[Pleaseinsertintopreamble]bersicht Einleitung Robuste Programmierung im Controller und Model Validierung von Inputs Views Demo Ende

FHNW APSI Lab 2: Html Form

Roland Hediger, Jonas Schwammberger

16. Dezember 2013

- Einleitung
- 2 Robuste Programmierung im Controller und Model

Ende

- Views
- Demo

- Tomcat 7, JDBC/MySQL, javax.mail
- Java HTTP Servlet Applikation
- MVC Pattern kein Framework
- 1 Servletklasse
- mehrere Methoden die auf 1 Controller aufrufe machen.
- 1 Model Company.java DAO mithilfe JDBC

- Information Hiding soviel private und final wie möglich
- Controller zustandslos behalten. Hat nur die Verantwortung für DB Verbindung
- Nach Setzen von Properties im DAO werden die Werte automatisch zuerst als nicht gültig betrachtet.
- Annotationen begrentzt eingesetzt

- Instanzvariable "Valid"
- List<String> messages wird gefüllt mit Benachrichtigungen beim Auftreten von Fehler.
- Nur falls valid = true ist im Company kann ein Eintrag gespeichert werden.
- Jeder Formulareintrag wird gegen ein "CleanString" (Regexausdruck) gematcht.
- Passwort Wechsel :
- Seperate Validerung , eingeben von altes Passwort vorrausgesetzt .

```
private static final String usrCleanString = "[ôŌēĒāÅeéÉĒāöūÄÖŪß\\-\\._\\w]{4,64}"; private static final String pwdCleanString = "[ôŌēĒāÅeéÉĒāoūÄÖŪß\\-\\._\\w\\d]{8,64}"; private static final String adrCleanString = "[ôŌēĒāÅeéÉĒāoūÄÖŪß\\-\\._\\w]{0,255}"; private static final String townCleanString = "[ôŌēĒāÅeéÉĒāoūÄÖŪB\\-\\._\\w]{0,255}"; private static final String companyCleahString = "[ôŌeĒāÅeéÉĒāoūÄÖŪB\\-\\._\\w]{0,20}";
```

[Pleaseinsertintopreamble]bersicht Einleitung Robuste Programmierung im Controller und Model Validierung von Inputs **Views** Demo

Einsatz von JSP:

```
request.setAttribute("messages", messages);
request.getRequestDispatcher(REGISTER).forward(
    request, response);
```

Beispiel JSP Seite:

```
<form method="POST" action='/apsi_lab2/</pre>
      RattleBits ' name="login">
   3 <%
     List < String > messages = (List < String >)
        request.getAttribute("messages");
 Iterator<String> it = messages.iterator();
     while (it.hasNext()) {
 %>
8 %= it.next() %>
 <% } %>
```

Ende

[Pleaseinsertintopreamble]bersicht Einleitung Robuste Programmierung im Controller und Model Validierung von Inputs Views Demo

Demo

[Pleaseinsertintopreamble]bersicht Einleitung Robuste Programmierung im Controller und Model Validierung von Inputs Views Demo

Danke für Ihre Aufmerksamkeit.