



## Softwarepraktikum im Grundstudium im Wintersemester 2002/2003

### 4. Übung, 21. November 2002

In dieser Übung soll die Funktionalität der Webanwendung um das Löschen und das Ändern von Aufgaben erweitert werden.

#### 1. Aufgabe (Detailansicht für Aufgaben)

Fügen Sie der Aufgabenliste in der JSP `ShowTasksView` für jeden Eintrag der Liste einen Link auf ein Servlet `DetailedTaskPrepare` hinzu. Dieses bekommt im Request den Key der gewählten Aufgabe übergeben (wie bei `http-Get` direkt in der URL), lädt diese aus der Datenbank und gibt die Anfrage an das JSP `DetailedTaskView` weiter. `DetailedTaskView` zeigt alle Daten der Aufgabe an und verfügt über einen Link zurück zur Aufgabenliste. Implementieren Sie das Servlet und die JSP.

Um dies zu ermöglichen, muss im `StorageAccess` eine Methode implementiert werden, die zu einem gegebenen key das entsprechende mit den Daten aus der Datenbank gefüllte Aufgabenobjekt zurückliefert.

#### 2. Aufgabe (Ändern von Aufgaben)

Erweitern Sie in der JSP `ShowTasksView` die Tabelle mit den Aufgaben um einen Link für das Ändern der jeweiligen Aufgabe. Dieser soll auf das Servlet `ModifyTaskPrepare` verweisen. Um die zu ändernde Aufgabe zu übergeben, soll wie oben direkt in der URL wie bei einem `http-GET` der Aufgaben-Schlüssel (`key`) übergeben werden. Das Servlet `ModifyTaskPrepare` lädt die anhand des Keys identifizierte Aufgabe aus der Datenbank, fügt diese in den Request ein und leitet die Anfrage an die JSP `ModifyTaskView` weiter. Dort wird eine Eingabemaske, vergleichbar mit der für neue Aufgaben angezeigt, wobei die Felder natürlich mit den aktuellen Werten vorbelegt werden müssen. Der Schlüssel (`key`) ist nicht veränderbar, muss aber in einem unsichtbaren Feld gespeichert werden, damit er im verarbeitenden Servlet wieder ausgelesen werden kann. Der Submit-Button soll das Servlet `ModifyTaskExecute` aufrufen, das die geänderte Aufgabe über den `StorageAccess` in die Datenbank schreibt. Jetzt kann mit einem Redirect wieder die Aufgabenliste angezeigt werden.

In der Klasse `StorageAccess` muss hierzu eine neue Methoden implementiert werden, die ein Aufgabenobjekt übergeben bekommt und den zugehörigen Termin in der Datenbank mit einem UPDATE-Statement ändert. Verwenden Sie auch hier ein `PreparedStatement`.

#### 3. Aufgabe (Löschen von Aufgaben)

Wie in den Aufgaben 1 und 2 muss zunächst der entsprechende Link in der JSP `ShowTasksView` angelegt werden. Dieser verweist hier direkt auf ein Servlet `DeleteTaskExecute`, das mithilfe des `StorageAccess` die gewünschte Aufgabe aus der Datenbank löscht. Anschließend soll ein Redirect auf die Aufgabenliste ausgeführt werden.

Erweitern Sie den `StorageAccess` um die Methode `deleteTask`, die die Aufgabe mit dem übergebene Key aus der Datenbank löscht. Verwenden Sie ein DELETE-Statement.

## **SQL-Befehle**

```
UPDATE person SET name='Becker', vorname='Simon M.' WHERE  
key=0001
```

```
DELETE FROM person WHERE key=0001
```

## **Ausgaben auf System.out und System.err**

Im Verzeichnis ~/i3\_spws/tomcat/logs befindet sich die Datei catalina.out, die alle Ausgaben des Tomcat enthält.

## **Fehlerbehandlung**

Vergessen Sie nicht, wie in der letzten Übung eingeführt, alle Fehler abzufangen und auf der Fehlerseite anzuzeigen.

## **Manuelle Übergabe eines Parameters in der URL**

Hierzu kann einfach an die URL über ein Fragezeichen ein Paar aus Name und Wert angehängt werden. Mehrere werden durch ein kaufmännisches Und abgetrennt:

```
http://www.google.com/search?hl=en&q=Simon+Becker
```

## **Testat**

Donnerstag, 28.11.2002 von 10.00 - 11.45 Uhr