Fachhochschule Nordwestschweiz
Hochschule für Technik

# Applikationssicherheit

15. Dezember 2013

Laborübung 2

---

## Inhaltsverzeichnis



Thomas Baumann & Egemen Kaba

# 1 Ressourcen

**JavaServerPages**
In der View-Schicht wurden JavaServerPages verwendet. In JavaServerPages kann Java-Code ausgeführt werden. Dies wurde vor allem benutzt, um eine Liste von Fehler- bzw. Erfolgsmeldungen anzuzeigen.
Eingabewerte in der View werden als Parameter im HttpServletRequest gespeichert und an das Servlet geschickt.
Auf dem Rückweg werden Informationen vom Servlet ebenfalls im HttpServletRequest als Attribut gespeichert, um diese in der JSP anzuzeigen.

**config.properties**
In diesem File werden Konfigurationen für die Mail-Adresse des Absenders, für das Mail-Template sowie für den Zugang zu der Datenbank aufgelistet.

**web.xml**
Da der Servlet als WebServlet annotiert ist, wird hier bloss das Welcome-File definiert, sowie ein Security Constraint, das Tomcat angibt wo er auf HTTPS umschalten soll.

**database.sql**
Mit diesem File kann die Datenbank für die Applikation aufgesetzt werden.

**https-einrichten.txt**
In diesem File wird beschrieben wie https auf tomcat aufgesetzt werden kann.

**build.xml**
Dieses Ant Script soll das Deployment auf Tomcat automatisieren.

# 2 Klassen

Implementierungsdetails siehe Sourcecode.

**RattleBitsServlet**
Diese Klasse erbt von HttpServlet, ist somit die Schnittstelle der Applikation zur Aussenwelt. In diesem Servlet wird im Konstruktor der Controller, der MailHelper sowie die Datenverbindung aufgesetzt. Hier wird ebenfalls auf GET- und POST-Anfragen reagiert. Beim Zerstören des Servlets wird ebenfalls darauf geachtet, dass abhängige Objekte sauber aufgeräumt werden.

**Controller**
Der Controller verarbeitet Anfragen, die an den Servlet gesendet wurden. Dabei prüfen die Methoden jeweils als erstes, ob der Benutzer bereits eingeloggt ist. Das Resultat dieser Anfrage wird benutzt, um festzustellen, ob der Benutzer die gewünschte Aktion durchführen kann, oder auf eine andere Seite weitergeleitet werden soll.
Da diverse Aktionen einen Zugriff auf die Datenbank voraussetzen, wird im Konstruktor ein Data Access Object, das CompanyDAO, mit der übergebenen Connection instanziert.
Das Weiterleiten auf Seiten, wird über den RequestDispatcher geregelt. Die Strings, welche Standorte der Views angeben sind dabei als Konstanten definiert.
Im Controller wird ebenfalls die Koordination zwischen diversen Komponenten, wie diversen Helpern und Utilities, mit der Applikation vorgenommen.

**CompanyDAO**
Das CompanyDAO ist die Schnittstelle der Applikation zur Datenbank. im Konstruktor wird deshalb auch ein Connection-Objekt übergeben, in dem Informationen enthalten sind, die für die Verbindung zur Datenbank notwendig sind.
In dieser Klasse werden einige Methoden bereit gestellt, welche Objetke des Models in der Datenbank persistieren, oder mit Informationen aus der Datenbank instanzieren und zurückgeben sollen.
Die Abfragen zur Datenbank werden dabei mittels PreparedStatements ausgeführt.

**Company**
Diese Klasse stellt das Model dar. Diese Klasse enthält Informationen, die aus der Datenbank stammen, oder in der Datenbank persistiert werden sollen. Sie bietet zudem Methoden an, um die enthaltenen Daten per RegEx zu validieren.

**MailHelper**
Im MailHelper werden die Properties in config.properties ausgelesen, um eine Message zu präparieren und zu senden.

**Utility**
In der Utility werden zentral statische Methoden angeboten, die an mehreren Stellen in der Applikation verwendet werden. Diese sind im Detail eine Methode zum generieren eines randomisierten Strings gemäss der Passwort-Validations-Methoden sowie eine Methode, um Strings mittels SHA-256 zu hashen.

**Verifiers**
Die Methoden dieser Klasse, haben die Aufgabe eine PLZ zu verifizieren. Damit die Verifikation nicht auf einen einzelnen Webdienst gestützt ist, wird zuerst versucht auf post.ch zu verifizieren. Schlägt diese Verifikation aufgrund von Verbindungsproblemen fehl, wird die PLZ auf postleitzahlen.ch verifiziert.
Bei beiden Varianten kann die PLZ als Parameter in der URL angegeben werden. Die Ant-

wort ist eine HTML-Page. Diese wird auf eine Meldung überprüft, welche darauf hindeutet, dass der Webdienst diese PLZ nicht gefunden hat.