UNIVERSITÄTSmedizin. MAINZ
Institut für Medizinische Biometrie, Epidemiologie und Informatik (IMBEI)

Andreas Borg, M.A.
Dipl.-Inf. Martin Lablans

55101 Mainz

Fon: +49 (0) 6131 17-8210
Fax: +49 (0) 6131 17-2968
E-Mail: info@mainzelliste.de

Mainzelliste – Installationsanleitung

Version 1.3 vom 24.08.2017

Versionshistorie

Version	Datum	Änderung/Status	Autor
1.0	05.05.2013	Erste Version	Andreas Borg, Martin Lablans
1.1	12.07.2013	Ergänzende Informationen zu Jersey	Andreas Borg, Maximilian Ataian
1.2	16.03.2015	Tomcat 6 → Tomcat 7, Hinweis zu alternativem Speicherort für Java Preferences, Hinweis auf Admin-Zugriff auf Patientendaten	Andreas Borg
1.3	24.08.2017	Obsolete Abschnitte bzgl. Java Preferences und manuelles Hinzufügen von Jersey entfernt; Hinweis auf mitgelieferte Datenbanktreiber ergänzt	Andreas Borg

1. Überblick

Grundlage für den Betrieb der Mainzelliste ist ein Webserver mit Apache Tomcat (Version 6 oder neuer). Empfohlen wird ein Ubuntu-Linux in der Serverversion (Ubuntu Server).

Die Installation erfolgt in Form einer WAR-Datei. Aus dem Eclipse-Projekt kann diese Datei mittels der Exportfunktion erstellt werden („File“ -> „Export“ -> „Web“ -> „WAR file“).

2. Zu installierende Pakete

- tomcat7
- Tomcat7-admin (Admin-Oberfläche, nicht nötig, erleichtert aber die Arbeit)
- Eine relationale Datenbank, z.B.:
  - PostgreSQL: Paket postgresql
  - MySQL: Pakete mysql-server und mysql-client (für Administration)

Für die gewählte Datenbank ist ein JDBC-Treiber im Classpath der Anwendung zu installieren. Treiber für MySQL und PostgreSQL werden mitgeliefert bzw. werden beim Build mit Maven heruntergeladen.

3. Konfiguration

Im folgenden Szenario sei „mainzelliste.example.org“ die Adresse des IDAT-Servers.

3.1 Tomcat

Benutzer (z.B. für Admin-Oberfläche) in /etc/tomcat7/tomcat-users.xml eintragen. Die Admin-Oberfläche benötigt Benutzerrollen „manager“ (für manager webapp) bzw. „admin“ (für host-manager webapp).

Die Änderungen werden mit Neustart von Tomcat aktiv.

3.2 SSL

Der Einsatz der Mainzelliste mit echten Patientendaten sollte nur über eine SSL-Verbindung erfolgen. Das folgende Beispiel skizziert die Erstellung und Installation auf einem typischen System. Für weitere Details sei auf die Dokumentation der verwendeten Programme sowie auf die zahlreichen Tutorials im WWW verwiesen.

Zertifikat und Key erzeugen:

- Erzeugen des Keys:

```
keytool -genkey -alias mainzelliste.example.org -keystore /etc/tomcat7/keystore -
keysize 4096 -keyalg RSA
```

  Im folgenden Dialog als "Common name" den Domainnamen des Servers (z.B. mainzelliste.example.org) angeben.
- Absichern des Keystore:

```
chmod 640 /etc/tomcat7/keystore
chown tomcat7:tomcat7 /etc/tomcat7/keystore
```

- Erzeugen des CSR:

```
keytool -certreq -alias mainzelliste.example.org -file patlist-req.pem -keystore
/etc/tomcat7/keystore
```

- Zertifikat via patlist-req.pem bei CA geholt, ggfls. erweitern um Zwischenzertifikate → patlist-crt.pem
- Importieren des Zertifikats:

```
keytool -import -alias mainzelliste.example.org -file patlist-crt.pem -keystore
/etc/tomcat7/keystore
```

Nutzung in Tomcat durch Einkommentieren des Connectors mit Port 8443. Port geändert auf 443. Damit ist die Anwendung im Browser ohne zusätzliche Angabe einer Portnummer aufrufbar. Ergänzen zweier Parameter keystoreFile und keystorePass (/etc/tomcat7/server.xml):

```
    <Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
               maxThreads="150" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS"
               keystoreFile="/etc/tomcat7/keystore"
               keystorePass="tomcat"
               ciphers="SSL_RSA_WITH_RC4_128_SHA, TLS_RSA_WITH_AES_128_CBC_SHA,
               SSL_RSA_WITH_3DES_EDE_CBC_SHA, SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA"
    />
```

Auskommentierung des normalen 8080-Connectors. Setzen von AUTHBIND=YES in /etc/default/tomcat7, damit tomcat7-Benutzer auf dem privilegierten Port lauschen kann.

