

### **Allgemeines:**

Literaturlisten können sowohl in der Datenbank gespeichert werden als auch dynamisch/temporär erzeugt werden.  
-> LitList-Anfragen müssen über eine in Stateful Session Bean abgehandelt werden!  
-> Änderungen werden in temporärer LitList gehalten.  
-> eine LitList kann von mehreren Usern temporär gehalten werden, aber je User maximal einmal!  
Rechte wurden von Multiusern geprüft und liegen bei Anfragen an AL auf jeden Fall vor!

### **Methode 5: Literaturliste umbenennen**

MU an AL: `save(LitListBean[String litListKey], TmpLitListBean[String name]) //temporär!`

AL an DB: existiert schon eine Liste unter dem neuen Namen?

AL: falls neuer Name noch nicht vergeben: LitList wird temporär umbenannt

*Probleme:*

-newName bereits vergeben (Meldung von DB?)

-Liste wird gerade bearbeitet

*Lösung:*

-Exception falls Name existent, evtl. Alternativ-Vorschläge oder Anhängsel

-"Listener", der alle Bohnen bei Namensänderung updatet

### **Methode 6: Eintrag aus Literaturliste löschen**

MU an AL: `delete(LitListBean[String litListKey], LitEntryBean[String litEntryKey]) //temporär!`

AL: Datensatz aus temporärer Literaturliste entfernen

*Anmerkungen:*

Es wird nur die Referenz in der Literaturliste gelöscht, nicht der Datensatz selbst!

*Probleme:*

-Literaturliste wird gerade bearbeitet

*Lösung:*

-"Listener", der die Bohnen updatet

### **Methode 7: Eintrag in Literaturliste einfügen**

MU an AL: `create(LitListBean[String litListKey], LitEntryBean[String litEntryKey]) //temporär!`

AL an DB: prüfen ob Datensatz existent (?)

AL: Referenz auf Datensatz in temporärer Literaturliste einfügen

*Probleme:*

-Datensatz existiert nicht/nicht mehr

-Literaturliste wird gerade bearbeitet

*Lösung:*

-Exception oder Alternativvorschläge existenter Datensätze

-"Listener"

### **Methode 8: Schließen einer Literaturliste**

MU an AL: `delete(TmpLitListBean[String litListKey]) //speichere/verwerfe temporäre Liste`

AL: die "Bohne" wird aus dem Speicher entfernt

//AL an JDBC/DB: Literaturliste zur Bearbeitung freigeben

*Probleme:*

-Es wurden Änderungen an Literaturliste vorgenommen

-Datenbank überlastet

*Lösung:*

-Exception ODER Änderungen verwerfen

-Exception UND (Liste offen halten ODER Bohne trotzdem verwerfen)

### **Methode 9: Speichern einer Literaturliste**

MU an AL: save(LitListBean[String litListKey]) //speichere temporäre Liste in DB  
//oder als neue Liste: save(LitListBean[String litListKey], LitListBean[String name])  
AL an DB: neue Werte speichern

*Probleme:*

- Änderungen in der Datenbank seit letztem Öffnen der Literaturliste
- neuer Name schon existent
- Rechte der bestehenden Literaturliste wurden geändert, User hat kein Recht mehr

*Lösung:*

- Änderungen jeweils nur einer Person gleichzeitig erlauben
- Anhängsel an neuen Namen und trotzdem speichern UND Exception
- evtl. unter neuem Namen speichern

### **Methode 10: Rechte für Literaturliste ändern**

MU an AL: save(LitListBean[String litListKey], LitListRightsBean[...]) //nicht temporär  
//vorher: getLitListRights ?  
AL an DB: Änderungen speichern //temporäre Speicherung wäre ein Mehraufwand!

*Probleme:*

- Liste wird gerade bearbeitet
- Owner versucht sich selbst andere Rechte zu geben

*Lösung:*

- "Listener", der alle Bohnen bei Namensänderung updatet
- Owner in Zusatzbenutzern ignorieren

### **Methode 11: Öffnen einer Literaturliste**

MU an AL: load(LitListBean[String litListKey]) //laden aus Datenbank  
AL an DB: gefüllte LitListBean anfordern  
AL an MU: gefüllte LitListBean

*Probleme:*

- Literaturliste existiert nicht/nicht mehr

*Lösung:*

- Exception

### **Methode 12: Literaturliste löschen**

delete(LitListBean[String litListKey]) //nicht temporär

*Probleme:*

- Literaturliste ist gerade geöffnet oder wird bearbeitet

*Lösung:*

- Exception/Warnung ODER ("Listener" UND Bohne trotzdem löschen) ODER ignorieren weil geöffnete Versionen temporär sind (Problem nur: alte ID könnte neu vergeben werden und beim Speichern würde dann eine völlig andere Literaturliste überschrieben)