Philipps-Universität Marburg
Informatik-Praktikum im Grundstudium
Prof. Dr. B. Seeger
22.05.2005

Konzeption: Beans/EJB
Zusammenfassung der Einzel-Analysen
von Manuel Haim

Siehe auch: http://www.mathematik.uni-marburg.de/~haim/it/ipg/docs/beans.htm

Nebenstehend eine Auflistung von unseren Beans und deren Properties.

Vier elementare Befehls-Bohnen:
Create (sinngemäß auch: Generate, Add)
-zum Erstellen eines Objekts bei MU/AL/DB
-Syntax: Create <WHAT> <FROM/FOR>

Save (sinngemäß auch: Set)
-zum Speichern eines Objekts bei MU/AL/DB
-Syntax: Save <WHAT> <FOR/AS/TO>

Load (sinngemäß auch: Get)
-zum Laden eines Objekts bei MU/AL/DB
-Syntax: Load <WHAT>

Delete (sinngemäß auch: Destroy, Remove)
-zum Vernichten eines Objekts bei MU/AL/DB
-Syntax: Delete <WHAT> <FROM>

Anwendungsfälle
1. *Anmeldung am System*
Create(Session, User) -> Session

2. *Abmeldung*
Delete(Session) -> void

3. *Passwort ändern*
Save(User, User) -> void

4. *Suche nach Datensätzen*
Load(LitEntry, Search) -> LitList

5. *Literaturliste umbenennen (temporär)*
Save(LitList) -> LitList

6. *Eintrag aus Literaturliste löschen (temporär)*
Delete(LitEntry, LitList) -> LitList

7. *Eintrag in Literaturliste einfügen (temporär)*
Create(LitEntry, LitList) -> LitList

8. *Schließen einer Literaturliste (temporär)*
Delete(LitList) -> void

9. *Speichern einer Literaturliste (persistent)*
Save(LitList, LitList) -> LitList

10. *Rechte für Literaturliste ändern (persistent)*
Load(LitListRights, LitList) -> LitListRights[]
Load(LitListRights, LitList, User) -> LitListRights
Save(LitListRights, LitList, User) -> void
Delete(LitListRights, LitList) -> LitListRights[]
Delete(LitListRights, LitList, User) -> LitListRights[]

11. *Öffnen einer Literaturliste (die temporäre LitList falls verändert, sonst von DB)*
Load(LitList, Search) -> LitList[]
Load(LitList) -> LitList

12. *Literaturliste löschen (persistent)*
Delete(LitList, LitList) -> void

13. *Generierung einer BibTex-Datei zu einer Literaturliste*
Create(BibTex, LitList) -> BibTex
Create(File, BibTex) -> File

14. *Generierung einer Text-Datei zu einer Literaturliste*
Create(Text, LitList) -> Text
Create(File, Text) -> File

15. *Generierung einer XML-Datei zu einer Literaturliste*
Create(Xml, LitList) -> Xml
Create(File, Xml) -> File

16. *Generierung von Datensätzen aus BibTex-Dateien*
Create(BibTex, File) -> BibTex
Create(LitEntry, BibTex) -> LitEntry
//Create(LitList, BibTex[]) -> LitList
//Create(LitList, File[]) -> LitList

17. *Datensatz manuell anlegen*
Create(LitEntry) -> LitEntry

18. *Datensätze editieren*
Load(LitEntry) -> LitEntry
Save(LitEntry) -> void

19. *Generierung eines BibTex-Eintrags zu einem Datensatz*
Create(BibTex, LitEntry) -> BibTex
Create(File, BibTex) -> File

20. *Datensatz löschen*
Delete(LitEntry) -> void

21. *Datei zu einem Datensatz hinzufügen*
Create(File, LitEntry) -> LitEntry

22. *Dateiausgabe*
Load(File, Search) -> File[]
Load(File, LitEntry) -> File[]
Load(File) -> File

23. *Datei löschen*
Delete(File) -> void

24. *Kommentar schreiben*
Create(Comment, LitEntry) -> Comment

25. *Kommentare anzeigen*
Load(Comment, Search) -> Comment
Load(Comment, LitEntry) -> Comment[]
Load(Comment) -> Comment

