

- Cotizar

Dynamodb

Crear tabla en dynamodb con el identificadar como uuid

## Crear una tabla de DynamoDB

Tutorial



DynamoDB es una base de datos sin esquema que solo necesita un nombre de tabla y una clave principal. La clave principal de la tabla está compuesta de uno o dos atributos que identifican de manera inequívoca cada elemento, efectúan la partición de datos y ordenan los datos dentro de cada partición.

Nombre de la tabla\* Tabla\_Cotizaciones



Clave principal\* Clave de partición

uuid

Cadena



☐ Añadir clave de ordenación

## Tabla creada

Tabla\_Cotizaciones Cerrar

Eliminar tabla

Información general **Elementos** Métricas Alarmas Capacidad Índices Tablas globales Copias de seguridad Contributor Insights Más

Crear elemento Acciones

Examen: [Tabla] Tabla\_Cotizaciones: uuid

Mostrando 0 de 0 elementos

Examen [Tabla] Tabla\_Cotizaciones: uuid

Añadir filtro

Iniciar búsqueda

uuid

## Ejemplo de la tabla

Tabla\_Cotizaciones Cerrar

Eliminar tabla

Información general **Elementos** Métricas Alarmas Capacidad Índices Tablas globales Copias de seguridad Contributor Insights Más

Crear elemento Acciones

Examen: [Tabla] Tabla\_Cotizaciones: uuid

Mostrando 1 de 1 elementos

Examen [Tabla] Tabla\_Cotizaciones: uuid

Añadir filtro

Iniciar búsqueda

uuid	Browser	Device	MonedaDestino	Usuario	UsuarioIP	Cantidad_Moneda	TimeStamp	Car
deb1d662-bc3b-4439-a2f4-9304c9c4bb39	chrome	Celular	BTC	Santiago	192.168.1.1	0.005	12:00	5

## Lambda

### Crear una función

#### Crear una función Info

Seleccione una de las siguientes opciones para crear la función.

**Crear desde cero** ☒

Empiece con un sencillo ejemplo "Hello World".

**Utilizar un proyecto** ☐

Cree una aplicación Lambda utilizando un código de muestra y los ajustes de configuración predefinidos de casos de uso comunes.

**Imagen del contenedor** ☐

Seleccione una imagen de contenedor para implementar para la función.

**Examinar el repositorio de aplicaciones sin servidor** ☐

Implemente una aplicación Lambda de ejemplo desde AWS Serverless Application Repository.

#### Información básica

**Nombre de la función**  
Escriba un nombre para describir el propósito de la función.

Utilice exclusivamente letras, números, guiones o guiones bajos. No incluya espacios.

**Tiempo de ejecución** Info  
Elija el lenguaje que desea utilizar para escribir la función. Tenga en cuenta que el editor de código de la consola solo admite Node.js, Python y Ruby.

Node.js 14.x

▼

### Usar un rol existente

▼ Cambiar el rol de ejecución predeterminado

**Rol de ejecución**  
Seleccione un rol que defina los permisos de la función. Para crear un rol personalizado, vaya a la [consola de IAM](#).

☐ Creación de un nuevo rol con permisos básicos de Lambda

☒ Uso de un rol existente

☐ Creación de un nuevo rol desde la política de AWS templates

**Rol existente**  
Seleccione un rol existente que haya creado para usarlo con esta función de Lambda. El rol debe tener permiso para cargar registros en Amazon CloudWatch Logs.

service-role/Tabla\_Cotizar\_Escritura-role-vkojais6

▼



[Consulte el rol Tabla\\_Cotizar\\_Escritura-role-vkojais6](#) en la consola de IAM.

```
index.js x +
1 const {v4: uuidv4} = require('uuid');
2 const AWS = require('aws-sdk');
i 3 AWS.config.update({region:'us-east-2'})
i 4 const docClient = new AWS.DynamoDB.DocumentClient()
5
6 exports.handler=(event,context,callback)=>{
7     var params={
8         TableName:"Tabla_Cotizaciones",
9         Item:{
10             "uuid":uuidv4(),
11             "Usuario":event.usuario,
12             "CantidadDolares":event.cantidad,
13             "MonedaDestino":event.CoinDest,
14             "UsuarioIP":event.Usuarioip,
15             "Browser":event.browser,
16             "Device":event.device,
17         }
18     };
19     docClient.put(params,function(err,data){
20         if(err){
21             callback(err,null);
22         }else{
i 23             callback(null,"Informacion Guardada")
24         }
25     });
i 26 }
```

API Gateway

## API REST

Desarrolle una API REST en la que obtenga control total de la solicitud y la respuesta, junto con las capacidades de administración de la API.

Funciona con lo siguiente:  
Lambda, HTTP, servicios de AWS

Importar

Crear

### Elegir el protocolo

Seleccione si desea crear una API de REST o una API de WebSocket.

☒ REST ☐ WebSocket

### Crear API nueva

En Amazon API Gateway, una API de REST hace referencia a una colección de recursos y métodos que se pueden invocar a través de puntos de enlace HTTPS.

