**Visa\_Exchange\_App**

**Documentation**

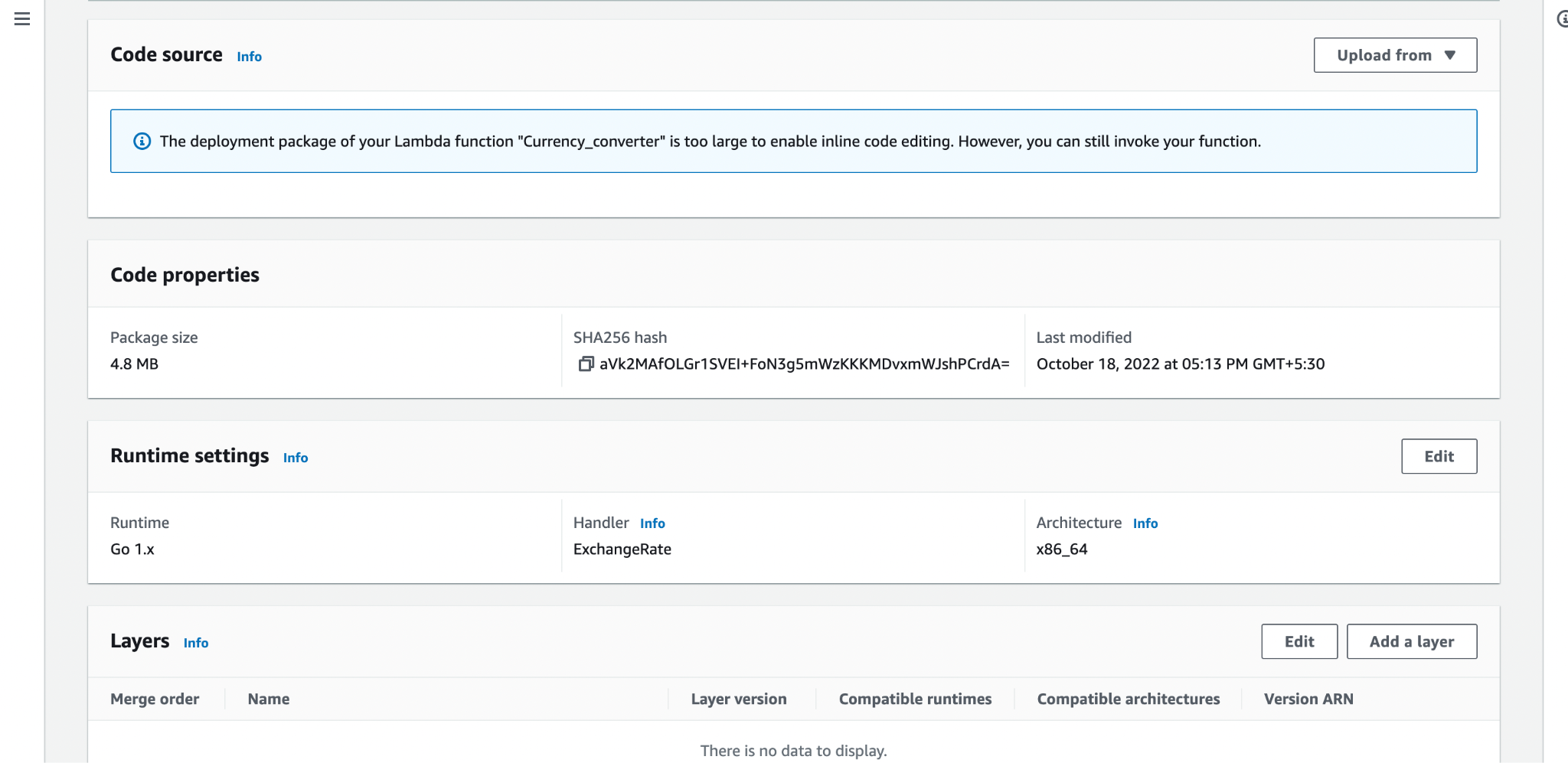
* Create an account on Visa Portal. You will have to choose a verification method from 2-way SSL verification and X-pay token. This app uses 2-way SSL verification .So, after entering your Project Name select 2-way SSL verification.
* While creating your Project you will have an option to download a certificate. It is the key certificate which we will use later. It will save as key.pem on your local machine.
* After that on your Visa Portal. Click on your Project Name. On the left hand side you will see a credentials tab. Click on that. You will have a Drop down on the main screen under Credentials Tab. There you will have a certificate .Download that and it will be saved as cert.pem on your local machine. Save your Username and Password which will be under the drop down option. Scroll down and download the Visa Platform Certificate . It will also be saved on your machine as Ca.pem.
* After that you will have some variable declared at the top of my Golang Code.
* var (
* // THIS IS EXAMPLE ONLY how will user\_id and password look like
* // userId = "1WM2TT4IHPXC8DQ5I3CH21n1rEBGK-Eyv\_oLdzE2VZpDqRn\_U";
* // password = "19JRVdej9";
* username = "K3PYCKYH05NYU0ICR4PF21wJsHcATCRBXGlN1a30FVwk\_B36A"
* password = "foMUNKg8TsqJIveUIgApgDLxPT0A3vxw8"
* // THIS IS EXAMPLE ONLY how will cert and key look like
* // clientCertificateFile = 'cert.pem'
* // clientCertificateKeyFile = 'key\_83d11ea6-a22d-4e52-b310-e0558816727d.pem'
* // caCertificateFile = 'ca\_bundle.pem'
* clientCertificateFile = GetCurrentPath() + "cert.pem" //"https://certifcates007.s3.ap-northeast-1.amazonaws.com/cert.pem"//arn:aws:lambda:ap-northeast-1:445192904874:layer:clientCertificateFile:1"//"/Users/rahul.kaushik/Downloads/cert.pem"
* clientCertificateKeyFile = GetCurrentPath() + "key.pem" //"https://certifcates007.s3.ap-northeast-1.amazonaws.com/key.pem"//"arn:aws:lambda:ap-northeast-1:445192904874:layer:clientCertificateKeyFile:1"///Users/rahul.kaushik/Downloads/key.pem"
* caCertificateFile = GetCurrentPath() + "ca.pem" //"https://certifcates007.s3.ap-northeast-1.amazonaws.com/ca.pem"//arn:aws:lambda:ap-northeast-1:445192904874:layer:caCertificateFile:1"///Users/rahul.kaushik/Downloads/ca.pem"
* )
* Enter your username and password which you got from credentials here.
* Also save all the .pem files in the same folder as the Golang code.You can change certificates name according to your need but you have to enter them in their respective fields.

