

OTH-Regensburg
Übungen zur Vorlesung
Softwareentwicklung

Übung Nr. 2
@RequestParam , @PathVariable @Model, @ModelAttribute

1) Aufgabe 1

Verwenden Sie in Ihrem Controller die Annotation `@RequestParam`, um alle Parameter (einzeln) aus einem Formular mit Login und Passwort zu erhalten, das in einer HTML-Ansicht geschrieben ist..

The screenshot shows a login interface with the following elements:

- A title "Login" at the top.
- A text input field labeled "Login" containing the placeholder "Login".
- A text input field labeled "Password" containing the placeholder "*****".
- A blue rectangular button labeled "Go!".

2) Aufgabe 2

Fügen Sie mithilfe des vorherigen Formulars ein Feld „Rolle“ hinzu, das die Werte „Administrator“ und „Benutzer“ haben kann. Verwenden Sie `@RequestParam` als Map, um alle Parameter aus demselben Formular zu sammeln.

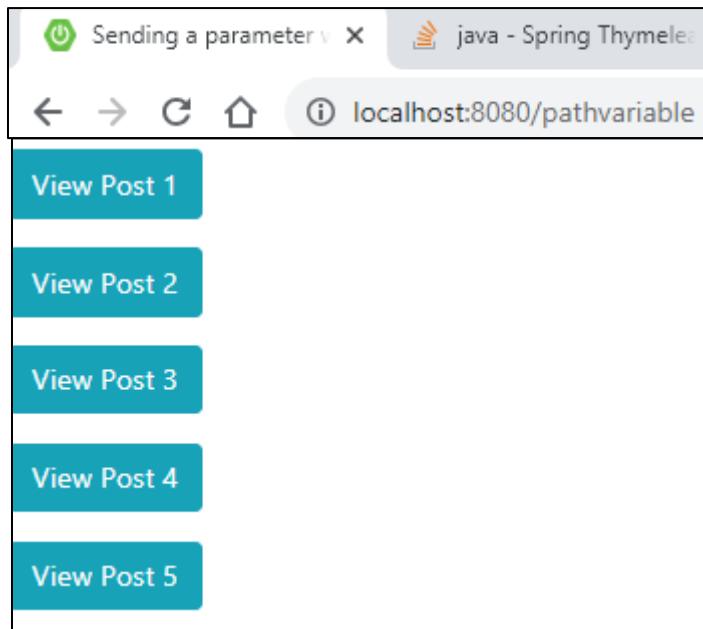
The screenshot shows an enhanced login interface with the following elements:

- A title "Login" at the top.
- A text input field labeled "Login" containing the placeholder "Login".
- A text input field labeled "Password" containing the placeholder "*****".
- A dropdown menu labeled "Role" with the value "Select" and a dropdown arrow icon.
- A blue rectangular button labeled "Go!".

3) Aufgabe 3

3.1 Verwenden Sie eine Collection vom Typ String im Controller, um mehrere „IDs“ zu erstellen, die mehrere Posts darstellen (z. B. Posts auf Facebook). Fügen Sie die Collection als Attribut in das Modellobjekt ein.

3.2 Zeigen Sie in dem View den Inhalt der Collection an, um anklickbare Links/Schaltflächen wie die in der folgenden Abbildung zu generieren.



3.3 Nach dem Klicken auf einen Link sollte die ID des Posts als **@PathVariable** an den Controller gesendet werden.

3.4 Verwenden Sie die **@PathVariable**, um im Controller den Wert der ID eines Posts zu empfangen, der angezeigt werden soll, nachdem der Benutzer auf seinen Link geklickt hat.

3.5 Wenn Sie auf einen Beitrag klicken, wird ein weiteres View erstellt, um die Anzeige des Beitrags wie folgt zu simulieren:



Aufgabe 4

Erstellen Sie eine Klasse „Student“ im neuen Paket „Modell“. Die Studentenklasse sollte dieselben Attribute wie das Formular „student-add.html“ haben: Name, Alter, E-Mail. Verwenden Sie die Taglib von Thymeleaf, um das Formular dem Objekt „Student“ zuzuordnen. Verwenden Sie auf der Controllerseite die Annotation @ModelAttribute, um die im Formular gesendeten Informationen zu empfangen.

Aufgabe 5

Erstellen Sie nun eine Klasse Address (street, city, state), die ein Attribut der Klasse STUDENT sein wird. Erstellen Sie das entsprechende Formular mit Studenten- und Klassenobjekten in dem View Student-add.html. Verarbeiten Sie anschließend die vom StudentController empfangenen Werte.