Übung 4: Create Formular implementieren

Ausgabe: 8.10.18
Besprechung: 15.10.18

Aufgabe 1: Subview "create.html" implementieren

Nutzen sie Listing 1, um die Subview "create.html" zu implementieren. Diese Subview muss in die Applikation aus AB8 integriert werden und das dort definierte Layout nutzen.

```
<form th:action="@{/questionnaires}" th:object="${questionnaire}" method="post">                    #1
    <div class="form-group row">
        <label class="col-md-2" for="title">Title:</label>
        <div class="col-md-10">
            <input type="text" id="title" class="form-control" th:field="*{title}" autofocus="autofocus" />    #2
        </div>
    </div>
    <div class="form-group row">
        <label class="col-md-2" for="description">Description:</label>
        <div class="col-md-10">
            <input type="text" id="description" class="form-control" th:field="*{description}" />
        </div>
    </div>
    <div class="form-group row">
        <a class="btn btn-default" th:href="@{/questionnaires}">Cancel</a>                          #3
        <button type="submit" class="btn btn-primary">Save</button>                                 #4
    </div>
</form>
```

Listing 1: Das Form-Element mit Bootstrap CSS-Klassen

Hinweise:
#1: Auf dieser Zeile wird der HTTP Request festgelegt. Das Attribute "th:action" legt die URL (hier relativ zur Root-URL) fest, wenn der Request abgeschickt wird. Mit "th:object" wird auf das Model-Bean verweist, das innerhalb der Form genutzt wird und das im POST Request mitgegeben wird.

#2: Das Attribure "th:field" referenziert den Titel des Questionnaire. Konkret wird die Methode "getTitle()" auf der Instanz bzw. Model-Bean Questionnaire aufgerufen.

#3: Bei einem Abbruch wird über den HTTP-Link ein GET Request auf "/questionnaires" ausgelöst.

#4: Beim Submit wird die in Zeile #1 definierte Aktion ausgeführt.

Aufgabe 2: QuestionnaireController um CREATE Funktionalität erweitern

Sie müssen den gesamten Ablauf in 2 Schritten ausführen:

1. Zuerst wird über einen **GET**-Request das create-Formulare aus Aufgabe 1 geladen. Der entsprechende Request kann wie folgt formuliert werden: ".../questionnaires?form"

   Warum braucht es diesen zusätzlich "form"-Parameter im HTTP-Request?

2. Nachdem das Formular ausgefüllt wurde, werden die Formulare-Daten mit Submit als **POST**-Request zur Webapplikation gesendet. Der QuestionnaireController muss diesen Request entgegennehmen und den Questionnaire mit Hilfe des "QuestionnaireRepository" abspeichern.

   WICHTIG: SpringMVC kann den Payload aus einem POST Request automatisch in eine Questionnaire-Instanz konvertieren. Man muss diese mit einer entsprechenden Methoden Signatur kennzeichnen, wie:

   public String create(**Questionnaire questionnaire**) { ... }

Aufgabe 3: Subviews "list.html" und "show.html" aktualisieren

Erstellen sie eine Kopie ihrer Subviews "show.html" und "list.html". Ersetzen sie dann die beiden durch die entsprechenden Versionen aus dem AD-Ordner "UB4-Files".

Was sind die Änderungen, die diese Files in das Projekt einbringen?

Aufgabe 4: Test

Testen sie in ihren Controller die beiden CRUD-Operationen:

- CREATE explizit mit der URL "localhost:8080/flashcard-mvc/questionnaires?form" und über den Link auf der Startseite "list.html".
- READ aus der Applikation.