Table of Contents

The integration of JVx in a Vert.x environment is very easy because JVx has a ready-to-use connector. Firstly, you have to start a Vert.x server (HttpServer or NetServer). We provide a simple class that starts a server and automatically uses JVx:

Simply execute (use ; instead of : on windows):

```
java -DConfiguration.basedir=/path/to/rad_parent -cp
jvx.jar:jvxvertx.jar:<your_application.jar>
com.sibvisions.vertx.NetSocketServer -cluster-host=<ip or hostname>
```

Set Configuration.basedir to the directory that contains your rad directory. This is important, otherwise JVx does not find your application(s). The rad directory structure is as usual.

In your client application, simply use NetSocketConnection if you are using NetSocketServer or JVx' standard HttpConnection if you're using HttpServer.

Example

```
NetSocketConnection con = new NetSocketConnection(<ip or hostname>);

MasterConnection macon = new MasterConnection(con);
macon.setApplicationName(<application>);
macon.setUserName(<username>);
macon.setPassword(<password>);
macon.open();
```

That's it.

Now your application works without application server like JBoss or Tomcat, but you're able to use all JVx features for your application.

Note

The source code of our vert.x connector is available on github.

From:
https://doc.sibvisions.com/ - **Documentation**

Permanent link:
https://doc.sibvisions.com/jvx/common/util/vertx

Last update: **2020/06/15 14:44**