

Entwicklerhandbuch

BWS-Sport-Choice

Lichtenfels, Lisa

Auszubildende zur Fachinformatikerin für Anwendungsentwicklung

Gutberlet, Joshua

Auszubildender zum Fachinformatiker für Anwendungsentwicklung

30.11.2018

Bundeskriminalamt Wiesbaden

## Inhaltsverzeichnis

1 Vorwort.....	3
2 Bibliotheken.....	3
3 Pakete.....	3
3.1 .ejb.....	4
3.1.1 de.bws.ctls.....	4
3.1.2 de.bws.data.....	4
3.1.3 de.bws.entities.....	4
3.1.4 de.bws.security.....	4
3.1.5 de.bws.sessionbeans.....	4
3.2 .war.....	4
3.2.1 de.bws.converter.....	4
3.2.2 de.bws.filter.....	4
3.2.3 de.bws.namedBeans.....	4
4 Klassendiagramme.....	4

## Abbildungsverzeichnis

Abbildung 1: Paketdiagramm.....	4
---------------------------------	---

# 1 Vorwort

Das vorliegende Dokument soll Ihnen bei der Weiterentwicklung oder Modifikation der Anwendung „BWS-Sport-Choice“ weiterhelfen.

Die Anwendung ist in der Programmiersprache Java implementiert. Die Methoden und Variablen der Klassen werden in den einzelnen Dateien ausführlich durch Kommentare erklärt. Nachfolgend wird Ihnen eine Übersicht über die benötigten Bibliotheken sowie die Pakete und deren Klassen der Anwendung gegeben.

## 2 Bibliotheken

Um diese Webanwendung auf einem Server bereitzustellen, werden neben dem Applikationsserver GlassFish Open Source Edition (OSE) Version 4.1.2 weitere Bibliotheken benötigt.

### JSF 2.0

Das Frontend wurde komplett mit Java Server Faces 2.0 realisiert. Der GlassFish 4.1.2 beinhaltet bereits das JSF 2.0.

### PrimeFaces 5.0

PrimeFaces ist ein Komponentenframework, welches Implementationen aus JSF 2.0 erweitert. Es vereinfacht unter Anderem dessen AJAX-Funktionalität, um die Anwendung benutzerfreundlicher zu gestalten.

### Foundation 6.4.2

Foundation ist ein freies Webframework des Unternehmens ZURB. Es wurde unter Anderem für das Design der Webanwendung ESTA benutzt, auf dem diese Webanwendung basiert. Es enthält Designvorlagen für Hypertext Markup Language (HTML) und Cascading Style Sheets (CSS) wodurch das Designen von Webseiten vereinfacht wird und vorgefertigte CSS-Klassen verwendet werden können.

## 3 Pakete

Die Anwendung ist nach dem Architekturmuster Model-View-Controller entwickelt worden. Zur besseren Trennung der Programmfunktionen sind daher die Klassen in Pakete unterteilt.

Zudem gebietet die Entwicklung in Java EE eine weitere Einteilung in ein Web-Modul, für die View-Komponenten sowie deren verwaltende Controller-Klassen, und ein EJB-Modul, für das Model aus Entity-Klassen sowie deren verwaltende Facades.

Das folgende Paketdiagramm veranschaulicht diese Strukturierung noch einmal.

