

Projektbericht

Johannes Wörndl

Horstware

Internetprogrammierung

16.01.2014

## **Inhaltsverzeichnis zum Projekt Internetprogrammierung**

<b>1.....</b>	<b>Projektvorstellung</b>	<b>3</b>
1.1.....	Allgemeine Projektvorstellung	3
1.2.....	Aufgabenverteilung	3
<b>2.....</b>	<b>Projektvorbereitung</b>	<b>3</b>
2.1.....	Einrichten des Git-Repository	3
2.2.....	Projektplanung	3
<b>3.....</b>	<b>Implementierung</b>	<b>4</b>
3.1.....	Ausarbeitung einer Vorlage	4
3.2	Ordnerstruktur für die REST-Schnittstelle	4
3.3 .....	Bekanntmachen der Klassen	4
3.4 .....	Get Request	5
3.5 .....	Post Request	6
3.6 ....	Entitätsklassen und DAO Klassen	6
<b>4.....</b>	<b>Schluß</b>	<b>7</b>

# 1. Projektvorstellung

## 1.1 Allgemeine Projektvorstellung

Im Fach Internetprogrammierung wurde über die Übungen zu diesen Fach eine Serverfunktionalität programmiert bzw. konfiguriert. Um den Server zu programmieren wurde als Plattform Java-EE Eclipse verwendet. Des Weiteren kamen Techniken wie JSF und EJB zum Einsatz. Ziel dieses Projekts war es eine Bestellplattform für Fahrzeuge auf diesen Server zu erweitern.

## 1.2 Aufgabenverteilung

Zu Beginn des Projekts wurden die jeweiligen Projekte auf die Teams verteilt. Unser Team hatte die Aufgabe eine Android Schnittstelle für den Server zu bauen. Des Weiteren soll es nun möglich sein neben Fahrzeuge erstellen und suchen einen Fahrzeugtyp auszuwählen. Meine Aufgabe in diesem Projekt war es eine REST Schnittstelle im Server zu bauen damit der Client darauf zugreifen kann.

# 2. Projektvorbereitung

## 2.1. Einrichten des Git-Repository

Um im Team arbeiten zu können benötigten wir eine Plattform um unsere Erstellten Daten auszutauschen. Hierbei verwendeten wir Git-Hub. Jedes Teammitglied erstellte einen Account. Mit diesen konnte dann auf das gemeinsame Repository zugegriffen werden.

## 2.2. Projektplanung

Als erstes erstellte ich Anwendungsfälle des Projekts erstellen und speicherte diese in einer Excelltabelle ab.

Anwendungsfall	Beschreibung
Anlegen	Als Nutzer eine Android Handys möchte ich in meiner App neue Bestellungen für Autos anlegen
Autosuche	Als Nutzer eine Android Handys möchte ich in meiner App nach meinen Bestellungen für Autos suchen
Anzeigen	Als Nutzer eine Android Handys möchte ich in meiner App meine Bestellungen anzeigen lassen
Eingeben	Als Nutzer eine Android Handys möchte ich in meiner App minen Vornamen, Nachnamen und die Menge derzu bestellenden Autos eingeben
Typfestlegung	Als Nutzer eine Android Handys möchte ich in meiner App den Fahrzeugtyp meiner Bestellung auswählen
Modellsuche	Als Nutzer eine Android Handys möchte ich in meiner App nach den Modellen für meine bestellten Autos suchen
Kundensuche	Als Nutzer eine Android Handys möchte ich in meiner App die Kunden für die Bestellungen anzeigen lassen

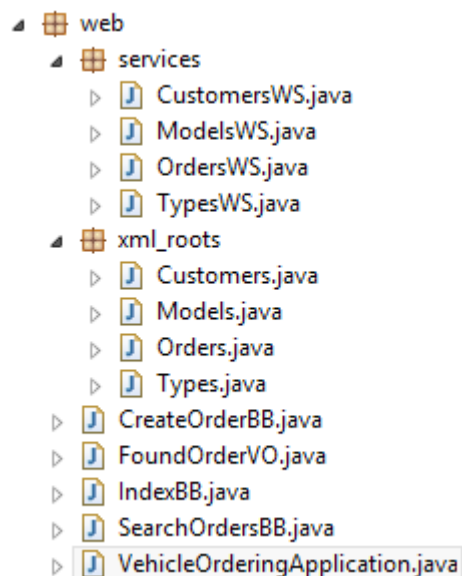
### 3. Implementierung

#### 3.1 Ausarbeitung einer Vorlage

Um den anderen Teammitgliedern eine Vorlage für die http Requests zu geben erstellte Herr Moschall eine Tabelle mit sämtlichen http Request welche für den Server implementiert werden soll.

