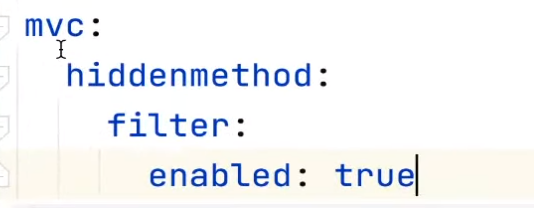
1. **За да работят hidden methods и delete се поставя настройка в .yml файла**

**Spring:**



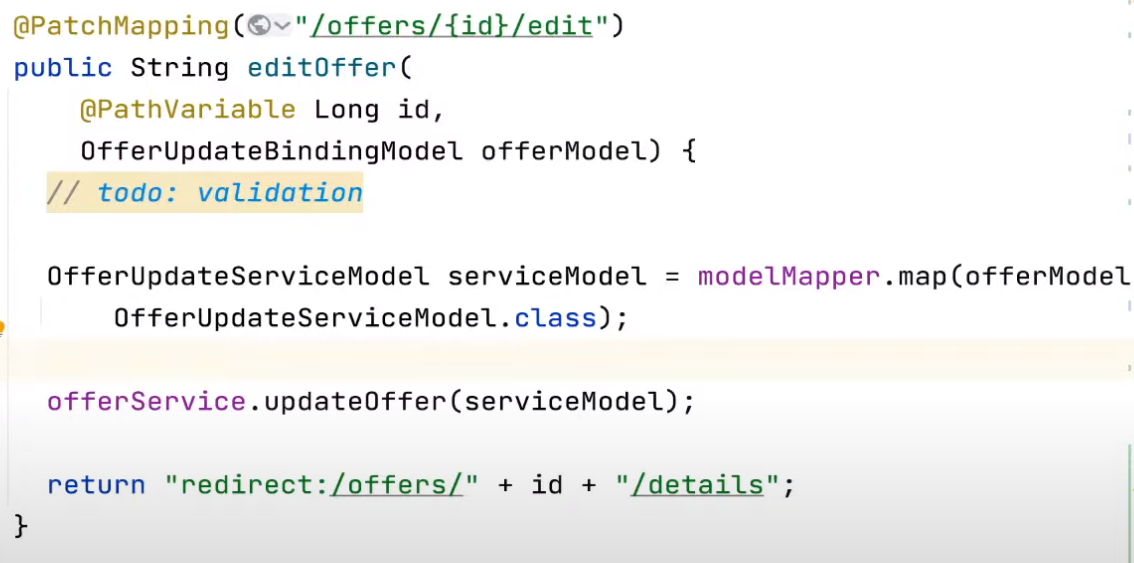
1. **@DeleteMapping**

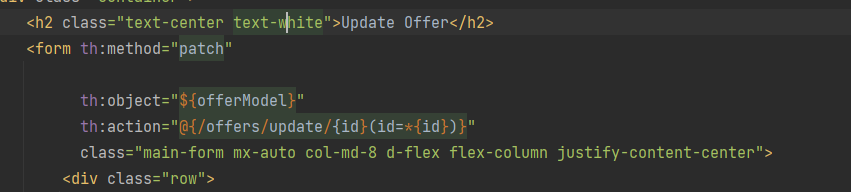
**Th:method=”delete”** се поставя винаги





