Maiken Winteberg Socket Registry

By Martin Alexander Thomsen on 02 September 2024

The Maiken Winteberg Socket Registry is a tool for registrating socket and subscription information. Its used by the Maiken Winteberg File Domain Jumper. Before you install the Maiken Winterberg File Domain Jumper you need to install a Socket Registry. The registry has a Java API called **com.maikenwinterberg.socketregistry.api.ClientRegisty**. This class is the main accesspoint when communicating with the registry. Through this class a Java programmer can register informatin and lookup information.

Installation

Easy installation in 5 steps:

1) Install java on your maschine

linux: sudo apt-get install openjdk-21-jdk

windows/macos: https://www.oracle.com/java/technologies/downloads/

2) Update config/registryConfig/registry.properties

A) Replace **localhost** with your domainname of your externalip

B) **Optionally**, You can have your own database by changing the database settings of JDBC (by default an H2 database is being used):

driver=org.h2.Driver
url=jdbc:h2:file:./registrydb
username=sa
password=

If you add a new driver you need to update the script files with a classpath of your driver.
bin/startRegistry.bat og bin/startRegistry.sh

3) Install the registry service

Linux:

Execute the file:

1. linux_service/installRegistryService.sh

Mac:

Execute the file:

1. bin/startRegistry.sh

Windows:

Metode 1: https://www.coretechnologies.com/products/AlwaysUp/

1. Install windows_service/AlwaysUp.exe
2. Add the application bin/startRegistry.bat

Metode 2:

1. Create a link of bin/startRegistry.bat to the startup folder: https://www.howtogeek.com/754239/how-to-access-the-windows-10-startup-folder/

4) Open for port 4554

5) You can look in the log to make sure no errors has happend.

Code example for the Java programmer

```
/*
 * Martin Alexander Thomsen den 29 August 2024
 */
package com.maikenwinterberg.socketregistry.test;

import com.maikenwinterberg.socketregistry.api.ClientRegistry;
import com.maikenwinterberg.socketregistry.api.ClientRegistryAddress;
import com.maikenwinterberg.socketregistry.api.RegistryConnectionFactory;
import com.maikenwinterberg.socketregistry.api.SocketRegistration;
import com.maikenwinterberg.socketregistry.security.RegistrySecurity;
import java.net.Socket;

/**
 * @author Martin Alexander Thomsen
 * @author Ron Georg Martin Richard
 * @see documentnetwork.com
 * @see maikenwinterberg.com
 */
public class Test {

    public static void main(String arg[]) throws Exception {
        //init
        String securityImpl = "com.maikenwinterberg.socketregistry.security.StaticRegistrySecurity";
        String publicKeyAsBase64 = RegistrySecurity.toBase64(securityImpl, RegistrySecurity.getKeyPair(securityImpl, Test.class).getPublic());
        ClientRegistry clientRegistry = ClientRegistry.getRegistryInstance("localhost", "localhost", 4554, true, false);
        //register
        ClientRegistryAddress registryAddress = clientRegistry.registerSocket("localhost", "fileReciever", "1", "localhost", "4445", publicKeyAsBase64,
securityImpl);
        if (registryAddress != null) {
            System.out.println("completed");
        } else {
            System.out.println("error");
        }
        //lookup
        SocketRegistration reg = RegistryConnectionFactory.getValidSocketRegistration(clientRegistry, "localhost", "fileReceiver");
        Socket s = reg.getSocket();
        //your socket programme comes next.
    }
}
```