

Installation von Tomcat und EDI-Portal

Installation TOMCAT

Java installieren

Sowohl für Tomcat als auch für das EDI Portal wird Java benötigt. Für das EDI-Portal Version 10 wird Java in der Version 8.0 benötigt. Ältere Versionen gehen nicht; Java 17 kann erst ab Portal in der Version 11 verwendet werden (Ältere Portal-Versionen haben mit Java 17 Probleme, Mails zu verschicken). Die Versionen dazwischen wurden nicht getestet, da in der Oracle-Ausgabe lizenztechnisch uninteressant.¹⁾ Im Falle von Oracle Java 8 empfiehlt sich die Version 1.8.0.202, welche die letzte lizenztechnisch unproblematische 1.8er Version ist. Die im Softzoll-Umfeld gerne eingesetzte Version 1.8.0.45 funktioniert zwar mit dem Portal, führt aber zu unregelmäßigen Abstürzen von Tomcat und ist deswegen nicht geeignet. Sowohl Oracle Java als auch OpenJDK funktionieren (OpenJDK ist von der Lizenz her im kommerziellen Umfeld unproblematisch).

Die für Tomcat benötigte Java-Version hängt von der eingesetzten Tomcat-Version ab, siehe <http://tomcat.apache.org/whichversion.html>

Die Installationsbeschreibung geht davon aus, dass die Tomcat-Version 7.0 verwendet wird. Für eine Tomcat-Version 7.x würde Java 6 reichen, nicht jedoch für das EDI-Portal. Installiert wird aber gleich Java 8, 64-Bit-Version.

Java 8 kann geladen werden unter:

<https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html>

OpenJDK gibt es hier:

<https://adoptopenjdk.net/>

Empfohlene Java-Version: 1.8.0_202

Bemerkung: Der Java-Installer einiger Versionen hat einen Fehler: Falls auf dem Rechner noch keine Java-Runtime vorhanden sein sollte, bricht er mit einem Fehler ab - und das, obwohl in den zu installierenden Dateien selbst eine komplette Runtime vorhanden ist! Daher muss vorher eine JRE installiert werden - Standard Edition genügt:

<http://www.java.com/de/download/>

Tomcat auswählen und herunterladen

Die Dokumentation basiert auf Tomcat in der Version 7.0. Die Versionen 8.0 und 9.0 können ebenfalls verwendet. Empfohlen wird Version 9.0. Tomcat 10.0 kann hingegen wegen der dort erfolgten Umstellung vom `javax-` auf den `jakarta-`Namensraum nicht direkt verwendet werden, aber mit einem Workaround geht es doch, siehe das nachfolgende Kapitel zu Tomcat 10.0.

Die Version kann geladen werden unter <http://mirrors.ae-online.de/apache/tomcat/tomcat-7/v7.0.50/bin> bzw. <https://tomcat.apache.org/download-90.cgi>.

In dieser Installationsanleitung wird die Datei apache-tomcat-7.0.50-windows-x64.zip benutzt.

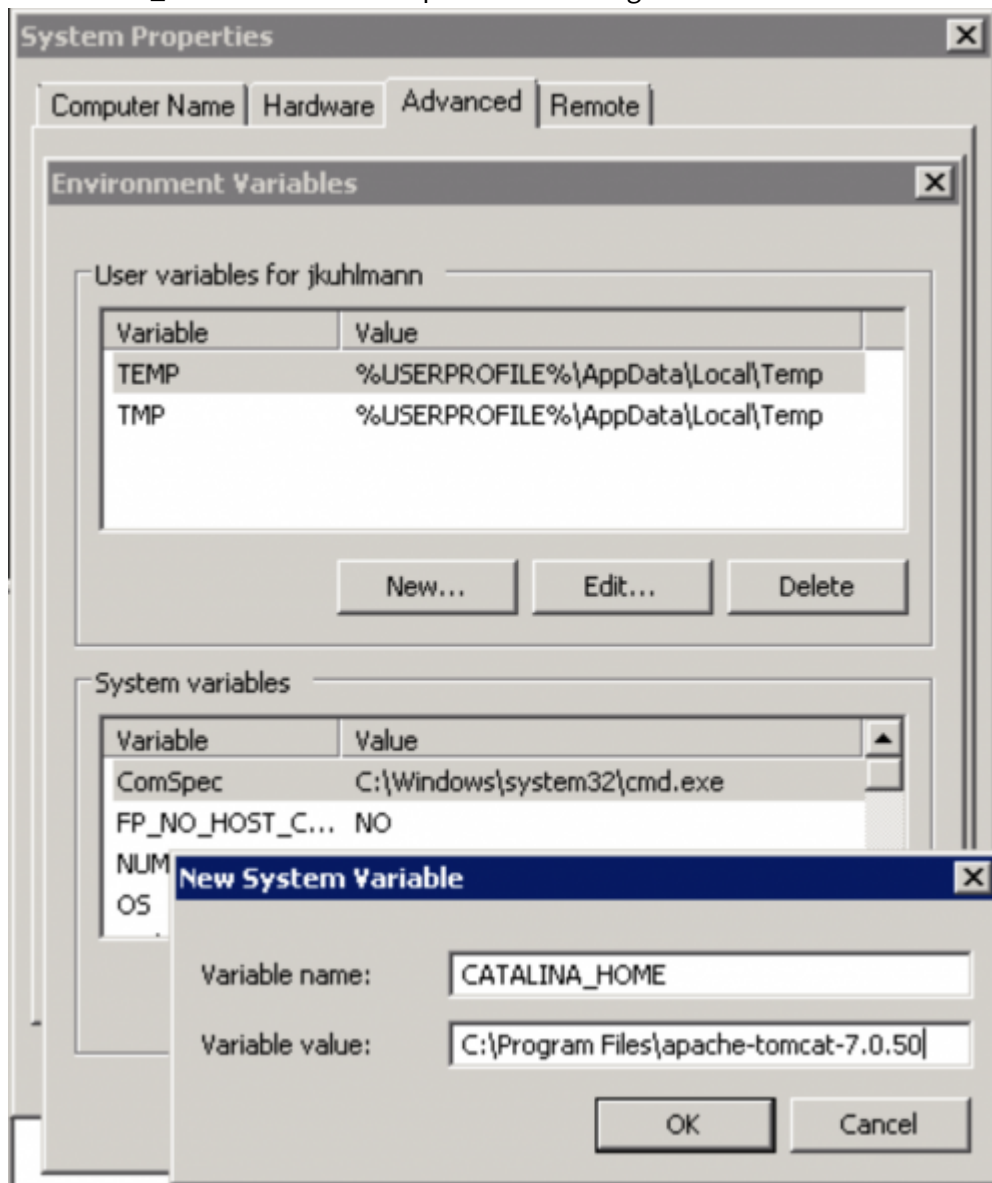
1. Zip-File entpacken
2. Weiteres Vorgehen wie in der Datei RUNNING.txt, welche sich im Hauptverzeichnis des Tomcat-Ordners befindet, beschrieben.

Zu 2.: Hier wird nur kurz die minimale Installation beschrieben, soweit sie für das EDI-Portal notwendig ist. Alles, was darüber hinausgeht, kann der Tomcat-Dokumentation entnommen werden.

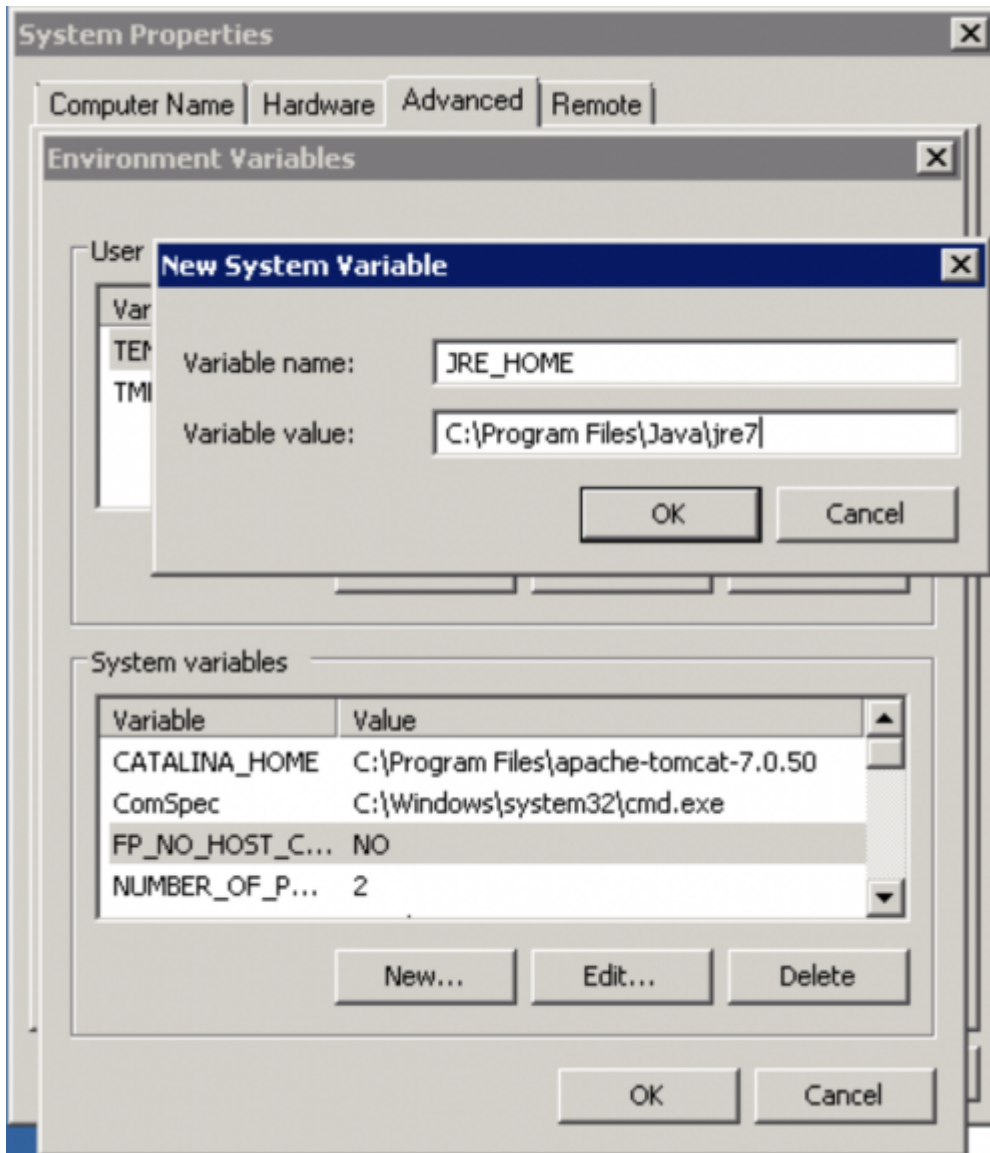
Die Zip-Datei wird entpackt. Das Verzeichnis apache-tomcat-7.0.50 wird dem entpacktem Ordner entnommen und unter \Program Files (oder wo man es gerne hätte) kopiert.