1. **@PatchMapping –** частичен ъпдейт

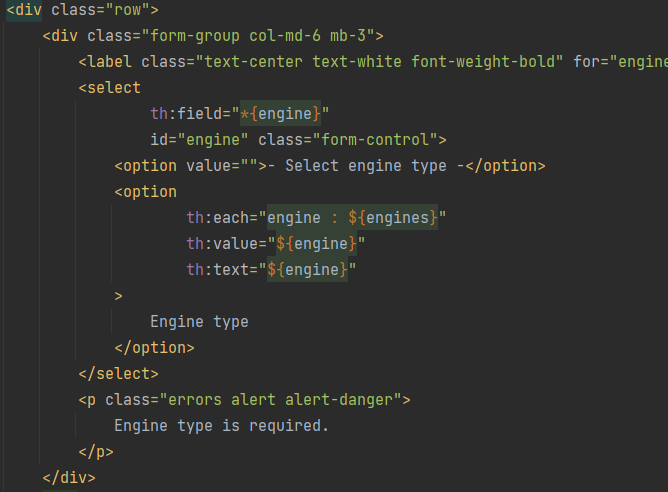




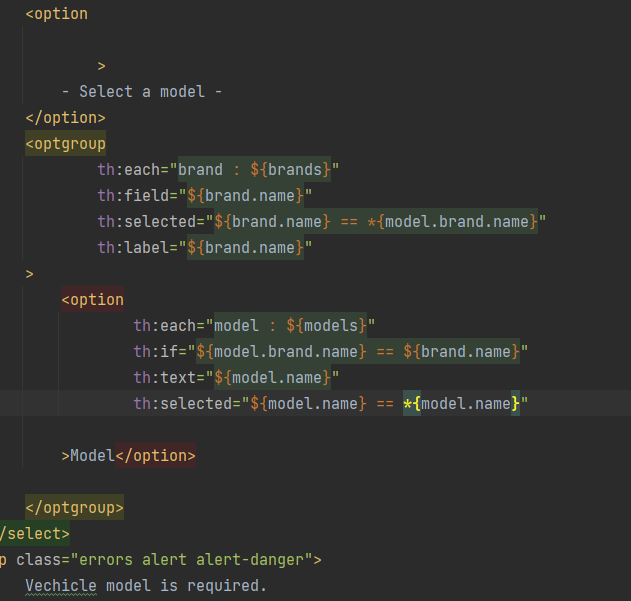
1. **Drop-down option thymeleaf**

**-** th:value = \*{engine} в горния <select> трябва да е въведен

**-** нужно е в контролера пред модел да се постави @ModelAttribute ако не се мапва обекта

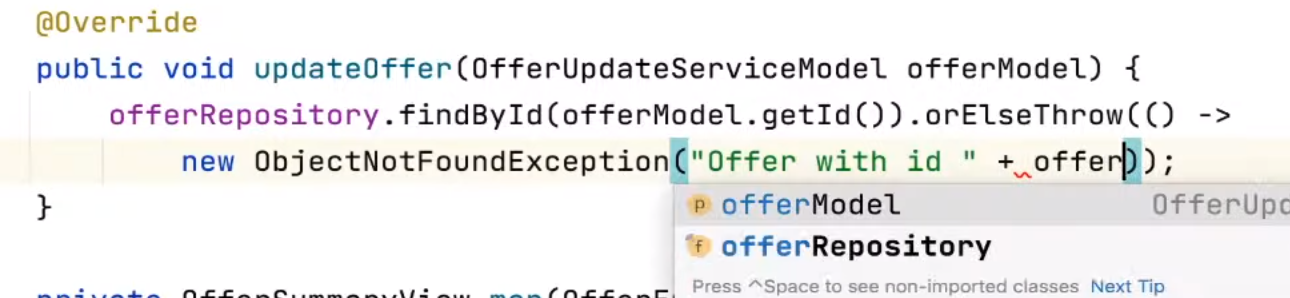
**-**необходими полета each, value, text, selected

* Drop-down меню с лист за всеки един бранд, всеки един модел



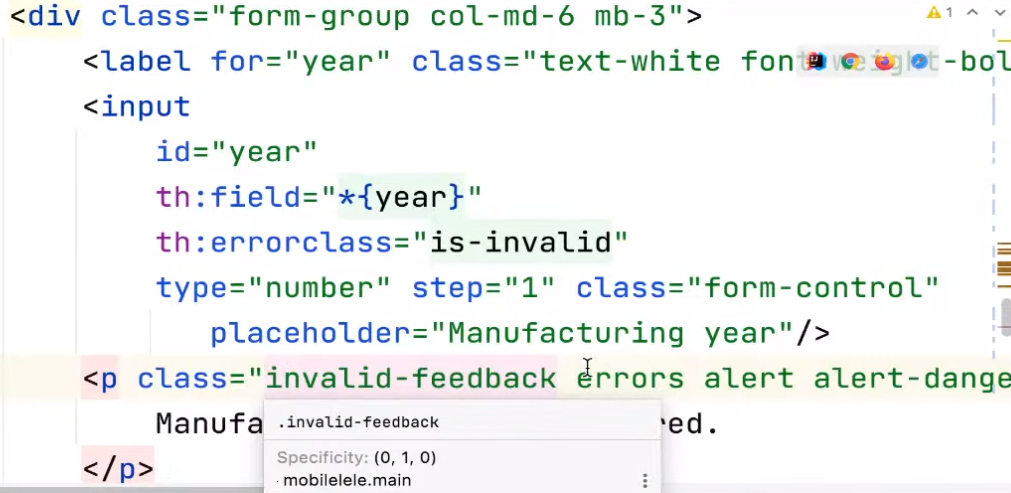
1. **NotFound exception – custom class**