A	B	C	D	E	F	G
ID	Methode	URL	Produces	Consumes	Beschreibung	
1	GET	vehicle-ordering/services/types	text/xml	-	Liste aller Typen	
2	GET	vehicle-ordering/services/types/id	text/xml	-	Typ mit entsprechender ID laden	
3	GET	vehicle-ordering/services/models?type=id	text/xml	-	Liste aller Modelle; optional nach Typ gefiltert	
4	GET	vehicle-ordering/services/models/id	text/xml	-	Model mit entsprechender ID laden	
5	GET	vehicle-ordering/services/customers?first-name&last-name	text/xml	-	Liste aller Kunden; optional nach Vor- und Nachname gefiltert	
6	GET	vehicle-ordering/services/customers/id	text/xml	-	Kunde mit entsprechender ID laden	
7	GET	vehicle-ordering/services/orders?first-name&last-name&model=id	text/xml	-	Liste aller Bestellungen; optional nach Kundenvor- und nachname sowie Model gefiltert	
8	GET	vehicle-ordering/services/orders/id	text/xml	-	Bestellung mit entsprechender ID laden	
9	POST	vehicle-ordering/services/customers	-	application/xml	Anlegen eines Kunden; XML muss der aus Schnittstelle 6 entsprechen	
10	POST	vehicle-ordering/services/orders	-	application/xml	Anlegen einer Bestellung; XML muss der aus Schnittstelle 8 entsprechen	

#### 3.2 Ordnerstruktur für die REST-Schnittstelle



#### 3.3 Bekanntmachen der Klassen

Um die dementsprechenden Klassen für eine Rest-Implementierung bekannt zu machen müssen diese in der java Klasse „VehicleOrderingApplication“ das ein Element

von Klassen in einer Liste bekannt zu machen von wo aus man mit dem Pfad „vehicle-ordering/services“ zugreifen kann.

```
package com.nttdata.emea.devschool.vehicleordering.web;

import java.util.HashSet;

@ApplicationPath("/services")
public class VehicleOrderingApplication extends Application
{
    @Override
    public Set<Class<?>> getClasses ()
    {
        Set<Class<?>> classes = new HashSet<Class<?>>();

        classes.add(TypesWS.class);
        classes.add(ModelsWS.class);
        classes.add(CustomersWS.class);
        classes.add(OrdersWS.class);

        return classes;
    }
}
```

### 3.4 Get Request

Um Elemente oder ganze Listen(siehe Tabelle, S.5) zu laden werden vom Client GET-Methoden gesendet. Der Server sendet dem Client ein XML- Dokument welche die gewünschten anfragen enthalten.

```
import java.util.ArrayList;

@Stateless
@Path("/types")
public class TypesWS
{
    @EJB private TypeDAO dao;

    @GET
    @Produces(MediaType.TEXT_XML)
    public Types getTypes ()
    {
        Types types = new Types();
        types.type = dao.retrieveAll();
        return types;
    }

    @GET
    @Path("/{id}")
    @Produces(MediaType.TEXT_XML)
    public Types getType (@PathParam("id") String id)
    {
        Type type = dao.findById(Long.valueOf(id));
    }
}
```

Beispiel: Mit dem Aufruf „vehicle-ordering/services/types“ lässt sich ein XML-Dokument aufrufen. Dieses beinhaltet alle Fahrzeugtypen. Der Aufruf „vehicle-ordering/services/types/4“ beinhaltet nur den Fahrzeugtyp mit der ID 4

### 3.5 Post Request

Um Elemente oder ganze Listen(siehe Tabelle, S.5) zu senden werden vom Client POST-Methoden gesendet. Der Server erhält von dem Client ein XML- Dokument welche die gewünschten Neuimplementierungen enthalten.

```
@POST
@Consumes(MediaType.APPLICATION_XML)
public void postOrders (Orders orders)
{
    for(Order o : orders.order)
    {
        Order newOrder = new Order(o.getCustomer(), o.getModel(), o.getAmount(), o.getDeliveryDate());
        dao.persist(newOrder);
    }
}
```

Beispiel: Um eine neue Bestellung aufzunehmen wird über den Html Request Post über den Aufruf des Pfads „vehicle-ordering/services/orders“ ein xml-Dokument versendet. Der Server erkennt die Elemente als Order über das XML-Rootelement Orders. Im Entity-manager werden diese Elemente dann in die Datenbank eingetragen. Dort werden die fehlenden Einträge in die Datenbank eingetragen.

### 3.6 Entitätsklassen und DAO Klassen

Über Entitätsklassen werden die jeweiligen Attribute vom Entity-manager unter Hilfenahme von Setter und Getter in die Datenbank gespeichert. Über DAO-Klassen werden die notwendige Datenbankabfragen implementiert.

```

import java.util.List;

@Stateless
public class TypeDAO extends AbstractDAO<Type>
{
    public List<Type> retrieveAll ()
    {
        String jpql = "SELECT t FROM Type AS t";
        Query query = entityManager.createQuery(jpql);
        @SuppressWarnings("unchecked")
        List<Type> result = query.getResultList();
        return result;
    }

    public Type findById (Long id)
    {
        String jpql =
            "SELECT t " +
            "FROM Type " +
            "AS t " +
            "WHERE t.id=?1";

        Query query = entityManager.createQuery(jpql);
        query.setParameter(1, id);

        @SuppressWarnings("unchecked")
        List<Type> result = query.getResultList();
        if(!result.isEmpty())
        {
            return result.get(0);
        }
        else
        {
            return null;
        }
    }
}

```

Beispiel TypeDAO: über die Methode retrieveAll wird eine Liste von allen Typen abgefragt. Über die Methode findById wird ein bestimmter Typ gesucht.

## 4. Abschluß

Abschließend wird noch die Präsentation noch vorbereitet. Ich werde noch meine Aufgaben dieses Projekts dieser Präsentation hinzufügen