PowerEnJoy

CODE INSPECTION

Flavio Primo, Hootan Haji Manoochehri
POLITECNICO DI MILANO | SOFTWARE ENGINEERING 2

Index

1 Descriptions of Classes

OFBiz "entity operators", implemented in the assigned EntityOperators class, are equivalent to SQL comparison operators. They allow for optimum flexibility in building entity condition statements. Entity operators are listed in the following table:

operator	operation EntityOperator
and	EntityOperator.AND
between	EntityOperator.BETWEEN
equals	EntityOperator.EQUALS
greater than	EntityOperator.GREATER_THAN
greater than equal to	EntityOperator.GREATER_THAN_EQUAL_TO
in	EntityOperator.IN
not in	EntityOperator.NOT_IN
less than	EntityOperator.LESS_THAN
less than equal to	EntityOperator.GREATER_THAN_EQUAL_TO
like	EntityOperator.LIKE
not like	EntityOperator.NOT_LIKE
not	EntityOperator.NOT
not equal	EntityOperator.NOT_EQUAL
or	EntityOperator.OR

EntityOperator class is part of the OFBiz Entity Engine contained in org.apache.ofbiz.entity package.

2 Functional Roles of the Assigned Class

Open for Business (OFBiz) is a suite of enterprise applications built on a common architecture using common data, logic and process components. The loosely coupled nature of the applications makes these components easy to understand, extend and customize.

The Ofbiz Entity Engine is a database agnostic application development and deployment framework seamlessly integrated into the OFBiz project code. It handles all the day-to-day data management tasks necessary to securely and reliably operate an enterprise. These tasks include, but are not limited to support for:

- Simultaneously connecting to an unlimited number of databases
- Managing an unlimited number of database connection pools
- Overseeing database transactions
- Handling database error conditions