Die folgenden Umgebungsvariablen müssen gesetzt werden:

CATALINA_HOME muss zum Hauptverzeichnis zeigen:



JRE_HOME muss zur Java-Runtime zeigen:



Beide Variablen bitte **ohne** abschließenden Backslash „\“, sonst funktioniert es nicht.

Falls ein normaler Windows-Benutzer ohne Administratorrechte Tomcat starten darf, dann benötigt er Schreibrechte auf das logs-Unterverzeichnis.

Manuell wird Tomcat gestartet mit dem folgenden Befehl:

```
"%CATALINA_HOME%\bin\startup.bat"
```

Gestoppt wird Tomcat mit dem folgenden Befehl:

```
"%CATALINA_HOME%\bin\shutdown.bat"
```

(Beide Skripte können über den Windows-Explorer aufgerufen werden.)

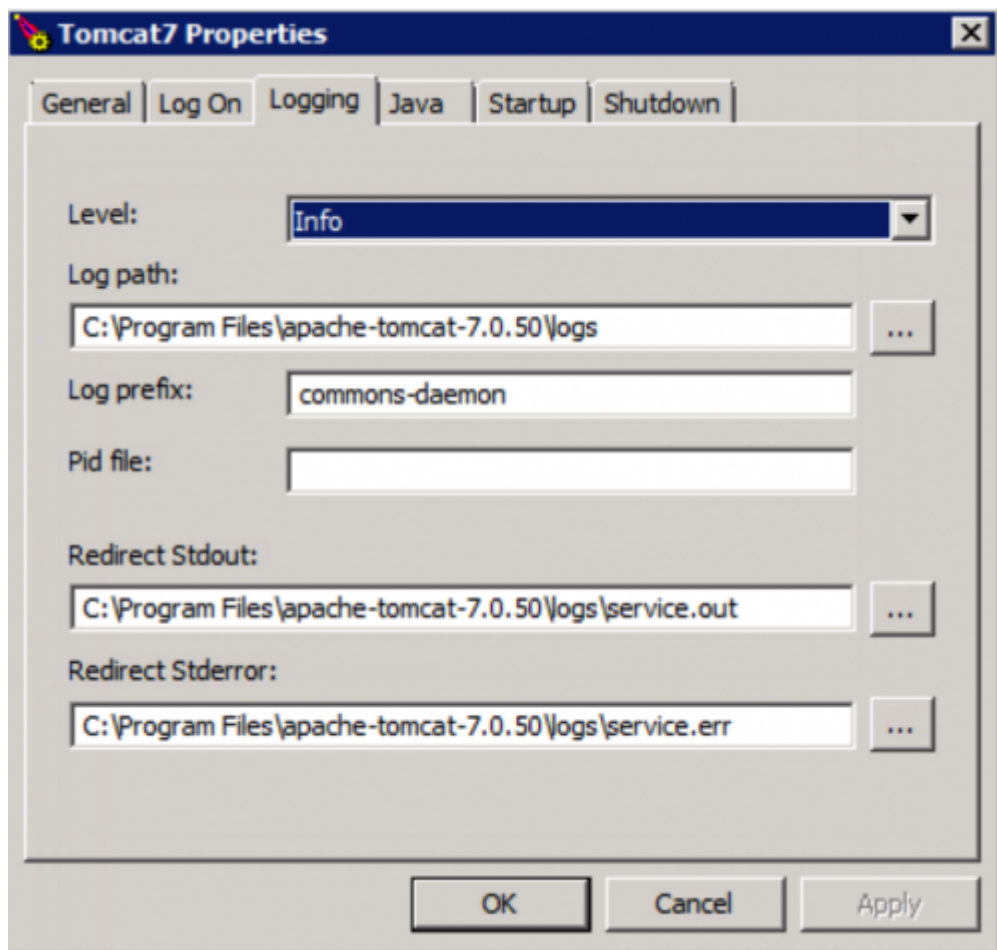
Tomcat kann in den Windows-Startup eingebaut werden. Dies ist z.B. dann sinnvoll, wenn Tomcat nach einem System-Neustart automatisch mitgestartet werden soll.

Dazu muss in der Kommandozeile das folgende Skript aufrufen werden:

```
%CATALINA_HOME%\bin\service.bat install
```

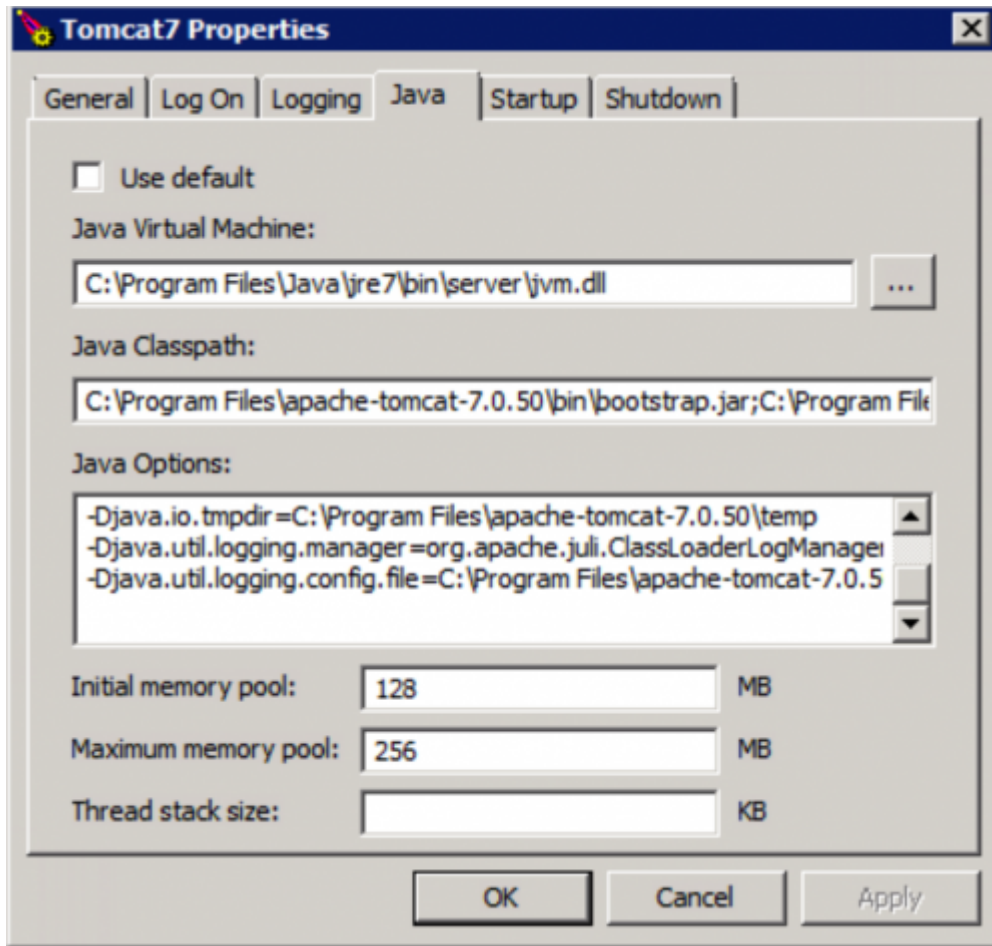
(Mit dem Skript kann Tomcat auch wieder aus dem Systemstart entfernt werden.)

Mit dem Programm %CATALINA_HOME%\bin\tomcat7w.exe muss der Log Path



und insbesondere der Classpath gesetzt werden zu

C:\Program Files\apache-tomcat-7.0.50\bin\bootstrap.jar;C:\Program Files\apache-tomcat-7.0.50\bin\tomcat-juli.jar (Der Pfad ist gegebenenfalls anzupassen, falls Apache woanders installiert wurde).



Tomcat mit HTTPS

Die Einschaltung von HTTPS folgt dem Tomcat SSL/TLS How-To, siehe <https://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html>.

Im Prinzip müssen die folgenden Schritte durchgeführt werden.

1. Zertifikat erstellen:

Sofern kein käuflich erworbenes Zertifikat vorliegt, kann ein Zertifikat mit OpenSSL oder mit dem Mac Zertifikatsassistent (siehe https://support.apple.com/kb/PH7173?locale=de_DE&viewlocale=de_DE) selbst erstellt²⁾ werden.

2. Java Keystore installieren

Der Keystore wird mit dem Befehl `$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA -keystore /path/to/my/keystore` installiert.

