LIFEBANK REST API

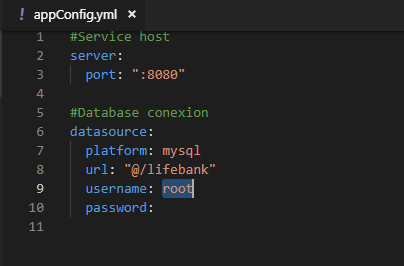
Desarrollado por: Jacquelin Gudalupe Angel Caballero

Las especificaciones del api son:

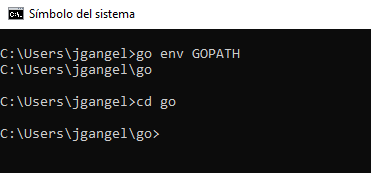
|  |  |
| --- | --- |
| LIFEBANK – API REST | |
| Lenguaje de desarrollo | Go (golang) |
| Editor de codigo | Visual Studio Code |
| Motor de Base de datos | Mysql |
| Puerto que usa el servicio | 8080 |
| Herramientas de prueba | Postman |

Los pasos para levantar la API son:

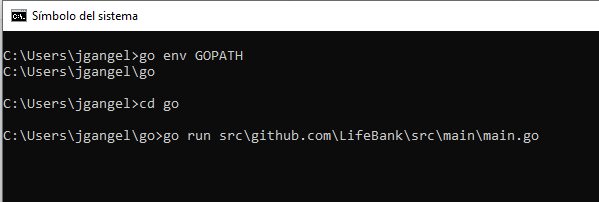
1. Correr el script de la base de datos (lifebank\_script.sql) que se encuentra en el zip (Jacqueline Angel - v3)
2. Para correr el api sin el compilado es necesario que posea el inteprete de Go y descargar desde la consola de git las siguientes librerías con los siguientes comandos:
   1. "github.com/gorilla/mux” comando: go get -u github.com/gorilla/mux
   2. "github.com/jinzhu/gorm" comando: go get -u github.com/jinzhu/gorm
   3. "github.com/olebedev/config" comando: go get -u github.com/olebedev/config
3. Ubicar el código fuente del api (carpeta lifebank) en el C:\Users\[Nombre Usuario]\go\src\github.com (si su GOPATH no es C:\Users\[Nombre Usuario]\go use el comando el en cmd de la pc: go env GOPATH para saber cuál es su GOPATH y ubicar el proyecto en GOPATH \src\github.com )
4. Antes de levantar el proyecto debe cambiar el username y password, al archivo appConfig.yml ubicado dentro de la carpeta GOPATH\src\github.com\LifeBank\src\resources con los valores que usted tiene configurado en su motor de base de datos



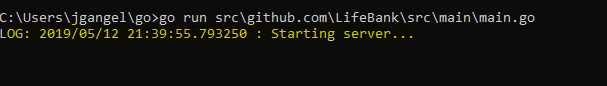
1. Luego de configurar las credenciales para la base de datos, abra una terminal y ubíquese en su GOPATH:



1. Ahora ejecute el siguiente comando: go run src\github.com\LifeBank\src\main\main.go



Si obtiene el siguiente mensaje el servicio ya se encuentra arriba:



PRUEBA DE SERVICIOS:

**Servicio de autenticación:** este servicio retorna un token con el estándar JWT, para consumir este servicio utilizamos postman, las indicaciones son las siguientes:

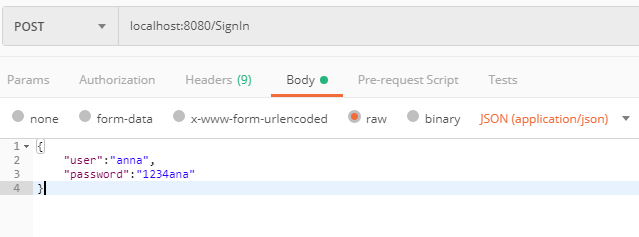
1. URL: localhost:8080/SignIn
2. Metodo: POST
3. Header: Content-Type application/json
4. Json:

{

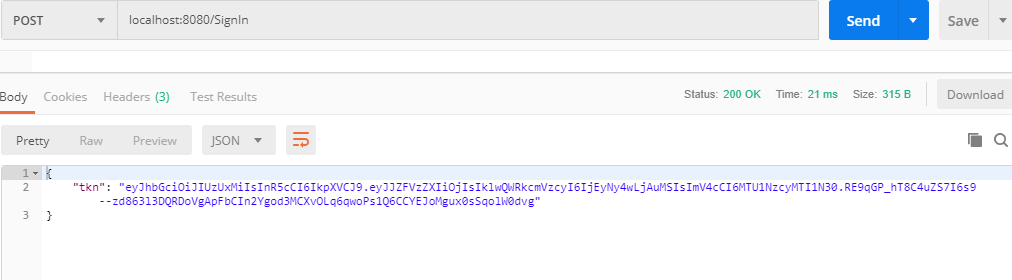
"user":"anna",

"password":"1234ana"

}

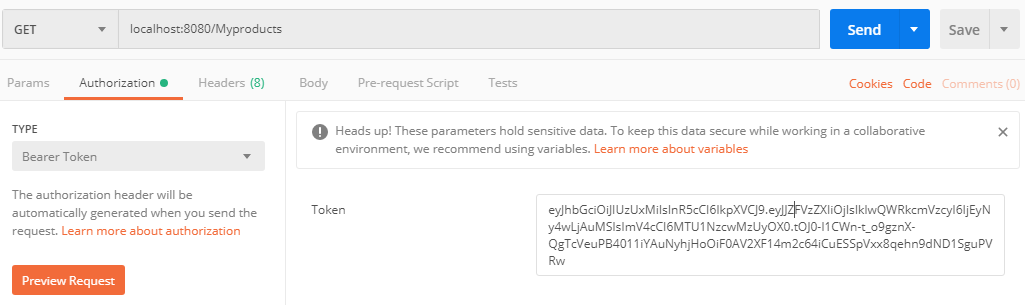


Esta petición si las credenciales son correctas debe retornar un toke jwt:

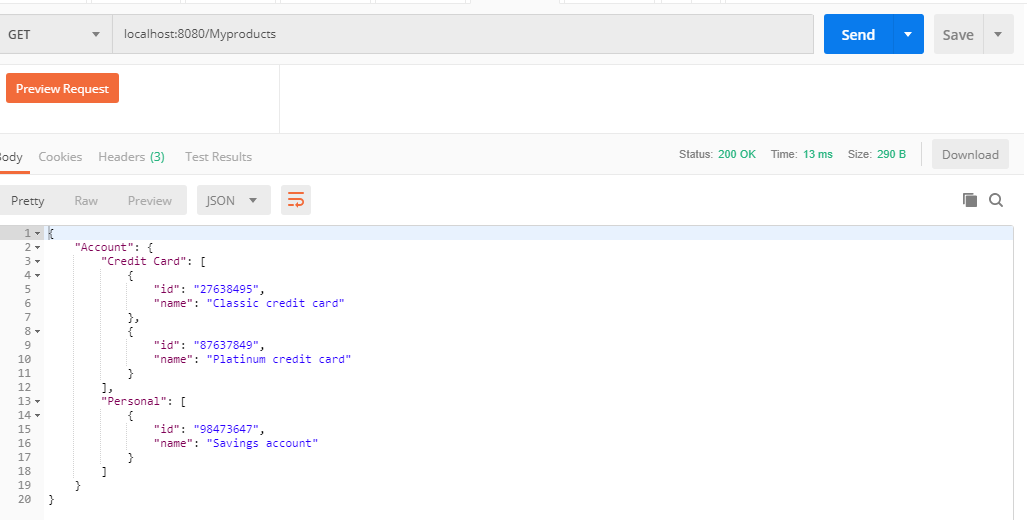


**Servicio de productos:** este servicio retorna una lista de productos que posee el cliente, para consumir este servicio utilizamos postman, las indicaciones son las siguientes:

1. URL: localhost:8080/Myproducts
2. Metodo: GET
3. Usar el Token obtenido en el servicio de autorización en el header Authorization type Bearer



Esta peticion debera retornar la lista de los productos que posee ese cliente:



**Servicio de afiliar Beneficiario:** este servicio afilia un beneficiaro una lista de productos que posee el cliente, para consumir este servicio utilizamos postman, las indicaciones son las siguientes:

1. URL: localhost:8080/AfBeneficiary
2. Metodo: GET
3. Usar el Token obtenido en el servicio de autorización en el header Authorization type Bearer