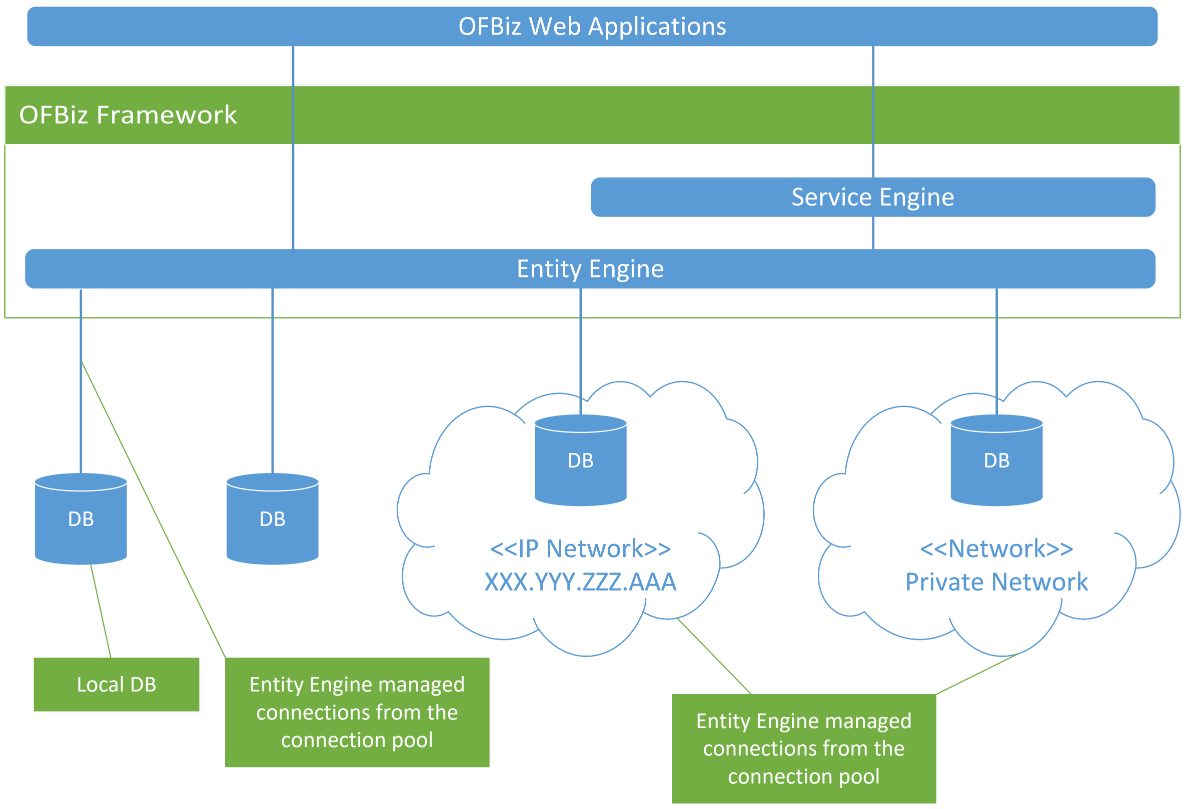


Figure 1: OFBiz Entity Engine

EntityOperator is contained in org.apache.ofbiz.entity.condition package. This package provides functionalities to constrain queries on entities. Various kind of constraints can be represented:

- EntityConditionList: list of entity conditions combined with the specified operator
- EntityExpr: simple expressions that combine entity conditions
- EntityFieldMap: map of fields where the field (key) equals the value, combined with the operator specified

Functionalities of the OFBiz project and the assigned cluster functionalities where studied through [1], [2] and [3] listed in the References section.

3 List of Issues

3.1 Naming Conventions

- “registerCase” method would be clearer if named “registerLowerAndUpperCase” (line 62)

3.2 Indentation

ok

3.3 Braces

- If statements not surrounded with curly braces (line 79-80, 86-87)

3.4 File Organization

Line length:

- Lines which exceed 80 characters but doesn’t exceed 120 characters:

  Codes: 60, 62, 67, 77, 80, 87, 105, 119, 176, 179, 181, 252, 275, 291, 292, 294.

  Comments: 101, 104, 285.

- Lines which exceeds 120 characters:

  Codes:

  95, 97, 99, 111, 113, 115, 128, 130, 136, 138, 144, 146, 152, 154, 160, 162, 164, 167, 169, 171, 174, 183, 185, 188, 190, 193, 195, 197, 248, 268, 293, 297, 318, 329.

  Comments: 104.

- The rest of the lines have less than 80 characters.

3.5 Wrapping Lines

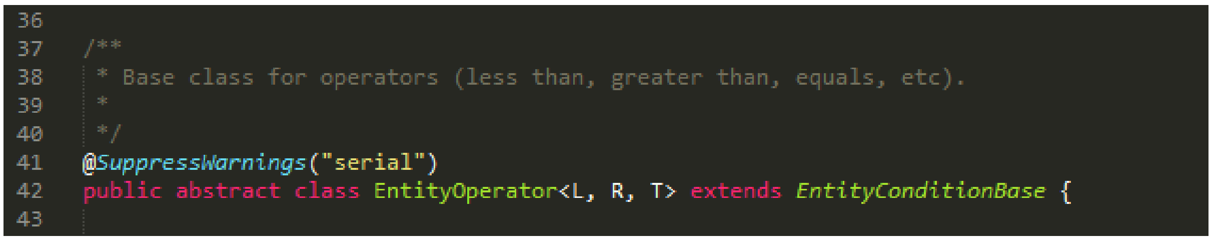
ok

3.6 Comments

- Code lines are commented without specifying the reason and the date which they should be deleted from the file at lines: **101**, **104**, **284-288**.

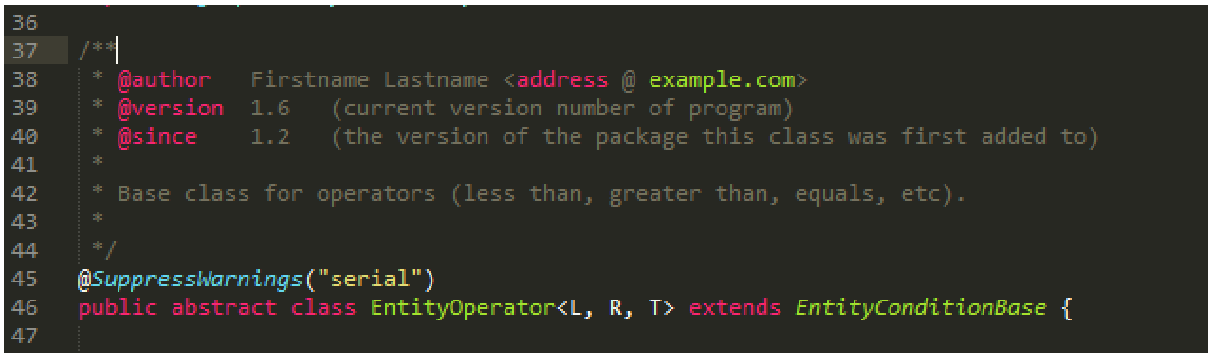
- Class comments should better describe the functionality of the class.