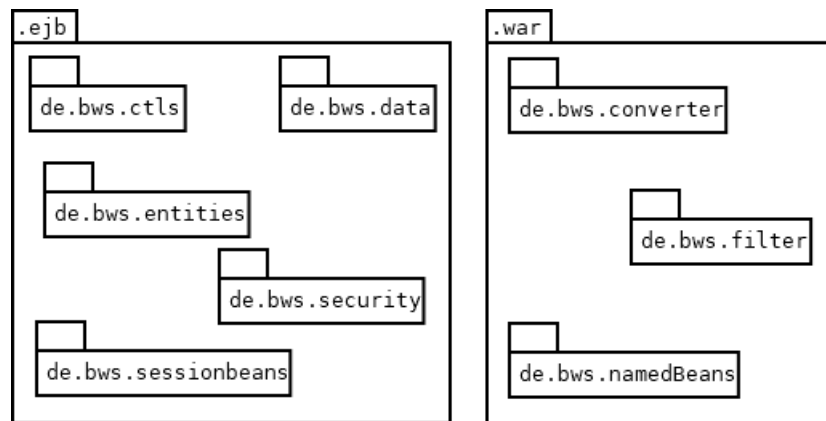


Abbildung 1: Paketdiagramm

Nachfolgend eine genauere Übersicht über die Pakete und deren Klassen.

### 3.1 .ejb

#### 3.1.1 de.bws.ctls

Die zu diesem Paket gehörende Klassen dienen als verwaltende Klassen der Entitäten des Model.

#### **BenutzerJpaController**

Diese Klasse verfügt über den sogenannten „Entity Manager“ der Datenbankzugriffe ermöglicht, um Entitäten vom Typ Benutzer zu erstellen, bearbeiten, löschen und finden.

#### **KursJpaController**

Diese Klasse verfügt über den sogenannten „Entity Manager“ der Datenbankzugriffe ermöglicht, um Entitäten vom Typ Kurs zu erstellen, bearbeiten, löschen und finden.

#### **PersonJpaController**

Diese Klasse verfügt über den sogenannten „Entity Manager“ der Datenbankzugriffe ermöglicht, um Entitäten vom Typ Person zu erstellen, bearbeiten, löschen und finden.

### **LehrerJpaController**

Diese Klasse verfügt über den sogenannten „Entity Manager“ der Datenbankzugriffe ermöglicht, um Entitäten vom Typ Lehrer zu erstellen, bearbeiten, löschen und finden.

### **SchuelerJpaController**

Diese Klasse verfügt über den sogenannten „Entity Manager“ der Datenbankzugriffe ermöglicht, um Entitäten vom Typ Schüler zu erstellen, bearbeiten, löschen und finden.

### **StufeJpaController**

Diese Klasse verfügt über den sogenannten „Entity Manager“ der Datenbankzugriffe ermöglicht, um Entitäten vom Typ Stufe zu erstellen, bearbeiten, löschen und finden.

### **ThemaJpaController**

Diese Klasse verfügt über den sogenannten „Entity Manager“ der Datenbankzugriffe ermöglicht, um Entitäten vom Typ Thema zu erstellen, bearbeiten, löschen und finden.

### **WahlJpaController**

Diese Klasse verfügt über den sogenannten „Entity Manager“ der Datenbankzugriffe ermöglicht, um Entitäten vom Typ Wahl zu erstellen, bearbeiten, löschen und finden.

### **WahlzeitraumJpaController**

Diese Klasse verfügt über den sogenannten „Entity Manager“ der Datenbankzugriffe ermöglicht, um Entitäten vom Typ Wahlzeitraum zu erstellen, bearbeiten, löschen und finden.

### **3.1.2 de.bws.data**

Dieses Paket verfügt über weitere, zum „Model“ gehörende, Klassen, die für die Webanwendung benötigt werden, jedoch keine Entitäten der Datenbank darstellen.

### **Rolle**

Bei der Rolle handelt es sich um eine Enumeration. Enumerationen sind Aufzählungstypen mit einem endlichen Wertebereich. Deren Werte sind Konstanten, welche eine implizite Aufzählungsfolge beinhalten. Diese kann nicht verändert werden, so dass man die Werte als Integer in der Datenbank abspeichern kann.

Die Klasse beinhaltet folgende drei Rollen: „ADMIN“, „LEHRER“ und „SCHUELER“

### **Eintrag**

Diese Klasse verwendet parametrisierte Typen und soll dazu dienen einem „Schlüssel“ einen Wert zuzuweisen. Als „Schlüssel“ werden die Entitäten der Datenbank genutzt, denen man dann Fallabhängig einen Wert zuweisen kann, der nicht zu deren Attributen gehört.

