

Labor der Vorlesung Enterprise Anwendungen 1, SS2018

**Hochschule Offenburg – Fakultät E+I
Studiengang AI, IWp, WIN**

Prof. Dr. Hartwig Grabowski

Bearbeitungshinweise:

1. Eigenständig – kein Copy & Paste. Den Nachbarn fragen ist natürlich erlaubt.
2. Bei Fragen darf und soll selbstständig im Internet recherchiert werden.
3. Zum Abgabetermin sind die Aufgaben selbstständig in Moodle hochzuladen. Es reicht ein ZIP-Datei des jeweiligen Workspaces aus.
4. Es werden keine Punkte vergeben. Es sind prinzipiell alle Aufgaben hochzuladen. Aufgabe die nicht zwingend hochzuladen sind, werden mit dem Zusatz *optional* versehen.

Hinweise:

- Buttons, CheckBox, etc. immer in ein Formular einbetten `<h:form> </h:form>`.
- Für Ajax zunächst mit `render="@all"` ausprobieren.
- `<h:body> <h:head>` verwenden.
- Den Server TomEE+ **nicht** in C:\Programme installieren sondern in c:\tomEE o.ä.
- Setter und Getter sind nie static
- Für Felder immer in Kleinbuchstaben verwenden
- Bei ManagedBean: `import javax.faces.bean.ManagedBean;`

1 Java Server Faces

1.1 „Hallo World“ - Anwendung

Erstellen Sie eine JSF-Anwendung, die „Hallo Wolrd“ ausgibt.

Erzeugen Sie dazu ein neues Projekt „L3_Hallo“ (Java Web -> Web Application) und wählen Sie dann Glassfish Server 4 und **Java EE 7 Web**. Aktivieren Sie in Frameworks „JavaServer Faces“.

a) Starten Sie die Anwendung `index.xhtml`.

b) Fügen Sie eine `<h:outputText>`-Komponente ein, die „Hallo again“ ausgibt.

c) Sie können die `index.xhtml`-Seite im Browser unter zwei verschiedenen URLs aufrufen:

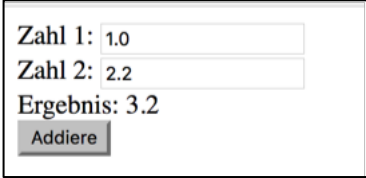
unter `http://localhost:8080/L3_Hallo/faces/index.xhtml` und

unter `http://localhost:8080/L3_Hallo/index.xhtml`.

Warum erhalten Sie unterschiedliche Ausgaben? Schauen Sie dazu auch in die `web.xml` Datei.

1.2 Taschenrechner - Anwendung

Erstellen Sie eine JSF-Anwendung, die einen Taschenrechner implementiert. Sie müssen nur den Addiere-Button implementieren, d.h. zwei Zahlen addieren.



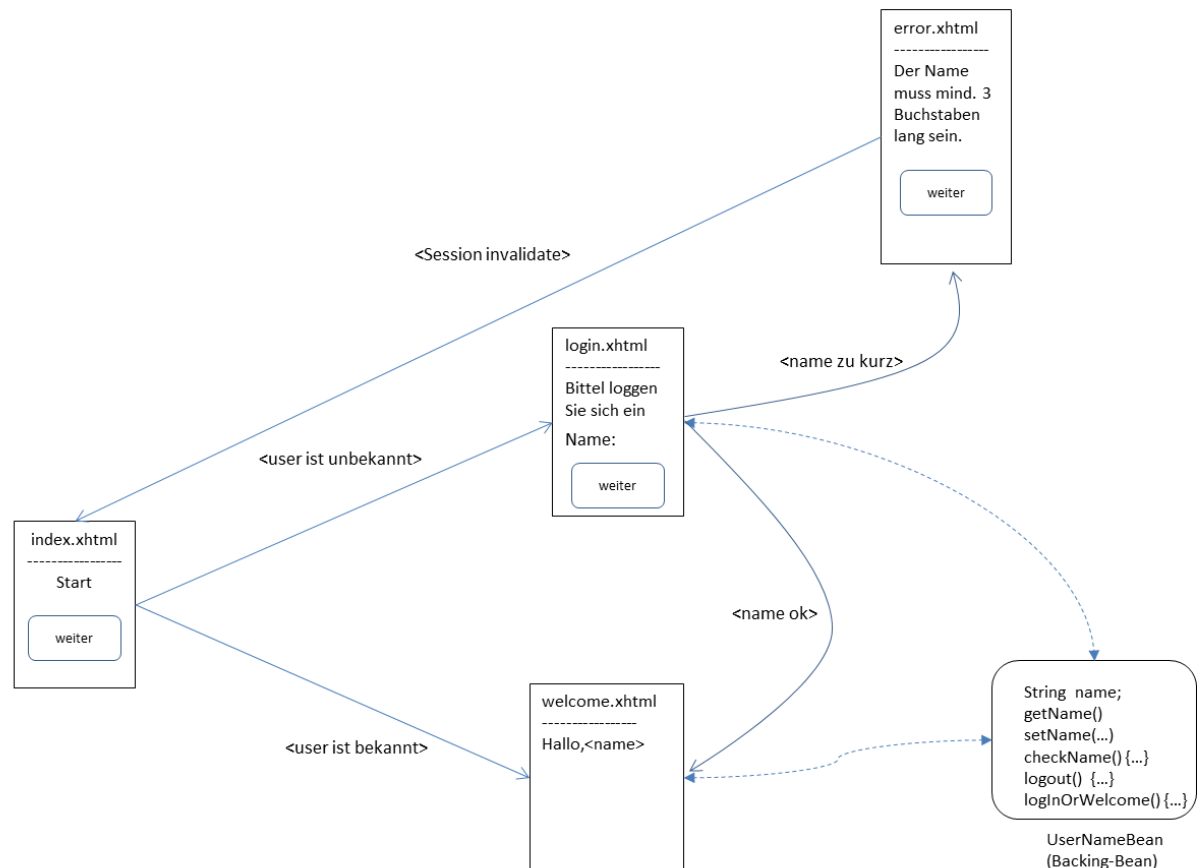
Sie benötigen dazu die folgende JSF-Komponenten:

```
<h:head>
<h:body>
<h:form>
<h:outputText />
<h:inputText />
<h:commandButton />
```

