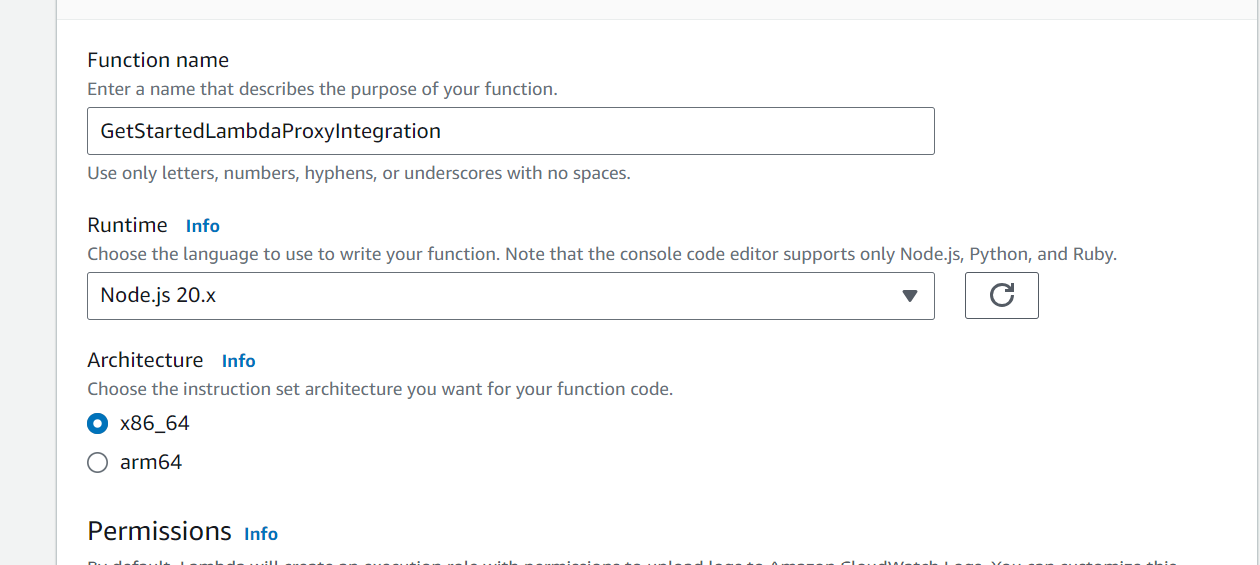
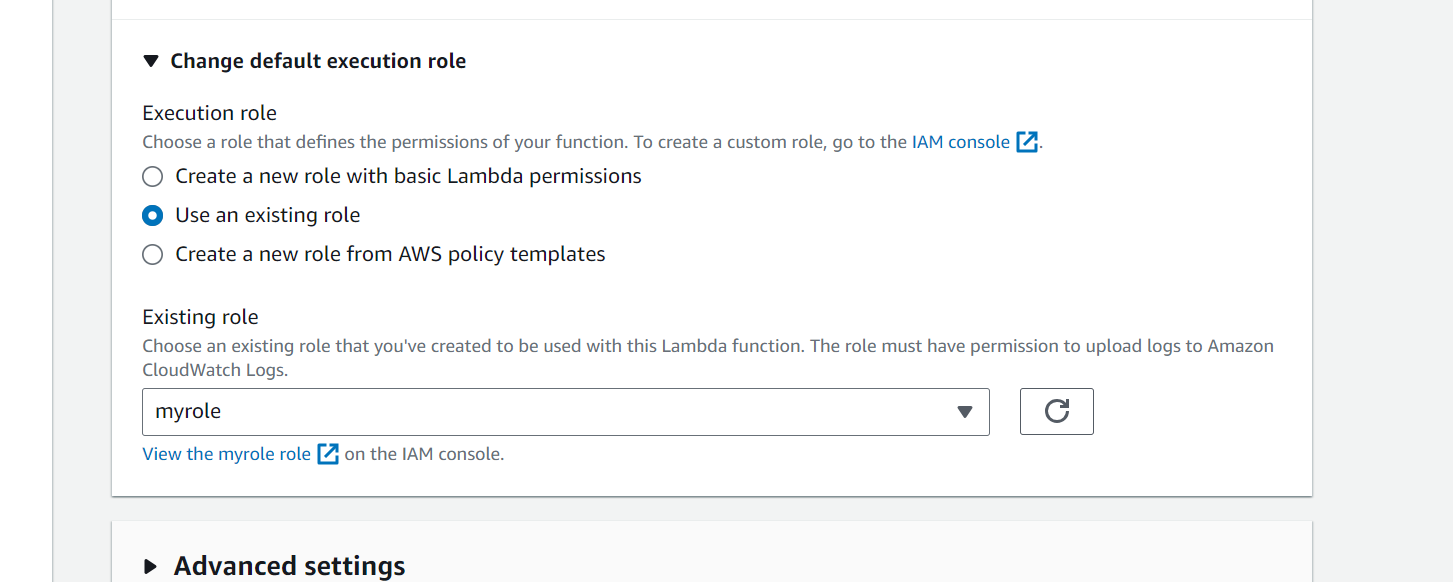
Lambda with Api Gateway

