Table of Contents

Für gewöhnlich werden die Daten für das Client UI bereits serverseitig eingeschränkt, damit der Client nur die Daten erhält die er auch benötigt. Eine gute Möglichkeit die Einschränkung durchzuführen ist über Views in der jeweiligen Datenbank. Der Nachteil dabei ist stets die Abhängigkeit zur Datenbank und das nicht alle Datenbanken diese Möglichkeit bieten.

Um die Daten Datenbankunabhängig einzuschränken existiert die sogenannte Restrict Condition. Dabei handelt es sich um eine oder mehrere Bedingungen die zur Einschränkung der Daten führen bevor diese zum Client übertragen werden.

Anwendungsbeispiel

Unsere Applikation stellt nur Daten des angemeldeten Benutzers dar und bekommt die Daten der übrigen Benutzer erst gar nicht übermittelt.

Wir lösen diese Anforderung wie folgt:

```
/**
 * Returns the user DBStorage.
 *
 * @return the user DBStorage.
 * @throws Exception if the user DBStorage couldn't be initialized.
 */
public DBStorage getUserData() throws Exception
{
    DBStorage userData = (DBStorage)get("userDefaults");

    if (userDefaults == null)
    {
        userData = new DBStorage();
        userData.setDBAccess(getDBAccess());
        userData.setWritebackTable("USERS");
        userData.setRestrictCondition(new Equals("ID", getUserId()));
        userData.open();

        put("userDefaults", userDefaults);
    }

    return userDefaults;
}

/**
 * Gets the ID of the current logged on user.
 *
 * @return the ID of the current logged on user.
 */
public Object getUserId()
{
    return get("userId");
}

/**
```

```
 * Creates a new database access object.
 *
 * @return the new database access object
 * @throws Exception if the connection can not be opened
 */
public DBAccess getDBAccess() throws Exception
{
   //configure DBAccess
   DBAccess dba = (DBAccess)get("dBAccess");

   if (dba == null)
   {
       dba = DBAccess.get...

       DBStorage dbsUser = new DBStorage();
       dbsUser.setDBAccess(dba);
       dbsUser.setFromClause("USERS");
       dbsUser.open();

       ISession session = SessionContext.getCurrentSession();

       IBean bnUser = dbsUser.fetchBean(new Equals("USERNAME",
session.getUserName()));

       Object userId =
bnUser.get(DBObjects.getColumnName(session.getConfig(),
                                                    "USERS", "ID"));

       put("userId", userId);
       put("dBAccess", dba);
   }

   return dba;
}
```