

Webentwicklung (WE) - FWPM

Übung 3: "Nutzerinteraktion über Controller und Views"

Controller

Nachdem ein Routing vorhanden ist, brauchen wir Controller Klassen, die die Verarbeitung unserer Requests übernehmen können. Der Einfachheit halber soll mit einer Controller Klasse begonnen werden, die eine Action zum Abrufen einer Detailansicht eines Blogbeitrags hat.

Da wir noch keine Persistenzschicht haben, werden die Blogbeiträge als sog. Fixtures als reiner Text in die Controller eingebaut.

1. Erstelle eine Klasse die als Controller für Blogbeiträge fungieren kann.
 - a. Erstelle die Action/Methode innerhalb der Klasse.
 - i. Erstelle mindestens zwei anonyme Klassen oder ein assoziatives Arrays die die Model von Blogbeiträgen simuliert
 - ii. Es sollte mindestens die Felder *Titel*, *Autor* und *Text* haben
2. Deine Controller Action muss unterscheiden können welchen der Beiträge sie an die View weiterleiten will. Daher muss deine Action das Request Objekt nutzen
 - a. Überlege dir welche Art von Eingabe (URL, Parameter, ...) den Blogbeitrag identifiziert
 - b. Mache testweise eine Ausgabe mit *echo* bevor die View erstellt wird

Routing

Dein eben erstellter Controller muss über eine URL erreichbar sein. Passe dazu deinen Controller/Route Mapping entsprechend an dass es diese Aufgabe übernehmen kann.

1. Implementiere einen Mechanismus zum Mapping deines Blog Controllers auf eine Route. Eine Route wie */blog/show* z.B.
 - a. Der vorgestellte Weg über Annotationen ist zwar architekturell sauber aber ohne eine weitere Bibliothek nur aufwändig umzusetzen. Solltest du dich dafür entscheiden nutze auf jeden Fall eine Bibliothek wie z.B. *doctrine/annotations*
 - b. Lass deinen Controller durch deinen Router aufrufen
2. Prüfe ob dein Controller mitsamt Action unter deiner gewählten Route erreichbar ist.
3. Vor allem für Blogbeiträge und ähnlichen Content bietet sich eine explizit eindeutige URL an. Stelle also sicher, dass dein Router mit einem URL Format wie */blog/show/{Titel des Blogbeitrags}* umgehen kann

View

Deine testweise eingebaute Ausgabe über *echo* innerhalb deines Controllers muss ersetzt werden. Erstelle dazu eine View Klasse und ein passendes Template zur Darstellung deines Beitrags in HTML.

1. Erstelle eine Viewklasse deren Aufgabe es ist, die Daten die durch den Controller bereitgestellt werden zu rendern
 - a. Deine View sollte eine Methode *render* anbieten der die Daten vom Controller übergeben werden können
 - b. Ergebnis der *render* Methode sollte ein String sein den dein Controller eventuell weiterverarbeiten kann. Integriere daher eine Möglichkeit auch HTML Templates als String zu erhalten (vergleiche Vorlesung)
2. Erstelle ein HTML Template zur Darstellung deines Blogbeitrags
 - a. Schaffe eine grundlegende HTML Struktur die deinen Datensatz repräsentiert. Mache es durch eingebundenes PHP dynamisch für verschiedene Blogbeiträge nutzbar
 - b. Binde das Template in deine Viewklasse mit ein und nutze es als Basis für dein Rendering
3. Ersetze das *echo* in deinem Controller durch das Rendering deiner Viewklasse
 - a. Nutze die Response Klasse um deine Ausgabe gezielt zu sammeln
 - b. Implementiere eine zentrale Ausgabe deiner Response Klasse. Dazu reicht im ersten Schritt ein *echo* deines Response Bodies z.B. am Ende der *index.php*. Später wird noch die Verarbeitung verschiedener Response Header und Status Codes mit einfließen