1. Sign in to the Lambda console at <https://console.aws.amazon.com/lambda>.
2. On the AWS navigation bar, choose an [AWS Region](https://docs.aws.amazon.com/general/latest/gr/apigateway.html).
3. Choose **Functions** in the navigation pane.
4. Choose **Create function**.
5. Choose **Author from scratch**.



1. Under **Basic information**, do the following:
   1. In **Function name**, enter **GetStartedLambdaProxyIntegration**.
   2. For **Runtime**, choose either the latest supported **Node.js** or **Python** runtime.
   3. Under **Permissions**, expand **Change default execution role**. Trainer will explain about roles.
   4. In **Role name**, enter **myrole**

.

* 1. Choose **Create function**.

1. Under **Function code**, in the inline code editor, copy/paste the following code:

export const handler = function(event, context, callback) {

console.log('Received event:', JSON.stringify(event, null, 2));

var res ={

"statusCode": 200,

"headers": {

"Content-Type": "\*/\*"

}

};

var greeter = 'World';

if (event.greeter && event.greeter!=="") {

greeter = event.greeter;

} else if (event.body && event.body !== "") {

var body = JSON.parse(event.body);

if (body.greeter && body.greeter !== "") {

greeter = body.greeter;

}

} else if (event.queryStringParameters && event.queryStringParameters.greeter && event.queryStringParameters.greeter !== "") {

greeter = event.queryStringParameters.greeter;

} else if (event.multiValueHeaders && event.multiValueHeaders.greeter && event.multiValueHeaders.greeter != "") {

greeter = event.multiValueHeaders.greeter.join(" and ");

} else if (event.headers && event.headers.greeter && event.headers.greeter != "") {

greeter = event.headers.greeter;

}

res.body = "Hello, " + greeter + "!";

callback(null, res);

};

