

Shutdown Enhydra application

Table of Contents

1. Using shutdown() method to perform some extra actions when Enhydra application is stopped

List of Examples

1.1. CalcApp.java:	1
--------------------------	---

Chapter 1. Using shutdown() method to perform some extra actions when Enhydra application is stopped

if you need to perform some extra actions during shutdown Enhydra application, you must override public synchronized void shutdown() method from StandardApplication class.

This is a sample code from Calculator (`${enhydra_root}/examples/calculator/calculator.CalcApp.java` application:

Example 1.1. CalcApp.java:

```
package calculator;

import com.lutris.appserver.server.*;
import com.lutris.appserver.server.httpPresentation.*;
import com.lutris.appserver.server.session.*;
import com.lutris.util.*;

import java.util.Date;

/**
 * This is a sample Enhydra application. It simulates a simple calculator.
 * A scan of a real calculator is used for the front end, with an image map.<P>
 *
 * A global counter of the number of buttons pushed (across all users) is
 * kept here. <P>
 *
 * Noramlly an Enhydra application would make note of various settings
 * in the config file, but this application is so simple it needs no
 * configuration.
 */
public class CalcApp extends StandardApplication {

    /**
     * A hit counter for the number of buttons pressed by all users.
     */
    public long buttonsPushed = 0;

    /**
```

```
* Show the number of buttons pressed. This will only be called by the
* Enhydra Multiserver (it won't show up in other servers).
*
* @return HTML that is displayed in the status page of the Multiserver.
*/
public String toHtml() {
    return "This is a sample Enhydra Application. " + buttonsPushed +
        " buttons have been pushed so far.";
}

/**
 * Shutdown the application. Override method from Standardapplication.
 */
public synchronized void shutdown() {
    if (state == STOPPED)
        return; // Handle multiple threads doing a shutdown.
    super.shutdown(); // must be called to perform necessary shutdown operations

    //Perform additional shutdown operations
    System.out.println("Application CALCULATOR stopped at: " + (new Date()).toString());
}
}
```

Note that you must check state of application and call super.shutdown() to perform default shutdown actions!