2 MVC

2.1 Implementieren Sie das Projekt L3_MVC_JSF

Sie sollen eine JSF- Anwendung erstellen;
 der Ablauf soll nach folgendem Schema erfolgen:



Die Methode `loginOrWelcome()` der Backing-Bean `UserNameBean` ermittelt, ob der User bekannt ist oder nicht, d.h. ob der Name der in der Backing-Bean bereits eingetragen ist oder nicht und gibt als Rückgabewert die entsprechende.xhtml-Seite aus.

Für `<session invalidate>` benötigen Sie die Methode `logout()` der `UserNameBean`. In der Methode müssen Sie dann die Methode `FacesContext.getCurrentInstance().getExternalContext().invalidateSession();` ausführen und zu `index.xhtml` weiterleiten.

Hinweis: Zum Weiterleiten auf eine Folgeseite verwenden Sie bitte immer die **action**-Komponente.

2.2 Backing-Bean

Kopieren Sie nun das Projekt **L3_MVC_JSF** und fügen es als **L3_MVC_JSF_2** ein.

Sie möchten die **UserNameBean** um die Einträge **Email** und **Age** erweitern. Ergänzen Sie die **UserNameBean**, damit Sie für den User seine Email und sein Alter eintragen können.

(Hinweis: getter und setter-Methoden können Sie schnell mit Source->InsertCode->... hinzufügen.)

Passen Sie die Login-Seite entsprechend an, so dass neben Name noch Email und Age eingetragen werden können.

2.3 Validation

Sie möchten auf Modell-Ebene verhindern, dass Age kleiner 1 oder größer 120 ist.

Sie möchten weiterhin verhindern, dass Email weniger als 3 Buchstaben oder mehr als 50 Buchstaben enthält

Fügen Sie dazu die entsprechenden Annotationen in die Backing-Bean ein.

(siehe: <http://docs.oracle.com/javaee/6/tutorial/doc/gircz.html>)

2.4 Life-Cycle

Sie möchten auf der Login-Seite einen cancel-Button einfügen, der die Session zurücksetzt **invalidateSession()** und zur Startseite weiterleitet.

Fügen Sie den „Cancel“-Button auf der Login-Seite ein und fügen Sie in die Backing-Bean eine entsprechenden Methode **cancel()** ein.

Lerneffekt: Sie müssen den Life-Cycle nach Phase 2 abbrechen, damit Ihre Einträge Age und Email nicht überprüft werden.

2.5 Events

Auf der Ausgabeseite **welcome.xhtml** soll nun optional neben dem Namen auch Alter und Email ausgegeben werden. Die Ausgabe von Email und Alter soll aber nur erfolgen, wenn die CheckBox „More“ angeklickt wurde.

Erstellen Sie dazu eine Formular, in das Sie die eine CheckBox-Komponente (**<h:select-BooleanCheckbox>**, http://www.jsftoolbox.com/documentation/help/12-TagReference/html/h_select-BooleanCheckbox.html) einfügen.

Der Wert dieser CheckBox-Komponente soll in dem (zu erstellenden) Feld **Boolean show** in der Backing-Bean **UserNameBean** gespeichert werden.

Damit direkt beim Anklicken der Check-Box eine Reaktion erfolgt, müssen Sie das onclick-Attribut der CheckBox setzen (**onclick="submit();"**) und dadurch ein erneutes Laden der Seite erzwingen.

Die Felder Email und Age sollen beim Laden der Seite nur dann ausgegeben werden, wenn **show=true** ist. Verwenden Sie für die Ausgabe eine **<h:outputText>**-Komponente, deren Ausgabe Sie über das Attribut **rendered="..."** steuern.

2.6 Ajax

Packen Sie die beiden Textfelder (Email und Age) in eine PanelGroup-Komponente `<h:panelGroup>` mit der ID `id="out"`.

Sie möchten nun, dass nicht immer die gesamte Seite erneut geladen wird, wenn Sie die Check-Box ändern, sondern nur die PanelGroup „out“. Somit muss `onclick="submit();"` durch einen Ajax-Request ersetzt werden.

Dazu löschen Sie `onclick="submit();"` und fügen in das Formular eine `<f:ajax>`-Komponente ein. Die Ajax-Komponente soll in diesem Formular (`@this`) auf den `click`-Event reagieren und es sollen alle Elemente „out“ neu dargestellt werden.

(http://www.jsftoolbox.com/documentation/help/12-TagReference/core/f_ajax.html)

Beachten Sie: Wenn die PanelGroup nicht mehr dargestellt wird, kann Sie auch mittels AJAX (und dem entsprechend generierten JavaScript) nicht mehr neu gezeichnet werden, weil die PanelGroup ja nicht mehr existiert. Deshalb muss die PanelGroup immer gezeichnet werden und die Elemente innerhalb der PanelGroup via `rendered="..."` ein- oder ausgeblendet werden.