Java Zulu 8.20.0.5-linux64

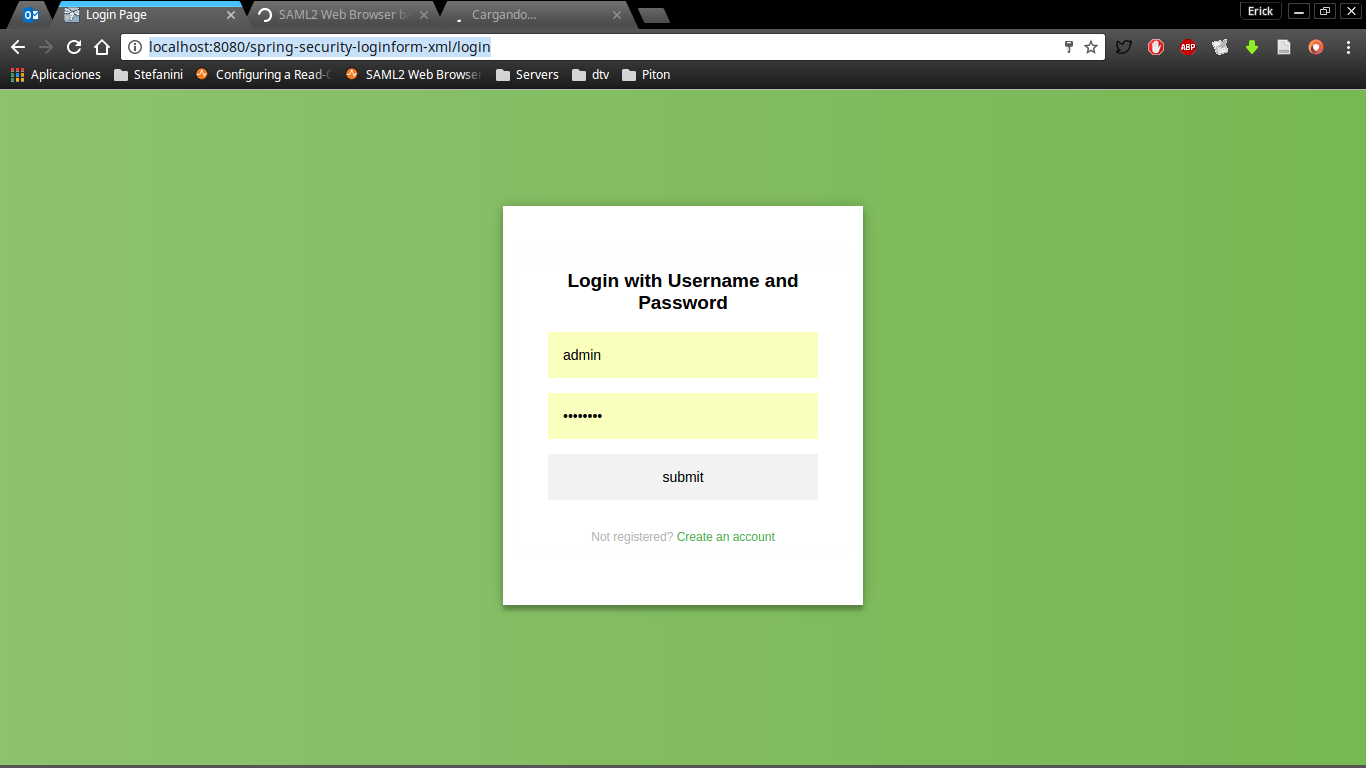
Jboss 10.Final

Spring 3.2.8.RELEASE

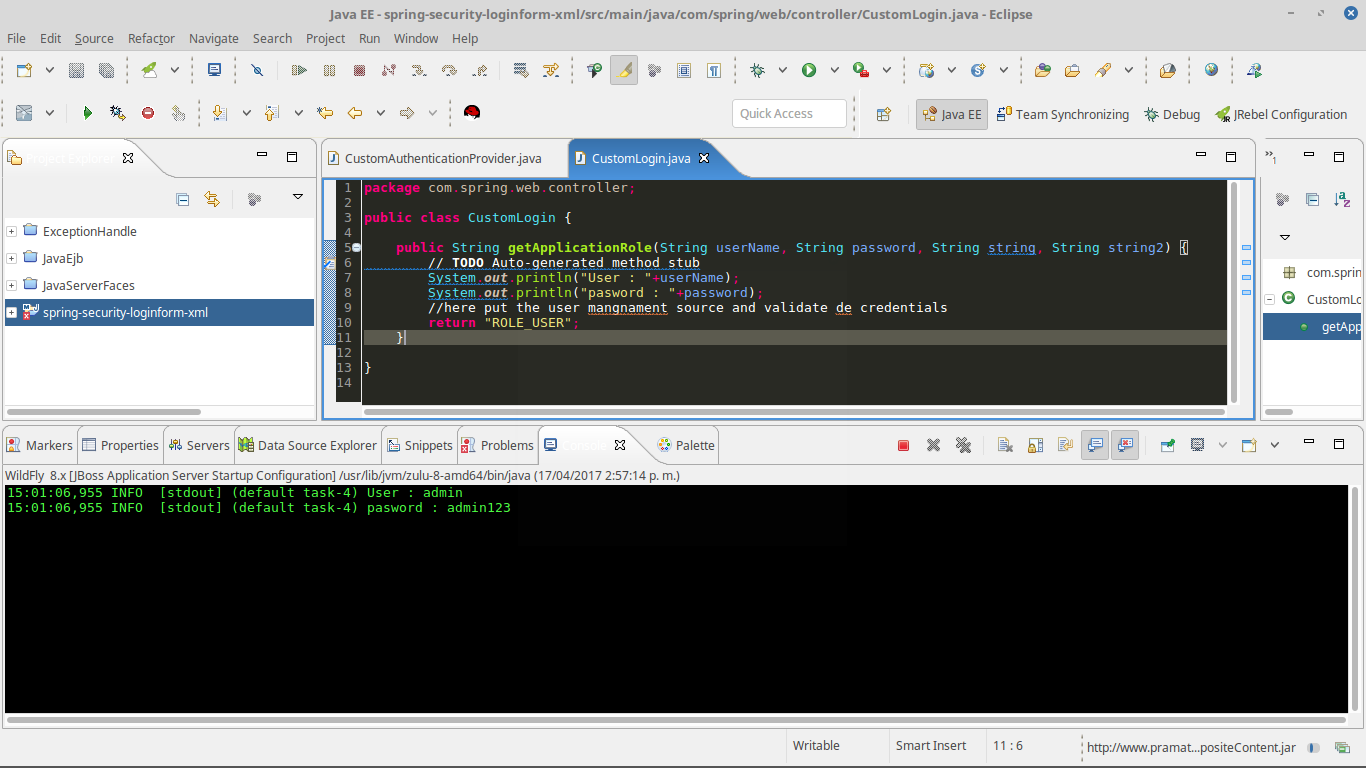