1. Choose **Deploy**.

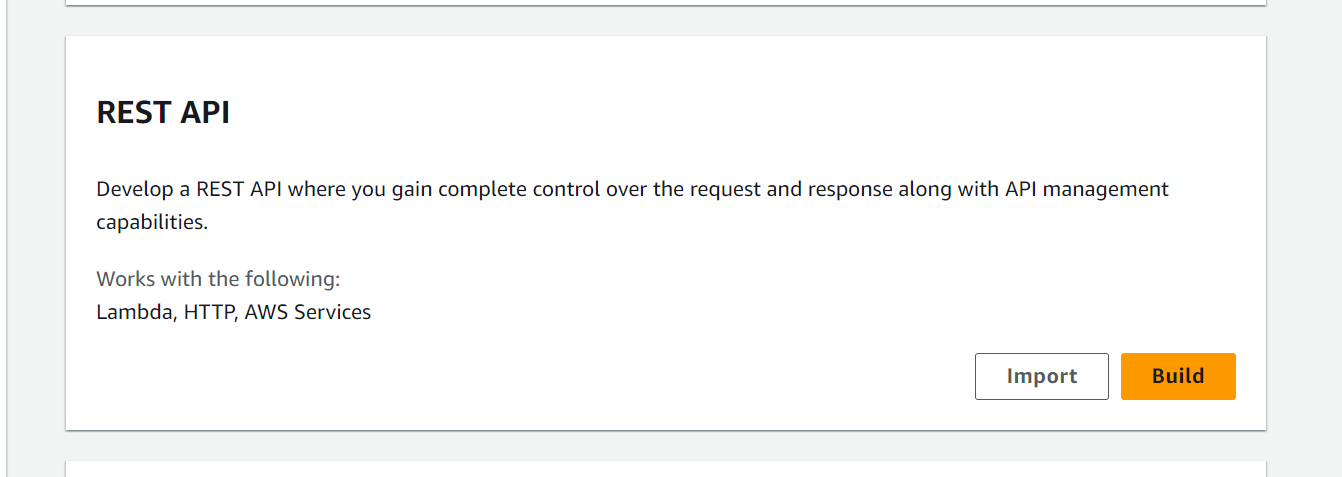
**Create a "Hello, World!" API**

Now create an API for your "Hello, World!" Lambda function by using the API Gateway console.

**To create a "Hello, World!" API**

1. Sign in to the API Gateway console at <https://console.aws.amazon.com/apigateway>.
2. If this is your first time using API Gateway, you see a page that introduces you to the features of the service. Under **REST API**, choose **Build**. When the **Create Example API** popup appears, choose **OK**.

If this is not your first time using API Gateway, choose **Create API**. Under **REST API**, choose **Build**.

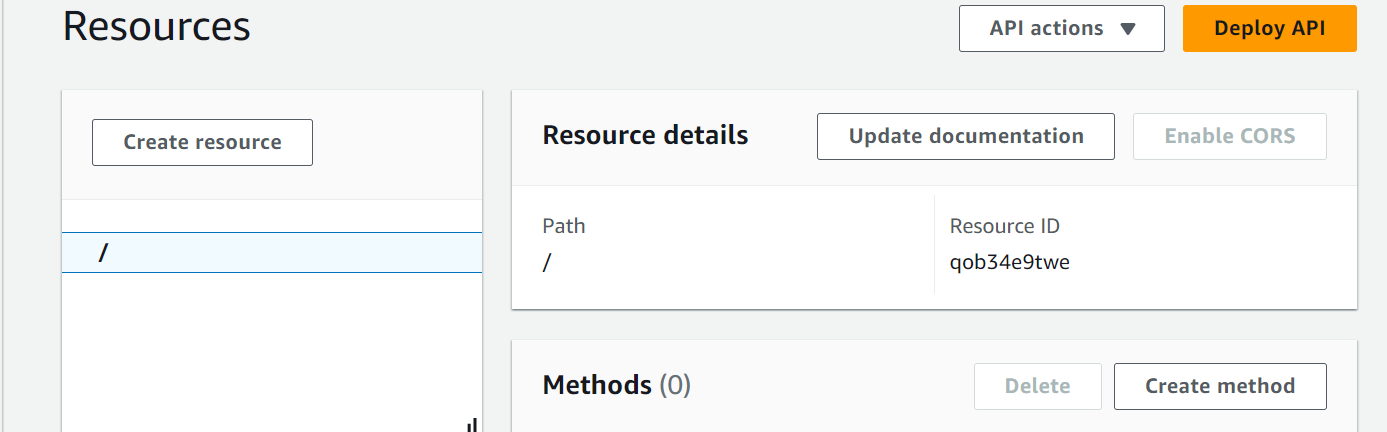


1. For **API name**, enter **LambdaProxyAPI**.
2. (Optional) For **Description**, enter a description.
3. Keep **API endpoint type** set to **Regional**.
4. Choose **Create API**.

