

Djigzo developers guide

Introduction

Some of the Java technologies Djigzo uses are:

- Hibernate
- Spring
- CXF
- James
- Tapestry 5
- Bouncycastle
- Freemarker

Requirements

- Djigzo requires Java 1.5 or Java 6 JDK¹.
- Ant and ant optional.

Djigzo has been built and tested on Ubuntu. It should however work on most systems that run Java. It should be possible to compile and build on Windows but this has not been tested.

An Eclipse project file is included in the source distribution (Web Tools Platform WPT is required)

Building Djigzo

The following steps describe how to build Djigzo.

Some commands are Ubuntu (Debian) specific so if you are using another platform you should use equivalent commands for your platform.

Install Java JDK and Ant

Install required packages

```
$ sudo apt-get install openjdk-6-jdk ant ant-optional
```

Untar and build Djigzo

Download the Djigzo source distribution files “djigzo-src_*.tar.gz” and “djigzo-web-src_*.tar.gz” from the Djigzo website².

¹ It has only been tested with SUN JDK and OpenJDK.

² Replace * with the current version.

Create two directories to which we will untar the sources

```
$ mkdir djigzo
$ mkdir djigzo-web
```

Untar the source distributions

```
$ tar xvzvf djigzo-src_*.tar.gz --directory djigzo
$ tar xvzvf djigzo-web-src_*.tar.gz --directory djigzo-web
```

(replace * with the current version)

Go to the djigzo directory and compile. Compilation should succeed without errors

```
$ cd djigzo
$ ant
```

Now go to the djigzo-web directory and compile. Compilation should succeed without errors

```
$ cd ../djigzo-web
$ ant
```

The djigzo-web project requires the jar file generated by the djigzo project. It is therefore required that djigzo project is build first. The djigzo-web build.xml ant script contains a relative reference to the djigzo jar file so if the two directories do not have the same parent directory, or the djigzo directory is named differently, you should change the `"djigzo.jar"` ant property.

Building the Debian files (.deb)

Building the Debian files requires that you install the following Debian packages

```
$ sudo apt-get install fakeroot debhelper cdb
```

Building the Debian files can be done with the debian ant target

```
$ ant debian
```

The djigzo and djigzo-web projects both have the debian ant target.

Running tests

MAKE SURE THAT THE JAVA "UNLIMITED STRENGTH JURISDICTION POLICY FILES" ARE PROPERLY INSTALLED BEFORE RUNNING THE TESTS³.

The tests require a running PostgreSQL server.

installing PostgreSQL

```
$ sudo apt-get install postgresql
```

³ See "Djigzo administration guide" for more info about the "unlimited strength jurisdiction policy files".

Create a test user and test database

Temporarily make yourself the postgres user

```
$ sudo su postgres
```

Create a djigzo-test user with password: “test” (without the quotes)

```
$ createuser --no-superuser --createdb --createrole --login --pwprompt djigzo-test
```

Create a test database

```
$ createdb --owner djigzo-test djigzo-test
```

Return to your own account

```
$ exit
```

To test if the database can be reached you can try to setup a connection with psql.

```
$ psql -h 127.0.0.1 -U djigzo-test djigzo-test
```

Use \q to quit psql

Running the tests

Go to the djigzo directory and run the tests

```
$ cd djigzo  
$ ant run-tests
```

The tests can also be run directly from Eclipse.

Notes:

- Testing can take a couple of minutes.
- Running the tests requires a lot of memory. "run-tests" ant target is set to 640M. Increase or decrease this when out-of-memory occurs during testing.
- During testing a lot of exceptions are thrown. This is normal behavior as long as all the tests succeed.
- The tests includes a CRL download test so don't be alarmed when connections are setup to CRL distribution points from which the CRLs are downloaded.
- Tests can also be run directly from Eclipse.

Running Djigzo

The “Djigzo installation guide” explains in details how to install Djigzo. Here we will explain how to run Djigzo from the project directory.

You need to setup a few things before Djigzo can be started from the project directory

Create a database user djigzo with password djigzo (if this is not already done).

```
$ echo "CREATE USER djigzo NOCREATEUSER NOCREATEDB ENCRYPTED PASSWORD  
'md5b720bc9de4ca53d53a4059882a0868b9';" | sudo -u postgres psql
```

Create a djigzo database (if this is not already done).

```
$ sudo -u postgres createdb --owner djigzo djigzo
```

build project

```
$ ant
```

Import the database schema. Use password “djigzo” (without the quotes).

```
$ psql -U djigzo -h 127.0.0.1 djigzo < ./conf/djigzo.sql
```

You can now start Djigzo with:

```
$ ./start-djigzo.sh
```

Notes:

- debugging can be enabled by enabling the debug properties in ./wrapper/djigzo.wrapper.conf.

Djigzo-web must be deployed in Jetty (or any other Servlet engine). Djigzo -web can be debugged from Eclipse by using run-jetty-run Eclipse plugin.