3. HTTPS in Tomcat-Konfiguration definieren

Die Datei `conf/server.xml` wird um die folgenden Zeilen erweitert:

```
<Connector port="8443"
  protocol="org.apache.coyote.http11.Http11NioProtocol"
  SSLEnabled="true"
  maxThreads="150" scheme="https" secure="true"
  clientAuth="false" sslProtocol="TLS"
  keystoreFile="C:/path/to/keystore/.keystore"
  keystorePass="password">
```

```
</Connector>
```

Gegebenenfalls kann die ungesicherte HTTP-Verbindung `<Connector port=„8080“ ... />` bei der Gelegenheit auskommentiert oder gelöscht werden.

1. Zertifikat in Keystore importieren.

Der Import ist im Apache-Howto beschrieben. Der Import ist scheinbar nicht nötig, wenn das exportierte Zertifikat im gleichen Verzeichnis wie der Keystore liegt.

Wenn jetzt zusätzlich normale Anfragen auf HTTPS umgeleitet werden sollen, ist noch Folgendes zu tun:

1. In `conf/server.xml` werden die Umleitungen deklariert, z.B:

```
<Connector port="8080" enableLookups="false" redirectPort="8443" />
```

Statt 8080 gegebenenfalls einen anderen Port angeben, der umgeleitet werden soll. Eventuell noch aktive andere Connector-Einträge zu jenem Port müssen natürlich entfernt werden.

2. In der Datei `conf/web.xml` (gleiches Verzeichnis wie `server.xml`) müssen unter dem Tag `<web-app>` die folgenden Zeilen eingefügt werden:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Protected Context</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

Tomcat auf Linux installieren

Bei der Installation des EDI-Portals unter Linux sind ein paar Besonderheiten zu beachten.

Java und Tomcat werden analog zu Windows installiert, siehe oben.

Die folgenden Umgebungsvariablen müssen gesetzt werden:

```
export JAVA_HOME=<Pfad zur Java-Runtime>
export JRE_HOME=<Pfad zur Java-Runtime>
export CATALINA_HOME=<Tomcat-Verzeichnis>
export softzoll_edi_cfg=<Pfad und Name der EDI-Portal-Konfigurationsdatei>''
```

Beispiel:

```
export JAVA_HOME=/etc/alternatives/jre_1.8.0
export JRE_HOME=/etc/alternatives/jre_1.8.0
export CATALINA_HOME=/usr/share/tomcat
export softzoll_edi_cfg=/home/ediuser/softzoll/ediconfig.properties
```

Alternativ zur Umgebungsvariable `$export softzoll_ed` kann in der Datei `$CATALINA_HOME/conf/catalina.properties` ein Java-Systemproperty gesetzt werden, welches auf die Konfigurationsdatei des EDI-Portals zeigt:

```
softzoll.edi.cfg=/usr/share/tomcat/softzoll/ediconfig.properties
```

Update auf eine neuere Tomcat-Version

Ein Update auf eine neuere Tomcat-Version ist möglich. Version 9.0 wird ohne Einschränkungen unterstützt, Version 10.0 mit Einschränkung (siehe nächstes Kapitel).

Folgende Punkte sind beim Update zu bedenken (hier exemplarisch für Windows):

- Im Verzeichnis `Apache-Tomcat-n.m.mm\logs` müssen die Schreibrechte überprüft werden, damit Log-Dateien geschrieben werden können.
- Webapps (also der Inhalt des Verzeichnis) muss aus dem alten Tomcat-Verzeichnis in das Neue kopiert werden.
- Die Datei `setenv.bat` muss aus dem alten `bin`-Verzeichnis in das neue `bin`-Verzeichnis kopiert werden.
- Die Datei `server.xml` im Verzeichnis `conf` muss mit dem Inhalt der alten `server.xml` in eine neue Datei migriert werden (nicht einfach den Inhalt der alten `server.xml` übernehmen; die enthält ein paar Einträge, mit denen Tomcat 9.0 nicht mehr funktioniert; also stattdessen ein Diff erstellen und das Passende herauskopieren)
- Der Service zum Tomcat-Start muss neu gemacht werden. (Ich habe diese Anleitung genutzt: tomcat.apache.org/tomcat-9.0-doc/windows-service-howto.html). Es hat hier allerdings Probleme gegeben und erst funktioniert, nachdem der alte Service komplett gelöscht und neu angelegt wurde.
- Nur für Tomcat 10.0.*: Der Workaround aus dem nächsten Kapitel zu Tomcat 10.0 muss in `conf/context.xml` eingebaut werden.

Hinweis: Das GWT-Entwicklungssystem läuft weder mit Tomcat 9 noch mit Tomcat 10. Hier empfiehlt sich weiterhin Tomcat 7. Das betrifft aber ausschließlich das Entwicklungssystem und hat keine Relevanz für produktive Installationen.

Besonderheiten mit Tomcat 10.0

Es gibt eine Namenstreitigkeit zwischen Oracle und der Eclipse Foundation um den `javax` Namensraum, welcher ab Tomcat 10 Auswirkungen hat. Tomcat 10 nutzt statt `javax` jetzt `jakarta`. Siehe https://tomcat.apache.org/migration-10.html#Specification_APIs und <https://www.heise.de/news/Apache-Tomcat-10-ist-konform-mit-Jakarta-EE-4986094.html>.

Dieser Namensraumwechsel hat zur Folge, dass bestehende Programme nicht mehr gestartet werden können, und das betrifft auch das EDI-Portal. Das ließe sich zwar leicht beheben, indem man die Referenzen anpasst, aber dann läuft der Code nicht mehr auf Tomcat 9.0 und älter (und vermutlich auch nicht auf alternativen Applikationsservern wie jBoss). Es gibt aber einen Workaround, indem man in der Tomcat-10-Konfiguration in `conf/context.xml` einen Loader definiert:

```
<Context>
  <Loader jakartaConverter="TOMCAT" />
```

```
...  
</Context>
```

Der Loader ist angeblich nicht perfekt, aber für das EDI-Portal sieht das gut aus.

Installation EDI-Portal

Installation Eclipse und GWT-Plugin

Falls das EDI-Portal in kompilierter Form vorliegt, kann dieses Unterkapitel übersprungen werden und mit dem Kapitel „Installation oder Updates des EDI-Portals“ fortgefahren werden.

Wenn das EDI-Portal noch nicht in kompilierter Form vorliegt, muss das Projekt erst einmal kompiliert werden. Dies geschieht am einfachsten über Eclipse.

Sofern Eclipse noch nicht installiert wurde, kann Eclipse unter <http://www.eclipse.org/downloads/> geladen werden und entsprechend der Anleitung (trivial) installiert werden. Es ist dabei übrigens egal, auf welchem Betriebssystem (Mac, Linux, Windows o.A.) Eclipse installiert wird und welche Version genommen wird, vorausgesetzt das GWT-Plugin steht für die Version zur Verfügung.

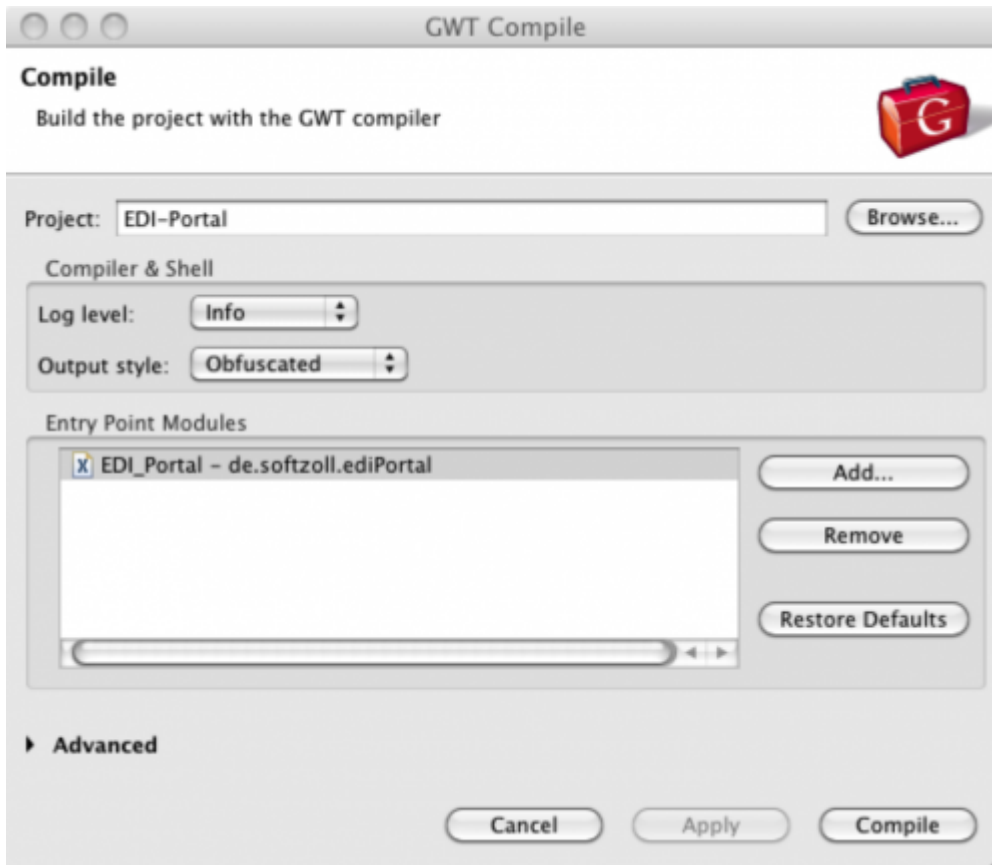
Das GWT-Plugin wird entsprechend dieser Anleitung über Eclipse geladen und installiert:
<https://developers.google.com/eclipse/docs/download>

Kompilation des Projektes

Das ausgecheckte EDI-Portal-Projekt muss kompiliert werden, und zwar sowohl für die Client-Seite als auch für die Server-Seite.

