JG|U UNIVERSITÄTSmedizin. MAINZ

Institut für Medizinische Biometrie, Epidemiologie und Informatik (IMBEI)

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

Obere Zahlbacher Straße 69
55131 Mainz

Fon: +49 (0) 6131 17-8210
Fax: +49 (0) 6131 17-2968
E-Mail: lablans@uni-mainz.de

Mainzelliste – Anleitung für Entwickler

Version 1.2 vom 28.01.2014

1. Versionshistorie

Version	Datum	Änderung/Status	Autor
0.1	03.05.2013	Erster Wurf	Andreas Borg
0.2	08.05.2013	Bitbucket statt SVN, Jersey nachladen	Andreas Borg
0.3	08.05.2013	Anpassung Benennung	Martin Lablans
1.1	21.08.2013	Anpassungen an Code-Release 1.1 (Maven)	Andreas Borg
1.2	28.01.2014	Korrekturen, Hinweis zu m2e-Konnektor	Andreas Borg, Benjamin Gathmann

2. Allgemeines

Beginnend mit Version 1.1 wird der Quellcode der Mainzelliste als Maven-Projekt veröffentlicht. Die Entwicklung kann prinzipiell also in jeder Entwicklungsumgebung mit Maven-Unterstützung erfolgen, ebenso ist ein Build mit Maven allein möglich. Im Folgenden wird der Import in Eclipse und Netbeans beschrieben. Für weitere Informationen zu Apache Maven sei auf http://maven.apache.org/ verwiesen.

3. Import in Eclipse

3.1 Eclipse einrichten

Benötigt wird „Eclipse IDE for Java EE Developers“. Getestet wurde Release „Juno“ in der 32-Bit-Version für Windows, verfügbar unter http://www.eclipse.org/juno/. Erfahrungsgemäß kann die 64-Bit-Version Probleme machen, also lieber die 32-Bit-Version vorziehen. Generell muss man in Eclipse mit einigen ‚Stolpersteinen‘ rechnen, wenn man das Projekt ans Laufen bringen will.

Es wird das Maven-Plugin *m2e* benötigt. Die Installation erfolgt folgendermaßen (vorher ist gegebenenfalls ein Proxy über *Window -> Preferences -> General -> Network Connections* zu konfigurieren):

1. Den „Eclipse Marketplace“ (*Help -> Eclipse Marketplace…*) öffnen.
2. Suchen nach „Maven”, dann „Maven Integration for Eclipse” installieren (**nicht** „Maven Integration for Eclipse WTP“!).

3.2 Import aus Datei

Zum Import des Projekts aus einer Datei (z.B. von BitBucket heruntergeladen) im Menü *File -> Import* wählen, dort *Maven -> Existing Maven Projects*.

3.3 Import von Bitbucket

Für den Import aus Bitbucket muss zunächst der Git-Connector für m2e installiert werden. Man wählt dazu *File -> Import -> Maven -> Check out Maven Projects from SCM*. Im folgenden Dialog klickt man auf den Link „Find more SCM connectos in the m2e Marketplace“. Dort „m2e-egit“ auswählen und installieren.[1]

Nach Installation und obligatorischem Neustart von Eclipse wiederum den Importdialog aufrufen. Nun kann in der ersten Auswahlliste unter *SCM URL* „git“ ausgewählt werden, als URL ist https://bitbucket.org/medinfo_mainz/mainzelliste.git einzutragen.

3.4 Projekteinstellungen

Eventuell muss Eclipse mitgeteilt werden, dass es sich um ein Projekt für eine Webapplikation handelt. Dazu im Kontextmenü (Rechtsklick auf Projekt) *Configure -> Convert to Faceted Form…* aufrufen (Falls dieser Menüeintrag fehlt, direkt in den Projekteinstellungen *Project Facets* auswählen). Im folgenden Dialog *Dynamic Web Module* wählen und bestätigen.

Zusätzlich muss in den Projekteinstellungen angegeben werden, dass die über Maven eingebundenen Bibliotheken beim Deployment mitexportiert werden. Dazu in den Projekteinstellungen unter *Deployment Assembly* mit *Add…* einen Eintrag hinzufügen. Im Dialog zuerst *Java Build Path Entries*, dann *Maven Dependencies* auswählen und mit *Finish* bestätigen.