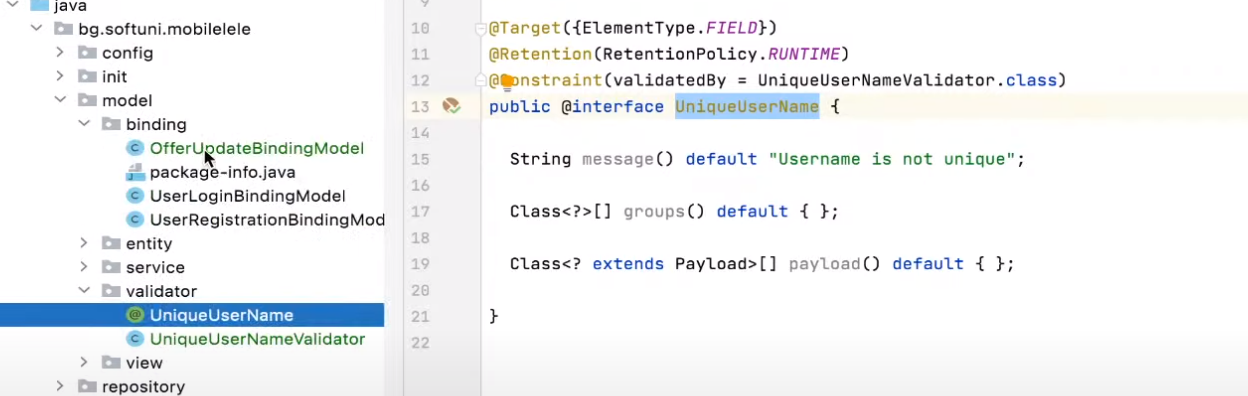
1. **Errorclass thymeleaf**

* При грешка във валидацията “is-invalid” = true и ще актвиира “invalid-feedback” и ще визуализира текста. Необходими са полета th:field и th:errorclass