Der GWT-Compiler übersetzt die Java-Programme nach JavaScript, welches dann im Browser ausgeführt wird.

Aufgerufen wird der Compiler über das Kontextmenü zum Projekt, Punkt *Google* → *GWT Compile*.



Die Vorgaben können so übernommen werden und die Kompilierung wird durch einen Klick auf „Compile,, angestartet.

Für die Fehlersuche auf Javascript-Ebene kann „Output Style“ zu Pretty geändert werden. Für die Auslieferung sollte aber Obfuscated verwendet werden.

Die Serverseite wird wie in Eclipse üblich über das Menü *Project* → *Build Project* kompiliert. Wenn, wie in Eclipse üblich, *Project* → *Build Automatically* ausgewählt ist, braucht dieser Schritt nicht mehr explizit ausgeführt zu werden, da Eclipse in dem Falle die Server-Seite bereits übersetzt hat.

Zusätzlich zu den Programmdateien muss noch der Dump mit den Datenbankinhalten erzeugt werden. Dazu gibt es im Projekt in dem Unterverzeichnis `db/dump` das Skript `export.sh`, welches unter Linux und MacOS X läuft (benötigt die Unix-Shell). Dieses hat die Aufrufsyntax

```
export.sh <export file> <database name> <user> <password> [<host>]
```

Sehr wichtig ist, dass diesem Skript eine Datenbank mitgegeben wird, welche einen aktuellen Stand aller Tabellen des EDI-Portals enthält. Dies ist normalerweise die Datenbank `atlas_qm` auf dem Softzoll Testsystem mit der internen IP `172.16.66.164`.

Das Skript `export.sh` hat die Parameter

`<export file>`: Name der erzeugten Dump-Datei. Wenn eine neue Version zusammengestellt wird, ist dies die Datei `edi-portal-data.sql` im Verzeichnis `db/dump` des Projektes.

`<database name>`: Name der Datenbank, üblicherweise ist das `atlas_qm`.

`<user>`: Benutzername zum Zugriff auf die Datenbank, z. B. `root`.

`<password>`: Passwort dieses Benutzers

`<host>`: Adresse des Datenbankservers, optional, üblicherweise ist dies `172.16.66.164`.

Der so erstellte Dump sollte im Projekt mit eingecheckt werden.

Installation oder Updates des EDI-Portals

Die eigentliche Installation der Dateien und Daten des EDI-Portals besteht aus den folgenden Schritten:

1. Installation Programmdateien
2. Einspielen Datenbanktabellen und -inhalte
3. Neustart des Portals

Die Reihenfolge der Schritte 1. und 2. ist unerheblich und kann vertauscht werden.

Updates neuer Versionen des EDI-Portals erfolgen gleichermaßen. Die verwendeten Skripte sorgen selbst dafür, dass noch benötigte Daten der vorangegangenen Installation erhalten bleiben.

Installation der Programmdateien

Für die Installation oder das Update der Programmdateien einer neuen EDI-Portal-Version müssen Dateien und Verzeichnisse aus dem Unterverzeichnis `war` des EDI-Portal-Projektes in das Verzeichnis `%CATALINA_HOME%\webapps\ROOT` kopiert werden. Dies sind die Dateien und Verzeichnisse

- `EDI-Portal.html`
- `WEB-INF`
- `ediportal`
- `favicon.ico`

Für fertige Releases befinden sich die Dateien ebenfalls im Installationsverzeichnis (Testsystem: `F:\install`) unter `EDI-Portal\<Datum>\program`.

Etwaige vorhandene alte Dateien sollten vorher gelöscht werden (dies betrifft insbesondere die Unterverzeichnisse `ediportal` und `WEB-INF\deploy`, da der GWT-Compiler bei jedem erneuten Kompilieren andere Dateinamen verwendet).

Im Falle des Demosystems lauten die Dateinamen teilweise anders:

- *`EDI-Demosystem.html`*
- *`EDI-Demosystem.css`*
- *`WEB-INF`*
- *`edi_demosystem`*
- *`favicon.ico`*

Die alten Dateien und Verzeichnisse müssen ebenfalls gelöscht werden.

Bei der erstmaligen Installation auf einem neuen Rechner muss, sofern noch nicht vorhanden, im Verzeichnis `%CATALINA_HOME%\webapps\ROOT` das Unterverzeichnis `tmp` angelegt werden. Falls dieses Verzeichnis schon vorhanden ist, darf es nicht geleert und auch nicht gelöscht werden.

Installation der Datenbank-Objekte

Im Verzeichnis db/dump des Projektes gibt es vier Dateien, die für eine Installation bzw. ein Update eingespielt werden müssen. Für fertige Releases befinden sich die Dateien ebenfalls im Installationsverzeichnis (Testsystem: F:\install) unter EDI-Portal\<Datum>\database. Die Dateien werden in mysql mit `source <Dateiname>` eingespielt.

Die Dateien können ohne Probleme mehrfach eingespielt werden, ohne dass es dadurch zu Problemen kommt. Auch die Reihenfolge, in der die vier Dateien in die Datenbank eingespielt werden, ist unerheblich.

`edi-portal-data.sql`: Dies ist der im Kapitel Kompilation erstellte Dump. Es handelt sich dabei um mehrere mit mysqldump erzeugte Einzeldumps, die zu einer großen Datei zusammengestellt wurden. Beim Einspielen darf kein Fehler auftreten, da ansonsten das EDI-Portal nicht korrekt funktioniert. Falls Fehler gemeldet werden, sind diese zu korrigieren und das Dump muss anschließend erneut eingespielt werden.

`updates.sql`: Dies sind Updates am Datenmodell bestehender Tabellen, welche nicht durch den Dump `edi-portal-data.sql` vorgenommen werden können, da die bestehende Tabelle Kundendaten enthalten kann. Beim Einspielen dieser Datei treten in der Regel viele Fehlermeldungen auf, die ignoriert werden können. Wenn die entsprechende Datenmodelländerung in der Datenbank bereits enthalten ist, gibt es eine Meldung, dass die Änderung nicht erneut vorgenommen werden kann (z. B. Spalte existiert schon), dies kann ignoriert werden, da an dieser Stelle nichts mehr zu tun ist. Falls dieses Skript vor `edi-portal-data.sql` eingespielt wird, kann es Fehlermeldungen geben, dass eine bestimmte Tabelle noch nicht vorhanden ist. Dies kann auch ignoriert werden, da beim anschließenden Einspielen von `edi-portal-data.sql` die fehlenden Tabellen gleich in der korrekten Form angelegt werden.

`updatesMustNotFail.sql`: Dies sind Updates am Datenmodell bestehender Tabellen, bei denen die Fehlermeldungen nicht ignoriert werden dürfen, sondern die Ursache dazu behoben werden muss. Es kann sich um Änderungen handeln, die einen manuellen Eingriff erfordern. Hinweise zur Behebung der Fehler sind dann im Skript zu finden. (Zum Zeitpunkt der Erstellung dieser Dokumentation war die einzige hierin sich befindliche Änderung die Entfernung der Mandanten-Spalte aus der Kommunikationspartnertabelle und deren Schlüssel, welche Anpassungen verlangt, um Dubletten zu entfernen.)

`views.sql`: Hier sind alle View enthalten, die vom EDI-Portal benötigt werden. Beim Einspielen darf kein Fehler auftreten, da ansonsten das EDI-Portal nicht korrekt funktioniert. Falls Fehler gemeldet werden, sind diese zu korrigieren und die Datei muss anschließend erneut eingespielt werden.

Neustart des EDI-Portals

Anschließend muss Tomcat neu gestartet werden, entweder über die oben erwähnten Start- und Stopp-Skripte, oder - sofern der Tomcat-Start im Windows-Start integriert ist - über einen Windows-Neustart. Es empfiehlt sich, zumindest nach Ende der Funktionstests, zumindest zum Schluss einmal den Rechner neu zu starten, da Windows ansonsten mit dem Ausloggen des Benutzers auch gerne mal alle von ihm gestarteten Prozesse mit beendet.

Im Falle von Linux kann ein Neustart von Tomcat erfolgen mit `sudo service tomcat restart` (oder in zwei Befehlen mit `stop` und `start` an Stelle von `restart`). Ein Reboot des Rechners ist prinzipiell nicht erforderlich.

Konfiguration des EDI-Portals

Nach der erstmaligen Installation muss das EDI-Portal konfiguriert werden.

Es gibt zwei Konfigurationsdateien, einmal allgemeine Einstellungen für das EDI-Portal und einmal die Verbindungsdaten für die Datenbank. Beide Dateien sind im Properties-Format Schlüssel = Wert.

Diese Properties-Dateien können an beliebiger Stelle stehen, der Pfad und Name muss aber dem EDI-Portal mitgeteilt werden. Dies kann über das Java-System-Properties `softzoll.edi.cfg` und `softzoll.jdbc.cfg` geschehen. Alternativ können die Umgebungsvariablen `softzoll_edi_cfg` und `softzoll_jdbc_cfg` gesetzt werden wie oben beschrieben über die Windows-Einstelldialoge oder durch einen Eintrag in der Datei `%CATALINA_HOME%\bin\setenv.bat` (siehe Tomcat-Doku). Wenn beide Einträge vorhanden sind, dann hat das Java-System-Property Vorrang. Im Falle der Konfigurationsdatei der Datenbankverbindung kann an Stelle von Systemproperty oder Umgebungsvariablen auch der Dateipfad und Name in der EDI-Konfigurationsdatei konfiguriert werden, was der empfehlenswerteste Weg ist.

Wichtig: Es muss der Pfad in der für Java üblichen Schreibweise angegeben werden, d.h. alle Backslashes „\“ werden durch normale Schrägstriche „/“ ersetzt.