3.5 Tomcat einrichten

Apache Tomcat, Version 7, herunterladen (http://tomcat.apache.org/) und installieren. In den Eclipse-Einstellungen *Server – Runtime Environments* wählen, mit *Add* einen neuen Eintrag hinzufügen. Als Typ *Apache – Apache Tomcat v7.0* wählen und den Pfad der Installation angeben.

Unter Ubuntu (möglicherweise auch in anderen Linux-Systemen) ist es nur umständlich möglich, eine Tomcat-Installation, die mit dem Paketmanager vorgenommen wurde, einzubinden. In diesem Fall wird empfohlen, Tomcat als Archiv herunterzuladen und manuell in ein Verzeichnis zu entpacken.

4. Import in Netbeans

Es wird die Netbeans IDE mit Unterstützung für Java EE, verfügbar unter https://netbeans.org/downloads/, benötigt. Getestet wurde Version 7.3.1 für Windows. Während der Installation wird abgefragt, welche Applikationsserver mitinstalliert werden sollen („Select the application servers to install with the IDE“), hier Apache Tomcat auswählen.

Nach der Installation kann das Projekt importiert werden. Dazu im Menü *Team -> Git -> Clone* wählen und als URL https://bitbucket.org/medinfo_mainz/mainzelliste.git angeben. Name und Pfad des Projekts können, falls gewünscht, angepasst werden.

5. Datenbank einrichten

Es wird eine relationale Datenbank mit JDBC-Treiber benötigt. Getestet wurde der Betrieb mit aktuellen Versionen von MySQL und PostgreSQL, für die folgende Treiber über Maven automatisch dem Projekt hinzugefügt werden:

[1] Falls die Installation fehlschlägt (beobachtet beim Release Kepler), ist eventuell die Installation über „Help“→“Install New Software“ möglich. Siehe dazu: http://stackoverflow.com/questions/11707971/cant-install-maven-scm-handler-for-egit-for-juno

- MySQL Connector/J (http://dev.mysql.com/downloads/connector/j/): Offizieller JDBC-Treiber für MySQL, unter der GPL veröffentlicht.
- PostgreSQL JDBC Driver (http://jdbc.postgresql.org/): Noch in Entwicklung befindlicher, aber in Bezug auf die JDBC-3-Spezifikation weitgehend vollständiger Treiber für PostgreSQL unter der BSD-Lizenz.

Treiber für andere Datenbanken müssen manuell heruntergeladen und im Classpath abgelegt werden. Treiber, die per Maven verfügbar sind, können in der pom.xml als Abhängigkeit mit dem Scope „runtime“ ergänzt werden.

6. Konfiguration

Für die Konfiguration sind die Konfigurationsdatei sowie der Context Descriptor (context.xml) anzupassen. Hierfür werden folgende Vorlagen mitgeliefert:

- Konfigurationsdatei: config/mainzelliste.conf.default. Das Verzeichnis *config* findet sich in Eclipse unter *Java Resources*, in Netbeans unter *Other Sources*. Eine angepasste Konfiguration lässt sich zum Beispiel durch Kopieren nach config/mainzelliste.conf anlegen, dieser Dateiname ist per .gitignore von der Versionsverwaltung ausgenommen. Ausgehend von der Beispielkonfiguration sind mindestens die Datenbankverbindung (Abschnitt „Database setup“) und der oder die zugreifenden xDAT-Server (Abschnitt „xDAT servers“) anzupassen. Eine ausführliche Dokumentation der Konfigurationsparameter findet sich im Dokument „Konfigurationshandbuch“.
- Context Descriptor: Datei *WebContent/META-INF/context.xml.default*, in Netbeans zu finden unter *Web Pages/META-INF*. Diese Datei im gleichen Ordner nach *context.xml* kopieren und dort den Pfad der Konfigurationsdatei angeben (Details siehe Vorlage). Für Entwicklungszwecke wird in der Regel die Konfigurationsdatei im Projekt (config/mainzelliste.conf) verwendet, dafür ist als Pfad */WEB-INF/classes/mainzelliste.conf* einzutragen.[2] In Netbeans ist darüber hinaus der Hinweis in der Vorlage (Attribut „path“ in Element „context“) zu beachten.

