

Criando o Servlet de File

Atender a requisição `/file/...` é simples, mas para isso, precisamos criar um `Servlet`.

Crie uma nova classe no pacote `br.com.casadocodigo.servlets` chamada `FileServlet`. Faça a classe herdar de `HttpServlet` e *sobrescreva* o método `service`.

O primeiro passo é mapear essa `Servlet`. Anote essa classe com `@WebServlet`. Dentro da *annotation*, coloque o valor `"/file/*"`. Esse valor significa que qualquer coisa começando com `/file/` será atendido pelo nosso `Servlet`.

Dentro do método `service`, obtenha a URL requisitada, dividindo o método do `request` pelo `/file`, e do array retornado, pegamos a posição `1` que é a posição após o `/file`.

Crie o `Path` e use o `FileNameMap` para obter o `contentType`. Para criar o `Path`, precisamos do caminho absoluto para o arquivo. Já temos o caminho relativo, só precisamos concatenar o local do servidor onde o arquivo está salvo. No `FileSaver` já temos o caminho do servidor, mas está em `private`. Vamos mudar a constante `SERVER_PATH` para ser `public`, o que não é um problema para nós, já que ela não pode ser alterada.

Os próximos passos são um pouco mais "burocráticos", já que teremos que tratar o `response` na mão.

Primeiro devemos limpar o `response` para remover possíveis sujeiras, depois informamos qual o `contentType`, *setamos* o Header para o tamanho `Content-Length` e por fim dizemos qual será o nome do arquivo com o Header `Content-Disposition`.

Mas ainda não pegamos o arquivo do disco para colocar no `response`. Para isso, vamos usar o `FileSaver` novamente, chamando `FileSaver.transfer`, e para ele, chame passe o `Path` que criamos antes e o `OutputStream` do `response`.

Ainda não criamos o método `transfer` do `FileSaver`, vamos fazer isso no próximo exercício.