05. Aplicando filtros

Date	@23/11/2022
Categoria	Java para Web
Curso	Java Servlet: autenticação autorização e o padrão MVC

Tópicos

- Projeto da aula anterior
- Primeiro filtro
- Filter x Servlet
- Filtro de autorização
- Ordem de execução
- Anotações x web.xml
- Onde está o problema?
- Para saber mais: Interceptor ou Filter
- Faça como eu fiz na aula
- O que aprendemos?

Filter x Servlet

No decorrer da aula, vimos que um filtro é bem semelhante a uma servlet, mas ele possui uma responsabilidade a mais. Qual seria essa responsabilidade?

- O filtro consegue parar a execução, já que ele vem como uma camada à frente da aplicação.
    - Essa é a diferença que podemos ver no nosso código, através do parâmetro `FilterChain`. Usamos o `FilterChain` para mandar a requisição para frente.

Anotações x web.xml

Existem casos em que anotações não são suficientemente "completas", como vimos na aula. Qual caso foi necessário o uso do **web.xml** ?

- Anotações não nos permitem definir a ordem dos filtros, para isso precisamos utilizar o **web.xml**. Ou seja, quando precisamos definir uma ordem de aplicação dos filtros, devemos optar pelo uso do **web.xml**.

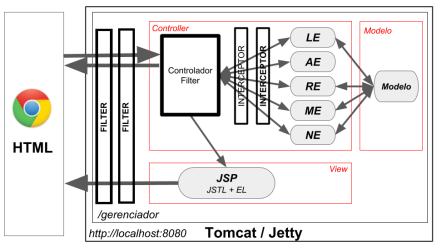
Para saber mais: Interceptor ou Filter

Muitas vezes, o desenvolvedor ou desenvolvedora usa as duas palavras como sinônimo. E realmente, os filtros podem interceptar o fluxo, ou seja filtros funcionam como *interceptadores*.

No entanto, você verá nos frameworks, como Spring, uma implementação própria chamada de `Interceptor`, além disso existem filtros específicos que o Spring implementou e usa.

Então, qual é a diferença entre Filter e Interceptor?

Para responder, veja a imagem abaixo, que mostra o uso do *Filter* e *Interceptor*:



Repare que o filtro fica antes do controlador e o interceptador depois. O filtro é um componente do mundo de Servlets e se preocupa em filtrar requisições (é ligado ao mundo web), enquanto um interceptador "filtra" chamadas de ações ou outros métodos. Os interceptadores são específicos do

framework (por exemplo, Spring) e até funciona sem Servlets. Ok?

O que aprendemos?

Nesta aula, conhecemos um novo recurso do mundo de Servlets, o `javax.servlet.Filter`. Os filtros realmente são muito úteis no dia a dia para diversas funções, então se lembre disso nos seus projetos web!

Aprendemos que:

- Um Filter e Servlet são bem parecidos
- Comparado com Servlet, o Filter tem o poder de parar o fluxo
- Para escrever um filtro, devemos implementar a interface `javax.servlet.Filter`
- Para mapear o filtro, usamos a anotação `@WebFilter` ou o **web.xml**
- Vários filtros podem funcionar numa cadeia (um chama o próximo, mas todos são independentes)
- Para definir a ordem de execução, devemos mapear os filtros no **web.xml**
- Um filtro recebe como parâmetro, do método `doFilter`, um `ServletRequest` e um `ServletResponse`
- Ambos, `ServletRequest` e `ServletResponse`, são interfaces mais genéricas do que `HttpServletRequest` e `HttpServletResponse`
- Para chamar o próximo filtro na cadeia, usamos o objeto `FilterChain`