Hinweis für Netbeans: In der Netbeans IDE lassen sich Dateinamenerweiterungen nicht ändern. Hier ist es sinnvoll, context.xml und mainzelliste.conf zunächst als leere Dateien anzulegen und den Inhalt der Vorlage zur weiteren Bearbeitung hineinzukopieren.

7. Start

7.1 Eclipse

Rechtsklick auf das Projekt, dann *Run As – Run on Server* wählen, im Dialog dann den *Apache Tomcat v7.0* auswählen und Haken setzen bei *Always use this server when running this project*. Im Erfolgsfall erscheint in der Konsole „INFO de.pseudonymisierung.mainzelliste.Initializer - #####Startup succeeded. Ready to take requests.“ und die Mainzelliste ist unter http://localhost:8080/mainzelliste erreichbar (Statusmeldung „This is mainzelliste running version x.y for …“).

7.2 Netbeans

Start des Projekts ist aus dem Kontextmenü (*Run*), dem Hauptmenü (*Run -> Run Project*) und mit dem grünen Pfeil in der Symbolleiste möglich. Beim ersten Start muss im Dialog „Select deployment server“ Tomcat als Server ausgewählt werden. Netbeans startet Tomcat standardmäßig auf Port 8084, die Mainzelliste ist also unter http://localhost:8084/mainzelliste/ erreichbar.

[2] Sowohl ein absoluter Pfad im Dateisystem als auch ein Pfad innerhalb der Webapplikation ist möglich. */mainzelliste.conf* findet die Konfigurationsdatei aus dem Projekt, die über die Build-Einstellungen mitdeployt wird.

8. Struktur der Anwendung

Die Mainzelliste ist ein JAX-RS/Jersey-Servlet mit einer REST-Schnittstelle. Als Persistenzmechanismus wird Open-JPA genutzt. Die Anwendung ist in folgende Pakete gegliedert:

- de.pseudonymisierung.mainzelliste: Grundlegende Datenstrukturen und Initialisierung.
- de.pseudonymisierung.mainzelliste.dto: Persistenz-Schnittstelle. Kapselt Datenbankzugriffe durch Methoden zum Laden und Speichern von Datenobjekten.
- de.pseudonymisierung.mainzelliste.exceptions: Eigene Exception-Klassen.
- de.pseudonymisierung.mainzelliste.matcher: Record-Linkage-Framework.
- de.pseudonymisierung.mainzelliste.webservice: Ressourcen und Methoden der Schnittstelle. Diese bilden die „Einstiegspunkte“ von außen ins Programm. Eine Ressource *{name}* wird in der Regel in der Klasse *{name}Resource* implementiert, zum Beispiel enthält die Klasse *SessionsResource* Methoden für Zugriffe auf */sessions*.

9. Tipps und Problemlösungen in Eclipse

- EclipseTomcat startet, aber die Mainzelliste nicht (ohne Fehlermeldung oder dergleichen): Dies tritt häufig beim ersten Start in der Eclipse-Sitzung auf. Abhilfe: Im Tab „Servers“ Rechtsklick auf den Tomcat-Eintrag, *Clean…* auswählen und die folgende Abfrage bestätigen. Dies erzwingt einen Redeploy.
- Die Anwendung startet nicht, in der Konsole erscheint die Fehlermeldung „- This configuration disallows runtime optimization, but the following listed types were not enhanced at build time” (o.ä.): Der Open-JPA-Enhancer muss erneut aufgerufen werden. Dazu *Project* -> *Clean* aufrufen und gegebenenfalls ein *Refresh* auf dem Workspace ausführen (F5 oder Kontextmenü; vgl. unten). Das Problem hängt anscheinend damit zusammen, dass Eclipse bei Änderung einzelner Dateien diese Dateien am Maven-Builder vorbei selbst kompiliert, wodurch der Aufruf des Enhancers entfällt.
- Der Start schlägt fehl mit der Meldung „Resource is out of sync with the file system“: Dieser Fehler kommt dadurch zu Stande, dass der Build-Prozess von Maven ausgeführt wird und damit außerhalb der Kontrolle von Eclipse. Die Ausgabedateien sieht Eclipse dann als externe Änderung, die zunächst mit dem Workspace synchronisiert werden muss.
- Auch bei anderen Fehlern (beobachtet zum Beispiel: Klasse aus Mainzelliste wird nicht gefunden) kann Abfolge aus *Clean* und *Refresh* abhelfen.