Spring Security 3.2.3.RELEASE

Maven 3.033

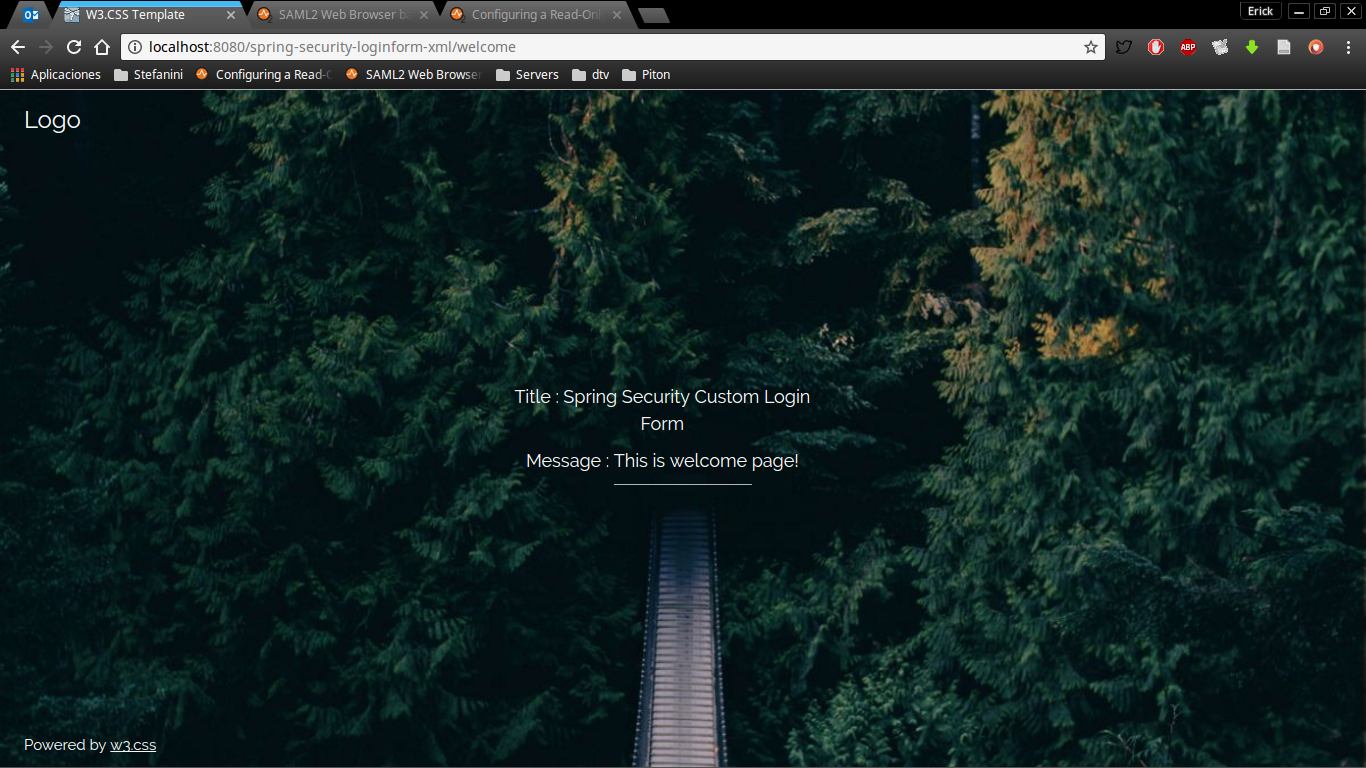
http://localhost:8080/spring-security-loginform-xml/login



