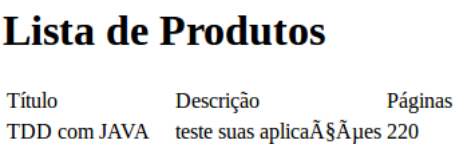
03

Encoding e ajustes no Controller

Transcrição

Note que nossa lista de produtos aparece com alguns caracteres estranhos. Isso acontece por que o **servidor** não conhece o **encoding** da requisição, então ele troca os caracteres especiais e com acentos por outros caracteres.



Há várias formas de resolver este problema, mas vamos usar uma das mais simples. Criando Filtros! Dessa forma, ao receber a requisição o **Spring** filtra a requisição com o **encoding** que vamos configurar. Em nossa classe `ServeltSpringMVC` dentro do pacote `br.com.casadocodigo.loja.conf`, vamos criar mais um método de configuração do **Spring**.

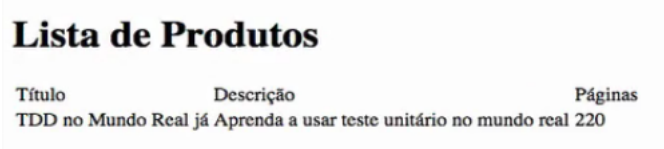
Existe um método chamado `getServletFilters` usado pelo **Spring** que espera receber um `array` de filtros. Então vamos criar um `CharacterEncodingFilter`, definir o `encoding` deste filtro usando o valor `"UTF-8"`, adicionar este filtro ao *array* de filtros e o retornar esse *array* para o **Spring**. Use os imports import javax.servlet.Filter e org.springframework.web.filter.CharacterEncodingFilter.

```
public class ServletSpringMVC extends AbstractAnnotationConfigDispatcherServletInitializer{

    //outros métodos omitidos

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

Agora quando cadastrarmos novos produtos, os caracteres estarão normais.



Melhorando rotas no controller

Vamos fazer agora um pequeno ajuste em nosso `ProdutosController` para deixar o mapeamento das rotas mais simples. Note que em todos os métodos usamos a anotação `@RequestMapping` passando sempre `/produtos`.

Para que não precisemos ficar passando `/produtos` em todos os métodos do controller, vamos pôr essa anotação em nossa classe. Assim podemos remover o `/produtos` de todos os métodos e o **Spring** se encarrega de carregar os

mapeamentos baseados no mapeamento da classe. Sendo assim nossa classe `ProdutosController` deve ficar parecida com o código abaixo:

```
@Controller
@RequestMapping("produtos")
public class ProdutosController {
    [...]

    @RequestMapping("/form")
    public ModelAndView form(){
        [...]
    }

    @RequestMapping(method=RequestMethod.POST)
    public String gravar(Produto produto){
        [...]
    }


    @RequestMapping(method=RequestMethod.GET)
    public ModelAndView listar(){
        [...]
    }

}
```

Dessa forma, se acessamos `/produtos` via `GET` , o método `listar` será chamado. Se o acesso for via `POST` o método gravar será chamado. E o `/produtos/form` continua chamando o método `form` . Bem mais simples, certo?

Teste novamente as páginas de listagem e de cadastro de produtos. Tudo deve estar funcionando normalmente. Cadastre novos produtos e verifique que os caracteres estranhos também não aparecem mais.