

Creating a Custom Login Page for Spring Security



Bryan Hansen

DIRECTOR OF SOFTWARE DEVELOPMENT

@bh5k

failureUrl

```
.formLogin()  
.loginPage("/login")  
.failureUrl("/login?error=true")  
.loginProcessingUrl("/perform_login")  
.permitAll()
```

Displaying Errors

```
<div class="error">  
  <c:if test="${not empty param.error}">  
    Invalid username and password.  
  </c:if>  
</div>
```

Remember Me



Remember Me Frontend

```
<div>
  <label> User Name : <input type="text" name="username" /> </
  label>
</div>
<div>
  <label> Password: <input type="password" name="password" /> </
  label>
</div>
<div>
  <label> Remember Me: <input type="checkbox" name="remember-me" />
  </label>
</div>
```

Remember Me Configuration

```
.rememberMe()  
.key("superSecretKey")  
.tokenRepository(tokenRepository());
```

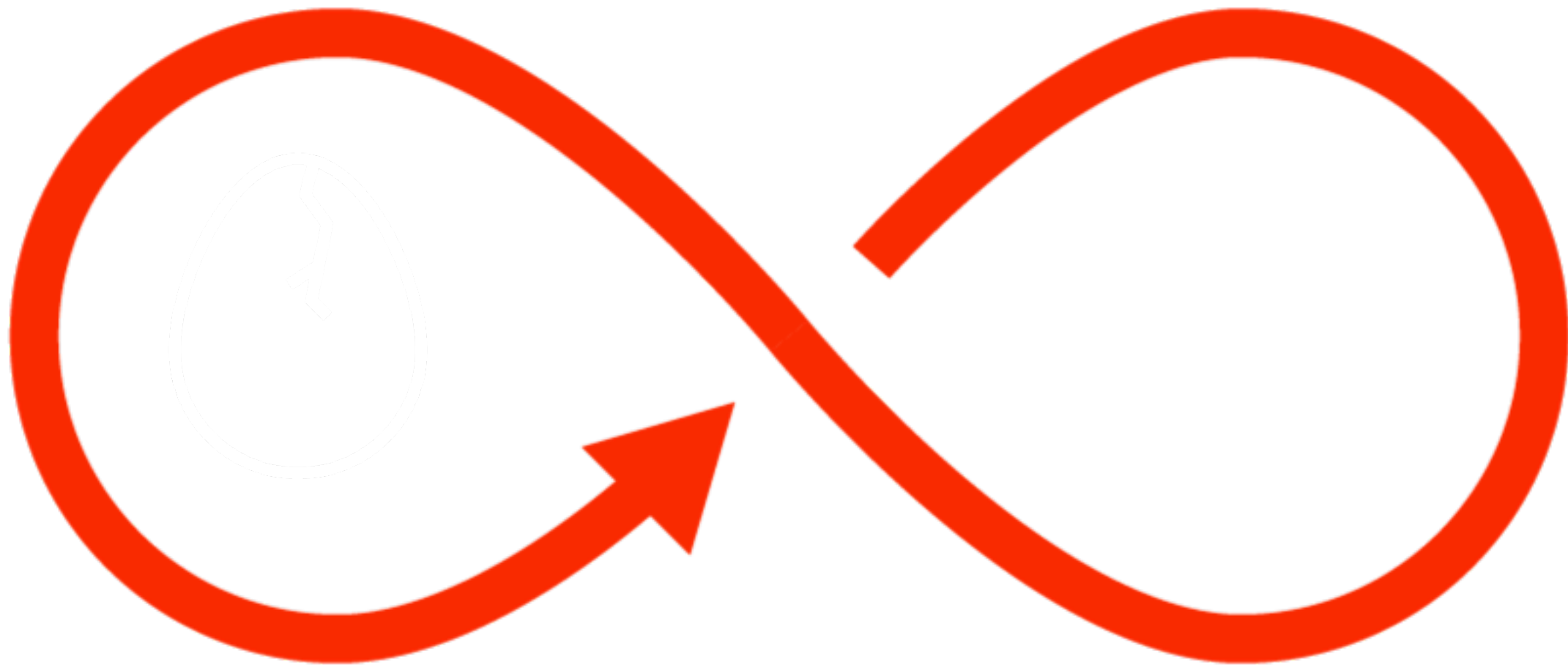
```
// ...
```

```
@Bean  
public PersistentTokenRepository tokenRepository() {  
    JdbcTokenRepositoryImpl token = new  
JdbcTokenRepositoryImpl();  
    token.setDataSource(dataSource);  
    return token;  
}
```

Remember Me SQL

```
CREATE TABLE persistent_logins(  
    username VARCHAR(50) NOT NULL,  
    series VARCHAR(64) PRIMARY KEY ,  
    token VARCHAR(64) NOT NULL,  
    last_used TIMESTAMP NOT NULL,  
    FOREIGN KEY (username) REFERENCES users(username)  
);
```

Logout



Logout Configuration

```
.and()  
.logout()  
.logoutSuccessUrl("/login?logout=true")  
.logoutRequestMatcher(new AntPathRequestMatcher("/  
perform_logout", "GET"))  
.invalidateHttpSession(true)  
.deleteCookies("JSESSIONID")  
.permitAll();
```

Logout Frontend

```
<a class="navbar-brand"  
  href="<c:out value='perform_logout' />">  
  Logout  
</a>
```

Displaying Success

```
<c:if test="{not empty param.logout}" >  
    <div class="alert alert-success" role="alert">  
        Logout was successful!  
    </div>  
</c:if>
```

Summary

Display Errors

Remember Me

Logout

GET vs POST

Bootstrap Messages