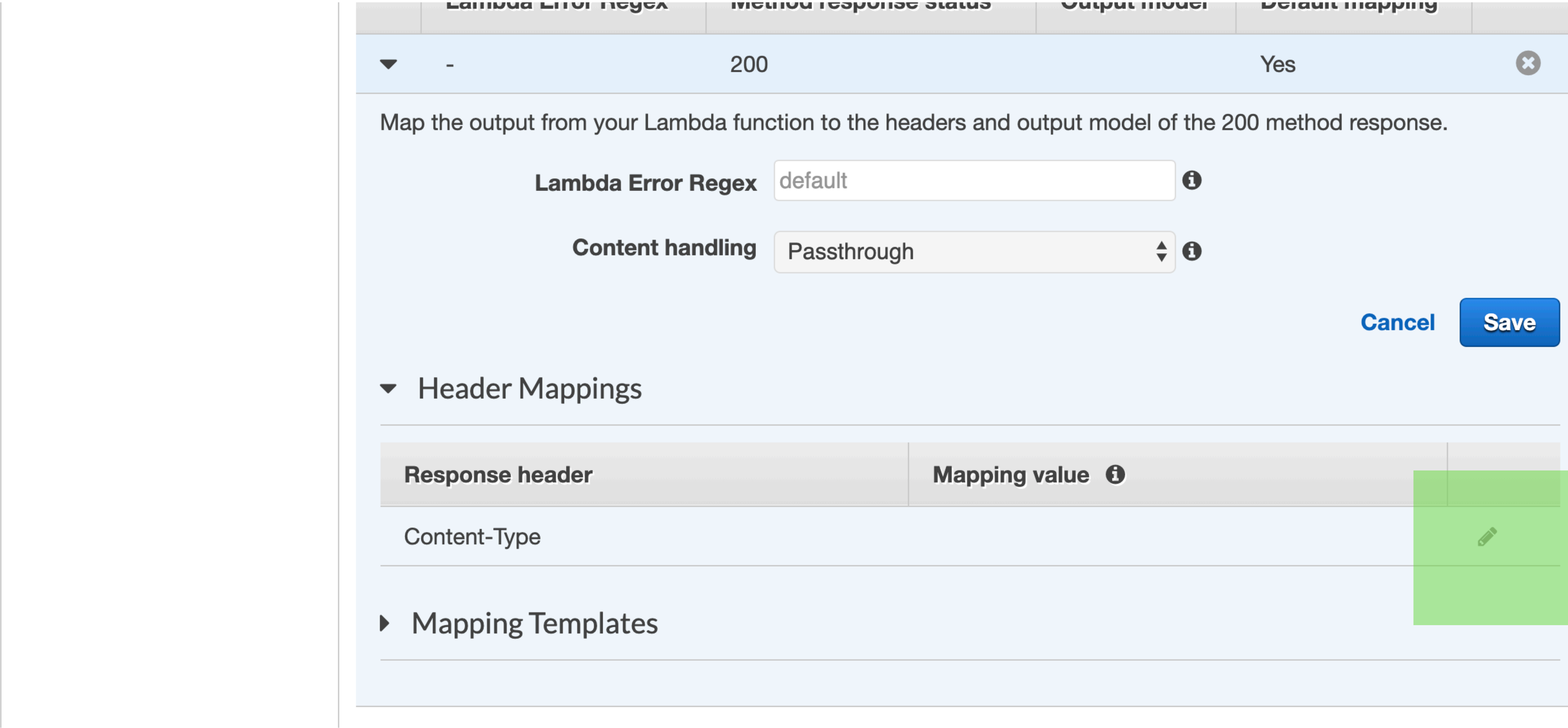
/ - POST - Integration Response

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: default