**Steps to build the go file and zip the folder.**

* Open your terminal. Go to your folder directory. And execute the following commands.
* #!/usr/bin/env bash
* GOOS=linux go build -o ExchangeRate main.go
* zip -r ExchangeRate.zip .
* # Handle value should be ExchangeRate
* Here ExchangeRate is the name for binary executable and zip folder. Make sure you use the same name for both build and zip command.

**Steps to Deploy your code using Lambda Integration and Amazon API Gateway.**

* Create an account on Amazon Aws.
* Open the Lambda interface from the search bar.
* Create a new Lambda function.



* Here you can see that you have to select Go 1.x as runtime. X86\_64 as Architecture. Your Handler name should be the same as your zip file and executable name as mentioned before.Upload your zip file by clicking on Upload From button on lambda interface.
* Now, you can check if your Lambda function is working by using the Test option . You can check for a sample request such as.

{

"fromCurrencyCode": "840",

"toCurrencyCode": "356",

"fromAmount": "100"

}

You will get a successful response in Lambda log if you have executed the steps correctly.Response will be like:

### Execution result: succeeded([logs](https://ap-northeast-1.console.aws.amazon.com/cloudwatch/home?region=ap-northeast-1#logStream:group=%252Faws%252Flambda%252FCurrency_converter))

{

"fromCurrencyCode": "840",

"toCurrencyCode": "356",

"fromAmount": "100",

"destinationAmount": "6439.87",

"conversionRate": "64.44381"

