



Apis - Práctico 2

Objetivo: Crear una API de REST que se integre a una lambda

Crearemos una lambda como backend y que reciba peticiones devolviendo un Hello-World con un saludo segun los parametros pasados y lo conectaremos a una API basado en una integración lambda

Utilizaremos un código escrito en Nodejs de ejemplo, pero recordemos que el objetivo no es centrarse en el código fuente de la lambda sino en su uso mediante APIS.

Nuestra función de backend devolverá un json con la forma
`"greeting": "Good {time}, {name} of {city}. [Happy {day}!]"`

Desarrollo:

Ingresa a la Consola de AWS y vamos a los servicios Lambda:

<https://console.aws.amazon.com/lambda>

Pasos:

1- Crear la función de backend

Elegir Create function

Elegir Author from scratch.

En Nombre colocar: GetStartedLambdaProxyIntegration

Elegir runtime 1.14

En Permissions expanda **Choose or create an execution role**

y luego Create new role from AWS policy templates

En rolename poner GetStartedLambdaBasicExecutionRole

y Create function

En Code function dentro del **index.js** poner:

```
'use strict';
console.log('Loading hello world function');

exports.handler = async (event) => {
    let name = "you";
    let city = 'World';
    let time = 'day';
    let day = '';
    let responseCode = 200;
    console.log("request: " + JSON.stringify(event));

    if (event.queryStringParameters &&
event.queryStringParameters.name) {
        console.log("Received name: " +
event.queryStringParameters.name);
        name = event.queryStringParameters.name;
    }

    if (event.queryStringParameters &&
event.queryStringParameters.city) {
        console.log("Received city: " +
event.queryStringParameters.city);
        city = event.queryStringParameters.city;
    }

    if (event.headers && event.headers['day']) {
        console.log("Received day: " + event.headers.day);
        day = event.headers.day;
    }

    if (event.body) {
```

```
        let body = JSON.parse(event.body)
        if (body.time)
            time = body.time;
    }

    let greeting = `Good ${time}, ${name} of ${city}.`;
    if (day) greeting += ` Happy ${day}!`;

    let responseBody = {
        message: greeting,
        input: event
    };

    // The output from a Lambda proxy integration must be
    // in the following JSON object. The 'headers' property
    // is for custom response headers in addition to standard
    // ones. The 'body' property must be a JSON string. For
    // base64-encoded payload, you must also set the
    'isBase64Encoded'
    // property to 'true'.
    let response = {
        statusCode: responseCode,
        headers: {
            "x-custom-header" : "my custom header value"
        },
        body: JSON.stringify(responseBody)
    };
    console.log("response: " + JSON.stringify(response))
    return response;
};
```

Elegir **Deploy**

2- Creación de la API

Vamos al servicio de Apis:

<https://console.aws.amazon.com/apigateway>

En Create new API (Crear nueva API), elija New API (Nueva API).

DIPLOMATURA DEVOPS - GUIA PRACTICA

En Settings (Configuración):

En API name (Nombre de la API), escriba LambdaSimpleProxy.

Seleccione Create API (Crear API)

Creamos el recurso:

Elija el recurso raíz (/) en el árbol Resources (Recursos).

En el menú desplegable Actions (Acciones), elija Create Resource (Crear recurso).

Deje Configure as proxy resource (Configurar como recurso de proxy) sin marcar.

En Resource Name escribir helloworld.

Deje Resource Path establecido en /helloworld

Deje Enable API Gateway CORS sin marcar

Elija Create Resource

Configuramos un método ANY

En la lista Resources elija /helloworld

En el menú Actions, elija Create method

Seleccione ANY en el menú desplegable y, a continuación, seleccione el icono de marca de verificación

Dejar Integration type establecido en Lambda Function

Seleccionar Use Lambda Proxy integration

Desde el menú desplegable Lambda Region, seleccione la región en la que ha creado la función de Lambda
GetStartedLambdaProxyIntegration

En el campo Lambda Function, escriba cualquier carácter y elija GetStartedLambdaProxyIntegration en el menú desplegable

Deje la casilla Use Default Timeout (Usar tiempo de espera predeterminado) desactivada.

DIPLOMATURA DEVOPS - GUIA PRACTICA

Seleccione Save.

Elija OK (Aceptar) cuando se le solicite Add Permission to Lambda Function

3- Deploy de la API

Realizar un deploy como en el practico 1 y darle Name: test

4- Prueba de la API:

Podemos utilizar curl o postman. Las consultas GET también las podemos probar desde un Navegador web.

```
curl -X POST  
'https://e44rzrbz6b.execute-api.us-east-1.amazonaws.com/test/h  
elloworld?name=Marce&city=Cordoba' -H 'content-type:  
application/json' -H 'day: Martes' -d '{ "time": "evening"  
'
```

```
curl -X GET  
'https://e44rzrbz6b.execute-api.us-east-1.amazonaws.com/test/h  
elloworld?name=Luis&city=Cordoba' -H 'content-type:  
application/json' -H 'day: Martes'
```