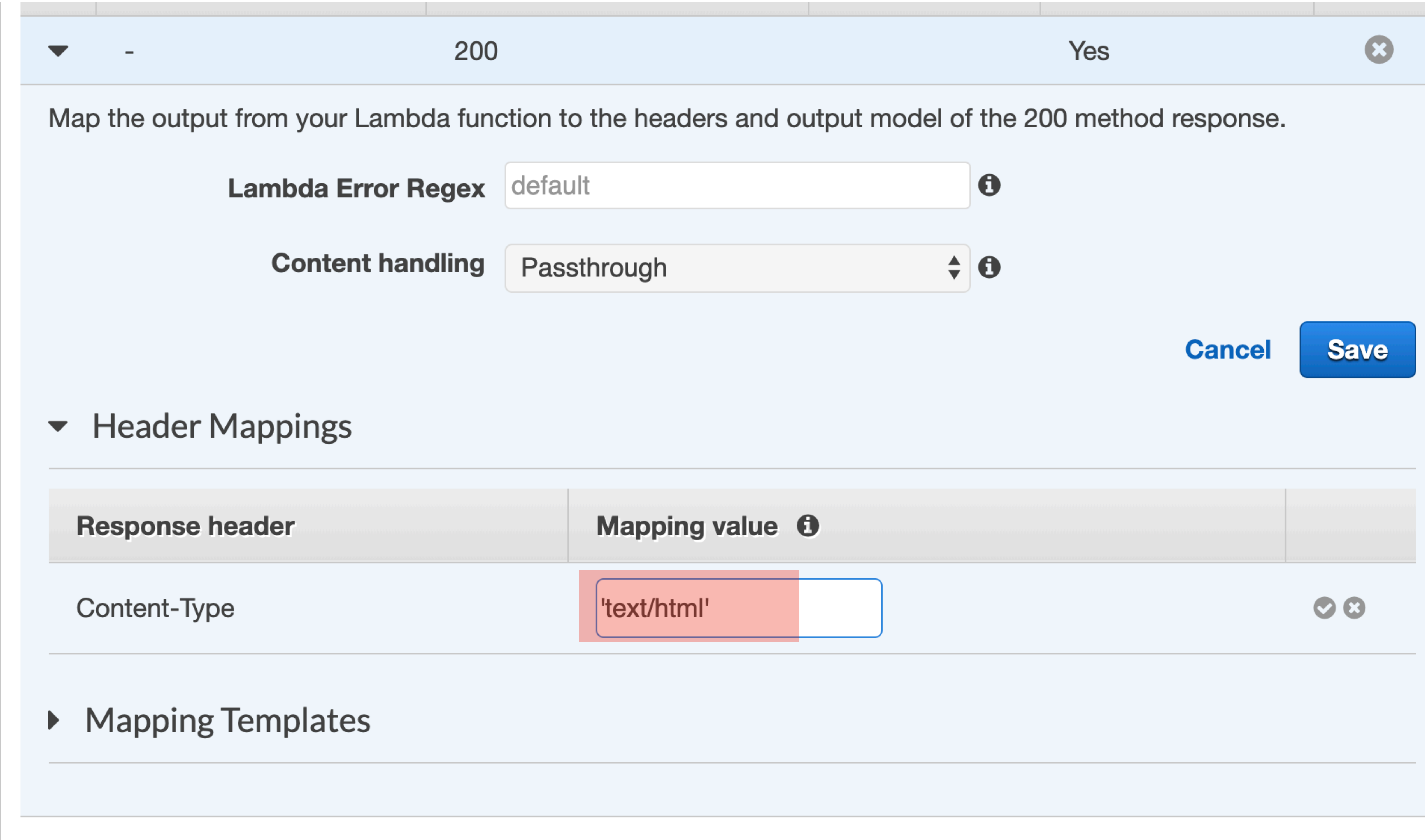
Content handling: Passthrough

Header Mappings

Response header	Mapping value
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