  Lines: **37-40**, **313-317**, **324-328**.
  As an instance:



```
36
37    /**
38     * Base class for operators (less than, greater than, equals, etc).
39     *
40     */
41    @SuppressWarnings("serial")
42    public abstract class EntityOperator<L, R, T> extends EntityConditionBase {
43
```

Would be better if described as:



```
36
37    /**
38     * @author   Firstname Lastname <address @ example.com>
39     * @version  1.6   (current version number of program)
40     * @since    1.2   (the version of the package this class was first added to)
41     *
42     * Base class for operators (less than, greater than, equals, etc).
43     *
44     */
45    @SuppressWarnings("serial")
46    public abstract class EntityOperator<L, R, T> extends EntityConditionBase {
47
```

- Methods and functions does not have the Javadoc standard.
  (Javadoc missing or doesn't follow the standards).

  List of methods without Javadoc:
    - Line 62: method void **registerCase** (String, EntityOperator)
    - Line 67: method void **register** (String, EntityOperator)
    - Line 73: method EntityOperator **lookUp** (String)
    - Line 77: method EntityComparisonOperator **lookupComparison** (String)
    - Line 84: method EntityJoinOperator **lookupJoin** (String)
    - Line 91: method int **requestId** ()
    - Line 204: constructor **EntityOperator** (int, String)
    - Line 209: method String getCode ()
    - Line 217: method int **getId** ()
    - Line 244: method boolean **entityMatches** (GenericEntity, L, R)
    - Line 248: method void **appendRHSList** (List, StringBuilder, ModelField, R)
    - Line 268: method void **appendRHSBetweenList** (List, StringBuilder, ModelField, X)
    - Line 290: method boolean **isEmpty** (L, R)
    - Line 291: method boolean **mapMatches** (Delegator, Map, L, R)
    - Line 292: method void **validateSql** (ModelEntity, L, R)
    - Line 293: method void **addSqlValue** (StringBuilder, ModelEntity, List, L, R
      DataSource)
    - Line 297: method void **addSqlValue** (StringBuilder, ModelEntity, List, boolean, L, R,
      DataSource)
    - Line 298: method EntityCondition **freeze** (L, R)
- Some methods override the super class behavior but neither child nor base class have Javadoc like:
    - Line 97: method boolean **compare** (Comparable, Object)
    - Line 99: method void **makeRHSWhereString** (ModelEntity, List, StringBuilder,
      ModelField, Object, Datasource)

  These methods repeated several times within the assigned class, we mentioned these to as demonstration. It's better to add Javadoc to the super class methods in order to keep the code DRY.

3.7 Java Source Files

- Multiple public class contained in "EntityOperator.java".
  "CollectionEntityComparisonOperator" and "ComparableEntityComparisonOperator" classes are present in addition to "EntityOperator" class.
- All methods miss a JavaDoc comments.
  Classes are commented but not in JavaDoc standard. Class's comments also fail to give a comprehensive description of the associated classes.

3.8 Package and Import Statements

Ok.

3.9 Class and Interface Declarations

- The actual class declarations order is the following:
    1. Class documentation comment (lines 37-40) (which suffers of the problem described in 3.7 section "Java Source Files")

2. Class statement (line 42)
3. Public static final attributes (lines 44-57)
4. Private static “dynamicId” and static final “registry” attributes (lines 59-60)
5. Private static “registerCase” (lines 62-65) method
6. Public static “register” (lines 67-71), “lookup” (lines 73-75), “lookupComparison” (lines 77-82), “lookupJoin” (lines 84-89), “requestId” (lines 91-93) methods
7. Public static final NOT_EQUAL (lines 111-122), LESS_THAN (lines 128-131), GREATHER_THAN (lines 136-139), LESS_THAN_EQUAL_TO (lines 144-147), GREATER_THAN_EQUAL_TO (lines 152-155), IN (lines 160-165), BETWEEN (lines 167-172), NOT (lines 174-177), AND (line 179), OR (line 181), LIKE (lines 183-186), NOT_LIKE (lines 188-191), NOT_IN (lines 193-198) attributes
8. Protected “idInt” (line 201), “codeString” (line 202) attributes
9. EntityOperator constructor
10. Public “getCode” (lines 209-215), “getId” (lines 217-219), “toString” (lines 222-224), “hashCode” (lines 227-229), “equals” (lines 233-242), “entityMatches” (lines 244-246) and protected “appendRHSList” (lines 248-266), “appendRHSBetweenList” (lines 268-281) methods
11. Public abstract isEmpty (line 290), mapMatches (line 291), validateSql (line 292) methods
12. Public addSqlValue (lines 293-295) method
13. Public abstract addSqlValue (line 297), freeze (line 298), visit (line 299) methods
14. Public static final “WILDCARD” (lines 301-311) attribute

