Introducción.

Gracias por darme la oportunidad de desarrollar este ejercicio, he realizado proyectos desde 0 pero siempre hay algo nuevo que aprender, y este no fué la excepción.

Les solicito visualizar el código, ver el uso de anotaciones, las configuraciones en application.properties, el uso de JPA, el uso de Security, un tema complejo que estoy estudiando arduamente; el manejo de excepción centralizado, entre otras muchos temas que ustedes bien conocen.

Quedo atento!. Muchas gracias por su tiempo y paciencia.

Diagramas

Los diagramas se encuentra en la carpeta diagramas/ del proyecto raiz.

Son documentos pdf:

- Diagrama-Componentes.pdf
- Diagrama-Secuencia.pdf

Notas

La version de Spring Boot elegida es la 2.7.17 para tener adherencia con java 1.8, la versión mas reciente de spring boot 3.1.5 migró a Java 17.

Spring Initializr (start.spring.io)

Dependencias instaladas

- Spring Web
- Spring Security
- Spring Data JPA
- Lombok
- Spring Boot DevTools
- H2 Database
- Spring Configuration Processor

GitHub Repository

Se puede descargar del siguiente sitio público y con el siguiente comando:

```
git clone https://github.com/goviedo/usermanagment-bci-smartjob.git
```

Java versión

- From Oracle used version: jdk-8u202-linux-x64.tar.gz

Setting Environment for Linux Platform

Esta es una configuracion de ejemplo para las variables de ambientes para JAVA en un entorno zsh / linux.

```
export JAVA_HOME="/opt/java/jdk1.8.0_202/"
path=('/opt/java/jdk1.8.0_202/bin' $path)
```

Reference URL to Visual Studio Code Lombok Installation

- https://projectlombok.org/setup/vscode

Extensions Visual Studio Code.

- Extension Pack for Java v0.25.15
- Lombok Annotations Support for VS Code v1.1.0
- Spring Boot Extension Pack

Entering h2 console

Si se desea investigar los datos en la bd h2, pueden seguir la ruta siguiente y la deficion de usuario y password dado por el archivo application.properties

Utiliza h2 en memoria, por lo que cada vez que se arranque spring, reiniciará todos los datos.

- http://localhost:8080/h2
- usario/password: sa/password

Version h2 sin problemas

Tuve que degradar la version de h2, debido a que me daba problemas con la creacion fisica de la base de datos.

- Version 1.4.199 seems OK.

```
runtimeOnly 'com.h2database:h2:1.4.199'
```

Explicacion aqui: https://stackoverflow.com/questions/68365221/h2-spring-boot-jpa-issue-with-retrieving-entity-with-offsettime-field?rq=2

User and Password Spring Security

En el archivo application.properties:

```
user: admin
password: admin
```

Como iniciar el proyecto

Ejecucion de endpoints

Primero, en la carpeta raiz del proyecto, iniciarlo:

```
./gradlew bootRun &
```

o bien ejecutar sin el ampersand y presionar ctrl-z y luego el comando bg para pasar el proceso a background.

Este inicia en el puerto por defecto 8080

Deje un comando para ahorrarse de escribir y ejecutar en foreground

```
./runApp
```

JUNIT 5 Tests.

En la carpeta raiz del proyecto, ejecutar:

```
./gradlew cleanTest test
```

o bien usar el comando en la carpeta raiz:

```
./testApp
```

Este ultimo filtra por la palabra TEST-APP para limpiar la salida

Ejecucion de api mediante comandos curl

Luego, en la carpeta raiz del proyecto existe una carpeta denominada curls/

```
cd ./curls/
```

Los siguientes ejecutables realizan:

- curl-sign-up : Una ejecucion normal del endpoint, ingresando el usuario y un telefono. Si se ejecuta de nuevo, saldra el error que el usuario ya existe, en este caso el correo ya existe ya que es único. CONFLICT, 409.
  - Aqui podemos ver el UUID que se genera automáticamente.
  - Si se edita el archivo y se cambia el correo, se puede ver como se genera un UUID nuevo.
  - Con lo anterior podemos ver como son creados los campos created y lastLogin con timestamp.

  - El token, también es autogenerado dependiendo del password que usemos, este token cambia.
  - isActive es representado por la herencia de UserDetailService de Spring Security con campos como accountNonExpired, accountNonLocked, credentialsNonExpired y enable que seria el equivalente a isActive.

- curl-sign-up-correo-incorrecto: El correo es incorrecto. BAD REQUEST. 400.
- curl-sign-up-clave-o-password-erroneo: La clave no tiene el formato esperado. BAD REQUEST. 400. Este sigue la expresión regular definida en application.properties. Mas información por favor ver clase cl.goviedo.usermanagment.validators.CustomPasswordValidatorImpl.java sobre el regex usado.
- curl-sign-up-nombre-telefono-opcionales: Sin la opcion de ingresar telefono alguno. Se ingresa correctamente. ACCEPTED. 200 OK.
- curl-sign-up-password-erroneo-y-email-incorrecto: Esto representa que es posible visualizar 2 errores ya que se van agregando a la salida, tanto del password incorrecto como el correo.

HELP.md

Este archivo son direcciones URLs que me han servido de referencia para el proyecto.

Nota Swagger

Por motivos de tiempo y lo expreso en el correo, dado que la prueba fue entregada recien hoy en la tarde, no he podido implementar Swagger. Mañana tengo un paseo por todo el día con mi hijo y pasado mañana me voy a Concepción. Si me dan un tiempo para el lunes, puedo entregarles con Swagger.

Reference Documentation

For further reference, please consider the following sections:

- Official Gradle documentation
- Spring Boot Gradle Plugin Reference Guide
- Create an OCI image
- Spring Web
- Spring Security
- Spring Data JPA
- Spring Boot DevTools
- Spring Configuration Processor

Guides

The following guides illustrate how to use some features concretely:

- Building a RESTful Web Service
- Serving Web Content with Spring MVC
- Building REST services with Spring
- Securing a Web Application
- Spring Boot and OAuth2
- Authenticating a User with LDAP
- Accessing Data with JPA

Additional Links

These additional references should also help you:

- Gradle Build Scans – insights for your project's build