### **3.1.3 de.bws.entities**

Die zum „Model“ gehörigen Klassen des MVC-Musters dienen der Beschreibung der im Datenbankmodell aufgeführten Entitäten mit ihren Attributen und Methoden.

#### **Benutzer**

Jeder Benutzer verfügt über eine eindeutige ID, einen Benutzernamen, ein Passwort, ein Salt zur Verschlüsselung des Passworts, eine Rolle und eine Person.

#### **Kurs**

Ein Kurs verfügt über eine eindeutige ID, einen Titel, ein Kürzel, eine Beschreibung, eine Bewertung, einen Hinweis, eine maximale Teilnehmerzahl, ein Jahr, einen themengleichen Kurs, eine Stufe, einen Lehrer, Unterthemen und Schüler als Teilnehmer des Kurses.

#### **Person**

Für eine Person wird eine eindeutige ID, ein Vorname und ein Nachname benötigt.

#### **Lehrer**

Jeder Lehrer verfügt über ein Kürzel. Diese Entität erbt außerdem von einer Person.

#### **Schueler**

Jeder Schüler verfügt über einen Tutor bzw. Lehrer und eine Stufe, sowie eine Wahl. Diese Entität erbt ebenfalls von einer Person.

#### **Stufe**

Eine Stufe verfügt über eine eindeutige ID und eine Bezeichnung.

#### **Thema**

Die Attribute eines Themas sind eine eindeutige ID, ein Anteil, eine Bezeichnung und ein Schwerpunkt.

#### **Wahl**

Die Entität Wahl beinhaltet eine eindeutige ID sowie eine Erst-, Zweit- und Drittwahl jeweils vom Typ Kurs.

### **Wahlzeitraum**

Ein Wahlzeitraum verfügt über eine eindeutige ID, ein Beginndatum und ein Enddatum.

### **3.1.4 de.bws.security**

Dieses Paket beinhaltet Klassen, die für die Sicherheit der Webanwendung stehen.

### **Passwort**

Diese Klasse stellt die Generierung und Verschlüsselung eines Passwortes zur Verfügung.

### **3.1.5 de.bws.sessionbeans**

In diesem Paket sind Facades zu finden, die als Schnittstelle für die Managed Beans des Web-Moduls dienen um auf die Datenbankverwaltungsklassen zugreifen zu können.

### **BenutzerFacadeLocal & BenutzerFacade**

Die Klasse BenutzerFacade implementiert das Interface BenutzerFacadeLocal und dient als Schnittstelle zum BenutzerJpaController.

### **KursFacadeLocal & KursFacade**

Die Klasse KursFacade implementiert das Interface KursFacadeLocal und dient als Schnittstelle zum KursJpaController.

### **PersonFacadeLocal & PersonFacade**

Die Klasse PersonFacade implementiert das Interface PersonFacadeLocal und dient als Schnittstelle zum PersonJpaController.

### **LehrerFacadeLocal & LehrerFacade**

Die Klasse LehrerFacade implementiert das Interface LehrerFacadeLocal und dient als Schnittstelle zum LehrerJpaController.

### **StufeFacadeLocal & StufeFacade**

Die Klasse StufeFacade implementiert das Interface StufeFacade Local und dient als Schnittstelle zum StufeJpaController.

### **ThemaFacadeLocal & ThemaFacade**

Die Klasse ThemaFacade implementiert das Interface ThemaFacadeLocal und dient als Schnittstelle zum ThemaJpaController.

### **WahlFacadeLocal & WahlFacade**

Die Klasse WahlFacade implementiert das Interface WahlFacadeLocal und dient als Schnittstelle zum WahlJpaController.