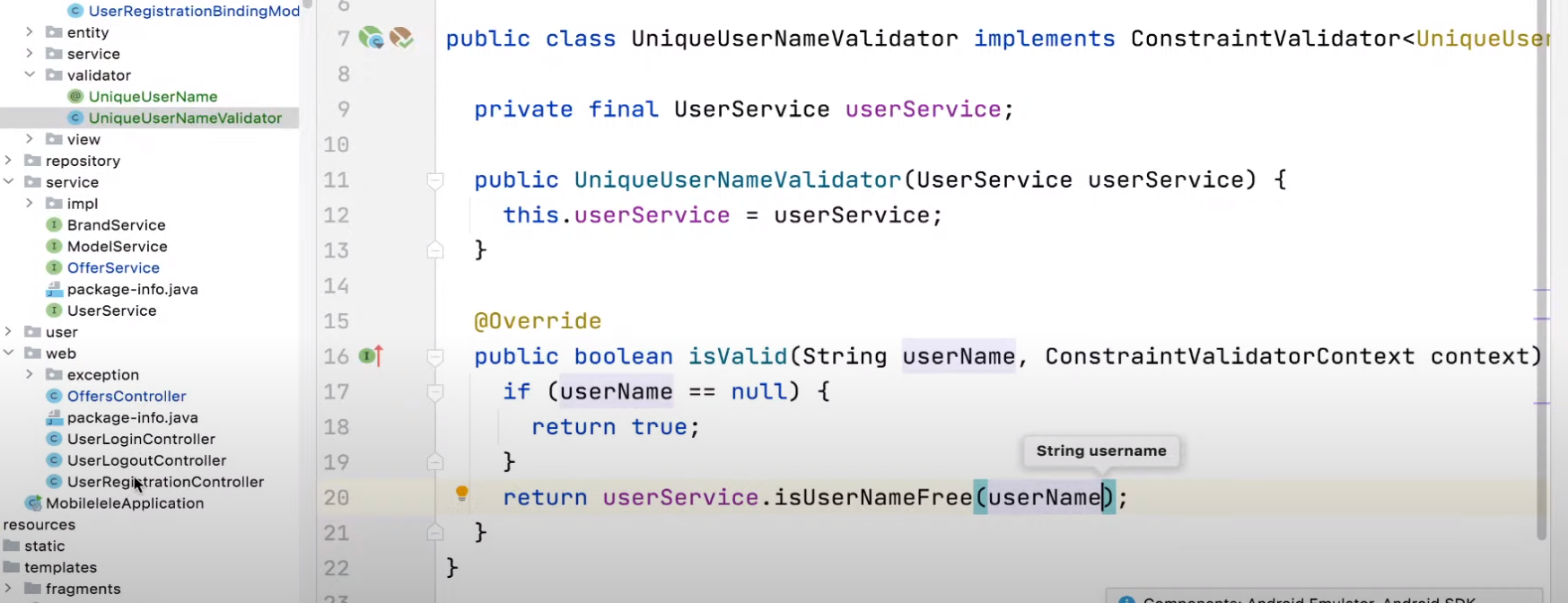
1. **Custom validator**

* Създаване на анотация



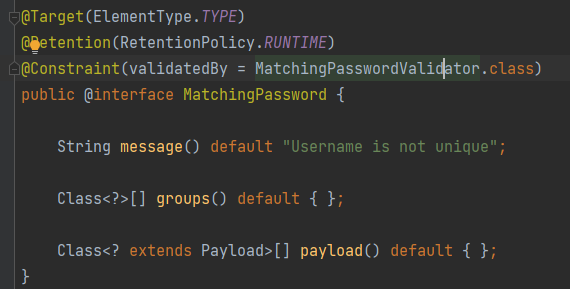
* Създаване на клас който имплементира ConstrainValidator<”annotation class”, “datatype”>

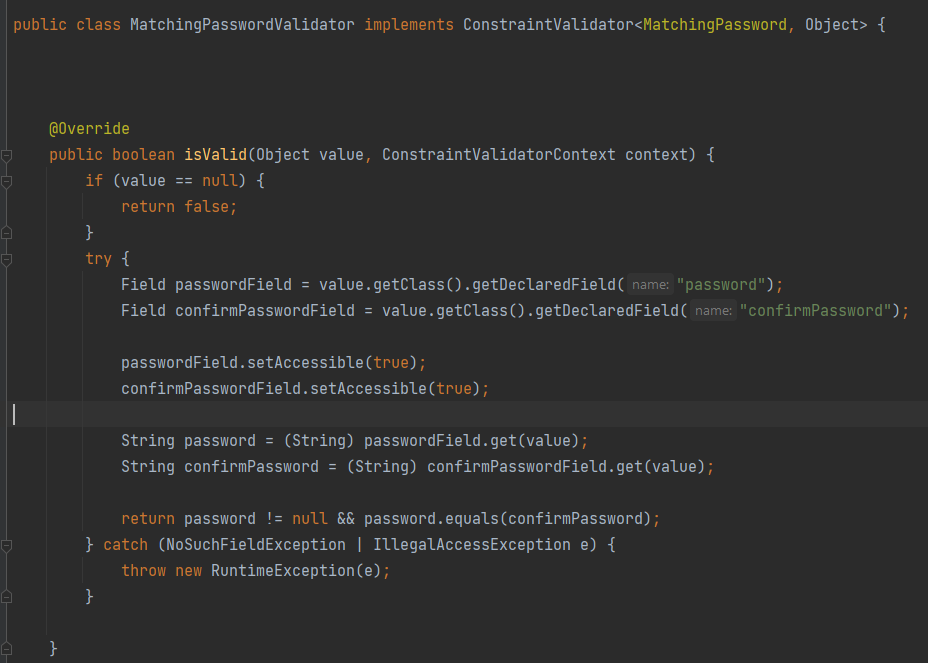
ConstraintValidator<UniqueUserName, String>

