**User Guide**

**jWebSocket**

**jWebSocket Watchdog Client**

**1.0**

**Control de versiones English please**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| *1/4/2012* | *1.0* | *Creating the Documment* | *Lester Alfonso Zaila Viejo.* |

**1. Characteristics of the solution**

The jWebsocket Watchdog Client is a desktop application that allows you to monitor the jWebsocket server to perform an action when it fails. It is a control mechanism that takes certain actions to be taken such as: notifying system administrators via e-mail or SMS. Letting them know the server status at the time it fails, contributing to efficiency and productivity of software firms.

This solution is different from other remote management tools because it uses the WebSocket protocol to make the test to the jWebSocket server.

**2. Main Features**

**Functionalities of jWebSocket Watchdog Client.**

1. Manage Test.
2. Manage Task.
3. Task run manually.
4. Automatic Task Run.
5. Notify

jWebSocket Watchdog Client is supposed to be used in environments where it is required to control applications running on the jWebSocket server. It is useful if you have multiple servers running jWebSocket thus avoid wasting time on that system administrators have to manually analyze the log of all servers.

**3. Problems to be solved**

The jWebSocket framework internally does not provide applications to be monitored and managed remotely separated, consequently resulting in insufficient control of the operation of the servers and its custom specific applications. Maintain proper operation of the servers and apps which use the jWebSocket framework is the top priority of network administrators. A malfunction could bring instability consequences services for customers and businesses unavailability and economic losses, conveying a lack of credibility to the companies using this framework. For many servers have the network administrator requires much effort and time to do manually review applications as they do not receive a notice from the current state. The network administrator's job becomes cumbersome and expensive in this situation because they do not know the current status of running applications.

With jWebSocket Watchdog Client the status of your server and its apps is constantly checked, once an instance fails, you will be notified via e-mail or SMS. Here I would like to read one more sentence about the benefits, e.g. info, warn, error, fatal messages, quick reaction and fix possible etc. What comes to your mind are the major benefits for the administrator? This is the solution part for the problem described above.