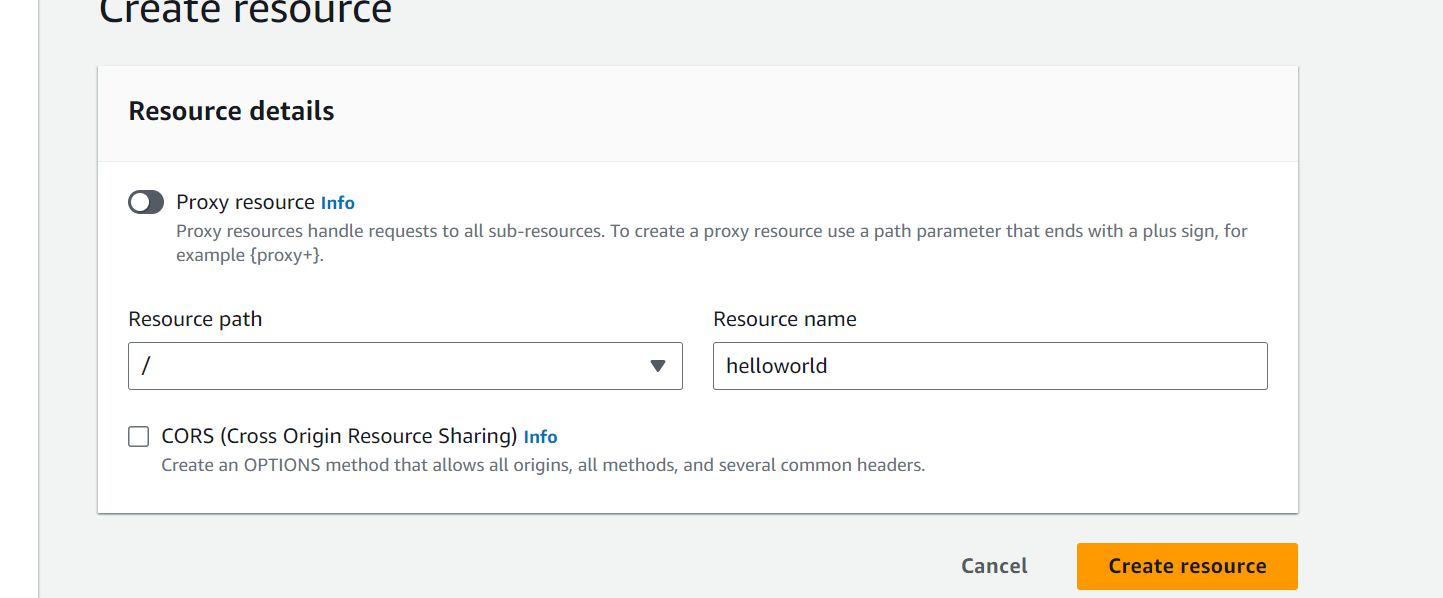
 For this example, you create a **/helloworld** resource.

**To create a resource**

1. Choose **Create resource**.



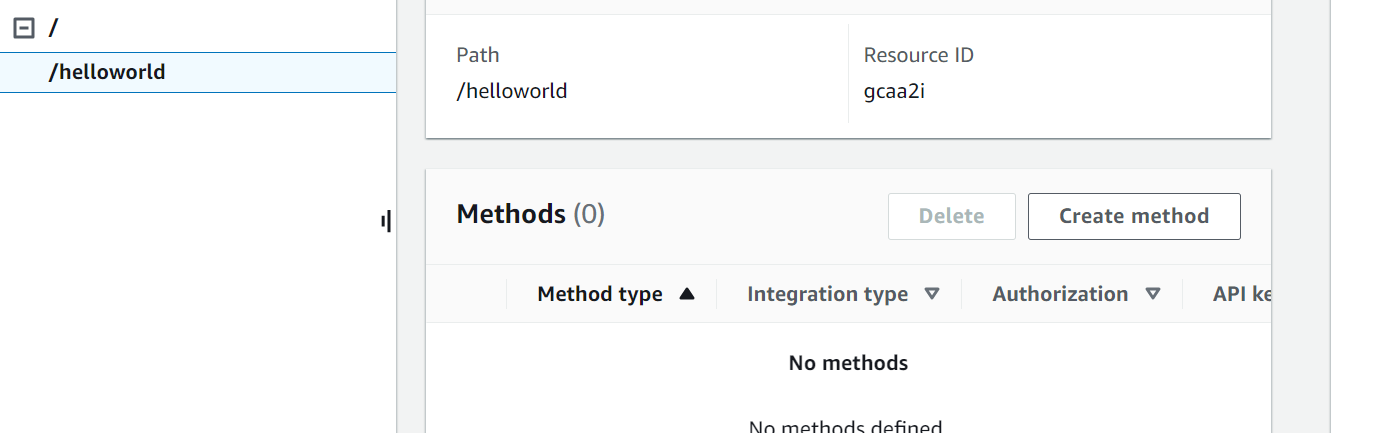
1. Keep **Proxy resource** turned off.
2. Keep **Resource path** as /.
3. For **Resource name**, enter **helloworld**.
4. Keep **CORS (Cross Origin Resource Sharing)** turned off.