Ab Version 10 des EDI-Portals können diese beiden Pfade zu den Konfigurationsdateien auch als Servlet-Kontextparameter definiert werden, wo sie dann Vorrang vor Java-Systemproperties und Umgebungsvariablen haben. Dies ist dann sinnvoll, wenn ein Tomcat-Server sich um mehrere Varianten des EDI-Portals (z. B. unterschiedliche Versionen oder Portal, Demo und Tätigkeitenerfassung) kümmert und unterschiedliche Konfigurationen benötigt werden. Dazu wird in `$CATALINA_HOME/<service>/<host>/<service>` und `<host>` sind der Datei `server.xml` der Tomcat-Konfiguration zu entnehmen) die Datei `ROOT.xml` mit dem folgenden Inhalt angelegt:

```
<!-- context parameters -->
<Context >
  <Parameter name="softzoll.edi.cfg" value="/path/to/ediconfig.properties"
  override="false"/>
  <Parameter name="softzoll.jdbc.cfg" value="/path/to/jdbc.properties"
  override="false"/>
</Context>
```

Die Konfigurationsdatei des EDI-Portals heißt üblicherweise `ediconfig.properties`. Sie enthält die folgenden Werte:

`jdbc.configuration = <Name und Pfad der Datenbank-Konfigurationsdatei>`

Optional wenn der Pfad auf die Datenbank-Konfigurationsdatei als Umgebungsvariable oder als Kontextparameter definiert ist oder wenn sich die Parameter zur Datenbank-Konfiguration in dieser Datei der Portal-Konfiguration befinden, sonst Pflicht.

`superuser = yes|no|limited-to <Liste von IP-Adressen mit Semikolon, * erlaubt>`

Bei `yes` darf sich ein Superuser einloggen, bei `no` sind alle Superuser-Zugänge gesperrt. `no` ist für Installationen, die von außen zugänglich sind, ausschließlich zur Konsultation und die besonders geschützt sein müssen. Mit `limited-to` werden die Rechner, d. h. deren IP-Adressen, von denen aus eine Anmeldung als Superuser möglich ist, eingeschränkt. Es werden diejenigen Adressen angegeben, die eine HTTP-Anfrage haben muss, damit ein Login möglich ist. Mehrere Adressen werden durch ein Semikolon getrennt. Adressen

dürfen einen * als Platzhalter haben. Zu beachten ist dabei, dass bei einem Zugriff über einen Proxy oder ein VPN oftmals nicht die Adresse des Rechners des Benutzers sondern diejenige des Proxys oder des VPNs am Server erscheint und für die Prüfung genutzt wird. Ebenfalls bedacht werden sollte, dass der Platzhalter * rein auf Textebene prüft ohne Kenntnis der Struktur einer IP-Adresse. (Beispiel: „172.1*“ akzeptiert auch „172.15.1.1“, deshalb gegebenenfalls „172.1.*“ verwenden.)

Der Parameter ist optional. Fehlwert ist no, d. h. kein Superuser erlaubt.

`system = test|production`

Bei `test` werden gewisse Einstellungen „laxer“ gehandhabt, wie z.B. ob Eingabefelder Pflichtfelder sind oder nicht. Insbesondere ist bei `test` die Angabe eines Use-Cases nicht verpflichtend, auf Systemen mit `production` hingegen schon. Optional, Vorgabewert ist `production`.

`system.title = < Browsertitel >`

Der Titel, der im Browser angezeigt wird, kann hier definiert werden. Damit wird das `<title>`-Tag im `<head>`-Teil des HTML-Dokumentes überschrieben. Die Angabe ist optional. Wenn kein Titel definiert wird, wird der Vorgabewert aus den übersetzbaren Properties-Dateien (`EDIText.properties` bzw. `DemoText.properties` für das Demosystem) verwendet.

`clientInstallation = yes|no`

Eine Kundeninstallation ist für nur einen Mandanten und erlaubt es deshalb, andere Benutzer als Superuser mit mehr Rechten auszustatten. Eine Installation für mehrere Mandanten, so wie es im Rechenzentrum der Fall ist, darf in keinem Falle hier mit `yes` gekennzeichnet sein, da ansonsten Benutzer teilweise auf Daten anderer Kunden zugreifen oder zumindest die Funktion von deren Verarbeitung stören könnten. Der Wert `yes` ist also vor allem für Inhouse-Systeme gedacht; bei `no` sind höhere Rechte auf Benutzer mit der Rolle „Super-Administrator“ beschränkt, sodass „Benutzer“ z.B. nicht das Recht zugewiesen bekommen können, Workflows zu bearbeiten.

Der Parameter ist optional. Vorgabe ist `no`.

`sessionLifetime = < Lebenszeit in Minuten >`

Lebenszeit einer Sitzung in Minuten bis ein inaktiver Benutzer automatisch ausgeloggt wird und die Sitzung beseitigt wird. Die Lebenszeit einer Sitzung wird beim Login festgelegt. Spätere Änderungen dieser Eigenschaft in der Properties-Datei haben keine Auswirkungen auf bereits eingeloggte Benutzer. Ein Wert von 0 bedeutet die maximale Lebenszeit. Optional; ein fehlender Wert oder ein negativer Wert bedeutet die Vorgabe-Lebenszeit, welche sich zwischen EDI-Portal und Demosystem unterscheidet (30 Min bzw. 240 Min). Nach spätestens zwölf Stunden wird eine inaktive Sitzung immer beendet.

`uploadFileBase = < Upload-Verzeichnis >`

Hier können Dateien vom Benutzer hin geladen werden.

Dies ist ein Pflichtparameter.

`base.base_A = < Template-Verzeichnis >|NONE`

Verzeichnis für alle Templates, die in der Tabelle `atTableColumnDef_s00`, Spalte `dataSource` mit `base_A`: gekennzeichnet sind. Statt `base_A` können in zukünftigen Versionen auch andere Kürzel vorkommen, deren Verzeichnisse analog definiert sein müssen.

An statt eines Verzeichnisses kann auch `NONE` definiert werden, dann ist ein Ändern der Templates nicht möglich und alle Template-Buttons im GUI sind deaktiviert.

Dies ist ein Pflichtparameter.

`base.base_Print = < Template-Verzeichnis >|NONE`

Verzeichnis für alle Templates, die in der Tabelle `atTableColumnDef_s00`, Spalte `dataSource` mit `base_Print`: gekennzeichnet sind.

An statt eines Verzeichnisses kann auch `NONE` definiert werden, dann ist ein Ändern der Templates nicht möglich und alle Template-Buttons im GUI sind deaktiviert.

Dies ist ein Pflichtparameter.

`base.incoming = <Eingangsverzeichnis (Archiv)>`

Verzeichnis für eingehende Kundendateien, und zwar das Verzeichnis, wo sie **nach** der Verarbeitung abgelegt werden und zu einem beliebigen späteren Zeitpunkt eingesehen werden können. Benötigt wird der Eintrag derzeit nur für den Progress-Monitor, dort ist es aber ein Pflichtparameter.

`base.outgoing = <Ausgangsverzeichnis (Archiv)>`

Verzeichnis für ausgehende Kundendateien, und zwar das Verzeichnis, wo sie **nach** der Verarbeitung abgelegt werden und zu einem beliebigen späteren Zeitpunkt eingesehen werden können. Benötigt wird der Eintrag derzeit nur für den Progress-Monitor, dort ist es aber ein Pflichtparameter.

`base.outgoingPdf = <Pdf-Ausgangsverzeichnis (Archiv)>`

Verzeichnis für ausgehende Kundendateien im Pdf-Format, und zwar das Verzeichnis, wo sie **nach** der Verarbeitung abgelegt werden und zu einem beliebigen späteren Zeitpunkt eingesehen werden können. Benötigt wird der Eintrag derzeit nur für den Progress-Monitor, dort ist es aber ein Pflichtparameter.

`fastListDirectory = NONE | [<path/>]DIR | [<path/>]ls -lT | [<path/>]ls -l -full-time`

Beschleunigtes Einlesen der Verzeichnisse (insbesondere der Template-Verzeichnisse) über das System-Kommando `DIR` (bei Windows) oder `ls` (bei Unix, Linux und Mac OS/X) an Stelle der verhältnismäßig langsamen Java-Bibliothek. Da sich das Ausgabeformat dieser Befehle im Prinzip in späteren Betriebssystemversionen ändern kann, ist dieses beschleunigte Einlesen potentiell gefährdet, irgendwann nicht mehr zu funktionieren. Daher sollte ein anderer Wert als `NONE` nur konfiguriert werden, wenn ein Performance-Problem beim Öffnen der Template-Dialoge besteht.

- `NONE`: Kein Beschleunigtes Einlesen; dies ist auch der Fall, wenn der Schlüssel in der Konfiguration ganz fehlt.
- `DIR`: Im Betriebssystem werden Verzeichnisse mit `DIR` angezeigt. Dies ist bei Windows der Fall.
- `ls -lT`: Als System-Kommando zum Anzeigen der Verzeichnisse kommt `ls` zum Einsatz, und die komplette Zeit wird über die Option `-T` angezeigt. Dies ist bei Mac OS/X und möglicherweise einigen Unix- und Linux-Varianten der Fall.
- `ls -l -full-time`: Als System-Kommando zum Anzeigen der Verzeichnisse kommt `ls` zum Einsatz, und die komplette Zeit wird über die Option `-full-time` angezeigt. Dies ist bei Linux und möglicherweise einigen Unix-Varianten der Fall.

Optionaler Parameter, Vorgabe ist `NONE`.

`logoDirectory = <Verzeichnis für Mandanten-Logos>`

Dies ist ein Pflichtparameter.

`comPartnerLogoDirectory = <Verzeichnis für Logos von Kommunikationspartnern>`

Dieser Parameter wird nicht mehr benötigt und kann entfallen. In alten Versionen war er aber ein Pflichtparameter.