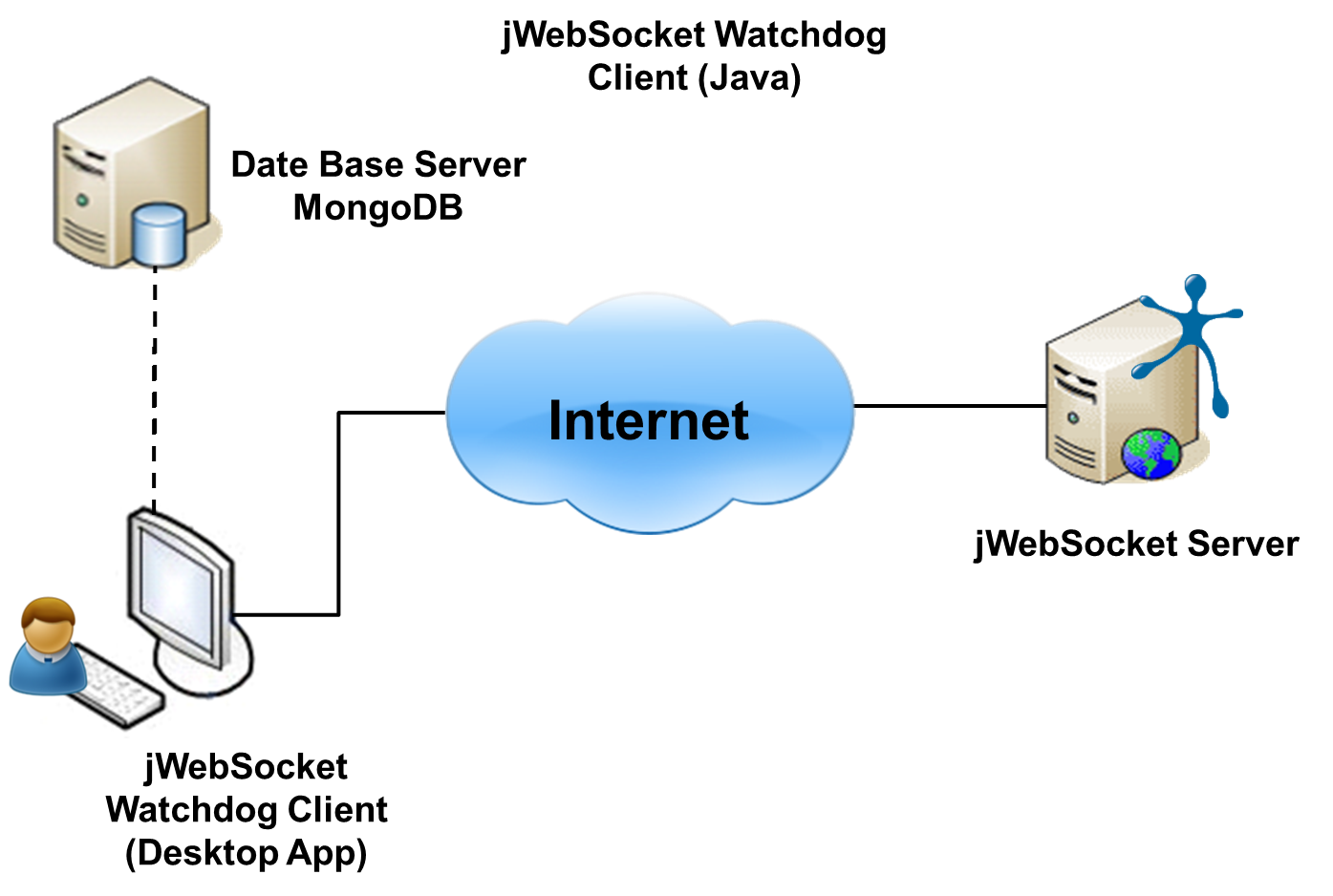
**4. Glossary**

**jWebSocket:** is a new technology aiming to develop WebSockets based applications that provide high levels of speed, scalability, security and real-time work, a key element for the web today and in the future. Add link in the footnote.

**Websocket:** WebSocket protocol defines the procedures to update the HTTP connection through a fully bidirectional connection by using TCP WebSocket. The client sends an HTTP GET request to establish communication with the server WebSocket. Subsequently, the communication remains active until it is closed, allowing exchange messages between the client and server. And link in the footnote.

**SMS:** *Short Message System (not Service?)*

**5. Solution Model**



The solution is also to check the availability the Apps and Plug-ins on the server, please add them in the diagram. To use jWebSocket Watchdog Client is required to have previously installed OpenJDK 1.7 or higher (Yes? Java 1.6 not sufficient anymore?). jWebSocket Watchdog Client connects to the database server to load or insert MongoDB tasks to execute the test to the jWebSocket server. The tasks can be run manually or automatically.

**7. Requirements for Use**

The application has the advantage of working on Linux, Windows and Mac OS, but to work you need:

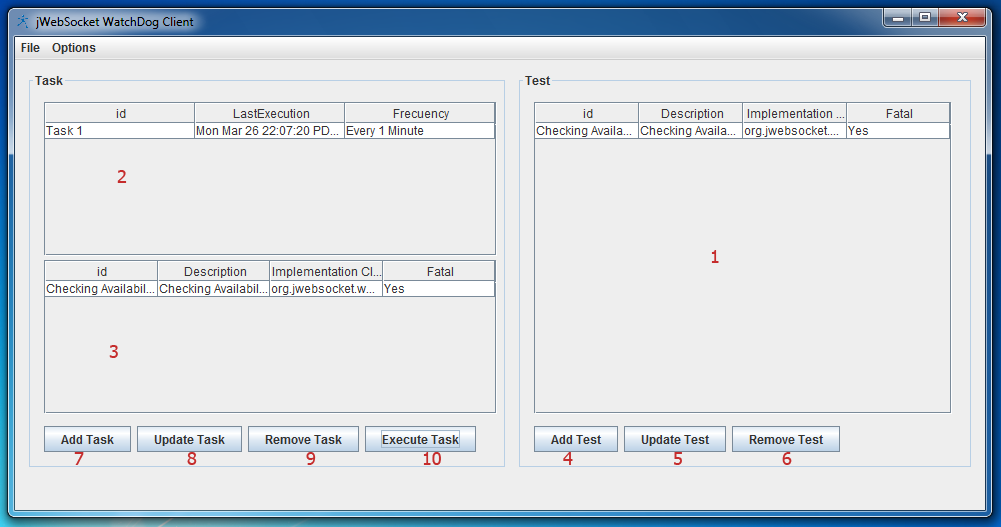
* OpenJDK 1.7 or higher installed, Java 1.6 or Java 1.7 possible? Tested?

