

# Setting up Your Own Server

## Download and Installation

If you want to use service-based layout inside your network, you can download the latest build of the server that fits to your platform from

<http://www.informatik.uni-kiel.de/rtsys/kieler/downloads/tools-libraries/>

The other things you need to run the server are:

- a java runtime environment, preferably the latest release of version 6 (<http://www.oracle.com/technetwork/java/javase/downloads/>),
- the latest release of the reference implementation of JAX-WS, currently version 2.2.6 (<http://jax-ws.java.net/>), and
- the current version of graphviz (<http://www.graphviz.org/Download.php>).



### Note for Mac OS Users

In case you start the server under Mac OS and have Security issues, try this [fix](#) described in the *Viewing Ptolemy Models* section.

The server of KWebS is implemented in java. Therefore we begin the installation with the java runtime environment. The server uses a SOAP web service based on JAX-WS and the java platform comes with support for JAX-WS already. The included version is quite old, therefore, after you installed the java runtime, you need to update the built-in version by using the endorsed-mechanism from java:

1. create a folder named `endorsed` in the `lib`-directory of the java runtime, and
2. extract the JAR-files from the JAX-WS reference implementation to the `endorsed`-folder.

Now the java runtime uses the current version of JAX-WS. When you have finished the java installation you continue by setting up the graphviz package. Then it is time to install and configure the server of KWebS. You begin by extracting the downloaded archive to the location you prefer. Now you can start the server by executing the `kwebs` binary, but it will not be fully operational because you have to set some installation dependent options. Therefore, the next step is the configuration of the server.

## Setting required configuration options

The server supports a set of configuration options. They are listed in [Configuration options for the server](#). For most of these the *default configuration*, which is part of the server, defines a default value. The one option that depends on your installation is the path to the `dot` executable of the Graphviz package. Therefore you need to tell the server where it can find the package. In order to do so, you use the *provider configuration*. It is located in the file `kwebs.user` in the root folder where you extracted the server to. For example, in the following listing the executable is located at `/usr/bin/dot`:

```
# Use this option to configure the path to the graphviz dot executable
de.cau.cs.kieler.kwebs.graphviz.path=/usr/bin/dot
```

The next step is to decide which *service variants* you want to provide. There are two of them (plus the HTTP Post which is available via the support server):

- SOAP over HTTP,
- SOAP over HTTPS

As the following listing shows, you can define the variants to be published by the server in the provider configuration. In the given example, SOAP over HTTP is available:

```
# Use this option to configure whether the JAXWS layout will be published HTTP based
de.cau.cs.kieler.kwebs.jaxws.publishHttp=true

# Use this option to configure whether the JAXWS layout will be published HTTPS based
de.cau.cs.kieler.kwebs.jaxws.publishHttps=false
```

The next step is to define how the variants are being published. All variants per default only accept requests from `localhost`. In order to make them publicly available you need to configure from which IPs the variants are accepting requests from. You can also define the port a variant is listening on and the HTTP context it is bound to. In the following example they are accepting requests from `0.0.0.0`, meaning that they accept incoming requests from any client. The SOAP over HTTP variant is bound to port 8442 and the context `layout`.

```
# Use this option to configure the address, port and context under
# which the SOAP variant will be published over HTTP
de.cau.cs.kieler.kwebs.jaxws.httpAddress=http://0.0.0.0:8442/layout
```

If you decide to publish the SOAP over HTTPS variant you need a server certificate in order to secure the connection. While a certificate that has been officially signed by a Certificate Authority (CA) is preferable, the server provides a java key store containing a self signed certificate. It is located in the folder `server/kwebs/security/keystores` beneath the root directory of your server installation, the name of the key store is `server.jks`. Furthermore, due to the nature of the SSL-implementation of the java platform, the server provides the corresponding client trust store that clients can use to add the server certificate to the list of certificates the java runtime implicitly trusts (`client.jks`). The server also contains a script (`GenerateCertsAndKeystores.sh`) that allows you to generate these by yourself. You can find it in the folder `server/cmd` beneath the root directory of your server installation. The script requires the presence of OpenSSL (<http://www.openssl.org/>). While it is generally already installed on Linux platforms, for a Windows based server you need to download and install the OpenSSL toolkit (<http://www.openssl.org/related/binaries.html>).

The server needs to know the location of the key store containing your certificate and the password required to access it. The following listing shows how you can configure these options:

```
# Use this option to configure the path to your servers key store
de.cau.cs.kieler.kwebs.httpsKeystore.jks.path=security/my.jks

# Use this option to configure the password for your servers key store
de.cau.cs.kieler.kwebs.httpsKeystore.jks.pass=MyPassword
```

In the given example the key store is located at `security/my.jks` relative to the servers root directory and the password is `MyPassword`.

## Using the Start/Stop Scripts

After an initial start of the `kwebs` binary, `kwebs_start.*` and `kwebs_stop.*` scripts will be unpacked to the root directory. They can be used to properly start and stop the server via a management application. The application can only be contacted via local host. For the stop script to work properly the correct password and keystore have to be passed. The default values are shown next.

```
-kp=mserver -kl=server/kwebs/security/keystores/mserver.jks
```