

# FinTechBank Client Management API

Este proyecto consta básicamente de un Web API para la gestión de datos de clientes bancarios en FinTechBank. A continuación, se detallan los pasos para ejecutar el proyecto.

**NOTA:** Puede encontrar el código fuente en el siguiente repositorio de GitHub: [client-manager-app](#).

## Pasos para Ejecutar el Proyecto

### 1. Levantar una Base de Datos PostgreSQL:

- En la carpeta `Backend/MyWebApi/Resources` encontrará un archivo llamado `init.sql`. Este archivo contiene un script SQL para crear un usuario para la aplicación, la base de datos, otorgar permisos al usuario, crear las tablas necesarias e insertar algunos datos de prueba.

### 2. Descargar la Imagen Docker:

- Ejecute el siguiente comando para obtener la imagen Docker desde Docker Hub y ejecutarla en un entorno de desarrollo:

```
# Get image from Docker Hub
docker pull ldavidflorez/client-manager-app:v3

# Start container
docker run -d \
--network host \
-e ClientConnection="Host=localhost;Port=5432;Database=clientsdb;Username=myuser;Password=mypassword;" \
--name \webapi client-manager-app
```

- Si está ejecutando en un entorno productivo, utilice el siguiente comando, reemplazando `YOUR_DB_HOST_IP` por la IP del servidor donde está alojada la base de datos:

```
docker run -d \
-e ClientConnection="Host=YOUR_DB_HOST_IP;Port=5432;Database=clientsdb;Username=myuser;Password=mypassword;" \
-p 5001:5001 \
--name webapi client-manager-app
```

- Si desean ver el archivo `Dockerfile` con el que se construyó la imagen, lo pueden encontrar en la ruta `Backend/MyWebApi/Dockerfile`

### 3. Probar la API:

- Utilice el archivo de colección de Postman presente en `Backend/MyWebApi/Resources` (archivo `PostmanCollection.json`). Este archivo define los siguientes endpoints:

- `POST /api/Auth`: para autenticación, donde se obtiene un JWT. Ejemplo de JSON de solicitud:

```
{
  "userName": "john_doe",
  "password": "pass1"
}
```

- `GET /api/Clients`: para listar todos los clientes existentes en la base de datos.
- `GET /api/Clients/{id}`: para consultar un cliente por su ID.
- `POST /api/Clients`: para crear un nuevo cliente. Ejemplo de JSON de solicitud:

```
{
  "name": "A Name",
  "lastname": "A Lastname",
  "accountNumber": "QWE123890",
  "balance": 0,
  "dateOfBirth": "2024-05-15T03:59:14.090Z",
  "address": "An address",
  "phone": "555-246",
  "email": "email.example@some.com",
  "clientType": "corporate",
  "maritalStatus": "single",
  "identificationNumber": "ID098123567",
  "profession": "Developer",
  "gender": "Male",
  "nationality": "Colombian"
}
```

- `PUT /api/Clients/{id}`: para actualizar los datos de un cliente existente. Ejemplo de JSON de solicitud:

```
{
  "name": "A Name",
  "lastname": "A Lastname",
  "accountNumber": "QWE123890",
  "balance": 1000,
  "dateOfBirth": "2024-05-15T03:59:14.090Z",
  "address": "An address",
  "phone": "555-246",
  "email": "email.example@some.com",
  "clientType": "corporate",
  "maritalStatus": "single",
  "identificationNumber": "ID098123567",
  "profession": "Developer",
  "gender": "Male",
  "nationality": "Colombian"
}
```

- DELETE /api/Clients/{id} : para eliminar un cliente.
- Es necesario configurar un header que contenga el JWT retornado por /api/Auth , como se muestra en el siguiente ejemplo:

```
Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

- Tenga en cuenta que existen dos tipos de roles: `admin` y `consultant` , teniendo ambos los mismos privilegios, excepto el rol `admin` que es el único que puede borrar clientes.

#### 4. Ejecutar Pruebas de la Aplicación:

- Diríjase al proyecto `Backend/MyWebApi.Test` y ejecute los siguientes comandos en la terminal:

```
dotnet restore
dotnet test
```

#### 5. Ejecutar el Proyecto sin Docker:

- Dentro del proyecto `Backend/MyWebApi` , ejecute los siguientes comandos en la terminal:

```
dotnet restore
dotnet run
```

## Soporte y Contribuciones

Si encuentra algún problema o tiene alguna pregunta, no dude en [abrir un issue](#). ¡Las contribuciones son bienvenidas!