**8. Roles of the solution**

**System User:** Interact with the application; you can perform all actions provided by the system. How and where is that specifie, what does the user need to consider here?

**9. System Operation**

How is the application started on the client machine?

**Main view of the application:** allows users to interact with the application.

1 -------- Area where the test are described.

2 -------- Area where the tasks and their test are described.

3 -------- Area where the test contained in the selected task with their descriptions.

4 -------- Add a test.

5 -------- Update a test.

6 -------- Remove a test.

7 -------- Add a task.

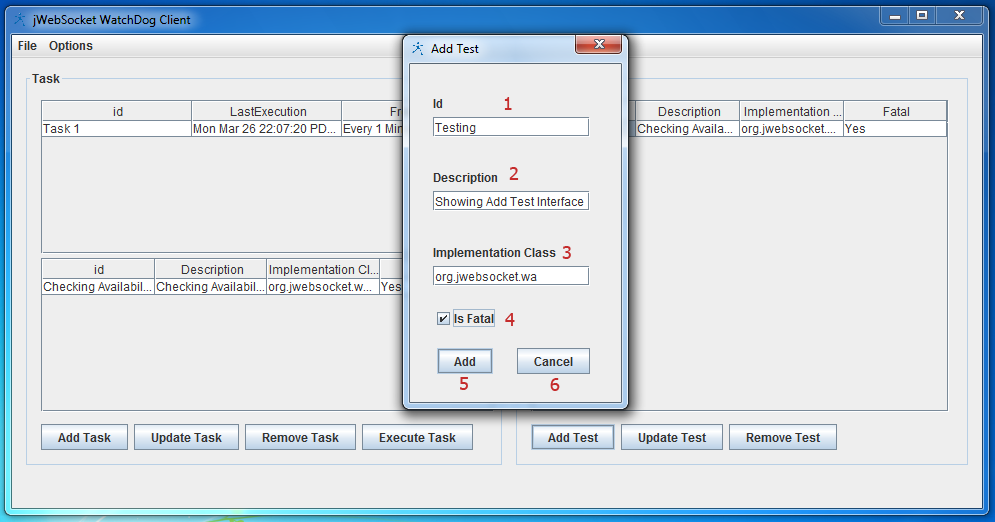
8 -------- Update a task.

9 -------- Remove Task.

10 ------ Executes a previously selected test.

**Add test View:**

Please explain the meaning of “test” and “task” here. I mean that a test is a single specific test, and that one task can run one or multiple tests.

****

Users can add a task. They will have to introduce the id (explain use, purpose, uniqueness?), the description of that test, which is that it is the test, especially if it is fatal or not the test to fail, to be used as the denominator when notifying the system administrator.