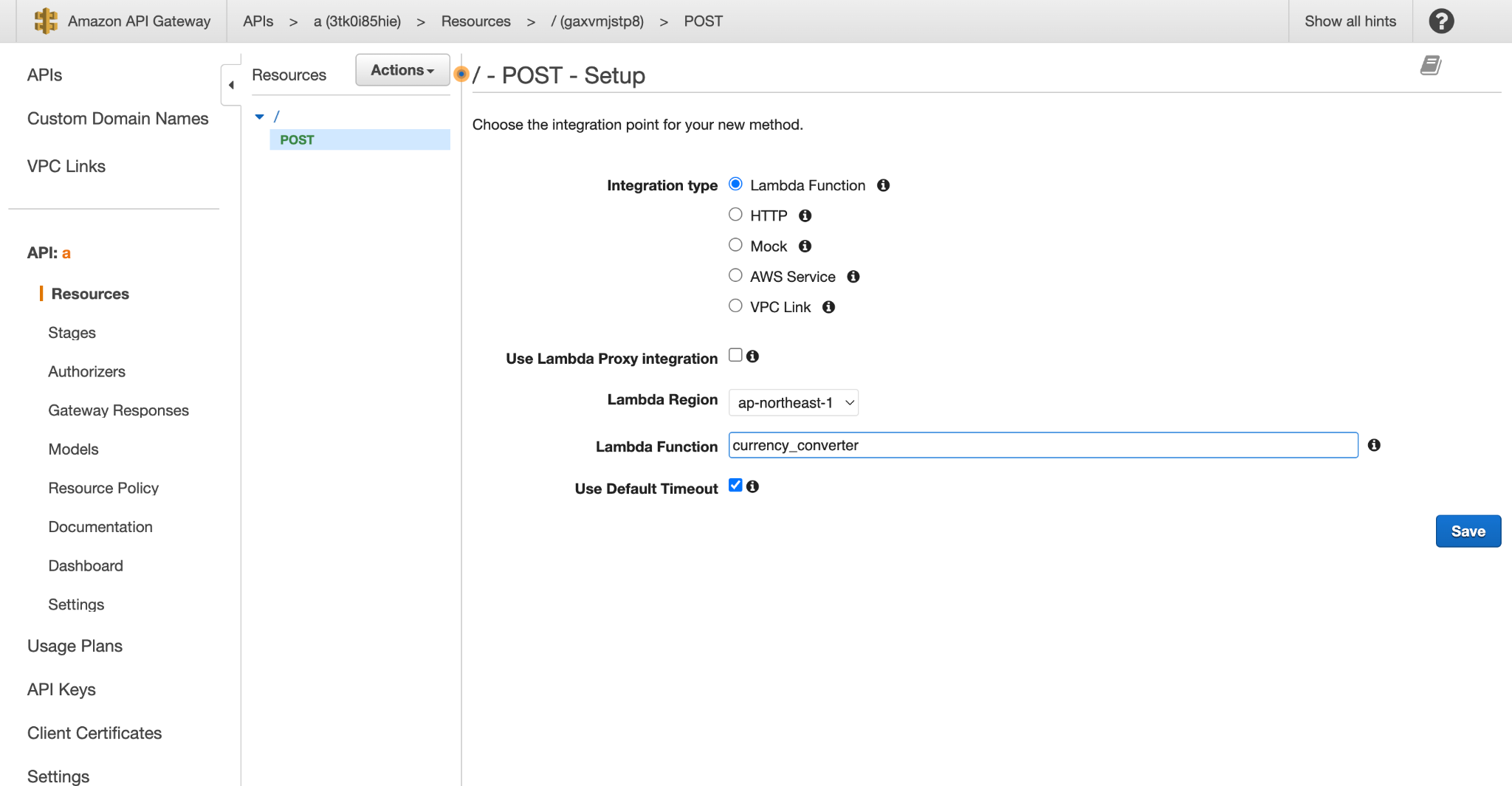
}

Now, it is sure that your code and lambda integration works fine.

