https://stackoverflow.com/questions/22749767/using-jdbcauthentication-in-spring-security-with-hibernate

|  |  |  |  |
| --- | --- | --- | --- |
| I just finished an Spring application which is based on the inMemoryAuthentication(), and now, after verified all are working perfectly, I want use the JDBC Authentication.  I have three kinds of classes to handle the connection to the database:   1. HibernateConfig, based on the code presented in this blog post. 2. DAO and Entity class, one for each table from my database (in this current state, I have three: User, Role and User\_Role) 3. And a UserService class, which implement UserDetailsService.   I read some articles on the Internet, and pratically all of them uses the following configuration for the class SecurityConfig:  public class SecurityConfig extends WebSecurityConfigurerAdapter {  @Autowired  private DataSource dataSource;  @Autowired  public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {  auth  .jdbcAuthentication()  .dataSource(dataSource)  .usersByUsernameQuery(getUserQuery())  .authoritiesByUsernameQuery(getAuthoritiesQuery());  }  What is an issue for me, since I don't have a Datasource class in my project. My current SecurityConfig is this:  @Configuration  @EnableWebSecurity  public class SecurityConfig extends WebSecurityConfigurerAdapter {  @Autowired  public void configure(AuthenticationManagerBuilder auth) throws Exception {  auth  .jdbcAuthentication()  .usersByUsernameQuery(getUserQuery())  .authoritiesByUsernameQuery(getAuthoritiesQuery());  }  protected void configure(HttpSecurity http) throws Exception {  http  .csrf()  .disable()  .authorizeRequests()  .antMatchers("/css/\*\*", "/fonts/\*\*", "/image/\*\*", "/js/\*\*").permitAll()  .anyRequest().authenticated()  .and()  .formLogin()  .loginPage("/spring/index").permitAll()  .loginProcessingUrl("/spring/login").permitAll()  .usernameParameter("login")  .passwordParameter("senha")  .successHandler(new CustomAuthenticationSuccessHandler())  .failureHandler(new CustomAuthenticationFailureHandler())  .and()  .logout()  .logoutUrl("/spring/logout")  .logoutSuccessUrl("/spring/index").permitAll();  }  private String getUserQuery() {  return "SELECT login as username, senha as password "  + "FROM usuario "  + "WHERE login = ?";  }  private String getAuthoritiesQuery() {  return "SELECT DISTINCT usuario.login as username, autorizacao.descricao as authority "  + "FROM usuario, autorizacao\_usuario, autorizacao "  + "WHERE usuario.id = autorizacao\_usuario.fk\_usuario "  + "AND autorizacao.id = autorizacao\_usuario.fk\_autorizacao "  + "AND usuario.login = ? ";  }  }  As you can see, I already made some adaptations for use JDBC Authentication, but I am still missing the part where I link this class to my Hibernate class.  Someone have any suggestions in how to make this?  **my HibernateConfig is this**  @Configuration  @EnableTransactionManagement  @PropertySource({ "classpath:persistence.properties" })  @ComponentScan({ "org.webapp.persistence" })  public class HibernateConfig {  @Autowired  private Environment env;  @Bean  public LocalSessionFactoryBean sessionFactory() {  LocalSessionFactoryBean sessionFactory = new LocalSessionFactoryBean();  sessionFactory.setDataSource(restDataSource());  sessionFactory.setPackagesToScan(new String[] { "org.webapp.persistence.model" });  sessionFactory.setHibernateProperties(hibernateProperties());  return sessionFactory;  }  @Bean  public DataSource restDataSource() {  BasicDataSource dataSource = new BasicDataSource();  dataSource.setDriverClassName(env.getProperty("jdbc.driverClassname"));  dataSource.setUrl(env.getProperty("jdbc.url"));  dataSource.setUsername(env.getProperty("jdbc.user"));  dataSource.setPassword(env.getProperty("jdbc.pass"));  return dataSource;  }  @Bean  @Autowired  public HibernateTransactionManager transactionManager(SessionFactory sessionFactory) {  HibernateTransactionManager txManager = new HibernateTransactionManager();  txManager.setSessionFactory(sessionFactory);  return txManager;  }  @Bean  public PersistenceExceptionTranslationPostProcessor exceptionTranslation() {  return new PersistenceExceptionTranslationPostProcessor();  }  Properties hibernateProperties() {  return new Properties() {  /\*\*  \*  \*/  private static final long serialVersionUID = 1L;  {  setProperty("hibernate.hbm2ddl.auto", env.getProperty("hibernate.hbm2ddl.auto"));  setProperty("hibernate.dialect", env.getProperty("hibernate.dialect"));  setProperty("hibernate.globally\_quoted\_identifiers", "true");  }  };  }  }  [spring](https://stackoverflow.com/questions/tagged/spring) [hibernate](https://stackoverflow.com/questions/tagged/hibernate) [spring-mvc](https://stackoverflow.com/questions/tagged/spring-mvc) [jdbc](https://stackoverflow.com/questions/tagged/jdbc) [spring-security](https://stackoverflow.com/questions/tagged/spring-security)   |  |  |  | | --- | --- | --- | | [share](https://stackoverflow.com/q/22749767)[improve this question](https://stackoverflow.com/posts/22749767/edit) | [edited Jun 29 at 13:11](https://stackoverflow.com/posts/22749767/revisions)  [[https://i.stack.imgur.com/46fVQ.jpg?s=32&g=1](https://stackoverflow.com/users/1278949/lambda)](https://stackoverflow.com/users/1278949/lambda)  [lambda](https://stackoverflow.com/users/1278949/lambda)  **1,807**11326 | asked Mar 30 '14 at 20:56  [[https://graph.facebook.com/772269476/picture?type=large](https://stackoverflow.com/users/2692962/kleber-mota)](https://stackoverflow.com/users/2692962/kleber-mota)  [Kleber Mota](https://stackoverflow.com/users/2692962/kleber-mota)  **1,509**154492 | |
|  | add a comment |

1 Answer

[active](https://stackoverflow.com/questions/22749767/using-jdbcauthentication-in-spring-security-with-hibernate?answertab=active#tab-top)[oldest](https://stackoverflow.com/questions/22749767/using-jdbcauthentication-in-spring-security-with-hibernate?answertab=oldest#tab-top)[votes](https://stackoverflow.com/questions/22749767/using-jdbcauthentication-in-spring-security-with-hibernate?answertab=votes#tab-top)

|  |  |  |
| --- | --- | --- |
| up vote9down voteaccepted | Ok, I solved it. All I needed to do was insert the following annotation in my class SecurityConfig:  @ComponentScan(value="org.webapp")  And now I can autowire my DataSource in this class  @Autowired  private DataSource restDataSource;  @Autowired  public void configure(AuthenticationManagerBuilder auth) throws Exception {  auth  .jdbcAuthentication()  .dataSource(restDataSource)  .usersByUsernameQuery(getUserQuery())  .authoritiesByUsernameQuery(getAuthoritiesQuery());  }   |  | | --- | | [share](https://stackoverflow.com/a/22759500)[improve this answer](https://stackoverflow.com/posts/22759500/edit) | |