

How to develop a new CRUD Object.

Table of Contents

How to develop a new CRUD Object.....	1
Aim.....	1
Global code package structure.....	1
Entity.....	2
Service.....	3
DAO.....	5

Aim

Define the standard in order to develop the CRUD command from database table to Java object.

Global code package structure

Entity		org.cerberus. crud.entity XXX.java
Service	Interface	org.cerberus. crud.service XXX.java
Service	Implementation	org.cerberus. crud.service.impl IXXX.java
DAO	Interface	org.cerberus. crud.dao XXXDAO.java
DAO	Implementation	org.cerberus. crud.dao.impl IXXXDAO.java
Factory	Interface	org.cerberus. crud.factory IFactoryXXX.java
Factory	Implementation	org.cerberus. crud.factory.impl FactoryXXX.java

Entity

Entity should match exactly the database structure.

All database entity should be in

`org.cerberus.crud.entity`

Any other need of entity should not be inside this package.

Service

```
public AnswerItem readByKey(String id);
// Get the XXX object by the Key. NB : the key could include more than 1 field.

public AnswerItem readByKeyTech(String id);
// Get the XXX object by the Technical Key. NB : Key tech should be only 1 field.

public AnswerList readAll();
// Get all the object XXX from database.

public AnswerList readByCriteria(int start, int amount, String colName, String dir, String searchTerm, String individualSearch);
// Get the list of objects XXX by pagination criteria.

public AnswerList readByYYY1(String yYY1);
// Get all the object XXX from database filtering by the specific field YYY1.

public AnswerList readByYYY1ByCriteria(String yYY1, int start, int amount, String colName, String dir, String searchTerm, String individualSearch);
// Get all the object XXX from database filtering by a specific field YYY1 and by pagination criteria.

public AnswerList readByYYY2(int yYY2);
// Get all the object XXX from database filtering by a specific field YYY2.

public AnswerList readByVarious1(int yYY3, String yYY4);
// Get all the object XXX from database filtering by a set of specific field (YYY3 and YYY4 in this case).

public AnswerList readByVarious2(int yYY3, String yYY7, int yYY8);
// Get all the object XXX from database filtering by a set of specific field (YYY3, YYY7 and YYY8 in this case).

public AnswerList readByObject(XXX xxx);
// Get all the object XXX from database filtering by the specific object XXX.
// If string null no filter.

boolean exist(String id);
// true if the object exist false if does not exist. null if error.

public Answer create(XXX xxx);
// Insert a XXX object in the database

public Answer delete(XXX xxx);
// Delete a XXX object from the database

public Answer update(String keyVal, XXX xxx);
// Update the XXX object in the database
// keyVal could be more than 1 field.
```

```

public Answer save(XXX xxx);

// Save the object in the database. Ie Insert if not exist or Update if already exist. This serice is to be used
if we don't know if the object exist in database at that stage.

public Xxx convert(AnswerItem answerItem) throws CerberusException;

// Convert AnswerItem to the object. Throw exception in case of issue reported in AnswerItem.

public List<Xxx> convert(AnswerList answerList) throws CerberusException;

// Convert AnswerList to the List of object. Throw exception in case of issue reported in AnswerList.

public void convert(Answer answer) throws CerberusException;

// Throw exception in case of issue reported in Answer.

```

Notes

- No more use of Cerberusexception (appart from convert methods).
- We get the final object and trapp potencial error with (or use the convert methods) :

```

AnswerList resp = xxxService.readXxxListByYYYByCriteria(YYY, startPosition,
length, columnName, sort, searchParameter, "");

JSONArray jsonArray = new JSONArray();

boolean userHasPermissions = request.isUserInRole("IntegratorRO");

if (resp.isCodeEquals(MessageEventEnum.DATA_OPERATION_OK.getCode())) { //the
service was able to perform the query, then we should get all values

    for (XXX xxx : (List<XXX>) resp.getDataList()) {

        jsonArray.put(convertApplicationToJSONObject(xxx)); // Perform your
operation here on xxx

    }

}

```

DAO

```
public AnswerItem readByKey(String id);
// Get the XXX object by the Key. NB : the key could include more than 1 field.

public AnswerItem readByKeyTech(String id);
// Get the XXX object by the Key. NB : the key could include more than 1 field.

public AnswerList readAll();
// Get all the object XXX from database.

public AnswerList readByCriteria(int start, int amount, String colName, String dir, String searchTerm, String
individualSearch);
// Get the list of objects XXX by pagination criteria.
// Optional as readXXXByCriteria service could also reuse readXXXByYYY1ByCriteria with null YYY1.

public AnswerList readByYYY1(String yYY1);
// Get all the object XXX from database filtering by the specific field YYY1.

public AnswerList readByYYY1ByCriteria(String yYY1, int start, int amount, String colName, String dir, String
searchTerm, String individualSearch);
// Get all the object XXX from database filtering by a specific field YYY1 and by pagination criteria.

public AnswerList readByYYY2(int yYY2);
// Get all the object XXX from database filtering by a specific field YYY2.

public AnswerList readByVarious1(int yYY3, String yYY3);
// Get all the object XXX from database filtering by a set of specific field (YYY3 and YYY4 in this case).

public AnswerList readByVarious2(int yYY3, String yYY3, int yYY5);
// Get all the object XXX from database filtering by a set of specific field (YYY3, YYY7 and YYY8 in this case).

public AnswerList readByObject(XXX xxx);
// Get all the object XXX from database filtering by the specific object XXX.
// If string null no filter.

public Answer create(XXX xxx);
// Insert a XXX object in the database

public Answer delete(XXX xxx);
// Delete a XXX object from the database

public Answer update(String keyVal, XXX xxx);
// Update the XXX object in the database
// keyVal could be more than 1 field.

public XXX loadFromResultSet(ResultSet rs) throws SQLException;
// Create the object XXX from the the ResultSet rs. Ie maps the database field to Java Object XXX.
```