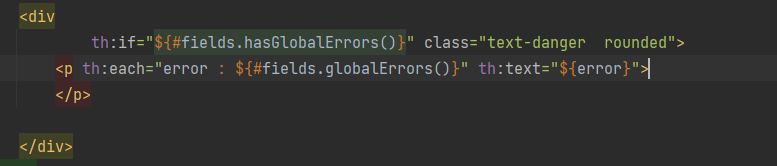


1. **Custom validator for matching passwords**

* Проверява се с .hasGlobalErrors() дали има грешка при валидиране на класа



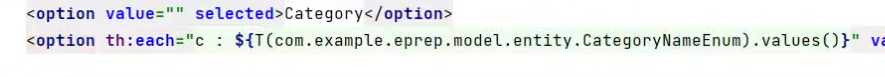




1. **IterStat при итерация с th:each**

**-**добавя се като 2-ри параметър след ,

1. **Accessing enumerations from thymeleaf**



1. **Multi class –** с единични кавички



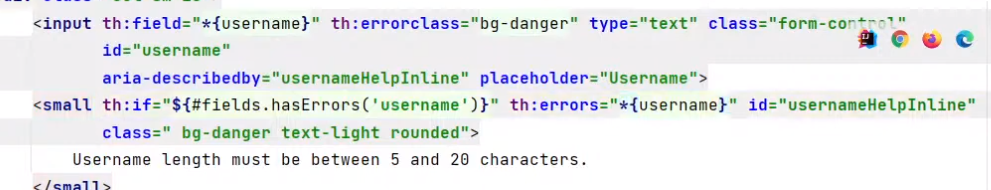
1. **Concatenation path with thymeleaf**