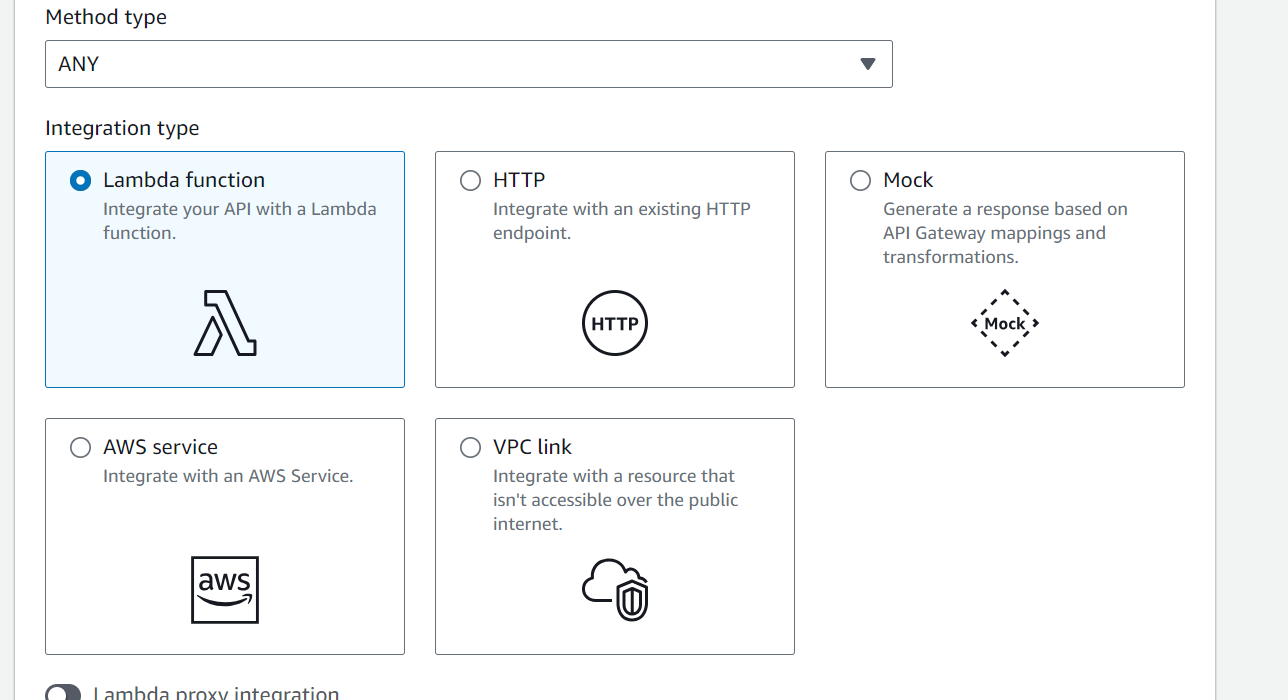
1. Choose **Create resource**.

**To create an ANY method**

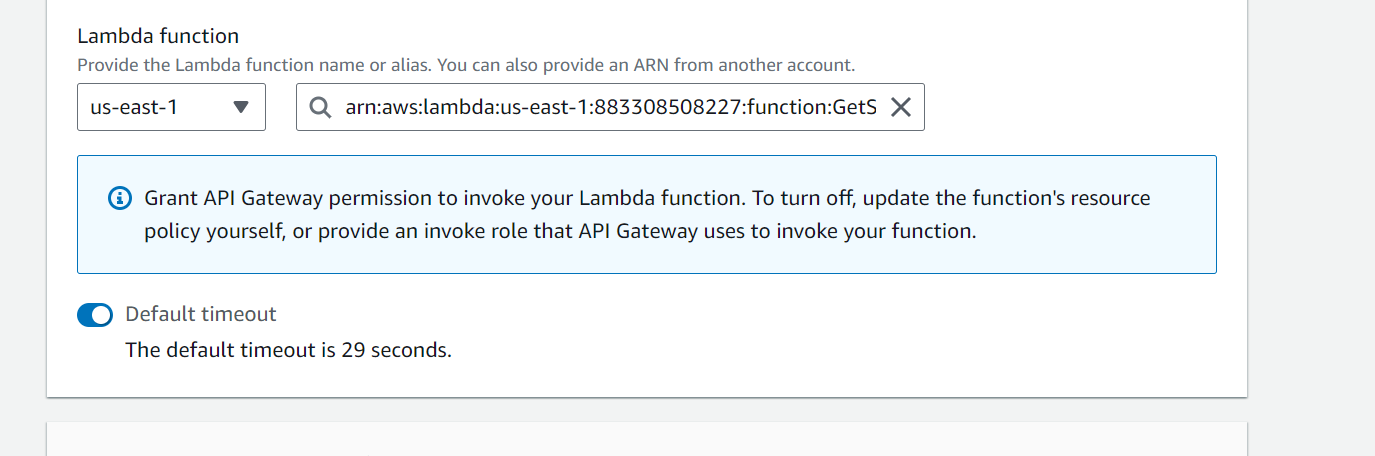
1. Select the **/helloworld** resource, and then choose **Create method**



