

Standalone Server Environment

For making CommunityMashup available it is recommended to set up a standalone server environment. This page describes how this can be done and how the server environment then can be maintained.

The basis of a standalone server environment is the Equinox OSGi implementation.

There are several alternatives for running the mashup and making the REST interface available via a Web interface:

- Hosting a Http server (e.g. Jetty) in Equinox
- Hosting of Equinox in a Servlet container (e.g. Jetty, Tomcat, Virgo)

In this How-To we describe how to host Jetty in Equinox.

More details about server side Equinox can be found at: <http://www.eclipse.org/equinox/server/>

Setup of a new server environment

Apache frontend, directory structure

In the following, we assume that the target system is running apache web server for serving standard web pages.

For every Mashup server environment, a new virtual host is created in Apache - in the directory `/var/www/domain-name/web/`

The OSGi installation will be placed in the directory `/var/www/domain-name/osgi/` - In the following sub-sections we usually refer to this directory as (OSGi-) home directory.

The minimal directory structure below the home directory is:

- `/osgi/`
 - `org.apache.felix.fileinstall-3.1.10.jar`
 - `org.eclipse.osgi_3.8.1.v20120830-144521.jar`
 - `configuration/`
 - `config.ini`
 - `load/` (Enthält alle Bundles die über Felix File Install installiert werden sollen)
 - `commons-logging_all-2.0.0.kf3.jar`
 - `org.eclipse.equinox.http.jetty_2.0.0.v20100503.jar`
 - `org.mortbay.jetty.server_6.1.23.v201004211559.jar`
 - `javax.servlet_2.5.0.v200910301333.jar`
 - `org.eclipse.equinox.http.servlet_1.1.0.v20100503.jar`
 - `org.mortbay.jetty.util_6.1.23.v201004211559.jar`
 - `org.apache.felix.webconsole-3.1.8.jar`
 - `org.eclipse.osgi.services_3.2.100.v20100503.jar`

Apache frontend, Apache configuration

If you are using an Apache frontend, the best way to forward requests to the OSGi installation is to use the Apache Reverse Proxy mode.

See the following example config file for Apache (it includes a non proxy handling for the sub-directory `/files` and an access restriction to an IP range).

```

<VirtualHost *:80>
    ProxyRequests Off
    ProxyPreserveHost On

    <Proxy *>
        Require ip 137.193
    </Proxy>

    ProxyPass "/files" "!"
    ProxyPass "/" "http://localhost:8181/"
    ProxyPassReverse "/" "http://localhost:8181/"
    <Location />
        Require ip 137.193
    </Location>

    ServerAdmin michael.koch@unibw.de
    ServerName awareness.communitymashup.de
    DocumentRoot /var/www/awareness.communitymashup.de/web

    ErrorLog /var/www/awareness.communitymashup.de/log/error.log
    CustomLog /var/www/awareness.communitymashup.de/log/access.log combined
</VirtualHost>

```

OSGi bundles

First we need to place all needed OSGi bundles in the sub-directory "plugins":

The following table lists all bundles CommunityMashup needs (most sources can be found in the plugins directory in the install directory of the local Eclipse development environment). Alternatively, you can download a minimal set of bundles together with the whole directory structure and other default files from TBD.

Bundle Id	Description	Source
org.eclipse.osgi_3.8.1.v20120830-144521	Equinox Framework	Eclipse Installation /plugins
org.eclipse.osgi.services_3.2.100.v20100503	Equinox Standard OSGi Services	Eclipse Installation /plugins
org.apache.felix.fileinstall_3.1.10	Bundle Installations und Start Service	http://felix.apache.org/site/downloads.cgi
org.apache.felix.webconsole_3.1.8	Felix Web Console	http://felix.apache.org/site/downloads.cgi
javax.servlet_2.5.0.v200910301333	Servlets, benötigt von Jetty	Eclipse Installation /plugins
org.knopflerfish.bundle.commons-logging_2.0.0.kf3	Verschiedene benötigte Log Implementierungen	http://www.knopflerfish.org/repo/index.html
org.mortbay.jetty.util_6.1.23.v201004211559	Utils, benötigt von Jetty	Eclipse Installation /plugins
org.eclipse.equinox.http.jetty_2.0.0.v20100503	Equinox Version des OSGi Http Service (basierend auf Jetty)	Eclipse Installation /plugins
org.mortbay.jetty.server_6.1.23.v201004211559	Mortbay Server, benötigt von Jetty	Eclipse Installation /plugins
org.eclipse.equinox.http.servlet_1.1.0.v20100503	Equinox Version der Http Servlets	Eclipse Installation /plugins

