**Schnittstellen**

Um die Daten abfragen zu können wird eine REST Schnittstelle definiert. Das Datenaustauschformat JSON oder XML kann wie von einer REST Schnittstelle erwartet in der Abfrage eingestellt werden. Derzeit ist noch kein Framework für die REST API definiert.

In der Schnittstelle ist die Zentralle Klasse DataAccess (Listing 1).

*/\*\*  
 \* The central class to query the Database  
 \*/***public class** DataAccess {  
  
 **private final** ILdapPermissionManager **\_permissionManager**;  
 **private final** IDatabaseAccess **\_databaseAccess**;  
  
  
 **public** DataAccess(ILdapPermissionManager permissionManager,

DatabaseAccess databaseAccess){  
 **\_permissionManager** = permissionManager;  
 **\_databaseAccess** = databaseAccess;  
 }  
  
 */\*\*  
 \* Query the Database  
 \** ***@param query*** *the query to execute  
 \** ***@return*** *a QueryResult with either Data or a ErrorMessage.  
 \*/* **public** QueryResult QueryDataSource(QueryBase query){  
 **if**(!**\_permissionManager**.IsAllowedToAccess(  
 query.getUserContext(),  
 query.getDataSourceContext())){  
 **return** createNotAllowedResult();  
 }  
 **return \_databaseAccess**.QueryDatabase(query);  
 }  
  
 **private** QueryResult createNotAllowedResult(){  
 **return new** QueryResult(**false**,  
 **"Not allowed to access datasource"**,  
 QueryStatusCode.***Error***) {  
 @Override  
 **public** Object GetData() {  
 **return null**;  
 }  
 };  
 }  
}

Listing : Klasse für den Datenzugriff

Kommt eine Anfrage aus der REST Schnittstelle wird zuerst über den ILdapPermissionManager überprüft ob der User anhand seiner Rolle berechtigt ist auf die gewünschten Daten zuzugreifen. Ist das erlaubt wird die IDatabaseAccess aufgerufen, welche dann die Abfrage an die Datenbank stellt und ein QueryResult zurückliefert. Der Vorteil dieser Implementierung ist, dass der Datenbankzugriff völlig unabhängig vom Berechtigungssystem ist, und beliebig ausgetauscht werden kann.

Um eine neue Abfrage hinzuzufügen muss lediglich die Methode getQuery() der Klasse QueryBase implementiert werden (Listing 2).

Listing : Basisklasse um Abfragen zu definieren

*/\*\*  
 \* A Baseclass to query the database  
 \*/***public abstract class** QueryBase {  
 **private** IUserContext **\_userContext**;  
 **private** IDataSourceContext **\_dataSourceContext**;  
  
  
 **public** QueryBase(IUserContext userContext,

IDataSourceContext dataSourceContext){  
 **\_userContext** = userContext;  
 **\_dataSourceContext** = dataSourceContext;  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *the user context of the query  
 \*/* **public** IUserContext getUserContext(){  
 **return \_userContext**;  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *the dataSourceContext of the query  
 \*/* **public** IDataSourceContext getDataSourceContext(){  
 **return \_dataSourceContext**;  
 }  
  
 */\*\*  
 \* Implement this method for the query.  
 \* It will get  
 \** ***@return*** *\*/* **public abstract** String getQuery();  
}

Die Schnittstellen und Klassen sind im Moment ein Gerüst und können sich im Laufe der weiteren Implementierung noch ändern. Auf Darstellung der REST Schnittstelle wurde derzeit noch verzichtet, da noch nicht klar ist, wie genau die Authentifizierung darüber funktioniert und in welcher Form die Abfragen darüber ausgewählt werden können.