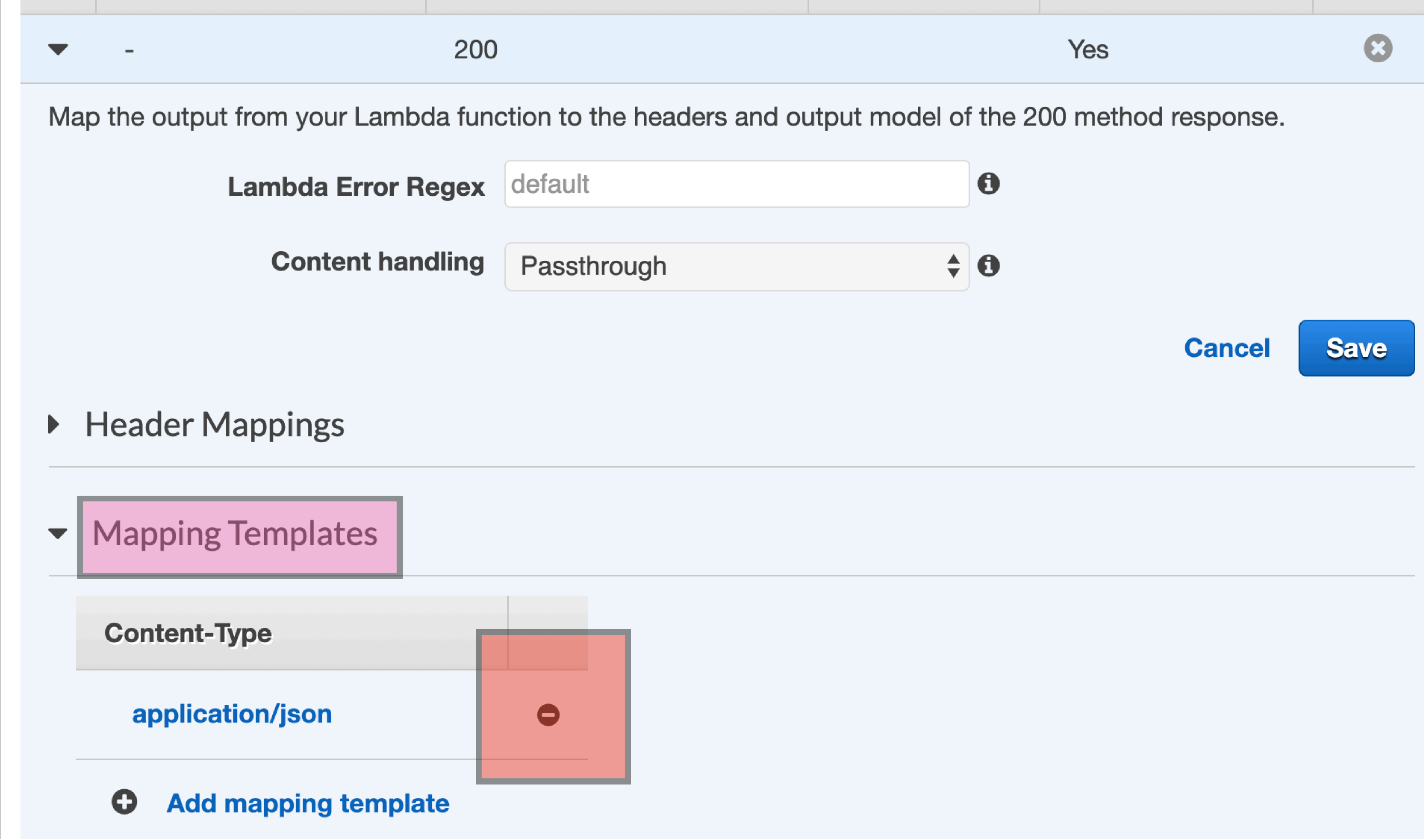
Content handling: Passthrough

Header Mappings

Response header	Mapping value	
Content-Type	'text/html'	