Configuration

- Die Datei configuration/config.ini enthält folgendes:

```
osgi.bundles=org.apache.felix.fileinstall-3.1.10.jar@start
eclipse.ignoreApp=true
org.osgi.service.http.port=8088
```

Dies bedeutet:

- Es wird der der Felix File Install Service gestartet, der anschließend alle Bundles im Verzeichnis **load** installiert und startet.
- Es wird keine Eclipse Anwendung gestartet.
- Der Http Service wird über port 8088 erreichbar sein.

Starten der OSGi Laufzeitumgebung

Für den Start von Equinox wird im Hauptverzeichnis der angelegten Verzeichnisstruktur der Befehl `java -jar org.eclipse.osgi_3.6.2.R36x_v20110210.jar` ausgeführt.

Anschließend sorgt der Felix File Install Service dafür, dass alle Bundles sowie deren Abhängigkeiten in der richtigen Reihenfolge in Equinox installiert und gestartet werden.

Für die Steuerung der OSGi-Umgebung steht die OSGi Console zur Verfügung.

Diese kann je nach Startparameter entweder direkt in der Kommandozeile oder über Telnet aufgerufen werden.

Startanleitung mit Console in Kommandozeile

- Equinox Framework mit OSGi Console starten

```
java -jar org.eclipse.osgi_3.8.1.v20120830-144521.jar -console
```

Startanleitung mit Telnet Console

- Framework mit über Telnet erreichbare OSGi Console starten

```
java -jar org.eclipse.osgi_3.8.1.v20120830-144521.jar -console 7777 &
```

- Mit OSGi Console verbinden

```
telnet localhost 7777
```

- Von OSGi Console trennen

```
disconnect
```

Felix Console

Nach dem Start von Equinox steht die Console auch über die Felix Webschnittstelle bereit:

```
http://service.communitymashup.net/system/console
```

OSGi Kommandos

- Status-Abfrage der Bundles
Zeigt den aktuellen Status und die jeweilige id der installierten bzw. aktiven Bundles an.
Diese sind nach der Installation *Installed*, nach der Auflösung durch andere Bundles *Resolved* und nach erfolgreichem Start *Active*.

```
status
```

- Status-Abfrage der CommunityMashup Bundles
Zeigt den aktuellen Status und aller Bundles an, in deren Namen `"*soc*"` enthalten ist.
Diese sind nach der Installation *Installed*, nach der Auflösung durch andere Bundles *Resolved* und nach erfolgreichem Start *Active*.

```
ss soc
```

- Installieren von Bundles
Installiert zur Laufzeit ein Bundle über den angegebenen Verzeichnis-Pfad {path} zur JAR-Datei.
Bei erfolgreicher Installation wird die Bundle id mitgeteilt.

```
install file:{path}
```

- Aktivieren von installierten Bundles
Unter Angabe der Bundle {id} wird das jeweilige Bundle im Resolved-Status gestartet.

```
start {id}
```

- Deaktivieren von aktiven Bundles
Unter Angabe der Bundle {id} wird das jeweilige Bundle im Resolved-Status gestopt.

```
stop {id}
```

- Starten der gesamten OSGi-Umgebung

```
launch
```

- Beenden der gesamten OSGi-Umgebung

```
shutdown
```

- Beenden der OSGi Console

```
quit
```

Download: Minimale OSGi Umgebung als Zip

[Minimale OSGi Umgebung \(ZIP\)](#)

Contents

- [vhost Konfiguration](#)