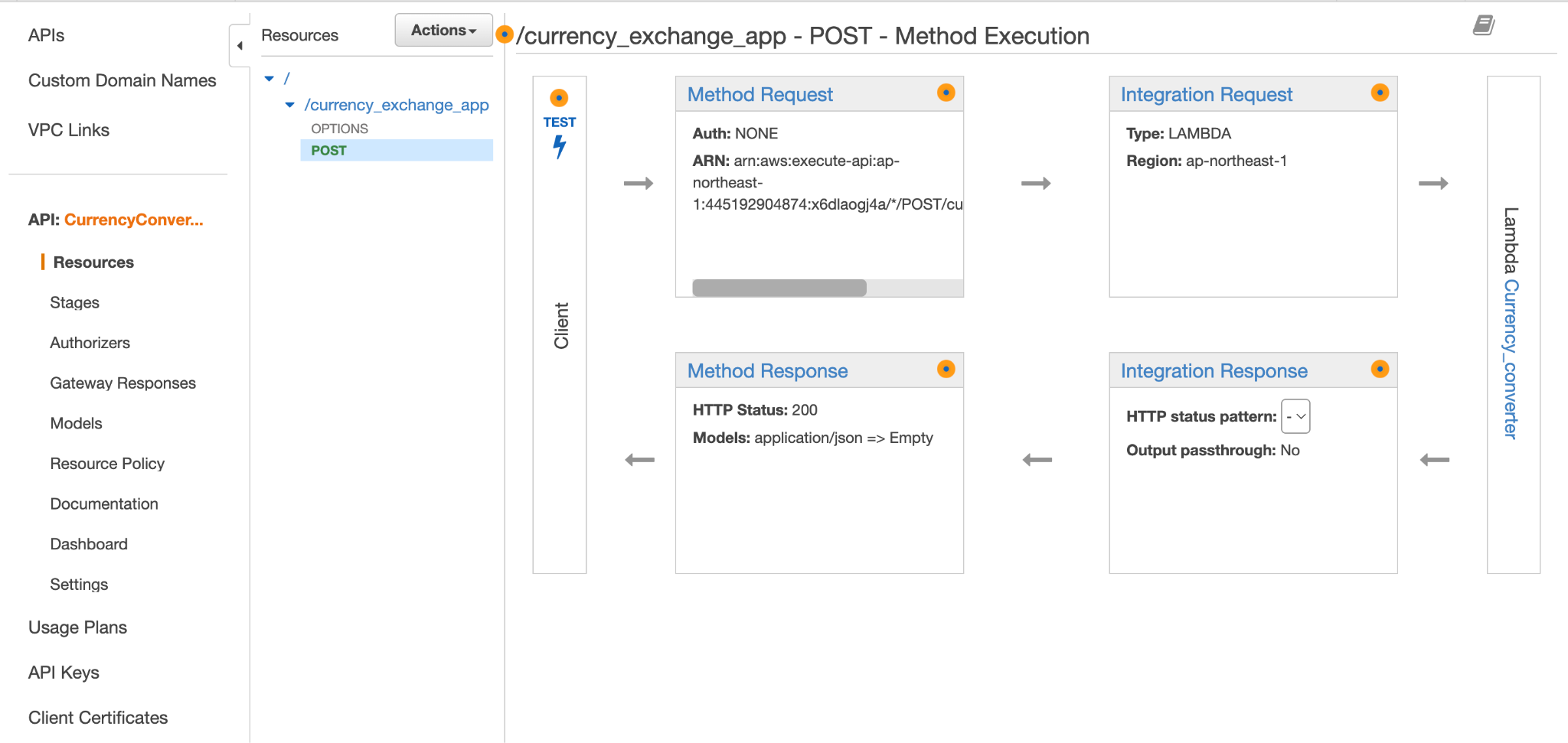
Now you have to expose your Lambda function on a Prod stage using Amazon Api Gateway.

