# Politecnico di Milano
# A.A. 2016–2017
# Software Engineering 2: “PowerEnJoy”
# *Code Inspection Document*

Pietro Ferretti, Nicole Gervasoni, Danilo Labanca

February 5, 2017

# Contents

# 1 Assigned Class

apache-ofbiz-16.11.01/framework/service/src/main/java/org/apache/ofbiz/service/job/JobManager.java

# 2 Functional Role

```

/**
* Job manager. The job manager queues and manages jobs. Client code can queue a job to be run immediately
* by calling the runJob({@link #runJob(Job)}) method, or schedule a job to be run later by calling the
* {@link #schedule(String, String, String, Map, long, int, int, int, long, int)} method.
* Scheduled jobs are persisted in the JobSandbox entity.
* <p>A scheduled job's start time is an approximation - the actual start time will depend
* on the job manager/job poller configuration (poll interval) and the load on the server.
* Scheduled jobs might be rescheduled if the server is busy. Therefore, applications
* requiring a precise job start time should use a different mechanism to schedule the job.</p>
*/
```

# 3 List of Issues

## 3.1 Naming Conventions

### 1. Meaningful Names

mi sembra che siano tutti buoni

**2. One-character variables** Ok. There are no one-character variables.

**3. Class names** Ok. Every class name is in mixed case and properly capitalized.

**4. Interface names** Ok. No interfaces are declared. (se ce ne sono) every interface used by the code is in mixed case and properly capitalized.

**5. Method names** Ok. Every method name is a verb. Every method name is camelCase and properly capitalized.

**6. Class variables** Ok. Every class variable is in mixed case and properly capitalized.

**7. Constants** NO. module and istanceId are immutable, so they can be considered constant. They should be capitalized. registeredManagers is fine because it's mutable

## 3.2 Indentation

**8. Number of spaces** Ok. The code is consistently indented with 4 spaces.

**9. No tabs for indentation** Ok. No tabs are used to indent the code.

## 3.3 Braces

**10. Consistent bracing style** Ok. The code is consistently braced following the *"Kernighan and Ritchie"* style.

**11. One-line statements bracing** NO "if" riga 326, 351, 354

## 3.4 File Organization

**12. Blank lines as separation** Ok. Blank lines are present between each method, around imports and variable declarations. Most of the methods also begin with a Javadoc.

**13. Where practical, line length under 80 characters** NOPE righe 73, 74, 89, 126, 147, 150, 154, 156, 161, 182, 186-190, 195, 198, 201, A great number of lines exceed 80 characters

**14. Line length always under 120 characters** NEPPURE righe 74, 186, 198, 217, 221, 222, 261-264, 273, 311, 315, 317, 387, 409, 429, 453, 498, 543, 560, 561 le dichiarazioni dei metodi sono lunghissime e wrappate poco

## 3.5 Wrapping Lines

**15. Line breaks after commas and operators** NO riga 152, la virgola dovrebbe stare sopra

**16. Higher-level breaks are used** Ok. Non ci sono line break con operatori

**17. Statements are aligned to previous ones** Ok. Sì, per tutti

### 3.6 Comments

### 3.7 Java Source Files

**20. Single public class or interface** Ok. Job manager is the only public class declared in the file. There are no other classes.

**21. The public class is the first class in the file** Ok. Job manager is the only public class declared in the file. There are no other classes.

**22. External program interfaces are consistent with the Javadoc**

**23. The Javadoc is complete** NO.

- No javadoc for 'module'! line 71
- No javadoc for 'instanceId'! line 71
- No javadoc for reloadCrashedJob!! line 304
- Missing @return tag on getInstance, line 88
- Missing @return tag on getDelegator, line 119
- Missing @return tag on getDispatcher, line 124
- Missing @param tag for 'limit' on poll, line 174
- Missing @return tag on poll, line 174
- Missing @param tag for 'job' on runJob, line 363
- Missing @throws tag for 'JobManagerException' on runJob, line 363
- Missing @throws tag for 'JobManagerException' on schedule, line 386, 408, 428, 453, 469, 498, 543

assertIsRunning, getRunPools sono private quindi non hanno necessariamente bisogno di javadoc

### 3.8 Package and Import Statements

## 4 Other Problems

## 5 Effort Spent

- Pietro Ferretti: hours of work
- Nicole Gervasoni: hours of work
- Danilo Labanca: hours of work

# 6 Revisions

## 6.1 Changelog

- CID v1.0, published on February 5, 2017