

Servicio de Creación y Consulta de Usuarios

Este servicio permite la creación y consulta de usuarios. Desarrollado como parte de una prueba de habilidades, el proyecto está en desarrollo y no se recomienda su uso en un entorno de producción.

Documentos

En la carpeta docs se encuentra el diagrama de componente y secuencia, además del enunciado de la prueba que generó este proyecto.

Estado del Proyecto

En desarrollo

Instalación del Proyecto

Siga estos pasos para instalar y ejecutar el proyecto en su máquina local.

Requisitos Previos

Asegúrese de tener instalados los siguientes elementos en su entorno de desarrollo:

- Java Development Kit (JDK) - versión 8 o superior.
- Gradle - herramienta de construcción y gestión de dependencias.

Pasos de Instalación

1. **Clonar el Repositorio:**

```
git clone https://github.com/tu-usuario/tu-proyecto.git
cd tu-proyecto
```

2. **Configurar la Base de Datos H2 (Opcional):** El proyecto puede utilizar H2 como base de datos embebida. Modifique la configuración en `src/main/resources/application.yml` según sus necesidades.

3. **Ejecutar la Aplicación con Gradle:**

```
./gradlew bootRun
```

Uso de la Aplicación

1. **Acceder a la Aplicación:** La aplicación estará disponible en <http://localhost:8091>.
2. **Probar la API:** Utilice herramientas como Postman o curl para interactuar con la API.
3. **Crear un Usuario:**

- Endpoint: `http://localhost:8091/sign-up`
- Método: `POST`
- Cuerpo de la Solicitud:

```
{
  "name": "Nombre Apellido",
  "email": "correo@dominio.com",
  "password": "contrasena",
  "phones": [
    {
      "number": 123456789,
      "citycode": 1,
      "countrycode": "US"
    }
  ]
}
```

- Ejemplo de Solicitud en CURL:

```
curl --location --request POST 'http://localhost:8091/sign-up' \
--header 'Content-Type: application/json' \
--data-raw '{
  "name": "Pedro Perez",
  "email": "pedro01@dominio.com",
  "password": "Pwd12abc",
  "phones": [
    {
      "number": 3332843,
      "citycode": 4,
      "countrycode": "5"
    }
  ]
}'
```

- Respuesta Exitosa, con el Token que se va a usar en el siguiente end-point:

```
{
  "id": 1,
  "name": "Pedro Perez",
  "email": "pedro01@dominio.com",
  "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJQZWRYbyBQZXJleiIsImhhdCI6MTcwMjQxNjQ1OSwiZXhwIjoxNzAyNDIwMDU5fQ.s1RWAmZzgV",
  "created": "2023-12-12T21:27:39.884+00:00",
  "lastLogin": null,
  "active": true
}
```

- Respuesta Fallida, validación de usuario único:

```
{
  "error": [
    {
      "timestamp": "2023-12-12T19:02:04.956101",
      "codigo": 400,
      "detail": "Ya existe otro usuario con el mismo correo"
    }
  ]
}
```

- Respuesta Fallida, password no cumple con el formato:

```
{
  "error": [
    {
      "timestamp": "2023-12-12T19:03:30.584868",
      "codigo": 400,
      "detail": "Fallo en validación [ password: El password debe tener Sólo una letra mayuscula, dos numeros y un"
    }
  ]
}
```

4. Consultar un Usuario:

- Endpoint: `http://localhost:8091/login`
- Método: `POST`
- Cuerpo de la Solicitud:

```
{
  "email": "correo@dominio.com"
}
```

- Cabecera:

```
Autenticación: Bearer {token}
```

- Ejemplo de Solicitud en CURL:

```
curl --location --request POST 'http://localhost:8091/login' \
--header 'Authorization: Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJQZWRYbyBQZXJleiIsImhhdCI6MTcwMjQxOTM2OSwiZXhwIjoxNzAyNDI
--header 'Content-Type: application/json' \
--data-raw '{
  "email": "pedro02@dominio.com"
}'
```

- Respuesta Exitosa:

```
{
  "name": "Nombre Apellido",
  "email": "correo@dominio.com",
  "password": "contrasena",
  "phones": [
    {
      "number": 123456789,
      "citycode": 1,
      "countrycode": "US"
    }
  ],
  "id": 1,
  "created": "2023-12-09T17:05:52.204+00:00",
  "lastLogin": "2023-12-09T17:08:45.114+00:00",
  "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJQZWRYb1BlcmV6IiwiaWF0IjoxNzAyMTQxNzI1LCJleHAiOiJlE3MDIxNDUzMjV9.R9wQSWbzYwJL",
  "active": true
}
```

- Estructura de datos de respuesta exitosa:

User

VARIABLE	TIPO DATO	DESCRIPCIÓN
id	Long	Identificador de registro
name	String	Nombre del usuario
email	String	Correo electrónico del usuario
password	String	Contraseña del usuario
phones	Phone	Lista de teléfonos del usuario
created	Datetime	Fecha de creación de usuario
lastLogin	Datetime	Fecha del último logueo
token	String	Token de login de usuario

Phone

VARIABLE	TIPO DATO	DESCRIPCIÓN
number	Long	Número de teléfono
citycode	Long	Código de área (Ciudad)
countrycode	String	Código de área (País)

- Respuesta Fallida, token ivalido:

```
{
  "error": [
    {
      "timestamp": "2023-12-12T19:16:44.051756",
      "codigo": 400,
      "detail": "Token no válido"
    }
  ]
}
```

- Estructura de datos de respuesta fallida:

Error

VARIABLE	TIPO DATO	DESCRIPCIÓN
timestamp	Datetime	fecha y hora del fallo
codigo	String	Código de status HTTP
detail	String	Descripción del fallo

Estructura del Proyecto

```
tu-proyecto/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   ├── com/
│   │   │   │   ├── microservice/
│   │   │   │   │   ├── glologic/
│   │   │   │   │   │   ├── GlobalLogicUserApplication.java
│   │   │   │   │   │   ├── controller/
│   │   │   │   │   │   │   ├── AuthController.java
│   │   │   │   │   │   │   ├── UserController.java
│   │   │   │   │   │   ├── dto/
│   │   │   │   │   │   │   ├── CreatedUserDto.java
│   │   │   │   │   │   │   ├── SignUpUserDto.java
│   │   │   │   │   │   │   ├── SignUpUserPhoneDto.java
│   │   │   │   │   │   │   ├── UserDto.java
│   │   │   │   │   │   ├── entity/
│   │   │   │   │   │   │   ├── User.java
│   │   │   │   │   │   │   ├── UserPhone.java
│   │   │   │   │   │   ├── exceptios/
│   │   │   │   │   │   │   ├── CustomException.java
│   │   │   │   │   │   │   ├── GlobalExceptionHandler.java
│   │   │   │   │   │   ├── repository/
│   │   │   │   │   │   │   ├── UserRepository.java
│   │   │   │   │   │   ├── service/
│   │   │   │   │   │   │   ├── JwtUtil.java
│   │   │   │   │   │   │   ├── UserService.java
│   │   │   │   │   │   │   ├── UserServiceTest.java
│   │   │   │   ├── resources/
│   │   │   │   │   ├── application.yml
│   │   │   │   │   ├── static/
│   │   │   │   │   ├── templates/
│   │   │   ├── test/
│   │   │   │   ├── java/
│   │   │   │   │   ├── com/
│   │   │   │   │   │   ├── microservice/
│   │   │   │   │   │   │   ├── glologic/
│   │   │   │   │   │   │   │   ├── GlobalLogicUserApplicationTests.java
│   │   │   │   │   │   │   │   ├── controller/
│   │   │   │   │   │   │   │   │   ├── AuthControllerTest.java
│   │   │   │   │   │   │   │   │   ├── UserControllerTest.java
│   │   │   │   │   │   │   ├── service/
│   │   │   │   │   │   │   │   ├── JwtUtilTest.java
│   │   │   │   │   │   │   │   ├── UserServiceImplTest.java
│   │   ├── build.gradle
│   └── README.md
```

Contribuyendo

Las contribuciones son bienvenidas. Para cambios importantes, primero abra un problema para discutir lo que le gustaría cambiar. Asegúrese de actualizar las pruebas según sea necesario.

Licencia