1. For **Method type**, select **ANY**.



1. For **Integration type**, select **Lambda function**.
2. Turn on **Lambda proxy integration**.
3. For **Lambda function**, select the AWS Region where you created your Lambda function, and then enter the function name.

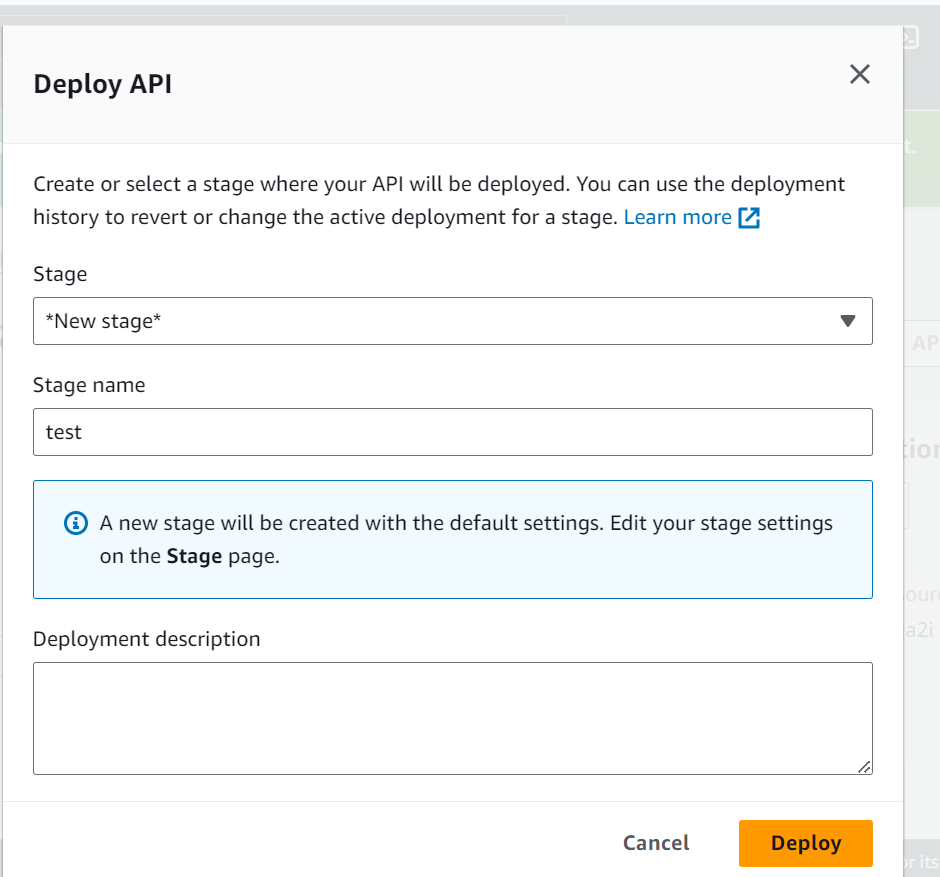


1. To use the default timeout value of 29 seconds, keep **Default timeout** turned on. To set a custom timeout, choose **Default timeout** and enter a timeout value between 50 and 29000 milliseconds.
2. Choose **Create method**.

**Deploy and test the API**

**To deploy your API**

1. Choose **Deploy API**.
2. For **Stage**, select **New stage**.
3. For **Stage name**, enter **test**.



1. (Optional) For **Description**, enter a description.
2. Choose **Deploy**.
3. Under **Stage details**, choose the copy icon to copy your API's invoke URL.

