

---

# Laborprotokoll

## Distributed Computing GK9.3

### "Cloud-Datenmanagement"

---

Systemtechnik Labor  
5BHIT 2017/18

Nicolaus Rotter

**Note:**  
**Betreuer: BORM**

**Version 0.2**  
**Begonnen am 5. Oktober 2017**  
**Beendet am 5. Oktober 2017**

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Ziele . . . . .	1
1.2	Voraussetzungen . . . . .	1
1.3	Aufgabenstellung . . . . .	1
<b>2</b>	<b>Ergebnisse</b>	<b>2</b>
2.1	Django-REST-Framework Tutorial . . . . .	2
2.1.1	1: Proejtk Setup . . . . .	2
2.1.2	Sync Database und Superuser . . . . .	2
2.2	Tabelle . . . . .	3
2.3	Aufzählung . . . . .	3
2.4	Code . . . . .	3

# 1 Einführung

Diese Übung zeigt die Anwendung von mobilen Diensten.

## 1.1 Ziele

Das Ziel dieser Übung ist eine Webanbindung zur Benutzeranmeldung umzusetzen. Dabei soll sich ein Benutzer registrieren und am System anmelden können.

Die Kommunikation zwischen Client und Service soll mit Hilfe einer REST Schnittstelle umgesetzt werden.

## 1.2 Voraussetzungen

- Grundlagen einer höheren Programmiersprache
- Verständnis über relationale Datenbanken und dessen Anbindung mittels ODBC oder ORM-Frameworks
- Verständnis von Restful Webservices

## 1.3 Aufgabenstellung

Es ist ein Webservice zu implementieren, welches eine einfache Benutzerverwaltung implementiert. Dabei soll die Webapplikation mit den Endpunkten /register und /login erreichbar sein.

Registrierung Diese soll mit einem Namen, einer eMail-Adresse als BenutzerID und einem Passwort erfolgen. Dabei soll noch auf keine besonderen Sicherheitsmerkmale Wert gelegt werden. Bei einer erfolgreichen Registrierung (alle Elemente entsprechend eingegeben) wird der Benutzer in eine Datenbanktabelle abgelegt.

Login Der Benutzer soll sich mit seiner ID und seinem Passwort entsprechend authentifizieren können. Bei einem erfolgreichen Login soll eine einfache Willkommensnachricht angezeigt werden.

Die erfolgreiche Implementierung soll mit entsprechenden Testfällen (Acceptance-Tests bez. aller funktionaler Anforderungen mittels Unit-Tests) dokumentiert werden. Verwenden Sie auf jeden Fall ein gängiges Build-Management-Tool (z.B. Maven oder Gradle). Dabei ist zu beachten, dass ein einfaches Deployment möglich ist (auch Datenbank mit z.B. file-based DBMS).

## 2 Ergebnisse

### 2.1 Django-REST-Framework Tutorial

Ich habe mit zur Bearbeitung dieser Übung die Django-REST-Schnittstelle ausgesucht.

#### 2.1.1 1: Proejtk Setup

Zuerst muss ein neues Django Projekt erstellt und die Quickstart app aufgerufen werden. Das Tutorial gibt hierbei vor, wie man über die Commandline ein Projekt erstellt und weitere Schritte. Diese haben jedoch nicht funktioniert, da das Command zum Erstellen eines neuen Projekts : **django-admin.py startproject tutorial** . keine Auswirkung hatte.

Lösung: Also wurde händisch über PyCharm ein neues Django Projekt angelegt und die Ordnerstruktur später manuell erweitert.

#### 2.1.2 Sync Database und Superuser

Anschließend wird die Database über das Command **python manage.py migrate** zum ersten mal Angeglichen und ein neuer Superuser wird mit **python manage.py migrate** angelegt.

Dieser hat den Username **admin** und das Passwort **password123**.

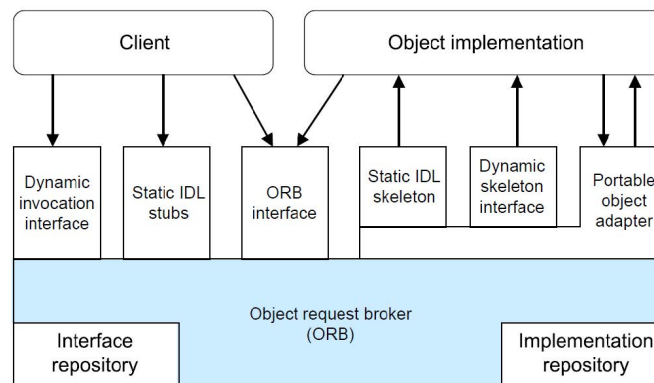


Abbildung 1: Common Object-Request-Broker Architecture [1]

Hier sollen die Schritte der Laborübung erläutert werden. Alle Fragestellungen der Lehrkraft müssen hier beantwortet werden. Etwaige Probleme bzw. Schwierigkeiten sollten ebenfalls hier angeführt werden.

Es kann gut möglich sein, dass Lehrkräfte hier auch noch andere Eckpunkte explizit verlangen. Diese können dann in der selben Hierarchiestufe wie die *Ergebnisse* eingeordnet werden. Viel Spass nun mit einer kleinen Übersicht von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Elementen.

Header	Kopf
<b>Lorem</b>	Ipsum dolor sit amet, consetetur sadipscing elitr
<b>Ipsum</b>	At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus
<b>Dolor</b>	Consetetur sadipscing elitr, sed diam nonumy

Tabelle 1: Lorem ipsum dolor sit amet [1]

## 2.2 Tabelle

## 2.3 Aufzählung

- **Lorem ipsum:** dolor sit amet, consetetur sadipscing elitr
- sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat
- ut labore et dolore magna aliquyam erat, sed diam voluptua

## 2.4 Code

At vero eos et accusam et justo duo dolores et ea rebum.

```

1  try{
    gTransCur.begin();
3   //Perform the operation inside the transaction
    not_registered =
5     gRegistrarObjRef.register_for_courses(student_id,selected_course_numbers);

7

9     if (not_registered != null)

11        //If operation executes with no errors, commit the transaction
        boolean report_heuristics = true;
        gTransCur.commit(report_heuristics);
13    } else gTransCur.rollback();
15

17 } catch(org.omg.CosTransactions.NoTransaction nte) {
    System.err.println("NoTransaction: " + nte);
19    System.exit(1);
    } catch(org.omg.CosTransactions.SubtransactionsUnavailable e) {
21    System.err.println("Subtransactions Unavailable: " + e);
    System.exit(1);
23 } catch(org.omg.CosTransactions.HeuristicHazard e) {
    System.err.println("HeuristicHazard: " + e);
25    System.exit(1);
    } catch(org.omg.CosTransactions.HeuristicMixed e) {
27    System.err.println("HeuristicMixed: " + e);
    System.exit(1);
29 }

```

Listing 1: Implizite Transaktion [1]

## Literatur

- [1] A.S. Tanenbaum and M. Van Steen. *Verteilte Systeme: Prinzipien und Paradigmen*. Pearson Studium. Addison Wesley Verlag, 2007.

## Tabellenverzeichnis

1	Lorem ipsum dolor sit amet [1] . . . . .	3
---	--	---

## Listings

1	Implizite Transaktion [1] . . . . .	3
---	-------------------------------------	---

## Abbildungsverzeichnis

1	Common Object-Request-Broker Architecture [1] . . . . .	2
---	---	---