### **WahlzeitraumFacadeLocal & WahlzeitraumFacade**

Die Klasse WahlzeitraumFacade implementiert das Interface WahlzeitraumFacadeLocal und dient als Schnittstelle zum WahlzeitraumJpaController.

## **3.2 .war**

### **3.2.1 de.bws.converter**

In diesem Paket befinden sich Konverter-Klassen, die von JSF benutzt werden um Eingaben zur Verwendung in den Entitäten aufzubereiten.

#### **DateConverter**

Dieser Konverter bereitet nach der Eingabe eines Datums dies in ein für die Datenbank gültiges Format um oder wirft eine Exception, wenn die Eingabe nicht valide ist.

#### **StufeConverter**

Dieser Konverter wird für die Aufbereitung einer angegebenen Stufe verwendet.

### **3.2.2 de.bws.filter**

#### **LoginFilter**

Diese Klasse dient dazu, dass bestimmte Seiten der Webanwendung ohne eine vorhandene, gültige Session nicht zu erreichen sind. Das heißt, dass der Benutzer sich einloggen muss, um auf diese Seiten navigieren zu können und dass er, sobald seine Session ausgelaufen ist, wieder zur Login-Seite navigiert wird, anstatt auf „sichere“ Seiten.

### **3.2.3 de.bws.namedBeans**



Das Paket beinhaltet die Managed Beans, die Verwalterklassen des EJB-Moduls. Diese Klassen stellen die Interaktion zwischen Oberfläche und den im Hintergrund ablaufenden Programmfunktionen her.

### **BenutzerAendernNB**

Stellt die Funktionalität für das Bearbeiten bzw. Ändern von Benutzerdaten bereit.

### **BenutzerAnlegenNB**

Stellt die Funktionalität für das Anlegen von Benutzern bereit.

### **BenutzerNB**

### **BenutzerVerwaltenNB**

Stellt die Funktionalität für das Löschen von Benutzern bereit, sowie das Zurücksetzen von Passwörtern der Benutzer. Außerdem behält sich die Bean die Auswahl des Benutzers, der zum Bearbeiten ausgewählt wurde und gibt diese weiter an die dafür zuständige Bean.

Das Hoch- und Abstufen von Schülern wird in der Bean ebenfalls verwaltet.

### **KursNB**

Stellt die Funktionalität für das Anlegen und Bearbeiten von Kursen bereit.

### **KursZuweisenNB**

Stellt die Funktionalität für das manuelle Zuweisen von Kursen und den Kurswechsel bereit.

### **KurslisteNB**

Gibt verschiedene Listen von Kursen zurück. Zum Beispiel Kurse von bestimmten Lehrern, Kurse für eine bestimmte Stufe etc.

### **LoginNB**

Stellt die Funktionalität zum Anmelden und Ausloggen bereit.

Außerdem wird der angemeldete Benutzer in der Session gespeichert.

### **MenueNB**

Über diese Managed Bean wird das Menü der Webanwendung und die Startseite nach Anmeldung gerendert. Für Admin, Lehrer und Schüler gibt es verschiedene Menüpunkte und die Startseite zeigt ebenfalls verschiedene Inhalte für verschiedene Rollen

### **PasswortAendernNB**

Stellt die Funktionalität für das Ändern von Passwörtern bereit.

### **StartupBean**

Wird beim Deploy der Webanwendung auf dem Server aufgerufen und legt in der Datenbank den Admin-Benutzer, sowie zwei Stufen, die von Beginn an vorhanden sein sollen, an, sofern diese noch nicht vorhanden sind.

### **StufeNB**

Gibt eine Liste mit allen Stufen zurück.

### **WahlNB**

Stellt die Funktionalität für das Anlegen und Speichern der Kurswahl eines Schülers bereit.

### **WahlzeitraumNB**

Stellt die Funktionalität für das Anlegen und Ändern eines Wahlzeitraumes bereit.

## **4 Klassendiagramme**

Die Klassendiagramme werden zu ihren Paketen zusammengefasst dargestellt.