1 -------- Enter test ID

2 -------- Enter description of the test

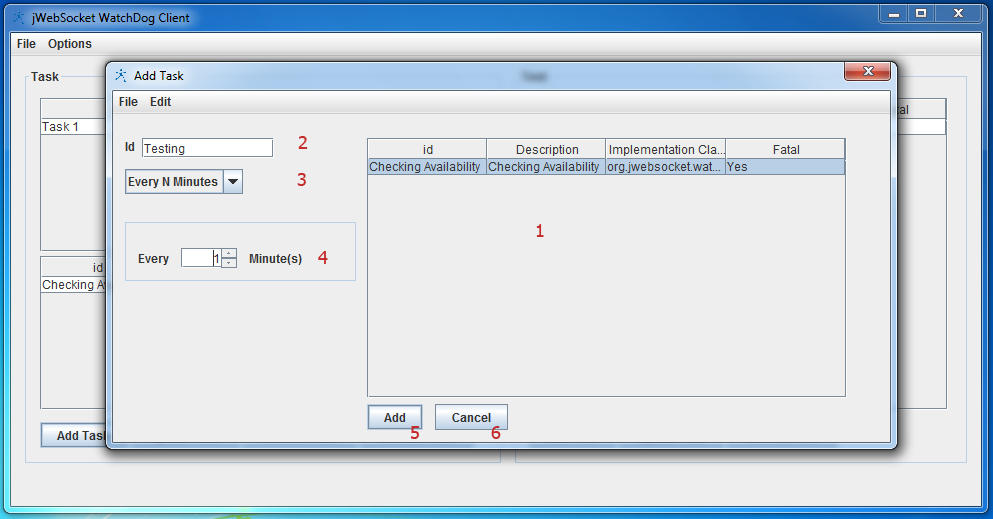
3 -------- Enter where the test is located. How does the user know the class name? More explanation here, please, or provide a list or similar?

4 -------- Indicate whether fatal or not the test to fail. What does this mean, effect for the user?

5 -------- Once all data is entered, we add the test.

6 -------- Cancel if you want to do it later.

**Add task view:**



1 -------- Task Area

2 --------Introduce Task ID (again explain, use, purpose, uniqueness etc.)

3 -------- Enter the frequency to execute the task.

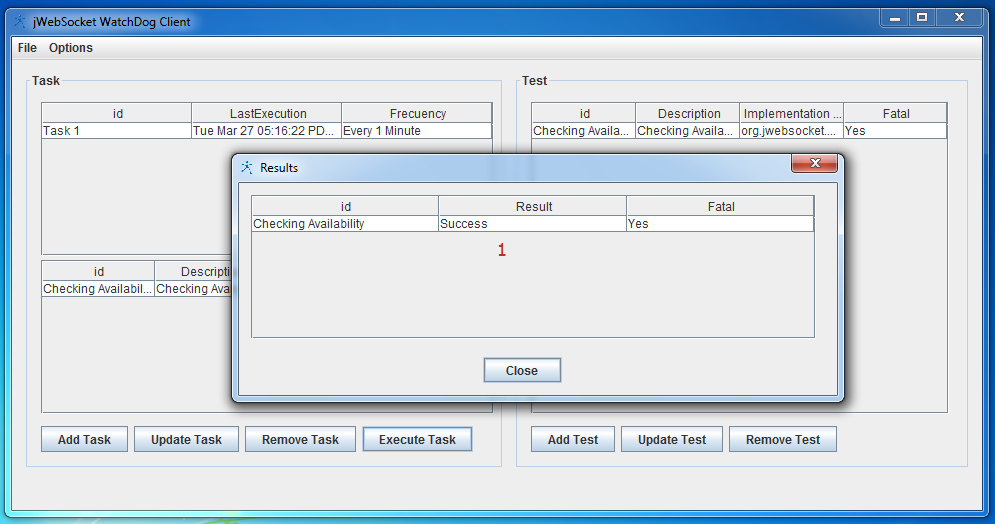
4 --------Enter each time given the frequency that will run the task.

5 --------  Add the task once all fields are entered.

6 -------- Cancel if you want to do it later. Well, saved or discarded then?

**Running Task View:**

How do I run it, you have an execute button, remember, the user does not know yet anything about your solution.

****

1 -------- Results area showing id of the test, the result of it and if it is fatal or not, the test to fail for it to be notified to the system administrator.

What does fatal mean here? Explain effects to the user.

**10. Rules of the solution**

I don’t understand this section, what does this mean to the user?

**R1:** If the user deletes a task, it cannot be recovered.

**R2:** If the user deletes a test, it cannot be recovered.

In the beginning of this document you talk about sending SMS and E-mails.

Only here I find no more words and examples about that.

When are which Messages sent? The user will for sure be interested in that!

Here I expect some more words.