The right class declarations order would be the following:

1. Class documentation comment (lines 37-40)
2. Class statement (line 42)
3. Public static final attributes (lines 44-57)
4. Public static final NOT_EQUAL (lines 111-122), LESS_THAN (lines 128-131), GREATHER_THAN (lines 136-139), LESS_THAN_EQUAL_TO (lines 144-147), GREATER_THAN_EQUAL_TO (lines 152-155), IN (lines 160-165), BETWEEN (lines 167-172), NOT (lines 174-177), AND (line 179), OR (line 181), LIKE (lines 183-186), NOT_LIKE (lines 188-191), NOT_IN (lines 193-198) attributes
5. Public static final “WILDCARD” (lines 301-311) attribute
6. Protected “idInt” (line 201), “codeString” (line 202) attributes
7. Private static “dynamicId” and static final “registry” attributes (lines 59-60)
8. EntityOperator constructor
9. Public static “register” (lines 67-71), “lookup” (lines 73-75), “lookupComparison” (lines 77-82), “lookupJoin” (lines 84-89), “requestId” (lines 91-93) methods
10. Public abstract isEmpty (line 290), mapMatches (line 291), validateSql (line 292) methods
11. Public abstract addSqlValue (line 297), freeze (line 298), visit (line 299) methods
12. Public “getCode” (lines 209-215), “getId” (lines 217-219), “toString” (lines 222-224), “hashCode” (lines 227-229), “equals” (lines 233-242), “entityMatches” (lines 244-246) and protected “appendRHSList” (lines 248-266), “appendRHSBetweenList” (lines 268-281) methods
13. Public addSqlValue (lines 293-295) method
14. Private static “registerCase” (lines 62-65) method

3.10 Initialization and Declarations
ok

3.11 Method Calls
ok

3.12 Arrays
ok

3.13 Object Comparison
- “rhs == GenericEntity.NULL_FIELD” (line 100, 116)
- “this == obj” checks for reference equality on an “Object” object type (line 234)

3.14 Output Format
Ok

Since there is no output string, we don’t have any issue.
We have some exception messages for debugging purposes, but no error displayed to the users.
line: 101,104.

3.15 Computation, Comparisons and Assignments
- Public static final int ID_* (lines 44-57) adhere to the “brutish programming” [4] “task 17: Define a finite set of named constants”. It should be substituted with Java enums [5]
- getCode instead of returning null if codeString is indeed null it returns a string with content “null”. If possible, a better and cleaner alternative would be to use the “Optional” type introduced in Java 8 that let handle variables that can be null or not.

3.16 Exceptions
Ok

No try-catch block is present and the exception catching is delegated to calling classes through “throw” exceptions.
We have 5 throw exception, and they have clear output.
Line: 80, 87, 176, 292, 303.

3.17 Flow of Control
ok

3.18 Files
Ok
No read and write from/to file.

4 Other Issues
- Class attributes “idInt” and “codeString” should be respectively called “id” and “code” since they are respectively of type “int” and “string”, is redundant (line 201-202)
- It would be preferable to move static blocks and static variables initialization (if possible) inside the constructor since it would improve readability and maintainability

5 References
1. *Apache OFBiz Cookbook* by Ruth Hoffman

2. *OFBiz Documentation* - https://ofbiz.apache.org/documentation.html
3. *OFBiz EntityOperator class Java Docs* - https://ci.apache.org/projects/ofbiz/site/javadocs/org/ofbiz/entity/condition/EntityOperator.html
4. *Brutish programming* - http://users.csc.calpoly.edu/%7Ejdalbey/SWE/CodeSmells/bonehead.html
5. Java Enum - http://docs.oracle.com/javase/tutorial/java/javaOO/enum.html

6 Hours Spent

Table describing the time management for the team.

Team member	**Hours**
Flavio Primo	3
Hootan Haji Manoochehri	3
	6 total