Create an Api using REST Api and enter some details regarding name etc

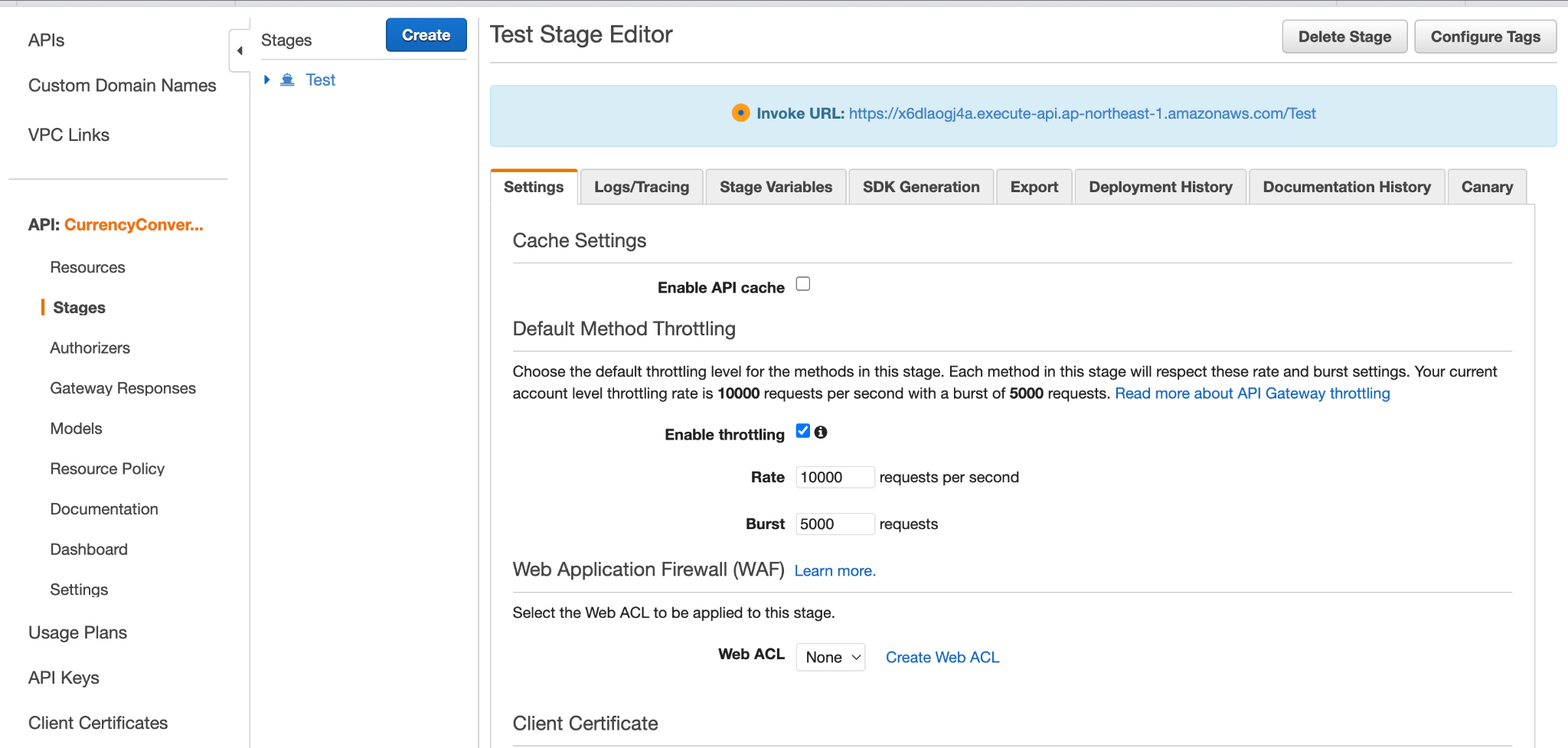
Then in your Api Interface . Click on the Action button. Select create method. Use Integration type as Lambda and method as POST.

Mention your lambda function In Lambda function option as shown below.

