Sign In or Up Ruby Python **JavaScript** Front-End Tools **▼** More Tips **Jobs** 

```
Coderwall
                                                  Embedded openMQ 5.0 (without installation)
                 #java
                         #broker
                                                                    #imq
    #example
                                   #embedded
                                                         #openmg
                                                                            #jms2
                                                                                     #openmq5
  It is possible to run openMQ 5.0 embedded without referencing a complete installation of openMQ.
  This can be usefull when creating tests. It does however initially require an installation, since some files
  need to be copied to your project.
  Do the following:
     1. Download and install openMQ 5.0
     2. In your project create in the 'src/test/resources' eg. an 'openmq' folder
     3. In the 'openmq' folder create a folder named 'lib/props/broker'
     4. Copy default.properties and install.properties from your openMQ installation here
     5. In the 'openmq' folder create a folder named 'var/instances/imqbroker/etc'
     6. Copy accesscontrol.properties and passwd from your openMQ installation here
  In your pom.xml add the following dependency:
  <dependency>
      <groupId>org.glassfish.main.extras
      <artifactId>glassfish-embedded-all</artifactId>
      <version>4.0
      <scope>test</scope>
  </dependency>
  Now you can create a (Java) class which creates an embedded broker. This looks like:
  package my.package;
  import com.sun.messaging.jmq.jmsclient.runtime.BrokerInstance;
  import com.sun.messaging.jmq.jmsclient.runtime.ClientRuntime;
  import com.sun.messaging.jmq.jmsserver.Globals;
  import com.sun.messaging.jmq.jmsservice.BrokerEvent;
  import com.sun.messaging.jmq.jmsservice.BrokerEventListener;
  import lombok.extern.slf4j.Slf4j;
  import java.util.Properties;
  @Slf4j
  public class EmbeddedBroker {
      private static final String IMQ_HOME = EmbeddedBroker.class.getClassLoader().getResource("op
      private static final String IMQ_VAR_HOME = IMQ_HOME.concat("/var");
      private static final String IMQ_LIB_HOME = IMQ_HOME.concat("/lib");
      private static final String IMQ_INSTANCE_HOME = IMQ_VAR_HOME.concat("/instances");
      private static final String IMQ_INSTANCE_NAME = "imqbroker";
      public static BrokerInstance start() throws ClassNotFoundException, IllegalAccessException,
          ClientRuntime clientRuntime = ClientRuntime.getRuntime();
          BrokerInstance brokerInstance = clientRuntime.createBrokerInstance();
          BrokerEventListener listener = new BrokerEventListener() {
              public void brokerEvent(BrokerEvent error) {
              public boolean exitRequested(BrokerEvent event, Throwable thr) {
                  return true;
          };
          Properties properties = brokerInstance.parseArgs(getArguments());
          updateConfigurationForBroker();
          brokerInstance.init(properties, listener);
          brokerInstance.start();
          return brokerInstance;
      }
      private static String[] getArguments() {
          return new String[]{
                    "-varhome", IMQ_VAR_HOME,
                    "-libhome", IMQ_LIB_HOME,
                    "-imqhome", IMQ_HOME
          };
      private static void updateConfigurationForBroker() {
          try {
              Globals.getConfig().updateProperty("imq.home", IMQ_HOME);
              Globals.getConfig().updateProperty("imq.libhome", IMQ_LIB_HOME);
              Globals.getConfig().updateProperty("imq.varhome", IMQ_VAR_HOME);
              Globals.getConfig().updateProperty("imq.instanceshome", IMQ_INSTANCE_HOME);
              Globals.getConfig().updateProperty("imq.instancename", IMQ_INSTANCE_NAME);
              Globals.getConfig().updateBooleanProperty("imq.persist.file.newTxnLog.enabled", fals
              Globals.getConfig().updateBooleanProperty("imq.cluster.enabled", false, true);
  //
              Globals.getConfig().updateProperty("imq.autocreate.destination.maxNumMsgs", "100");
          } catch (Exception e) {
              EmbeddedBroker.log.warn("Unable to set the configuration for the broker", e);
      }
  }
  By calling the EmbeddedBroker.start() method, the broker will be started and you can do
  whatever you want to do, like creating queues/topics and send/receive JMS messages.
    #example
                 #java
                        #broker
                                   #embedded
                                                                    #imq
                                                                                     #openmq5
                                                 #jms
                                                         #openmq
                                                                            #jms2
Written by Marcel van den Brink
 Recommend
                 Say Thanks
                                 Update Notifications Off
```

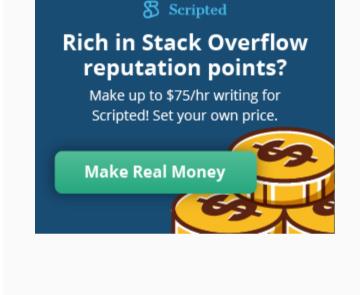
Have a fresh tip? Share with Coderwall community! Post a tip **Best #Example Authors** leviter ★ 1.71K #example #java #Open Source josefnpat ★ 1.289K #example #Lua #PHP codeship **†** 615 #example #Clojure #Python ashnur ★ 394 #example #JavaScript #CoffeeScript niklongstone \* 318 #example #git #Phyton **Related Tags** #example #java #broker #embedded #openmq #imq #jms2

#openmq5

Post a job for only \$299

Awesome Job

See All Jobs >



Recursively Expand Tabs to Spaces

● 1.289K ● 0

Behavior-Driven Integration and Unit Testing **●** 617 **●** 0

@coderwall ¥

I've started a repo about how really hard is to code asynchronously **③** 396 **9** 0

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.