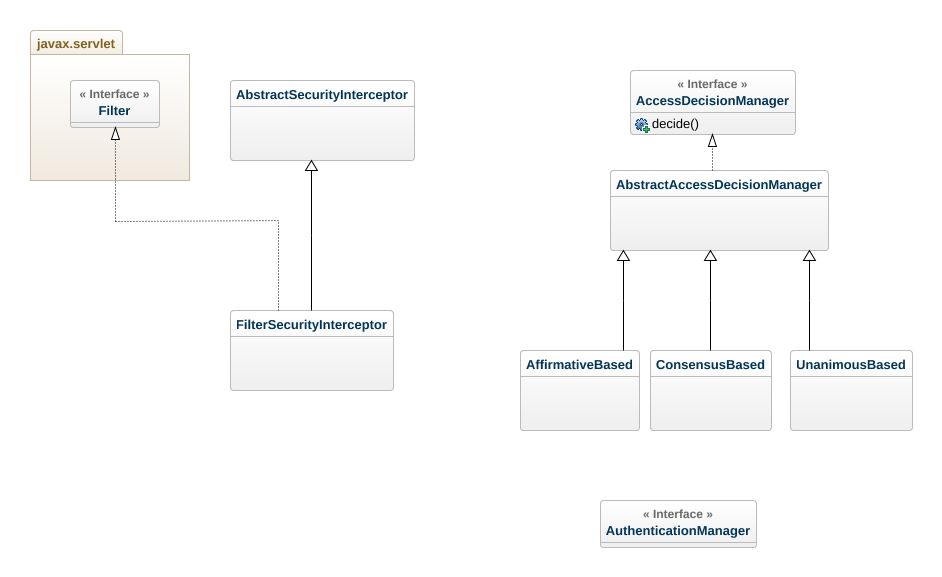
Nota : al aun no tener integrado un proveedor de identidad, el codigo en el metodo de validacion deja pasar cualquier usuairo y contraseña imprimiendolos por consola



Welcome Page



Crear autenticacion con Spring Security, en la cual se validen las credenciales ingresadas contra un proveedor de identidad externo.



El Autentication Manager hace uso de la clase CustomAuthenticationProvider en el metodo **authenticate**que recibe un objeto de tipo org.springframework.security.core.Authentication :

String **userName** = authentication.getName().trim();

String **password** = authentication.getCredentials().toString().trim();

Authentication **auth** = **null**;

CustomLogin **login** = **new** CustomLogin(); //Authenticate the user based on your custom logic

String **role** = login.getApplicationRole(userName, password, "ADMIN","DEVELOPER");

En la clase CustomLogin se encuentra el codigo para validar el usuario y la contraseña contra proveedores de identidad externos en el metodo

**public** String **getApplicationRole**(String userName, String password, String string, String string2) { Auto

//TO DO : GET USER’s FROM LDAP-generated method stub

System.***out***.println("User : "+userName);

System.***out***.println("pasword : "+password);

**return** "ROLE\_USER";

}