Host für „mainzelliste.example.org" anlegen (/etc/tomcat7/server.xml):

```
      <Host name="mainzelliste.example.org"  appBase="webapps-mainzelliste"
            unpackWARs="true" autoDeploy="true"
            xmlValidation="false" xmlNamespaceAware="false"/>
```

Verzeichnis webapps-mainzelliste anlegen und Berechtigungen korrekt setzen.

Prüfung via https://www.ssllabs.com/ssltest/analyze.html?d=mainzelliste.example.org&hideResults=on.

3.3 Datenbank

Die Datenbankverbindung ist gemäß dem Konfigurationshandbuch einzustellen. Da beim ersten Start der Applikation die Datenbank initialisiert wird, muss der gewählte Datenbankbenutzer alle Berechtigungen auf der gewählten Datenbank erhalten. Am praktikabelsten ist es, pro Instanz der Mainzelliste einen eigenen Benutzer anzulegen, der Eigentümer der entsprechenden Datenbank ist.

3.4 Mainzelliste

Der Pfad der Anwendung entspricht dem Dateinamen des WAR ohne Erweiterung, z.B.: Deployen von „mainzelliste.war“ auf „mainzelliste.example.org“ -> Anwendung läuft unter https://mainzelliste.example.org/mainzelliste. Die Datei sollte also mit dem gewünschten Namen erzeugt oder umbenannt werden. Mit dem speziellen Namen „ROOT.war“ kann auch direkt auf dem Root-Pfad deployt werden (im Beispiel also https://mainzelliste.example.org/). Das Deployen erfolgt durch Kopieren in das webapps-Verzeichnis, z.B. /var/lib/tomcat7/webapps-mainzelliste.

Die Datei context.xml.default (in WebContent/META-INF im Eclipse-Projekt bzw. META-INF im WAR) ist in das xmlBase-Verzeichnis (z.B. /etc/tomcat7/Catalina/mainzelliste.example.org/) zu unter dem Namen [Name der Applikation].xml (z.B. mainzelliste.xml) zu kopieren. Darin ist der Pfad zur Konfigurationsdatei anzupassen – empfohlen: /etc/mainzelliste/[Name der Applikation].conf. Eine Vorlage für die Konfiguration wird in unter WEB-INF/classes/mainzelliste.conf.default bereitgestellt.

Neben der Datenbank ist mindestens der zugreifende Server zu konfigurieren, siehe Abschnitt „Zugriffsberechnungen“ im Konfigurationshandbuch.

Abschließend Tomcat neustarten. https://mainzelliste.example.org/mainzelliste aufrufen. Die Mainzelliste meldet sich.

4. Administrativer Zugriff auf Patientendaten

Unter der Ressource /html/admin/editPatient, im Beispiel also https://mainzelliste.example.org/mainzelliste/html/admin/editPatient, steht ein Formular zum Bearbeiten von Patientendaten durch den Administrator zur Verfügung. Dies erlaubt es auch, Patienten zu löschen, einen Datensatz als Duplikat eines anderen zu kennzeichnen und den Status „vorläufig“ eines Patienten zu entfernen (dient bei unsicheren Fällen der Bestätigung, dass es sich um einen neuen Patienten handelt).

Die Authentifizierung erfolgt standardmäßig[1] über Tomcat, der zugreifende Benutzer muss die Benutzerrolle „admin“ einnehmen. Die einfachste Möglichkeit, einen solchen Zugang einzurichten, besteht darin, folgende Einträge in die Datei tomcat-users.xml[2] aufzunehmen:

```
<role rolename="admin"/>
<user username="{Hier Namen eintragen}"
 password="{Hier Passwort eintragen}" roles="admin"/>
```

[1] Definiert im Deployment Descriptor (web.xml).
[2] Standardverzeichnis unter Linux: /etc/tomcat7.