`wfPortTool.import = <Import-Befehl für das WfPortTool>`

Das Kommando in der Befehlszeile (Shell) für den Aufruf des `WfPortTool` für Importe wird festgelegt. Das Kommando muss den kompletten Pfad inklusive Dateinamen des `WfPortTool` enthalten sowie sämtliche Parameter, die verlangt werden. Zur Übergabe von Parametern zwischen EDI-Portal und `WfPortTool` werden in diesem Kommando die folgenden Variablen ersetzt:

- `$file`: Wird durch den Namen der zu importierenden Datei ersetzt.
- `$replaceTemplate`: Erhält YES, BACKUP oder NO je nachdem, ob Template-Dateien beim Import ersetzt werden, ersetzt werden mit vorherigem Backup oder nicht ersetzt werden.
- `$replaceWorkflow`: Erhält REUSE oder NO je nachdem, ob ein bestehender Workflow wiederverwendet wird oder zu einem Fehler führt.
- `$user`: Erhält die ID des Benutzers, der den Import im EDI-Portal aufgerufen hat. Diese ID wird als `f_createUserId` oder `f_modUserId` in den Audit-Spalten der neu erstellten oder geänderten Zeilen in der Datenbank eingetragen.
- `$date`: Wird mit dem aktuellen Datum im Format `yyyy-MM-dd` ersetzt.
- `$time`: Wird mit der aktuellen Uhrzeit im Format `HHmmss` ersetzt.

Ein Beispiel für den Wert `wfPortTool.import`:

```
wfPortTool.import = /dies/ist/der/vollständige/Pfad/WfPortTool IMPORT -o
/Pfad/zu/Logs/wfPortLog.$date.txt -db DEFAULT_DB -base /Pfad/zu/Basis -
replTmpl $replaceTemplate -replWf $replaceWorkflow -createUser $user -file
$file
```

Der Parameter ist optional. Ist er nicht vorhanden, ist kein Workflow-Import möglich, und der entsprechende Button wird in der GUI nicht angezeigt.

`wfPortTool.export` = <Export-Befehl für das `WfPortTool`>

Das Kommando in der Befehlszeile (Shell) für den Aufruf des `WfPortTool` für Exporte wird festgelegt. Das Kommando muss den kompletten Pfad inklusive Dateinamen des `WfPortTool` enthalten sowie sämtliche Parameter, die verlangt werden. Zur Übergabe von Parametern zwischen EDI-Portal und `WfPortTool` werden in diesem Kommando die folgenden Variablen ersetzt:

- `$file`: Wird durch den Namen der Export-Datei ersetzt, die das `WfPortTool` beim Export schreiben soll.
- `$workflow`: Erhält die ID des Workflows, der exportiert werden soll.
- `$date`: Wird mit dem aktuellen Datum im Format `yyyy-MM-dd` ersetzt.
- `$time`: Wird mit der aktuellen Uhrzeit im Format `HHmmss` ersetzt.

Ein Beispiel für den Wert `wfPortTool.export`:

```
wfPortTool.export = /dies/ist/der/vollständige/Pfad/WfPortTool EXPORT -o
/Pfad/zu/Logs/wfPortLog.$date.txt -db DEFAULT_DB -base /Pfad/zu/Basis -wf
$workflow -file $file
```

Der Parameter ist optional. Ist er nicht vorhanden, ist kein Workflow-Export möglich, und der entsprechende Button wird in der GUI nicht angezeigt.

`wfPortTool.export.edibus` = <Export-Befehl des Exports mit der EDIBUS-Verzeichnisstruktur für das `WfPortTool`>

Das Kommando in der Befehlszeile (Shell) für den Aufruf des `WfPortTool` für Exporte mit EDIBUS-Verzeichnisstruktur wird festgelegt. Das Kommando muss den kompletten Pfad inklusive Dateinamen des `WfPortTool` enthalten sowie sämtliche Parameter, die verlangt werden. Zur Übergabe von Parametern zwischen EDI-Portal und `WfPortTool` werden in diesem Kommando die folgenden Variablen ersetzt:

- `$workflow`: Erhält die ID des Workflows, der exportiert werden soll.
- `$replaceDir`: Erhält NO, YES, BACKUP oder UPDATE je nachdem, ob bei bestehendem Zielverzeichnis der Export abgebrochen wird, das Verzeichnis überschrieben wird, das Verzeichnis umbenannt wird oder nur die F50 aktualisiert wird.
- `$date`: Wird mit dem aktuellen Datum im Format `yyyy-MM-dd` ersetzt.
- `$time`: Wird mit der aktuellen Uhrzeit im Format `HHmmss` ersetzt.

Ein Beispiel für den Wert `wfPortTool.export.edibus`:

```
wfPortTool.export = /dies/ist/der/vollständige/Pfad/WfPortTool EXPORT -o  
/Pfad/zu/Logs/wfPortLog.$date.txt -db DEFAULT_DB -base /Pfad/zu/Basis -wf  
$workflow -file /Pfad/zum/EDIBUS/Verzeichnis -replDir $replaceDir
```

Der Parameter ist optional. Ist er nicht vorhanden, ist kein Workflow-Export möglich, und der entsprechende Button wird in der GUI nicht angezeigt.

`tmpDir` = <temporäres Verzeichnis>

Verzeichnis für temporäre Dateien. Optionaler Parameter, Vorgabe ist `/tmp` für Unix/Linux und `C:\temp` für Windows. Das Verzeichnis muss existieren. Derzeit wird das Verzeichnis nur vom Workflow-Import und -Export genutzt, kann aber in Zukunft auch andersweitig genutzt werden.

`template.pdf` = <Pdf-Template Sendungsverfolgung>

Kompletter Pfad und Dateiname der Template-Datei für das Template des Pdf-Exports in der Sendungsverfolgung. Optional. Sollte dieser Eintrag nicht vorhanden sein, wird das vorgegebene Template aus dem EDI-Portal genommen, was der Normalfall sein dürfte.

`template.xls` = <Excel-Template Sendungsverfolgung>

Kompletter Pfad und Dateiname der Template-Datei für das Template des Excel-Exports in der Sendungsverfolgung. Optional. Sollte dieser Eintrag nicht vorhanden sein, wird das vorgegebene Template aus dem EDI-Portal genommen, was der Normalfall sein dürfte.

`template.partner.pdf` = <Pdf-Template Export Partner>

Kompletter Pfad und Dateiname der Template-Datei für das Pdf-Template des Partner-Exports. Der Partner-Export ist in der Anwendung „Use-Cases“ aufrufbar. Exportiert werden dort Informationen zu Partnern und Workflows. Optional. Sollte dieser Eintrag nicht vorhanden sein, wird das vorgegebene Pdf-Template aus dem EDI-Portal genommen, was der Normalfall sein dürfte.

`template.partner.xls` = <Excel-Template Export Partner>

Kompletter Pfad und Dateiname der Template-Datei für das Excel-Template des Partner-Exports. Der Partner-Export ist in der Anwendung „Use-Cases“ aufrufbar. Exportiert werden dort Informationen zu Partnern und Workflows. Optional. Sollte dieser Eintrag nicht vorhanden sein, wird das vorgegebene Excel-Template aus dem EDI-Portal genommen, was der Normalfall sein dürfte.

`template.cpas.xls` = <Excel-Template Export Workflow-Zusammenfassung>

Kompletter Pfad und Dateiname der Template-Datei für das Excel-Template des Exports von Workflow-Zusammenfassungen. Dieser Export ist im Dialog „Workflow-Zusammenfassung“ in der Anwendung „Workflows“ aufrufbar. Optional. Sollte dieser Eintrag nicht vorhanden sein, wird das vorgegebene Excel-Template aus dem EDI-Portal genommen, was der Normalfall sein dürfte.

`template.timer.xls` = <Excel-Template Export Tätigkeiten>

Kompletter Pfad und Dateiname der Template-Datei für das Excel-Template des Tätigkeiten-Exports. Der

Tätigkeiten-Export ist in den Anwendungen „Archiv“ und „Prüfung“ zu finden. Optional. Sollte dieser Eintrag nicht vorhanden sein, wird das vorgegebene Excel-Template aus dem SZ-Timer genommen, was der Normalfall sein dürfte.

`template.timer.order.xls` = <Excel-Template Export Tätigkeiten für AB-Nummern>

Kompletter Pfad und Dateiname der Template-Datei für das Excel-Template des Tätigkeiten-Exports in der Anwendung „AB-Nummern“ im Dialog „Buchungen“. Dieses Template wird für AB-Nummern verwendet, die kein Zeitkontingent besitzen. Optional. Sollte dieser Eintrag nicht vorhanden sein, wird das vorgegebene Excel-Template aus dem SZ-Timer genommen, was der Normalfall sein dürfte.

`template.timer.orderctg.xls` = <Excel-Template Export Tätigkeiten für AB-Nummern mit Kontingent>

Kompletter Pfad und Dateiname der Template-Datei für das Excel-Template des Tätigkeiten-Exports in der Anwendung „AB-Nummern“ im Dialog „Buchungen“. Dieses Template wird für AB-Nummern verwendet, die ein eigenes Zeitkontingent besitzen. Optional. Sollte dieser Eintrag nicht vorhanden sein, wird das vorgegebene Excel-Template aus dem SZ-Timer genommen, was der Normalfall sein dürfte.

`exportXls.maxRows` = <max. Anzahl Zeilen>

Hiermit wird die maximale Anzahl von Zeilen für ein Arbeitsblatt in Excel festgelegt. Werden mehr Zeilen exportiert, werden die überzähligen Zeilen in weitere Arbeitsblätter geschrieben.

Optional. Normalerweise ist die Angabe überflüssig, da der Standardwert von 65.000 die Limitierungen des Excel-Dateiformats abfängt. Der Parameter kann gesetzt werden, wenn Arbeitsblätter kurz gehalten werden sollen oder für Tests.

