Code Inspection Document

POLITECNICO
MILANO 1863

Figure 1: Logo Politecnico di Milano

PowerEnjoy

Figure 2: Logo PowerEnjoy

- Maria Chiara Zaccardi
- Nicola Sosio
- Riccardo Redaelli

5/02/2017

Contents

Chapter 1

Code Description

1.1 Introduction

The purpose of this document is to report on the quality status of selected code from the Apache OFBiz project, an open source product for the automation of enterprise processes that includes framework components and business applications for ERP (Enterprise Resource Planning), CRM (Customer Relationship Management) and other business-oriented functionalities. The inspection is based on using the checklist for Java code inspection, a document that deal with of all possible issues that could be found in a java code.

1.2 Assigned class

This is the path of the assigned class:

```
../apache-ofbiz-16.11.01/framework/webapp/src/main/java/
    org/apache/ofbiz/webapp/stats/ServerHitBin.java
```

1.3 Functional role

The class `ServerHitBin` belong to the framework OFBiz components. These components are responsible for database access, caching, rendering the screens, managing transactions, and many more low-level tasks used by the application components. The main task of `ServerHitBin` is described at the beginning of the class

> *" Counts server hits and tracks statistics for request, events and views. Handles total stats since the server started and binned stats according to settings in the serverstats.properties file. "*

Thus the purpose of `ServerHitBin` is to provide statistics functionalities through the following main methods

countRequest(String id, HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin)

countEvent(String id, HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin)

countView(String id, HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin)

countEntity(String id, HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin)

countService(String id, HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin)

Chapter 2

Code issues

2.1 Checklist issues

Following all the issues found in ServerHitBin.java class:

- **C1** The function "count" at L.74, L.78, L.82, L.86, L.90, L.94 are misleading: from a count function we expect an int instead of a void return.
- **C11** There are a number of if statements with only one statement to execute which aren't surrounded by curly braces: at L.96, at L.112, at L.114, at L.460, at L.462.
- **C12** Blank line at L.26, L.28, L.40, L.51, L.57, L.59, L.66, L.121, L.129, L.133, L.332, L.363 are not useful. At L.469, L.492, L.494, L.520 we think that a blank line is needed.
- **C13** There are a number of occurrences in which the line length exceeds the 80 character limit but doesn't exceed 120 character limit: at L.96, at L.109, at L.140, at L.144, at L.257, at L.263, at L.271, at L.338, at L.493, at L.519, at L.522.
- **C14** There are a number of occurrences in which the line length exceeds the 120 character limit: between L61 and L65, between L.68 and L.72, at L.74, at L.78, at L.82, at L.86, at L.90, at L.94, at L.137, at L.216, at L.466, at L.469, at L.479, at L.482, at L.489, at L.469, at L.500, at L.519.
- **C23** Javadoc is not complete. In particular, the following public methods are not documented:
  - *public static void countRequest(String id, HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin)* at L.74.
  - *public static void countEvent(String id, HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin)* at L.78.

  - *public static void countEntity(String id, HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin)* at L.86.
  - *public static void countService(String id, HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin)* at L.90.
  - *public Delegator getDelegator()* at L.377.
  - *public String getId()* at L.381.
  - *public int getType()* at L.385.
  - *public synchronized long getNumberHits()* at L.420.
  - *public synchronized long getMinTime()* at L.424.
  - *public synchronized long getMaxTime()* at L.428.
  - *public synchronized long getTotalRunningTime()* at L.432.
  - *public double getMinTimeSeconds()* at L.436.
  - *public double getMaxTimeSeconds()* at L.440.
  - *public synchronized double getAvgTime()* at L.444.
  - *public double getAvgTimeSeconds()* at L.448.

- **C25** The variable *typeIds* at L.58 is declared before the end of public static variable declaration.

- **C25** The variables *delegator* at L.325, *id* at L.326, *type* at L.327, *limitLenght* at L.328, *binLenght* at L.329, *startTime* at L.330, *endTime* at L.331, *numberHits* at L.333, *totalRunningTime* at L.334, *minTime* at L.335 and *maxTime* at L.331 should have been declared after public static variable declaration.

- **C25** The class declarations at L.338 and at L.357 should have been after the class variables declaration.

- **C26** Methods are not grouped by functionality rather than by accessibility. They should be reordered following for example accessibility listing at first all the public methods and then private methods.

- **C27** There are mainly two methods which are too long and should be splitted

  - *private static void countHit(String baseId, int type, HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin, boolean isOriginal)* at L.137
  - *private void saveHit(HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin)* at L.466

- **C33** There are several occurrences of variable declarations which are not at the beginning of blocks: at L.147, at L.149, at L.150, at L.208, at L.477, at L.486, at L.492, at L.502, at L.509, at L.517 and at L.520.

- **C43**
  - At L.234 "Could not save ServerHitBin:" a space is missing after colon.
  - "Visit delegatorName=" and ", ServerHitBin delegatorName=" is missing a space after the equal.
- **C44** Between L.52 and L.56 constants are defined as static final int.
- **C44** At L.96 *UtilProperties.propertyValueEquals(String resource, String name, String compareString)* is a boolean function that returns a boolean value, so it doesn't need to be compared to true in order to produce a boolean expression.
- **C46** At L.130 *cal.getTime().getTime() < (curTime - binLength)* the use of parentheses on the right-side of the relational operator is useless.
- **C47** At L.445 it is not checked if *numberHits* is equal to zero.
- **C47** At L.445 it is not checked if *getNumberHits*() is equal to zero.
- **C55** The switch statements at L.152, at L.177, at L.274 and at L.299 don't have a default branch.

2.2 Other issues

- There are several occurrences of switch at L.152, at L.177, at L.274 and at L.299 which could be handled in a different properly way.
- Between L.74 and L.90 several methods that invoke the same function *countHit(String id, int type, HttpServletRequest request, long startTime, long runningTime, GenericValue userLogin)* where the only difference is the input type

Chapter 3

Used tools

- Github: for version controller
- Lyx: to format this document

Chapter 4

Effort spent

For redacting and writing this document we spent 12 hours per person