1. **Escaping white spaces with | |**

* Th:text=” |Number of orders: \*{count} | “

1. **Background color when error with custom message**

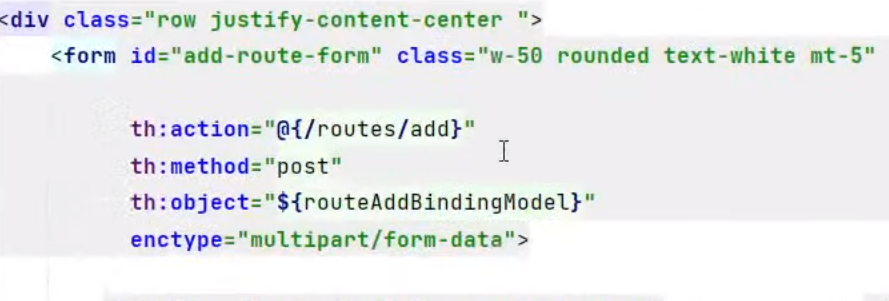


1. **Upload file -option**

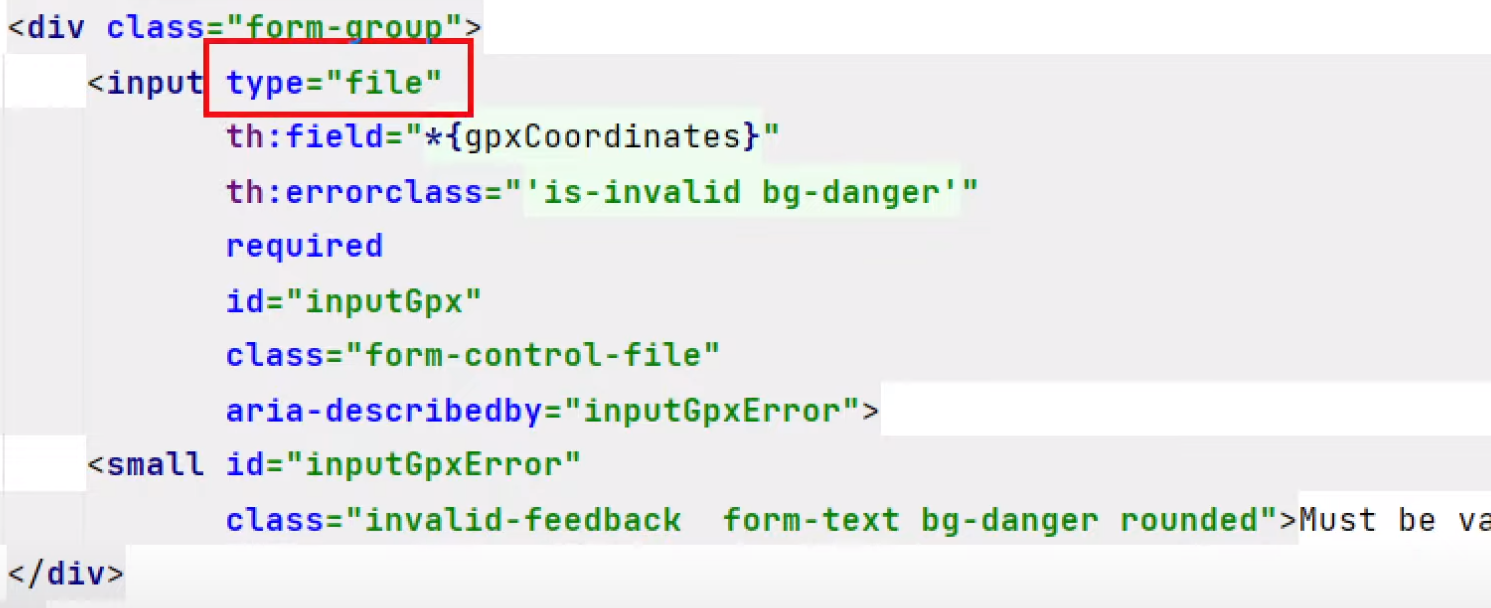
* Съдава се поле от тип MultipartFile



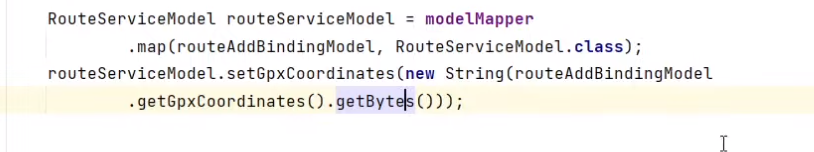
* Дефинира се че във формата има мултипарт файл с enctype



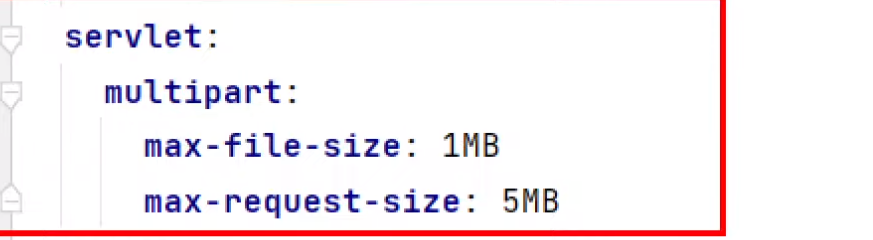
* Задава се на <input> type=”file”



* Обръща се в String мултипарт файла



* Лимитиране на MultipartFile



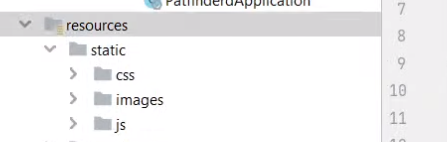
1. **Fetch REST API**

* “box-a-img” е ID от HTML файла



* Добавя се скрипт в HTML и в директория ресурси .js файл в папка js
* Посочва се пътя след /resources/static

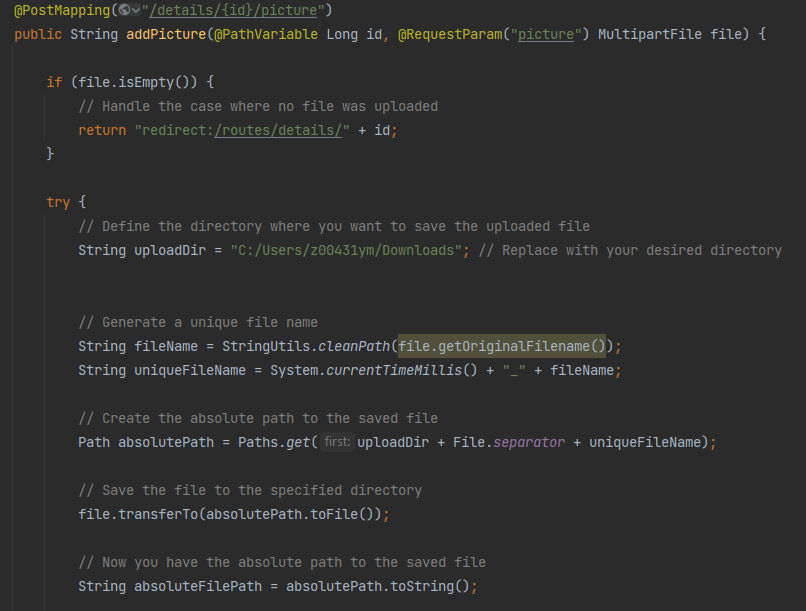




1. **Upload multipart file with @RequestParam(“name”) and transferTo different location + get path**

* Важно е в @Requestparam да се подаде името от HTML темплейта в този случай

name=”picture”





1. **Pageable and PageRequest to LIMIT selection of JPQL Query**

* Лимитира резултатите според входните параметри PageRequest.of()