☒ API nueva ☐ Clonar de API existente ☐ Importar de Swagger u Open API 3 ☐ API de ejemplo

### Configuración

Elija un nombre o una descripción fáciles de recordar para su API.

Nombre de API\*   
Descripción   
Tipo de punto de enlace



Amazon API Gateway API > Tabla\_Cotizaciones (top2rb438h) > Recursos > / (qlgp05i262) > Crear

Mostrar todas las sugerencias



Recursos

Acciones

### Nuevo recurso secundario

Utilice esta página para crear un nuevo recurso secundario para su recurso.

Configurar como [recurso de proxy](#)

☐

Nombre del recurso\*

Ruta de recurso\*

Puede agregar parámetros de ruta utilizando llaves. Por ejemplo, la ruta de recurso **(username)** representa un parámetro de ruta denominado "username". Al configurar **(proxy+)** como un recurso de proxy, se recuperan todas las solicitudes en sus recursos secundarios. Por ejemplo, funciona en una solicitud GET a /foo. Para controlar las solicitudes a /, agregue un nuevo método ANY en el recurso /.

Habilitar API Gateway CORS

☒

\* Obligatorio

Cancelar

Crear recurso

Amazon API Gateway

API > Tabla\_Cotizaciones (top2rb438h) > Recursos > /escribir (v45ckw) > POST

Mostrar todas las sugerencias ?

Recursos

Acciones

/escribir - POST - Configuración

/

/escribir

OPTIONS

POST

Elija el punto de integración del nuevo método.

Tipo de integración

☒ Función Lambda

☐ HTTP

☐ Simulación

☐ Servicio de AWS

☐ Enlace de VPC

Usar la integración de proxy Lambda

☐

Región Lambda

us-east-2

Función Lambda

Lambda\_Cotizar

Usar tiempo de espera predeterminado

☒

Guardar

Implementar la API

×

Elija una etapa donde se implementará su API. Por ejemplo, una versión de prueba de su API se podría implementar en una etapa denominada beta.

Etapas de implementación

[Nueva etapa]

▼

Nombre de la fase\*

Escribir

Descripción de etapa

⌵

Descripción de implementación

⌵

Cancelar

Implementación

- # Comprar

Dynamodb

Tabla de Compras



Crear una lambda para enviar a la tabla



```

1  const {v4: uuidv4} = require('uuid');
2  const AWS = require('aws-sdk');
3  AWS.config.update({region: 'us-east-2'})
4  const docClient = new AWS.DynamoDB.DocumentClient()
5
6  exports.handler=(event,context,callback)=>{
7      let uuid_dato =uuidv4()
8      var params={
9          TableName:"Tabla_Compras",
10         Item:{
11             "uuid":uuid_dato,
12             "uuid_Cotizacion":event.id_Cotizador,
13             "TimeStamp":event.tiempo,
14             "Usuario":event.usuario,
15             "Tipo_Cambio":event.cambio,
16             "MonedaDestino":event.CoinDest,
17             "Cantidad_Original":event.cantidad,
18             "Cantidad_Moneda_Comprada":event.resultado,
19             "UsuarioIP":event.Usuarioip,
20             "Browser":event.browser,
21             "Device":event.device,
22             "Geolocalizacion":event.geolocalizacion
23         },
24         headers:{
25             "Access-Control-Allow-Headers":"Content-Type",
26             "Access-Control-Allow-Origin":"*",
27             "Access-Control-Allow-Methods":"POST"
28         },
29     };
30     docClient.put(params,function(err,data){
31         if(err){
32             callback(err,null);
33         }else{
34             callback(null,uuid_dato)
35         }
36     });
37 }

```



Elegir el protocolo

Seleccione si desea crear una API de REST o una API de WebSocket.
 

☒ REST
 ☐ WebSocket

Crear API nueva

En Amazon API Gateway, una API de REST hace referencia a una colección de recursos y métodos que se pueden invocar a través de puntos de enlace HTTPS.
 

☒ API nueva
 ☐ Clonar de API existente
 ☐ Importar de Swagger u Open API 3
 ☐ API de ejemplo

Configuración

Elija un nombre o una descripción fáciles de recordar para su API.
 

Nombre de API\*

Lambda\_Comprar

Descripción

funciona con Lambda\_Compra

Tipo de punto de enlace

Regional

?

## Consultar la tabla

```

1  const AWS = require('aws-sdk');
2  AWS.config.update({region: 'us-east-2'});
3  const ddb = new AWS.DynamoDB.DocumentClient();
4  exports.handler = (event, context, callback) => {
5      let params = {
6          TableName: "Tabla_Compras",
7          headers: {
8              "Access-Control-Allow-Headers": "*",
9              "Access-Control-Allow-Origin": "*",
10             "Access-Control-Allow-Methods": "*"
11         },
12         Key: {
13             "uuid": event.Id
14         }
15     };
16     ddb.get(params, function(err, data) {
17         if(err) {
18             callback("un error" + err, null);
19         } else {
20             callback(null, JSON.stringify(data));
21         }
22     })
23 };

```



Ventana

Cotizador

[INICIO](#)

De

Moneda

A

Moneda

COTIZAR

COMPRAR 00

CONSULTAR COMPRA