

Mãos na massa: Envio de arquivos

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

- 1) Permita que o usuário envie um sumário do livro em PDF. No **form.jsp**, dentro do formulário, antes do botão de cadastro, crie um novo campo do tipo `file`. Além disso, defina o valor do `enctype` do formulário para `multipart/form-data`:

```
<form:form action="${s:mvcUrl("PC#gravar").build()}" method="POST"
  commandName="produto" enctype="multipart/form-data">

  <!-- Outros campos omitidos -->

  <div>
    <label>Sumário</label>
    <input name="sumario" type="file" />
  </div>
  <button type="submit">Cadastrar</button>
</form:form>
```

- 2) Na classe `Produto`, adicione um atributo chamado `sumarioPath` do tipo `String`, juntamente com seu *getter* e *setter*:

```
@Entity
public class Produto {

    private String sumarioPath;

    public String getSumarioPath() {
        return sumarioPath;
    }

    public void setSumarioPath(String sumarioPath) {
        this.sumarioPath = sumarioPath;
    }

    // restante do código omitido
}
```

- 3) Configure o Spring para trabalhar com arquivos. Na classe `AppWebConfiguration`, crie o seguinte método:

```
@Bean
public MultipartResolver multipartResolver() {
    return new StandardServletMultipartResolver();
}
```

- 4) Além disso, altere a configuração das *servlets*. Então, na classe `ServletSpringMVC`, sobrescreva o seguinte método:

```

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

```

5) Crie a lógica de salvar um arquivo. Para isso, crie a classe `FileSaver`, no pacote `br.com.caelum.loja.infra`, com um método chamado `write`, que deverá salvar o arquivo em uma pasta recebida por parâmetro. Lembre-se que você pode usar o método `transferTo` da classe `MultipartFile` para realizar essa tarefa, e para pegar o caminho completo até a pasta, use um `HttpServletRequest`, utilizando a anotação `@Autowired` para o Spring injetar essa dependência:

```

@Component
public class FileSaver {

    @Autowired
    private HttpServletRequest request;

    public String write(String baseFolder, MultipartFile file) {

        try {
            String realPath = request.getServletContext().getRealPath("/") + baseFolder;
            String path = realPath + "/" + file.getOriginalFilename();
            file.transferTo(new File(path));

            return baseFolder + "/" + file.getOriginalFilename();

        } catch (Exception e) {
            throw new RuntimeException(e);
        }

    }
}

```

6) Com a classe criada, use-a no *controller*. Adicione um novo parâmetro no método `gravar`, em `ProdutosController`, chamado `sumario`, do tipo `MultipartFile`. Dentro do método, chame o método `write`, da classe `FileSaver`, salvando o arquivo na pasta `arquivos-sumario`. Lembre-se que para tudo funcionar, corretamente o Spring deve injetar um `FileSaver`:

```

@Controller
@RequestMapping("/produtos")
public class ProdutosController {

    @Autowired
    private FileSaver fileSaver;

    @RequestMapping(method=RequestMethod.POST)
    public ModelAndView gravar(MultipartFile sumario, @Valid Produto produto,
        BindingResult result, RedirectAttributes redirectAttributes) {

        if(result.hasErrors()) {
            return form(produto);
        }

        String path = fileSaver.write("arquivos-sumario", sumario);
    }
}

```

```
produto.setSumarioPath(path);

produtoDao.gravar(produto);

redirectAttributes.addFlashAttribute("sucesso", "Produto cadastrado com sucesso!");

return new ModelAndView("redirect:produtos");
}

// restante do código omitido
}
```

7) Crie a pasta **arquivos-sumario** dentro de **src/main/webapp/**.

8) Por fim, para o Spring encontrar o `FileSaver`, adicione-o na anotação `@ComponentScan`, na classe `AppWebConfiguration`:

```
@EnableWebMvc
@ComponentScan(basePackageClasses={HomeController.class, ProdutoDAO.class,
    FileSaver.class})
public class AppWebConfiguration {

    // restante do código omitido

}
```