```java
package edu.wsu.gridstat.publisher.interfaces;

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

// GridStat packages.
import edu.wsu.gridstat.publisher.common.PublisherException;

public interface PublisherInterface
{
  /**
  * The <code>connectToEdge</code> method is used to connect to a edgeStatusRouter.
  * <BR>
  * @return Returns <code>0</code> if connected, error code otherwise.
  */
  public short connectToEdge();

  /**
  * The <code>disConnectFromEdge</code> method is used to disconnect from a edgeStatusRouter.
  * <BR>
  * @return Returns <code>0</code> we have unregisteded all the publications
  * and we have disconnected, error code otherwise.
  */
  public short disConnectFromEdge();

  /**
  * The <code>reConnectToEdge</code> method is used to connect to a edgeStatusRouter.
  * <BR>
  * @return Returns <code>0</code> if connected, error code otherwise.
  */
  public short reConnectToEdge();

  /**
  * The <code>isConnected</code> method is used return if we are connected to the edgeStatusRouter or not.
  * <BR>
  * @return Returns <code>true</code> if we are connected, <code>false</code> otherwise.
  */
  public boolean isConnected();

  /**
  * The <code>publish</code> method is used to publish a userdefined event.
  * <BR>
  * @param variableId The unique Id of this status variable
  * @param data The data to be published.
  * @return Returns <code>true</code> if message is published, <code>false</code> otherwise.
  */
  public boolean publish(int variableId, byte[] data);

  /**
  * The <code>publish</code> method is used to publish a userdefined event.
  * <BR>
  * @param variableId The unique Id of this status variable
  * @param data The data to be published.
  * @param timeStamp The timestamp that this event will get.
  * @return Returns <code>true</code> if message is published, <code>false</code> otherwise.
  */
  public boolean publish(int variableId, byte[] data, long timeStamp);

  /**
  * The <code>publish</code> method is used to publish a int event.
  * <BR>
  * @param variableId The unique Id of this status variable
  * @param value The value to be published.
  * @return Returns <code>true</code> if message is published, <code>false</code> otherwise.
  */
  public boolean publish(int variableId, int value);

  /**
  * The <code>publish</code> method is used to publish a int event.
  * <BR>
  * @param variableId The unique Id of this status variable
  * @param value The value to be published.
  * @param timeStamp The timestamp that this event will get.
  * @return Returns <code>true</code> if message is published, <code>false</code> otherwise.
  */
  public boolean publish(int variableId, int value, long timeStamp);

  /**
  * The <code>publish</code> method is used to publish a float event.
  * <BR>
  * @param variableId The unique Id of this status variable
  * @param value The value to be published.
  * @return Returns <code>true</code> if message is published, <code>false</code> otherwise.
  */
  public boolean publish(int variableId, float value);
```