

```
package edu.wsu.gridstat.qosBroker.common;
```

```
/**  
 * <p>Title: QoSBrokerBaseInterface </p>  
 * <p>Copyright: Copyright (c) 2002</p>  
 * <p>Company: Washington State University </p>  
 * @author Kjell Harald Gjermundrod  
 * @version 1.0  
 */
```

```
public abstract interface QoSBrokerBaseInterface
```

```
{  
    /**  
     * The <code>start</code> method is used bootstrap the (Leaf) QoS Broker.  
     * <BR>  
     * @return Returns <code>>true</code> if it is started <code>>false</code> otherwise.  
     */  
    public abstract boolean start();  
  
    /**  
     * The <code>stop</code> method is used stop the (Leaf) QoS Broker, will stop all the threads and kill the objects.  
     * <BR>  
     * @return Returns <code>>true</code> if it is stopped <code>>false</code> otherwise.  
     */  
    public abstract boolean stop();  
  
    /**  
     * The <code>isRunning</code> method is used to find out if this (Leaf) QoS Broker is  
     * running or not.  
     * <BR>  
     * @return Returns <code>>true</code> if it is running <code>>false</code> otherwise.  
     */  
    public abstract boolean isRunning();  
  
    /**  
     * The <code>setFlooding</code> method is used to set a variable to be flooded.  
     * <BR>  
     * @param pubName The name of the publisher of the variable.  
     * @param variableName The name of the variable.  
     * @param modes For which modes would this variable be flooded.  
     * @throws InterruptedException If during the execution of the command the thread dies.  
     * @return Returns <code>>true</code> if the variable is set to be flooded, <code>>false</code> otherwise.  
     */  
    public abstract boolean setFlooding(String pubName, String variableName, int modes) throws InterruptedException;  
  
    /**  
     * The <code>removeFlooding</code> method is used to remove a variable that is flooded.  
     * <BR>  
     * @param pubName The name of the publisher of the variable.  
     * @param variableName The name of the variable.  
     * @param modes For which modes is this variable flooded.  
     * @throws InterruptedException If during the execution of the command the thread dies.  
     * @return Returns <code>>true</code> if the variable is not flooded anymore, <code>>false</code> otherwise.  
     */  
    public abstract boolean removeFlooding(String pubName, String variableName, int modes) throws InterruptedException;  
  
    /**  
     * The <code>changeMode</code> method is used to change the mode.  
     * <BR>  
     * @param mode The new mode.  
     * @throws InterruptedException If during the execution of the command the thread dies.  
     * @return Returns <code>>true</code> if the mode is changed, <code>>false</code> otherwise.  
     */  
    public abstract boolean changeMode(int mode) throws InterruptedException;  
  
    /**  
     * The <code>getMode</code> method is used return the current mode.  
     * <BR>  
     * @throws InterruptedException If during the execution of the command the thread dies.  
     * @return Returns the current mode.  
     */  
    public abstract int getMode() throws InterruptedException;  
}
```