

| Guía :

API RESTful CRUD en AWS EC2 t2.micro con pruebas en Postman

Sigla	Asignatura	Experiencia de Aprendizaje
DSY1104	Desarrollo Full Stack II	EA Integración RESTful con Google Cloud Platform
Tiempo	Modalidad de Trabajo	Indicadores de logro
2 h	Individual	IL 4.3



Antecedentes generales

Esta guía tiene como objetivo enumerar las acciones necesarias para dar solución a los problemas planteados.



Requerimientos para esta actividad

Para el desarrollo de esta actividad deberás disponer de:

- Computador
- AWS – EC2



Actividad

Esta actividad consiste en enumerar las acciones necesarias para dar solución a los casos que se verán a continuación, para ello los estudiantes deberán realizar la actividad de forma individual.

Sigue las Instrucciones

Configuración de AWS EC2

1. Inicia sesión en la consola de AWS.
2. Navega a EC2 y haz clic en "Launch Instance".
3. Selecciona "Amazon Linux 2 AMI".
4. Elige el tipo de instancia t2.micro.
5. Configura los detalles de la instancia (usa la configuración por defecto para empezar).
6. En el paso de configuración del grupo de seguridad, asegúrate de abrir el puerto 22 (SSH) y el puerto 3000 (para nuestra API).
7. Crea o selecciona un par de claves existente y lanza la instancia.

Conexión a la instancia EC2

1. Abre tu terminal.
2. Usa el siguiente comando para conectarte (reemplaza con tu ruta de clave y dirección IP pública):

```
...  
  
ssh -i "tu-clave.pem" ec2-user@tu-ip-publica  
...
```

Configuración del entorno

1. Actualiza los paquetes:

```
...  
  
sudo yum update -y  
...
```

2. Instala Node.js:

```
...  
  
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash  
.  
~/.nvm/nvm.sh  
nvm install node  
...
```

3. Verifica la instalación:

```
...  
  
node -v  
npm -v  
...
```

Creación de la API RESTful

1. Crea un nuevo directorio y navega a él:

```
...  
  
mkdir crud-api  
cd crud-api  
...
```

2. Inicializa un nuevo proyecto Node.js:

```
...  
  
npm init -y  
...
```



3. Instala las dependencias necesarias:

```
...  
  
npm install express body-parser  
  
...
```

4. Crea un archivo `app.js`:

```
...  
  
nano app.js  
  
...
```

5. Añade el siguiente código a `app.js`:

```
````javascript  
const express = require('express');
const bodyParser = require('body-parser');

const app = express();
const port = 3000;

app.use(bodyParser.json());

let items = [];

// Create
app.post('/items', (req, res) => {
 const item = req.body;
 item.id = items.length + 1;
 items.push(item);
 res.status(201).json(item);
});

// Read (all)
app.get('/items', (req, res) => {
 res.json(items);
});

// Read (one)
app.get('/items/:id', (req, res) => {
 const item = items.find(i => i.id === parseInt(req.params.id));
 if (item) {
 res.json(item);
 }
});
```

```
 } else {
 res.status(404).send('Item not found');
 }
 });

// Update
app.put('/items/:id', (req, res) => {
 const index = items.findIndex(i => i.id === parseInt(req.params.id));
 if (index !== -1) {
 items[index] = { ...items[index], ...req.body, id: items[index].id };
 res.json(items[index]);
 } else {
 res.status(404).send('Item not found');
 }
});

// Delete
app.delete('/items/:id', (req, res) => {
 const index = items.findIndex(i => i.id === parseInt(req.params.id));
 if (index !== -1) {
 items.splice(index, 1);
 res.status(204).send();
 } else {
 res.status(404).send('Item not found');
 }
});

app.listen(port, () => {
 console.log(`API running on port ${port}`);
});
...

```

6. Guarda y cierra el archivo (en nano: Ctrl+X, luego Y, luego Enter).

## Ejecución de la API

1. Inicia la API:

```
...
node app.js
...
```

2. La API ahora debería estar corriendo en el puerto 3000.

## Configuración de Postman

1. Descarga e instala Postman desde:

[postman.com](<https://www.postman.com/downloads/>).

2. Abre Postman y crea una nueva colección llamada "CRUD API".

## Pruebas con Postman

Usa la IP pública de tu instancia EC2 en lugar de ``tu-ip-publica``.

1. Create (POST):

- URL: ``http://tu-ip-publica:3000/items``

- Method: POST

- Body (raw JSON):

```
```json
{
  "name": "Item 1",
  "description": "This is item 1"
}
...`
```

2. Read All (GET):

- URL: ``http://tu-ip-publica:3000/items``

- Method: GET

3. Read One (GET):

- URL: ``http://tu-ip-publica:3000/items/1``

- Method: GET

4. Update (PUT):

- URL: ``http://tu-ip-publica:3000/items/1``

- Method: PUT

- Body (raw JSON):

```
```json
{
 "name": "Updated Item 1",
 "description": "This is the updated item 1"
}
...`
```

#### 5. Delete (DELETE):

- URL: `http://tu-ip-publica:3000/items/1`
- Method: DELETE

### Mantenimiento y Seguridad

#### 1. Mantén tu instancia EC2 actualizada:

...

```
sudo yum update -y
```

...

#### 2. Considera implementar HTTPS para conexiones seguras.

#### 3. Implementa autenticación y autorización para proteger tu API.

#### 4. Monitorea el uso de recursos de tu instancia t2.micro para asegurarte de que satisface tus necesidades.

#### 9. Cierre de la instancia

Cuando hayas terminado de usar la instancia:

1. Ve a la consola de EC2 en AWS.
2. Selecciona tu instancia.
3. Haz clic en "Instance State" > "Terminate" para detener y eliminar la instancia.

Recuerda que terminar una instancia eliminará todos los datos asociados a ella.

### Recursos de apoyo

AWS. (s/f). Amazon Web Services, Inc. Recuperado el 11 de agosto de 2024, de <https://aws.amazon.com/es/>