09

Habilitando o spring security

Agora que já adicionamos as nossas dependências precisamos configurar o spring security. Crie as classes SpringSecurityFilterConfiguration, SecurityConfiguration conforme o vídeo e adicione a classe SecurityConfiguration no método getRootConfigClasses. Cole aqui o código das classes que foram criadas/alteradas.

Opinião do Instrutor

Feito isso ao tentar acessar a home do nosso site será solicitado um login e senha.

```
package br.com.caelum.loja.conf;

public class SpringSecurityFilterConfiguration extends AbstractSecurityWebApplicationInitializer {

}
```

```
package br.com.caelum.loja.conf;


@EnableWebSecurity
public class SecurityConfiguration  {

}
```

```
package br.com.caelum.loja.conf;

import javax.servlet.Filter;

import javax.servlet.MultipartConfigElement;
import javax.servlet.ServletRegistration.Dynamic;

import org.springframework.web.filter.CharacterEncodingFilter;
import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;

public class ServletSpringMvc extends AbstractAnnotationConfigDispatcherServletInitializer{

    @Override
    protected Class<?>[] getRootConfigClasses() {
        return new Class[] {SecurityConfiguration.class};
    }

    @Override
    protected Class<?>[] getServletConfigClasses() {
        return new Class[] { AppWebConfiguration.class , JPAConfiguration.class};
    }

    @Override
    protected String[] getServletMappings() {
```

```
    protected String[] getServletMappings() {
        return new String[] {"/"};
    }

    @Override
    protected Filter[] getServletFilters() {
        CharacterEncodingFilter encodingFilter = new CharacterEncodingFilter();
        encodingFilter.setEncoding("UTF-8");

        return new Filter[] {encodingFilter};
    }

    @Override
    protected void customizeRegistration(Dynamic registration) {
        registration.setMultipartConfig(new MultipartConfigElement(""));
    }
}
```