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

Content handling Passthrough

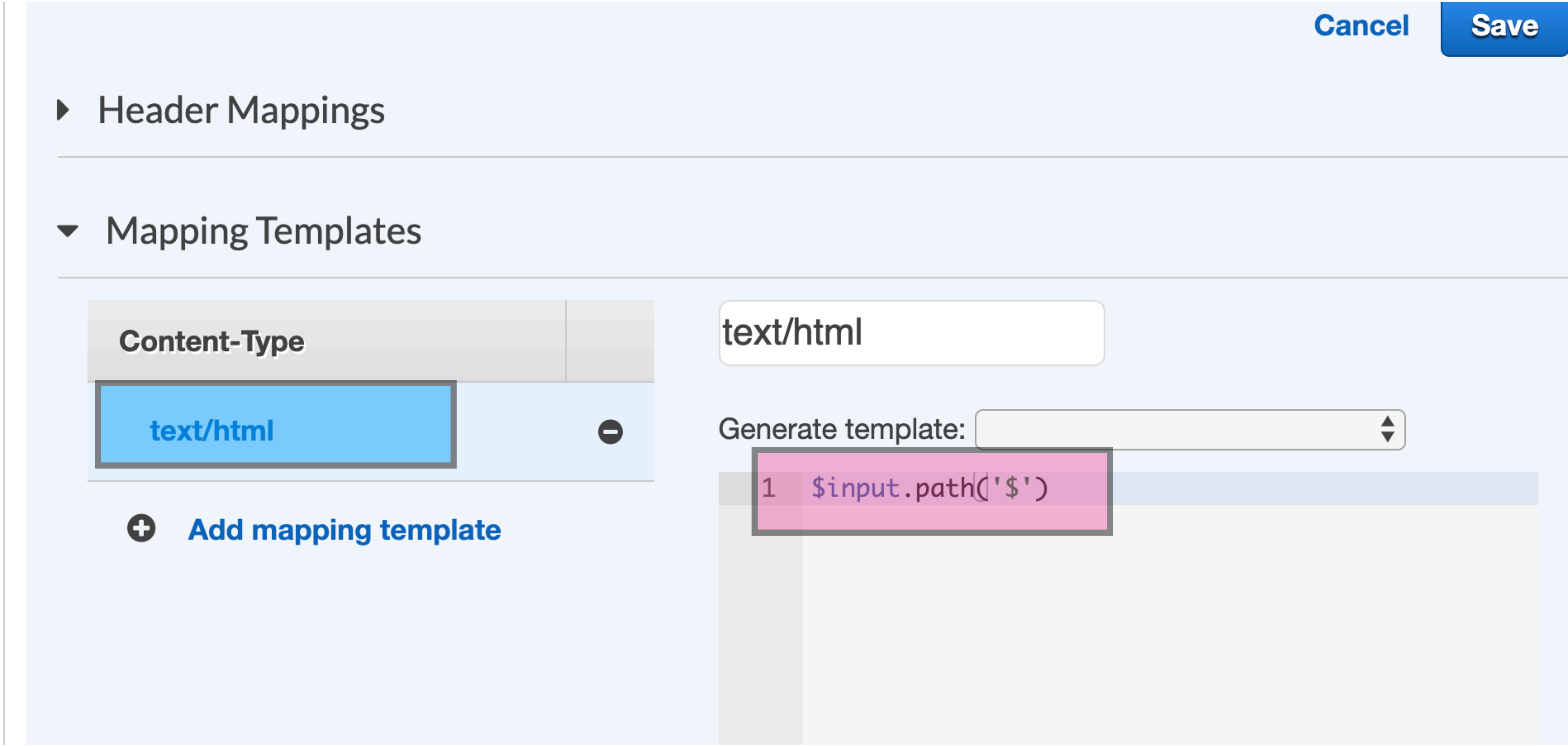
▸ Header Mappings

▾ Mapping Templates

Content-Type	
application/json	⊖

⊕ Add mapping template

POST - Integration Response

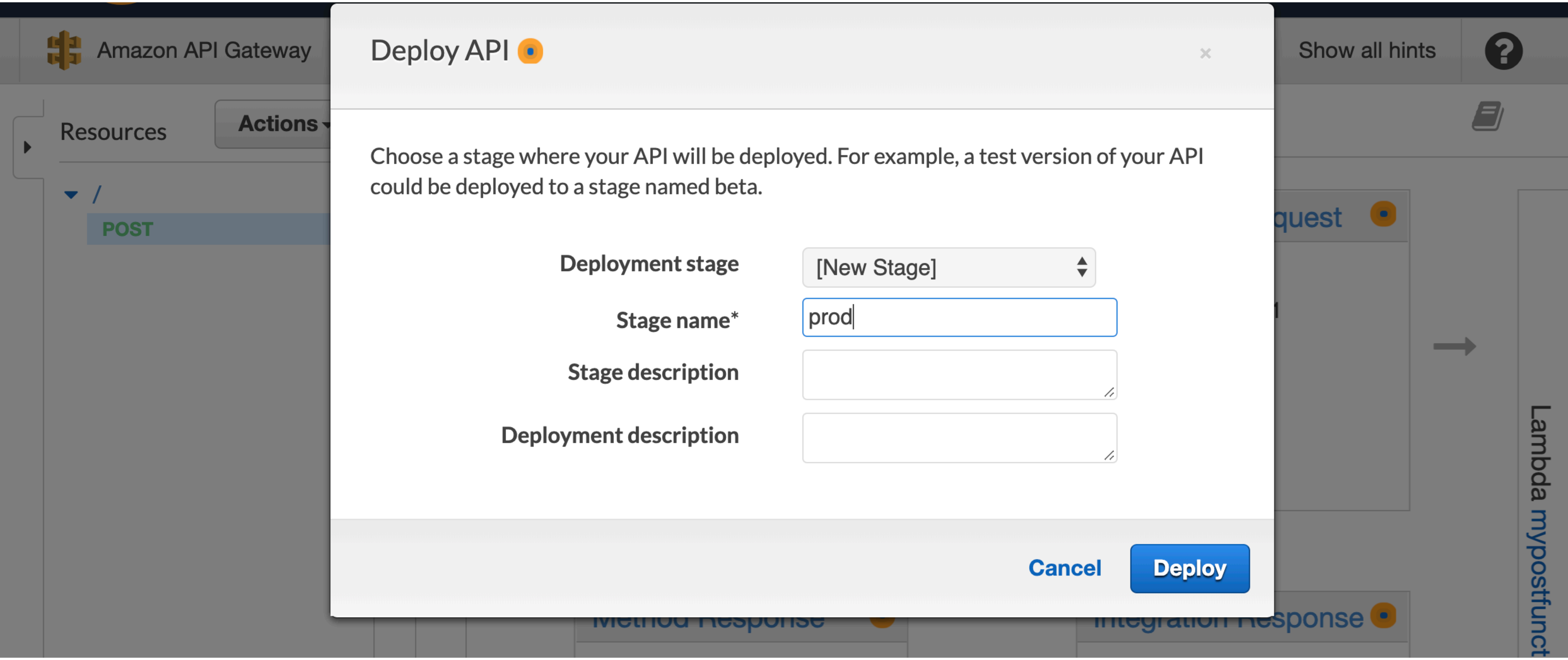


text/html

Generate template:

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

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: