Time Tracking Local and Pro Code Differences

By the moment the tool is not using a different properties file depending on which environment it is deployed. So, to be able to work properly from local, test or production, some changes must be made.

- **/src/main/resources/application.properties**

From this file what interest us is the database connection. There are 3 connections defined (prod, test and local) but the production is the same than the test one. And obviously the local settings depend on the developer configuration. So, to deploy it in local these properties must be uncommented. Like this:

```
#Spring DB configuration Open Shift Prod
spring.jpa.hibernate.ddl-auto=create-drop
spring.datasource.url=jdbc:mysql://timetrackingdb-
v1:3306/time_tracking_db?useSSL=false&allowPublicKeyRetrieval=true
spring.datasource.username=pperezcivit
spring.datasource.password=tTracking2019


#Spring DB configuration Open Shift Test
#spring.jpa.hibernate.ddl-auto=create-drop
#spring.datasource.url=jdbc:mysql://timetrackingdb-
v1:3306/time_tracking_db?useSSL=false&allowPublicKeyRetrieval=true
#spring.datasource.username=pperezcivit
#spring.datasource.password=tTracking2019


#Spring DB configuration local
#spring.jpa.hibernate.ddl-auto=create-drop
#spring.jpa.database=mysql
#spring.datasource.url=jdbc:mysql://localhost:3306/time_tracking?serverTimezo
ne=UTC
#spring.datasource.username=root
#spring.datasource.password=cactus
```

- **/src/main/java/com/Zurich/security/SecurityConfig.java**

From this file the connection to LDAP is configured. These changes are because if the application is running inside OpenShift platform needs some encoding.

For production environment must be like follows:

```
	// LOCAL
//	final private String LDAP_SERVER	=
"ldap://emea.zurich.corp:3268/dc=emea,dc=zurich,dc=corp";
//	final private String SEARCH_BASE	= "ou=3RDPty,ou=zUsers";
//	final private String SEARCH_FILTER	= "sAMAccountName={0}";
//	final private String MANAGER_DN =
"CN=Y1017834,OU=svcUID,OU=zUsers,DC=emea,DC=zurich,DC=corp";
//	final private String MANAGER_PWD= "Wp04RR9QbQy9QLI";
//	final private String REFERRAL		= "follow";

	// PROD and TEST
	final private String LDAP_SERVER	=
"ldap://zurich.com:3268/dc=zurich,dc=com";
    final private String SEARCH_BASE   = "";
    final private String SEARCH_FILTER       = "sAMAccountName={0}";
    final private String MANAGER_DN    = "CN=svc-Jira-LDAP,OU=Service
Accounts,OU=Spain,OU=EMEA,DC=zurich,DC=com";
    final private String MANAGER_PWD   = "2xoDN5m11GhGex24L6Bq6b5CPw7wWv";
    final private String REFERRAL            = "follow";
```

If wanted, from local, to simulate different types of users easily is possible to load some info and not ask permission to LDAP.

The way it is for production (o ask to LDAP) is as follows:

```
@Override
public void configure ( AuthenticationManagerBuilder auth ) throws Exception
{
	auth.authenticationProvider ( ldapAuthenticationProvider() );

}
```

But we can change to this:

```
@Override
public void configure ( AuthenticationManagerBuilder auth ) throws Exception
{
	auth.inMemoryAuthentication()
	.withUser("contractor").password(passwordEncoder().encode("pass")).rol
es("USER").and()
	.withUser("manager").password(passwordEncoder().encode("pass")).roles(
"USER").and()
	.withUser("admin").password(passwordEncoder().encode("pass")).roles("A
DMIN").and()
	.withUser("manadmin").password(passwordEncoder().encode("pass")).roles
("ADMIN");

}
```

- **/src/main/java/com/zurich/utils/Utils.java**

Here we define the environment, if production/test or local. As it is, it’s for production environment. (For testing will be true and false and in local false and has no impact because there is no access to mail server)

```
final public static boolean PRODUCTION_ENVIRONMENT_FOR_REPORT = true;
final public static boolean PRODUCTION_ENVIRONMENT_FOR_MAIL   = true;
```

This will affect when generating reports and when sending mails.