26. *Kommentar ändern*
Save(Comment) -> void

27. *Kommentar löschen*
Delete(Comment) -> void

28. *Benutzer anlegen*
Create(User) -> User

29. *Rechte für Benutzer ändern*
Load(User, Search) -> User[]
Load(UserRights, User) -> UserRights
Save(UserRights) -> UserRights
Load(LitListRights, User) -> LitListRights[]
Load(LitListRights, User, LitList) -> LitListRights
Save(LitListRights, User, LitList) -> void
Delete(LitListRights, User) -> void
Delete(LitListRights, User, LitList) -> void

30. *Benutzer löschen*
Delete(User) -> void

31. *Passwort für Benutzer setzen*
Save(User) -> void

32. *Besitzer einer Literaturliste ändern*
Save(User, LitList) -> void

33. *System installieren*
Create(System) ?
Create(System, File) ?

34. *Image erstellen*
Create(File, System) -> File ?

35. *System wiederherstellen*
Save(File, System)

36. *Volltextsuche in Datensätzen*
Load(LitList, File, Search) -> LitList ?

37. *Extraktion eines Datensatzes aus einer Datei*
Create(LitEntry, File) -> LitEntry (?LitList)

Allgemeines zur Funktionsweise
Funktions-Aufruf
Im Allgemeinen erzeugt das Web-Interface eine CommandBean, welche sie an MU/AL sendet, um einen Befehl auszuführen. Die CommandBean enthält eine SessionBean mit dem Schlüssel der aktuellen Session (zur Identifikation) sowie eine der vier CmdBean's mit dem eigentlichen Befehl. Eine CmdBean hat ein, zwei oder drei DataBean's, welche die Parameter darstellen. Als Parameter wird eine DataBean mit entsprechend gesetzten Properties übergeben, wobei bei Befehlen mit mehr als einer DataBean oft eine leere DataBean an erster Stelle den Zieltyp der Operation angibt..

Anmeldung
Zur Anmeldung eines Users sendet das Webinterface eine CommandBean mit folgendem Inhalt:
-CmdCreateBean(SessionBean(empty), UserBean(userName, password))
-SessionBean(empty)
Rückgabewert ist bei erfolgreicher Anmeldung eine ResultBean mit folgendem Inhalt:
-Exception null
-SessionBean(sessionKey, userKey)

Rechte-Überprüfung
Für das Web-Interface kann es interessant sein, zu wissen, ob ein User in einem bestimmten Kontext eine Funktion ausführen darf. Hierzu gibt es die CmdTryBean:
-CmdTryBean(CmdDeleteBean(User(userKey)))
Mit „empty“ ist gemeint, dass eine DataBean erzeugt und nicht gefüllt wurde. Dazu könnte eine Funktion „isEmpty“ eingebaut werden.

Dateiupload
Das Web-Interface sollte die Rechte prüfen, bevor sie einem User gestattet, etwas hochzuladen. Hat man das getan und der User lädt eine Datei hoch, so speichert das Web-Interface diese in ein temporäres Verzeichnis und erzeugt eine FileBean(fileName, path). Diese wird als Parameter der jeweiligen Funktion an MU/AL weitergegeben. AL überlegt sich einen neuen Namen/Pfad für die Datei, kopiert sie dorthin und erzeugt eine neue FileBean mit den aktuellen Name/Pfad-Angaben, welche in der Datenbank gespeichert wird. Für einen Download kann die FileBean aus der Datenbank beibehalten werden, wenn WI auf den neuen Pfad Zugriff hat.

Zur einfacheren Handhabung der CmdBean's...
könnten verschiedene Konstruktoren angegeben werden, die alle Möglichkeiten mit JavaDoc abdecken, z.B.

```
/**
 * Login. For creating a new session using username+password.
 * @param sessionBean An empty SessionBean.
 * @param userBean A UserBean containing userName, password.
 */
public CmdCreateBean(SessionBean sessionBean, UserBean userBean) {...}
```

Exceptions
werden einfach in der ResultBean mitgegeben und können so vom Web-Interface als Property abgefragt werden.