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: Real XML Stream Processing) Consider the use case that an XML Stream really consists of independent events (e.g. from RFID sensor inputs). Here, this is again "faked" by feeding the XML events from Mondial.

- 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 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, use a SAX/StAX writer to write individual events.
- Are there documents where the response is the same as the document?
- Real-world setting: in case it does not work like this (which is the expected case): analyze what is the problem, and try to find a solution to it – the Web can be helpful.

Exercise 5.3 (Application: (Fragments of) the Web-Services of the der university infrastructure)

Design four Web services that implement EXA/e-campus, StudIP, FlexNow and Ilias – focusing on the data exchange:

- EXA: data about lecturing persons, rooms, courses and scheduled exams
  - Simple user interface for creating lecturers (name, department), courses (rooms+times, lecturer), exams (course, date, room (or Ilias)),
  - read such bulk data from an XML file an integrate with existing data (add only the new onbes)
  - change dates and rooms during the semester (and communicate this to StudIP),
  - for each course, add the information in which studies it can be credited (optional: sketch of the module catalog, but this contains lots of structural and user interface stuff).
  - show the weekly schedule of a lecturer, the usage of a room, or all data about a course+exam.
  - optional as a use case for document management: module description for each course. Then, FlexNow could create for each final certificate a PDF with all module decriptions the student has taken.
  - optional modeling extension: define studying programs as lists of groups of mandatory/optional courses.

- StudIP: data about students (name, studies):
  - data about teaching persons and courses is imported from EXA (bulk import + updates during the semester),
  - user interface: students can register for courses,
    optional: if study programs exist, suggest/filter appropriate courses for registration,
  - show the weekly schedule of a student.
- FlexNow: exam organisation and grades:
  - Scheduled exams are communicated from EXA.
  - registration and deregistration for exam.
  - input of grades: separately via user interface, or as an XML document (created by the lecturer or from Ilias).
  - show/print BSc/MSC certificates and transcript of records (HTML or LateX),
    (in case that study programs have been defined: grouped as mandatory/optional course etc.)
  - Implement functionality that supports testing and presenting.
- ILIAS als stub service:
  - Communicate/update participants from FlexNow,
  - optional (and keep it simple): lecturers can create exams, students -during appropriate time slot?- cann fill in answers, lecturers enter grading+comments.
  - user interface for entering grades (not the correction process of individual exams).
  - communicate final grades to FlexNow.
- Data storage:
  - in XML files? (what must be considered then?)
  - there is an Oracle account "xmlp": See SQL-Praktikum (`http://www.dbis.informatik.uni-goettingen.de/Teaching/DBP/`); "Einführung in die Benutzung von Oracle am IFI" for direct calls by `sqlplus` from the CIP pool or via JDBC (see Slides for JDBC).
- FlexNow: When the registration for an exam is closed, the lecturer can activate that a corresponding "exam course" is automatically created in StudIP with all students who have registered for the exam.

Run the Web services on one or more CIP-Pool computers to be able to communicate (at home, the IP acresses are usually dynamic).