`hostApplicationAddress` = `https://<URL>`

Diese Adresse wird in den vom System verschickten E-Mails (z. B. nach Zurücksetzen des Passwortes) als Adresse des Systems mitgeteilt.

Dies ist ein Pflichtparameter. Fehlt er, sind die Angaben in der verschickten E-Mail unvollständig.

`email.command` = <Pfad und Name einer Batch-Datei>

Ist hier eine Batch-Datei definiert, so wird diese ausgeführt, anstatt eine E-Mail zu verschicken. Die Batch-Datei muss sich dann um den Versand der Mail kümmern. Die notwendigen Daten (Mail-Adresse des Empfängers, Benutzername, Hostname, Passwort) werden dem Batch als Parameter übermittelt. Falls die Batch-Datei genutzt wird, werden alle folgenden Properties zur E-Mail-Konfiguration ignoriert.

Dies ist ein optionaler Parameter.

`email.hostname` = <Mail-Host-DLD> (z.B. `smtp.domain.tld`)

Adresse des Mail-Servers (Ausgang)

Dieser Parameter ist Pflicht, es sei denn, der alternative Parameter `email.command` wird genutzt, dann entfällt er.

`email.port` = `nnn`

Port des Mail-Servers (Ausgang)

Optionaler Parameter, Vorgabewerte sind 465 für SSL, 587 für Starttls und 25 ohne Verschlüsselung.

`email.authentication.username` = <name>

Login-Name für Zugang am Mail-Server

Pflichtparameter, es sei denn, es ist keine Authentifizierung am SMTP-Server nötig.

`email.authentication.password` = <Passwort>

Passwort für Zugang am Mail-Server

Pflichtparameter, es sei denn, es ist keine Authentifizierung am SMTP-Server nötig.

```
email.from = edi-sender@itb-kuhlmann.com<E-Mail-Adresse Server>
```

E-Mail-Adresse, welche im from-Feld eingetragen wird.

Dies ist ein optionaler Parameter, wenngleich dringend empfohlen, da eine Absenderangabe in einer E-Mail erwartet wird.

```
email.replyTo = editest@itb-kuhlmann.com<E-Mail-Adresse Server>
```

E-Mail-Adresse, welche im replyTo-Feld eingetragen wird.

Dies ist ein optionaler Parameter (auch das Feld ReplyTo ist üblicherweise optional).

```
email.security = SSL|STARTTLS|NONE
```

Verschlüsselung der E-Mail auf dem Weg zum Mail-Server.

Dies ist ein optionaler Parameter, Vorgabe ist SSL.

Es ist möglich, einzelne Tabellengruppen von einer externen (d. h. einer anderen als der für den Rest genutzten) Datenbank zu beziehen. Dies geht mit den folgenden drei Parametern (die mehrmals mit unterschiedlichen Aliassen auftauchen können, wenn von mehreren externen Datenbanken Daten gelesen werden):

```
jdbc.configuration.<Alias> = <Name und Pfad der Datenbank-Konfigurationsdatei>
```

```
external.table.<Gruppenname> = <Alias> external.store.<Gruppenname> = FEDERATED
```

Die Parameter sind optional und werden genau dann verlangt, wenn eine Tabellengruppe extern gespeichert ist. external.store.<Gruppenname> ist immer optional.

Siehe [Externe Tabellen im EDI-Portal](#) für eine Beschreibung.

Die Verbindung zur Datenbank muss in der Datei `jdbc.properties` konfiguriert werden. Ein Beispiel befindet sich im Hauptverzeichnis des Projektes. Der Aufbau der Datei ist wie folgt:

```
connection.username = <MySQL User>
```

```
connection.password = <MySQL Passwort>
```

```
connection.driver = org.gjt.mm.mysql.Driver
```

```
connection.url = jdbc:mysql://<Rechner-URL>/<Datenbankname>
```

Verwenden Sie `localhost` als „Rechner-URL“, wenn sich die Installation der Datenbank auf demselben System befindet wie die Installation des EDI-Portals (z.B. auf Inhouse-Systemen), also z.B. `connection.url = jdbc:mysql://localhost/atlas_test?autoReconnect=true`.

Diese Parameter zur Datenbankverbindung sind Pflichtangaben.

Da sich in dieser Datei ein MySQL-Passwort im Klartext befindet, sollte der Zugriff auf die Datei über die Dateisystem-Rechte des Betriebssystems begrenzt werden. Der Tomcat-User benötigt natürlich das Leserecht.

Installation der Datenbank

Für den Betrieb des EDI-Portals muss eine MySQL-Datenbank vorhanden sein. Diese wird gegebenenfalls wie in der MySQL-Dokumentation beschrieben installiert. Auf der Datenbank (Datenbank-Managementsystem) muss ein Account für das EDI-Portal erstellt werden, dessen Zugangsdaten in die im vorigen Kapitel beschriebene Properties-Datei eingetragen werden (mit diesen Daten meldet sich die Portal-Software an der Datenbank an).

Für die benötigten Tabellen sind Skripte im EDI-Portal-Projekt im Unterverzeichnis `db\dmd` enthalten. Diese

werden einfach auf der Datenbank angelegt.

Ein erster Mandant muss in der Tabelle `atclient00` per Insert angelegt werden. Beim Insert sollte die `clientID` zu 0 gesetzt werden. Durch die Autoinkrement-Eigenschaft der Spalte erhält die ID einen gültigen Wert. Ebenso muss ein erster Anwender in der Tabelle `atuser00` angelegt werden. Dabei sollte die `UserID` wieder zu 0 (Autoinkrement) gesetzt werden, die `f_clientID` zu der ID des eben erzeugten Mandanten. Als `role` sollte A gewählt werden, sodass der Anwender diesen Mandanten verwalten kann und selbst über das Portal neue Anwender anlegen kann.

Das Passwort in der Tabelle `atclient00` ist über einen Hash verschlüsselt. Dieser Hash kann erzeugt werden über den Aufruf:

```
java -jar pwdHash.jar <username> <passwort>
```

Die hierzu nötige Datei `pwdHash.jar` kann in Eclipse über die im Hauptverzeichnis des Projektes liegende `createPwdHash.jar`desc (Aufruf mit *Open With* → *JAR Export Wizard*) erzeugt werden.

Templates für den PDF- und Excel-Export

In einigen Anwendungen besteht die Möglichkeit für einen Export der Daten in das PDF- oder in das Excel-Format. Für die Ausgabedateien gibt es Templates, dies sind PDF- und Excel-Dateien, welche die Struktur, Rahmenelemente, eine Titelseite u. Ä. enthalten, für die eigentlichen Daten aber nur Platzhalter. Beim Export wird dann die Template-Datei kopiert und die Platzhalter werden mit den aktuellen Daten ersetzt.

Die Templates sind im EDI-Portal mitgeliefert. Es besteht aber auch die Möglichkeit, über die Konfiguration eigene Templates einzubinden. Beiden Fällen ist gemein, dass die Template-Dateien gewissen Anforderungen genügen müssen, damit der Export das gewünschte Ergebnis bringt.

Die Sendungsverfolgung („Track & Trace“) zeigt die verarbeiteten Belege zusammengefasst je übertragene Datei (Interchange) an. Auch erst auf Datenbank-Ebene erzeugte Belege (z.B. Rechnungslisten) werden hier einsortiert. Es kann ein Dokument zur Verarbeitungshistorie eines Datenaustauschs („Lieferschein“) im PDF- und Excel-Format heruntergeladen werden.

Ebenso ist in der Anwendung „Use Cases“ ein PDF-Export möglich (ein Excel-Export allerdings nicht). Hier werden sinnvollerweise für einen Mandanten alle Workflows und Partner im Export angezeigt. Wenn ein Benutzer auf mehrere Mandanten zugreifen darf, dann muss der Mandant über einen Filter ausgewählt werden. (Ein Export für mehrere Mandanten ist auch möglich, aber dann werden Mandantennamen und -logo nicht in die Exportdatei eingesetzt.)

Des Weiteren gibt es im SZ-Timer die Möglichkeit, erfasste Tätigkeiten nach Excel zu exportieren. Dies kann sowohl über die Anwendungen „Archiv“ und „Prüfung“ geschehen als auch in der Anwendung „AB-Nummern“ unter „Buchungen“. Für die Anwendung „AB-Nummern“ gibt es zwei eigene Templates, einmal für AB-Nummern ohne Zeitkontingent, einmal mit Zeitkontingent, und das Template mit Zeitkontingent kann ein paar zusätzliche Platzhalter enthalten.

Sollten die Platzhalter nicht mit eindeutigen Werten belegt werden können (z. B. weil vom Anwender kein Filter gesetzt wurde, der eine eindeutige Angabe ermöglicht), werden die Platzhalter mit generischen Werten wie z. B. „alle“ belegt.

Bei den Platzhaltern in den Template-Dateien handelt es sich im Falle der PDF-Templates um Formularfelder

mit einem festgelegten Namen, der gleich dem hier definierten Namen sein muss. Bei Excel-Templates handelt es sich um eine normale Zelle mit dem Inhalt \$name (also Name des Platzhalters mit vorangestelltem \$-Zeichen). Die Platzhalter bestimmen zum einen Position und Größe des Bereichs, wo die Daten eingesetzt werden, zum anderen definieren sie Zeichensatz, Schriftgröße und -farbe sowie Ausrichtung (links, Mitte, rechts).

Im Template brauchen nicht alle Platzhalter vorhanden zu sein. Fehlt ein definierter Platzhalter, werden die entsprechenden Daten auch nicht in die Ausgabedatei eingesetzt. Dies kann manchmal gewünscht sein. Es gibt keine Muss-Felder, aber ein Template ohne das `table`-Feld ist unsinnig.