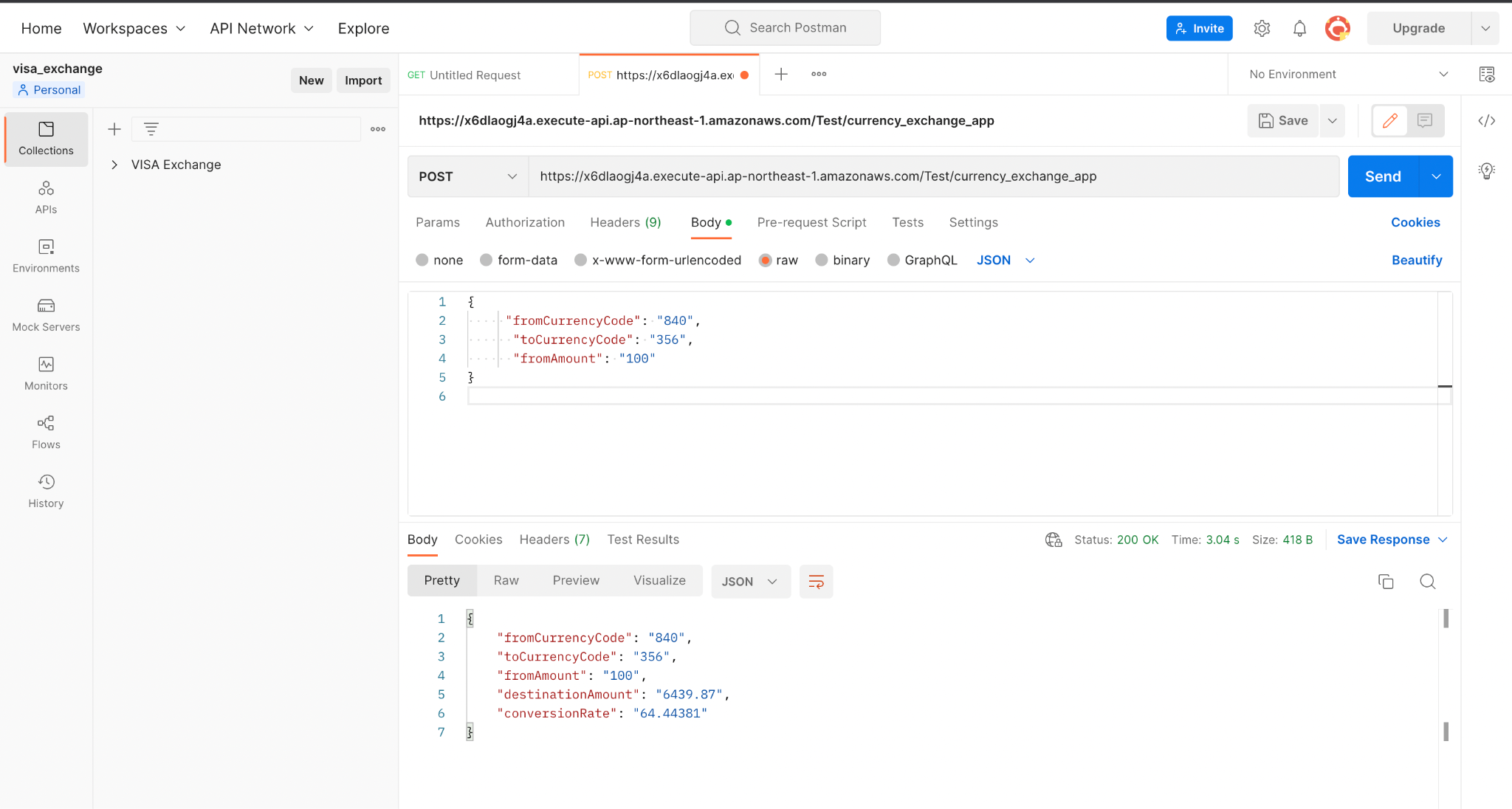
Click on save.



* Your Api Interface should look like this. Here click on Actions .Then select Deploy Api.
* Enter the deployment stage and description.



* Copy the invoke url . Your Api endpoint will be /You Api-name. Like mine is currency\_exchnage\_app which you can see above.Test with the same JSON request you used while Lambda testing on online Api platforms like Postman.
* Here is the Postman Response for the Api:-



If You get the same response as above then your Exchange app is live and can be used with any person with the above given Api .Please Make sure you enter valid Currency codes(ISO\_4217) which you can find here:-

<https://en.wikipedia.org/wiki/ISO_4217>

**End.**