

Webentwicklung (WE) - FWPM

Übung 5: "Input/Output Management"

In dieser Übung geht es darum, wie wir zuverlässig mit Nutzereingaben umgehen, und den Nutzern eine saubere Ausgabe garantieren können. Unter Nutzereingaben sind dabei nicht nur Formulareingaben unter Zuhilfenahme eines Browsers zu verstehen. Auch Webservice Aufrufe, das Einlesen von hochgeladenen Dateien o.ä. muss entsprechend behandelt werden.

1. Eingabeformular

Um unsere Anwendung (im Beispiel weiter ein Blog) mit Inhalten zu füllen, wollen wir den Nutzern erlauben über ein Formular Inhalte an die Anwendung zu senden.

1. Überlege dir welche Felder du für das Schreiben eines Blog Posts brauchst und erstelle ein HTML Formular um diese Felder zu erfassen.
 - a. Die Felder werden sich vermutlich mit deinem BlogPost Model decken.
 - b. Nutze natives HTML um das Formular zu gestalten. Eine gute Referenz ist die Seite der w3school: https://www.w3schools.com/html/html_forms.asp
 - c. Mach dir bereits Gedanken welche einfache Validierung du brauchen könntest.

2. Formular-View

1. Erstelle eine View Klasse mit der sich das Formular rendern lässt.
 - a. Dazu musst du im Prinzip nur das vorbereitete HTML Formular ausgeben.

Bonus: Über das HTML *value* oder das *placeholder* Attribut kannst du den aktuellen Stand der Daten aus der Datenbank einblenden. So kannst du die View auch für das Editieren nutzen sofern der Controller dies anbietet.

3. Formular-Controller

1. Erstelle einen Controller der sowohl die View rum Rendern des Formulars nutzt, als auch die über das Formular gesendeten Daten in die Datenbank speichert.

- a. Die Unterscheidung was davon gerade gebraucht wird kannst du über die Abfrage nach gesendeten Formulardaten machen. Wird nichts gesendet rendere die leere View ansonsten erzeuge ein Model und schreibe es in die Datenbank.
- b. Nutze deine Persistenzschicht und dein Model um die Daten zu speichern.
- c. Integriere die Route deines Controllers in deine Anwendung und in das *action* Attribute deines HTML Formulars.

4. Validierung

1. Baue eine sinnvolle Validierungsschicht in deinen Controller.
 - a. Diese sollte die übertragenen Formulardaten validieren und wenn nötig filtern bevor sie weiter verarbeitet werden.
 - b. Nutze am Besten eine Validierungsbibliothek wie sie in der Vorlesung angesprochen wurde.
2. Wir wollen die Ergebnisse von dem sehen was die Nutzer über die Formulare hinzugefügt haben. Erweitere deinen BlogController so, dass du die neuen Blogbeiträge abrufen kannst
 - a. Falls du bereits ein dynamisches System z.B. über URL-Slugs implementiert hast, ist nichts mehr zu tun
3. Innerhalb von Controller oder View solltest du dir Gedanken zum Encoding/Filtering/Escaping deiner Ausgabe machen. Die Nutzung von *htmlspecialchars()* auf alle Felder sollte das Mindeste sein.