Name	Beschreibung	Wo verwendet
title	In das Formular wird ein Titel eingetragen. Derzeit wird als Titel ein Text aus der properties-Datei <code>ServerTexts.properties</code> genommen.	Nur Sendungsverfolgung, PDF und Excel
table	In dieses Formularfeld wird die eigentliche Tabelle mit den Daten eingetragen. Die Tabelle nutzt nur die Größe, die das Formularfeld hat. Üblicherweise wird das Formularfeld fast die ganze Seite einnehmen. ⚠ Die Seite mit diesem Formularfeld wird so oft wie nötig kopiert, um alle Daten in der Tabelle unterbringen zu können.	Alle
table-title	Dieses Feld legt Zeichensatz, -größe und -farbe für die Titelzeile der Tabelle fest (diese kann sich vom Rest der Tabelle unterscheiden). Eine Besonderheit ist aber, dass Position und Größe dieses Feldes unwichtig sind und nicht berücksichtigt werden. Der Titel der Tabelle befindet sich <i>immer</i> im <code>table</code> -Feld, welches die ganze Tabelle definiert.	Alle PDF-Exporte
tableHeader	Für die Spaltenüberschriften der Tabelle (wird dynamisch nach rechts erweitert)	Nur Excel-Export
pagenum	In das Formular wird die Seitenzahl eingetragen. Das Formular muss sich zwingend auf der gleichen Seite wie das Formular „table“ befinden. Sollte die Seite kopiert werden, wird auch dieses Formular kopiert und die Seitenzahl wird entsprechend hochgezählt. Auf den Seiten, die keine Tabelle enthalten, wird keine Seitenzahl gesetzt, dies kann aber leicht „per Hand“ bei der Erstellung des Templates erfolgen. ⚠ Dieses Formularfeld auf „unsichtbar, aber drucken“ setzen (in Adobe Acrobat), dann erscheint in der Ausgabe kein sichtbares Formularfeld hinter der Seitennummer. (Ist ab Version 8 des EDI-Portals nicht mehr nötig.)	Alle PDF-Exporte
client	Hier kommt der Mandantennamen rein. Das Feld wird nur gefüllt, wenn die Daten auf einen Mandanten eingeschränkt sind.	Nur PDF-Export in Use-Cases
logo	Hier kommt das Mandantenlogo (eine Grafik) rein. Das Feld wird nur gefüllt, wenn die Daten auf einen Mandanten eingeschränkt sind.	Nur PDF-Export in Use-Cases

Name	Beschreibung	Wo verwendet
host-address	Das Feld bekommt den Server-Namen aus dem HTTP-Request-Header (also DNS-Name oder IP-Adresse).	PDF-Export und Excel-Export in Use-Cases und Workflow-Zusammenfassung
system-title	Hier kommt der Titel der Installation rein. Dies ist aus der Konfiguration der Wert des Schlüssels <code>system.title</code> , welcher auch als Titel des Browser-Fensters angezeigt wird. Sollte dieser Wert nicht gesetzt sein (dies ist ok, der Schlüssel ist optional), so wird auch hier der Server-Name wie im Feld <code>host-address</code> eingesetzt.	PDF-Export und Excel-Export in Use-Cases und Workflow-Zusammenfassung
workflow-id	Id des Workflows, dessen Workflow-Zusammenfassung exportiert wird	Export Workflow-Zusammenfassung
use-case	Lesbarer Name des Use-Cases, für dessen Workflow die Zusammenfassung exportiert wird	Export Workflow-Zusammenfassung
ext-process	Externe Prozessbezeichnung des Workflows, dessen Workflow-Zusammenfassung exportiert wird	Export Workflow-Zusammenfassung
user	Anwender, der die Tätigkeiten erfasst hat	Export Tätigkeiten im SZ-Timer
time-start	Startzeit, ab der die Tätigkeiten aufgelistet sind	Export Tätigkeiten im SZ-Timer
time-end	Endzeit, bis zu der die Tätigkeiten aufgelistet sind	Export Tätigkeiten im SZ-Timer
task-spec	Projekt oder Aufgabe, für das die Tätigkeiten erfasst wurden.	Export Tätigkeiten im SZ-Timer
customer	Kunde, für den die Tätigkeiten vorgenommen wurden.	Export Tätigkeiten im SZ-Timer
ticket	Ticketnummer, die den Tätigkeiten zugeordnet ist.	Export Tätigkeiten im SZ-Timer
order-number	AB-Nummer, auf die die Tätigkeiten erfasst wurden.	Export Tätigkeiten im SZ-Timer
com-date	Kaufmännisches Erstellungsdatum der AB-Nummer, für die die Tätigkeiten erfasst wurden	Export Tätigkeiten im SZ-Timer
ctg-amount	Betrag des Zeitkontingents der zugehörigen AB-Nummer	Export Tätigkeiten im SZ-Timer, nur wenn AB-Nummer mit Kontingent zugeordnet
ctg-used	Genutzter Betrag des Zeitkontingents	Export Tätigkeiten im SZ-Timer, nur wenn AB-Nummer mit Kontingent zugeordnet
ctg-avail	Restbetrag des Zeitkontingents, 0 wenn überbucht	Export Tätigkeiten im SZ-Timer, nur wenn AB-Nummer mit Kontingent zugeordnet
ctg-overbooking	Überbuchter Betrag des Zeitkontingents, 0 wenn keine Überbuchung vorliegt	Export Tätigkeiten im SZ-Timer, nur wenn AB-Nummer mit Kontingent zugeordnet

Das mitgelieferte PDF-Template hat in der Sendungsverfolgung den Dateinamen `PdfTemplate.pdf`, in den Use-Cases den Dateinamen `ClientUseCaseTemplate.pdf`. Es können weitere Templates angelegt werden, die dann verwendet werden, wenn in den Tabellen im Dokument eine bestimmte Anzahl Spalten erreicht bzw. überschritten wurde. Auf diese Weise kann verhindert werden, dass die einzelnen Spalten zu schmal werden. Die Anzahl der Spalten, ab der ein bestimmtes Template verwendet wird, wird im Dateinamen vor der Dateinamenserweiterung eingetragen; z.B. würde in der Sendungsverfolgung dieses Template ab 7 Spalten verwendet werden: `PdfTemplate.7.pdf`.

Für Sendungsverfolgung: Die Spalten, die in der PDF-Ausgabe angezeigt werden, sind nicht die Spalten, die auf der GUI angezeigt werden, sondern es werden immer alle diejenigen Spalten der Tabelle `atStddatFile00` angezeigt, die in `atTableColumnDef_s00` festgelegt wurden. D.h. Die Auswahl der Spalten im PDF ist durch den Benutzer nicht dynamisch veränderbar; der Benutzer kann nur die am Bildschirm anzuzeigenden Spalten auswählen.

Für Use-Cases: Die Spalten, die in der PDF-Ausgabe angezeigt werden, ist im Programm hinterlegt und kann nicht beeinflusst werden.

In der Excel-Ausgabe werden folgende Platzhalter verwendet:

- `$title` für den Titel der Ausgabe (analog zu PDF)
- `$tableHeader` für die Spaltenüberschriften der Tabelle (wird dynamisch nach rechts erweitert)
- `$table` für den Tabelleninhalt (wird dynamisch nach rechts und nach unten erweitert)

Erstellen Sie eine XLS-Datei (97-2003). Tragen Sie die Platzhalter als gewöhnlicher Text an den gewünschten Stellen und mit der gewünschten Formatierung ein. Speichern Sie die Datei unter dem Namen „XlsTemplate.xls“.

Die Templates müssen im Projekt im Verzeichnis „src/de/softzoll/ediPortal/public/resources/templates“ abgelegt werden, in der Installation im Verzeichnis „war/ediportal/resources/templates/“. Üblicherweise werden die Template-Dateien vom GWT-Compiler automatisch in dieses Verzeichnis kopiert. Alternativ kann in der Konfiguration des EDI-Portals der Name einer Template-Datei, die dann außerhalb der Installation liegen kann, angegeben werden.

Demo-System

Das „Demo-System“ ist ein dediziertes System mit einer speziellen Erweiterung für „Demonstrationen“, d.h. einem Modul zur Vorführung der Funktionen der Softzoll-Lösung.

Dateiauswahl

In der EDI-Konfigurationsdatei werden für das Demosystem Verzeichnisse definiert, in denen Dateien der Präsentationen abgelegt werden können.

Auszug aus der Konfiguration auf dem Testsystem:

```
# base for files for the demo
base.demo = C:/Softzoll/demo
# import directory to start the processing
base.demo_import = E:/AUSTAUSCH/in
# resources for presentations
base.demo_resources = C:/Softzoll/demo-resources
```

- `base.demo` ist das Basisverzeichnis, aus dem während der Demo Dateien ausgelesen werden können
- `base.demo_import` ist das Verzeichnis, in das die Präsentation die ausgewählte Datei kopiert, damit sie verarbeitet wird
- `base.demo_resources` ist das Basisverzeichnis, in dem Bilder, die in der Diaschau gezeigt werden, abgelegt werden.

Sämtliche Pfadangaben in der Konfiguration verstehen sich relativ zu diesem Basisverzeichnis. Es ist nicht möglich, auf Dateien außerhalb dieser Basisverzeichnisse zuzugreifen. Grund für diese Beschränkung ist, dass dem Ersteller einer Demo kein Kompletzzugriff auf den Rechner gegeben werden soll.

Verwandte Themen

- [Dedizierte Systeme](#)

¹⁾

Oracle Java ist von Version 9 bis einschließlich Version 16 für kommerzielle Anwendungen kostenpflichtig.

²⁾

Selbst erstellte Zertifikate haben den Nachteil, dass sie vom Anwender beim ersten Besuch der Seite einmalig im Browser installiert werden müssen.