Table of Contents

With JVx, the business logic can be developed in either the database or the middle tier. For the development of the business logic entirely in the middle tier, so-called server-side triggers are required. These triggers work just like the DataBook events at the client.

For example, a server-side trigger is called when a new record was inserted, modified, or deleted.

Example

We use a RemoteDataBook at the client for the creation of user accounts. A user is defined by username, password, first name, and last name. The password, however, has to be encrypted before it is stored in the database.

We define the storage:

```
public DBStorage getUsers() throws Exception
{
   DBStorage users = (DBStorage)get("users");
   if (users == null)
   {
      users = new DBStorage();
      users.setDBAccess(getDBAccess());
      users.setWritebackTable("USERS");
      users.open();

      users.eventBeforeInsert().addListener(this, "doEncryptPwd");
      users.eventBeforeUpdate().addListener(this, "doEncryptPwd");

      put("users", users);
   }
   return users;
}
```

and the trigger to insert and update:

```
public void doEncryptPwd(StorageEvent pEvent) throws Exception
{
   IBean bn = pEvent.getNew();

   String sNew = (String)bn.get("PASSWORD");
   String sOld;

   IBean bnOld = pEvent.getOld();

   if (bnOld != null)
   {
      sOld = (String)bnOld.get("PASSWORD");
   }
   else
   {
      sOld = null;
   }
```

```
   if (!CommonUtil.equals(sOld, sNew))
   {
      bn.put("PASSWORD", AbstractSecurityManager.getEncryptedPassword(
                         SessionContext.getCurrentSessionConfig(), sNew));
   }
}
```

In our example, we use IBean to access the properties. The event also allows the use of POJOs, as shown in the following example:

```
public void doEncryptPwd(StorageEvent pEvent)
{
    ...
    ...
    User user = pEvent.getNew(User.class);

    user.setPassword(AbstractSecurityManager.getEncryptedPassword(
                     SessionContext.getCurrentSessionConfig(),
                     user.getPassword()));
}
```

Note

Any POJO can be used. The implemented mechanism attempts to align the properties via the database´s column identifiers. In our previous example, we could also use an Address POJO, as long as it contains the relevant properties. Only feasible properties are aligned.