

Programowanie komponentowe – ćwiczenie 8

Cel ćwiczenia:

Obsługa formularzy

1. Przygotować tabelę users zgodną z klasą domenową User.

2. Do wygenerowania projektu proszę użyć Spring Initializr z wybraną listą starterów:

spring-boot-starter-web

spring-boot-starter-thymeleaf

spring-boot-starter-data-jpa

3. Utworzenie kontrolera w pakiecie .controllers

@Controller

```
public class UserController {
```

```
    @GetMapping("/adduser")
```

```
    public String userForm(Model model) {
```

```
        model.addAttribute("user", new User());
```

```
    return "add-user";
```

```
    }
```

```
    @PostMapping("/adduser")
```

```
    public String userSubmit(@ModelAttribute @Valid User user, BindingResult
```

```
bindingResult, Model model) {
```

```
    if (bindingResult.hasErrors()) {
```

```
        return "add-user";
```

```
    }
```

```
    // TODO zapis User do bazy
```

```
    model.addAttribute("user", user);
```

```
    return "user-info";
```

```
    }
```

```
    // TODO dodać usługę GET /userinfo do pobrania danych usera. Parametrem usługi ma być id usera.
```

```
}
```

4. Klasa domenowa w pakiecie .domain

// TODO uzupełnić adnotacje JPA

```
public class User {
```

```
    // TODO dodać walidacje w postaci adnotacji
```

```
    private long id;
```

```
    private String name;
```

```
    private String email;
```

```
    // standard constructors / setters / getters / toString
```

```
}
```

5. Utworzenie widoku

wszystkie pliki widoków umieszczamy w katalogu: src/main/resources/templates/

5.1. add-user.html

```
<!DOCTYPE HTML><html xmlns:th="https://www.thymeleaf.org">
```

```
<head>
```

```
<title>Rejestracja użytkownika</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

```

</head>
<body>
<h1>Formularz rejestracji</h1>
<form action="#" th:action="@{/adduser}" th:object="${user}"
method="post">
<p>Id: <input type="text" th:field="*{id}" /></p>
<p>Name: <input type="text" th:field="*{name}" /></p>
<p>Name: <input type="text" th:field="*{email}" /></p>
<p><input type="submit" value="Submit" /> <input type="reset"
value="Reset" /></p>
</form>
</body>
</html>

```

```

// TODO uzupełnić wyniki walidacji, np.
<p class="<ZDEFINIOWANY STYL DLA OPISU BŁĘDU>" th:if="${
#fields.hasErrors('name')}" th:errors="*{name}" />

```

5.2. user-info.html

```

<!DOCTYPE HTML>
<html xmlns:th="https://www.thymeleaf.org">
<head>
<title>Informacja o użytkowniku</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
<h1>Użytkownik</h1>
<p th:text="'id: ' + ${user.id}" />
<p th:text="'name: ' + ${user.name}" />
<p th:text="'email: ' + ${user.email}" />
<a href="/adduser">Kolejny użytkownik</a>
</body>
</html>

```

6. Dodać repozytorium UserRepository w pakiecie .repository

7. Uruchomienie aplikacji

Po uruchomieniu aplikacji należy z przeglądarki wywołać adres:

<http://localhost:8080/adduser> w celu dodania użytkownika lub
<http://localhost:8080/userinfo?id=<id usera>> w celu pobrania danych