

Klassendesign

Durch die Verwendung des Spring Web Frameworks wird die grobe Klassenstruktur bereits vorgegeben, nämlich in Form des MVC-Pattern für Webanwendungen.

In der Anwendung wird die Präsentationsschicht von den Daten getrennt. Das heißt für unser Projekt: JSP sorgt rein für die grafische Benutzeroberfläche (ohne Business Logic) und die Datenübermittlung zwischen dieser View und der Datenbank (Model) wird über einen Controller gesteuert.

Unsere Applikation wird folgendermaßen eingeteilt:

- **Model:**
 - Produkt (alle Attribute, die ein Produkt auf pandabay haben kann, und die dazugehörigen Get-/Set-Methoden)
 - Auktion (beinhaltet Daten, die nicht direkt zum Produkt gehören, eher zum Auktions-Workflow, z.B. Laufzeit, höchstes Gebot, etc., ein Vorteil dieser Trennung ist, das man zu einem späteren Zeitpunkt mehrere Produkte zu einer Auktion zusammenfassen kann)
 - Benutzer (Zugangsdaten, Adresse, usw. - alles was für ein Benutzerprofil wichtig ist - und wieder Getter/Setter um auf diese privaten Variablen zugreifen zu können)
- **View:** zeigt die Daten des Models in einer benutzerfreundlichen Web-Applikation. Das GUI wird mittels JSP realisiert
 - Login
 - Register
 - Home
 - ListProduct
 - DetailProduct
 - NewProduct
 - NewBid
 - uvm.
- **Controller:**
 - NewUserController
 - NewProductController
 - NewBidController
 - LoginController
 - RegisterController
 - uvm.

Um die Business Logic von den eigentlichen Daten zu trennen, greifen wir noch auf das sogenannte DAO Pattern (Data Access Object) zurück. Dabei lagern wir die Datenzugriffslogik (z.B. von Produkt oder User) in eine separate Klasse aus. Die eigentliche Model-Klasse ist nicht mehr von einer bestimmten Implementierung (Datenbankzugriff) abhängig. In unserem Fall gibt es dann austauschbare JdbcDao-Klassen, in denen die Implementierung auf eine MySQL-Datenbank erfolgt. Um die Abhängigkeiten zu minimieren, führen wir noch Interfaces für die jeweiligen Dao-Klassen ein.

Durch das zentrale DispatcherServlet (Spring Framework) können wir den MVC-Teil vom Rest unserer Applikation trennen. Die Implementierung erfolgt in der Serverkonfiguration (web.xml).