

5. Unit: Web Services

Exercise 5.1 (Web Services: tomcat)

Install tomcat. You can do this on your own computer, and/or in the CIP Pool in your own account. Starting it makes it available under the URL `localhost:8080` from the browser.

Copy the `.war` file for the XQuery Demo Web Service into `tomcat/webapps` and adapt the configuration (and restart tomcat again) to check whether it basically works.

Exercise 5.2 (Web Services: Stream Processing)

Write a Web Service that reads an XML document from the HTTP input (request) and returns (into the response stream) an XML document that has the same structure, but all element and attribute names are backwards. E.g., for

```
<country car_code='D'>
  <name>Germany</name><population year="2011">80219695</population>
</country>
```

the response is

```
<yrtnuoc edoc_rac='D'>
  <eman>Germany</eman><noitalupop raey="2011">80219695</noitalupop>
</yrtnuoc>
```

It should do this by streaming – i.e. whenever something is read, the result is immediately added to the answer. Log every action to the `System.out` to illustrate the proceeding.

Write a Java program that calls this Web Service, e.g., using `mondial.xml` (or any other XML data). It should also log its activity (sending and also receiving the answer) to `System.out`. For the writing of the xml to the request, consider the output method of JDOM (which will not log every step, but should be good for seeing that already during writing, the response is received), and a SAX/StAX writer.

Are there documents where the response is the same as the document?

Exercise 5.3 (Anwendung: Web-Services der Uni-Infrastruktur)

Entwickeln Sie drei Web-Services, die die Aufgabenbereiche von StudIP, FlexNow und EXA -mit dem Fokus auf den Datenaustausch- implementieren.

- EXA: Lehrendendaten, Räume, Lehrveranstaltungen, Prüfungen
 - Einfaches Benutzerinterface zum Anlegen von Lehrenden (Name, Fakultät), Lehrveranstaltungen (Stundenplanslots+Raum, Lehrender), Prüfungen (Lehrveranstaltung, Datum, Raum (oder Ilias))
 - Möglichkeit, solche Daten aus einer XML-Datei zu lesen und mit dem Datenbestand abzugleichen (bzw. nur die neuen hinzufügen).
 - Möglichkeit, die Termine einer LV noch im Semester zu ändern (und an StudIP weiterzugeben)
 - Zu jeder LV ablegen, in welchen Studiengängen die LV anrechenbar ist; (Modulkatalog optional, das ist viel Struktur- und Benutzerinterface-Kleinkram).
 - Optional: Modulbeschreibung zu jeder einzelnen LV wie im Modulkatalog als Dokumentverwaltungs-Beispiel. Dann könnte man z.B. für jedes Zeugnis ein PDF mit den absolvierten Modulbeschreibungen erzeugen.

- Anzeigen des Wochenplanes eines Lehrenden und des Belegungsplanes eines Raumes sowie der Basisdaten einer Lehrveranstaltung+Prüfung.
- StudIP: Studierendendaten (Name, Studiengang/gänge) werden hier verwaltet.
 - Lehrendendaten und Lehrveranstaltungen werden von EXA importiert.
 - Benutzerinterface: Studierende können sich hier zu LVs anmelden.
 - Anzeigen des Wochenplanes eines Studierenden
- FlexNow: Prüfungsorganisation und Noten.
 - Die Prüfungen werden aus StudIP übermittelt.
 - Anmeldung, Abmeldung zu Prüfungen.
 - Noteneingabe. Einzel, oder durch Upload eines XML-Dokuments aller Teilnehmer einer LV (das der Lehrende oder Ilias erstellt)
 - Zeugnis erstellen (HTML oder LateX).
 - Implementieren Sie ggf. zusätzliche Funktionalität, die das Testen/Vorführen erleichtert.
- Optional: ILIAS als Mini-Service (alle Teilnehmer dort hin übermitteln, Interface zum Erzeugen von Noten, diese an FlexNow übermitteln).
- Datenspeicherung:
 - XML-Dateien? (auf was muss man dann aufpassen?)
 - Es steht ein Oracle-Account “xml-p” zur Verfügung. Siehe SQL-Praktikum (<http://www.dbis.informatik.uni-goettingen.de/Teaching/DBP/>); “Einführung in die Benutzung von Oracle am IFI” für direkte Aufrufe mit sqlplus von den Pool-Rechnern und über JDBC (siehe Folien zu JDBC)
- FlexNow: Wenn die Anmeldung zu einer Prfung abgeschlossen ist, kann der Lehrende veranlassen, dass eine entsprechende Prüfungslehrveranstaltung mit allen angemeldeten Studierenden in StudIP angelegt wird.

Lassen Sie die Webservices auf einem oder mehreren CIP-Pool-Rechnern laufen, um miteinander kommunizieren zu können (zuhause hat man üblicherweise immer wechselnde IP-Adressen).