Project 2

Enterprise Application Integration (MIEBIOM)
Department of Informatics Engineering
Delivery date: see Inforestudante/Submissao de Trabalhos

Objectives

Gain familiarity with the development of tiered enterprise applications using the **Java Enterprise Edition (Java EE)** model. This includes the development of applications based on **Enterprise JavaBeans (EJB)**, the development of **Web Services,** the use of a **Persistence Engine,** and the creation of a **web front-end**. Overall, you should build at least one Enterprise Archive, which can be deployed in an Application Server.

Final Delivery

- You must submit your project in a zip file using Inforestudante. Do not forget to associate your work colleague during the submission process.
- The submission contents are:
    - Source code of the requested applications ready to compile and execute.
    - Report

The REPORT is expected to be complete in the sense that it needs to contain all necessary and sufficient information for the teacher to give the score to the evaluation item, even without having to run the code itself and without having to meet with the group in person for evaluation. For that you can include descriptions, screenshots, code extracts, whatever is needed for a complete and thorough evaluation. If the report is absent the score is 0, and if it is incomplete the score is significantly affected. This is to make sure you do have a complete report.

Before the main body, the REPORT starts with the complete identification of the students and group, then a table of contents, then the following:

a. Lists of what the group succeeded to do and what is missing
b. self-evaluation of the group (0-100%)
c. List of what each student contributed (no repetitions)
d. self-evaluation of each student in the group (0-100%)
e. hours of effort by each student separately

These items a,b,c are important for the teacher to check whether his evaluation coincides more or less with what the group and student thinks.

Grading

- REPORT (confirmed later by defense)
    - Overall quality of the data model used for representing data (XML/XSD);
    - Quality of the code (organization, modularity, formatting, code conventions, etc.);
    - Simplicity of the solution;
    - Final presentation of the work.

Software

Java EE Platform

For this project you are required to use a Java EE platform. We will be using the **WildFly Application Server** (previously known as JBoss AS). This platform is open-source and is available at: http://wildfly.org/downloads (check the exact software versions in the tutorials associated with this project). **Eclipse IDE for Java EE Developers** is the recommended IDE. (http://www.eclipse.org/). Eclipse will allow you to control the application server easily once you download the appropriate server adapter. The Eclipse configuration is quite easy. Refer to your Professor if you need additional help.

Data Persistence

When programming in Java EE it's quite common to use a database or persistence engine for saving the data. In this project students will have to use one persistence engine. We will be using the Java Persistence API (**JPA**) as it is a Java EE standard and provides Java developers with an object/relational mapping facility for managing relational data in Java applications. The JPA engine will be **Hibernate**. The recommended database is **PostgreSQL**.

References

Enterprise Applications Tutorial

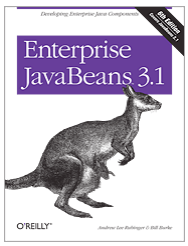
A tutorial on how to create Enterprise Applications will be **available at Inforestudante.** It allows you to train JPA in a standalone environment and also includes notes for using JPA (and EJB) in a container-managed environment. Note that the tutorial is only a starting point and students will need to read the additional resources.

Web Applications Tutorial

A tutorial on how to create web applications, using Servlets, JSPs, and JSTL will be **available at Inforestudante**. It describes a fast setup for creating dynamic web applications and provides an overview over the very basic concepts involved in this topic.

Books

There is extensive bibliography online about Java EE. Nevertheless, for this assignment, we strongly suggest that students start by reading a book that gives an overview about the basic concepts of Java EE. Contacting with a structured approach to Java EE development, like what is traditionally presented in books, is important. Although students may feel overwhelmed by the size and complexity of these books, don't despair. For this assignment, you don't need to worry about transactional and security requirements. Thus, you can skip the corresponding chapters. **The following book is recommended**:



Enterprise JavaBeans 3.1 (6th Edition)
by Andrew Lee Rubinger and Bill Burke

O'Reilly Media
ISBN 0596158025
September 24, 2010

Online Resources

There are many online resources about Java EE and EBJ 3.1. One of the most important resources is the **Java EE Tutorial**, available at:
https://javaee.github.io/tutorial

You can skim through at least: **Part One**, Chapter 1 – Overview; **Part Three** – The Web Tier; **Part Six** – Web Services; **Part Seven** – Enterprise Beans; **Part Eight** – Persistence.
You should also refer to the blog of this course. It may save you a lot of time:
Web service with JAX-WS 2.2.4
Java Persistence API with MySQL, Eclipse and JBoss 7
A Simple Enterprise JavaBeans 3.1 Example with JBoss (and Eclipse)
Creating an Enterprise Application Repository in Eclipse

Finally, all documentation for WildFly is available at:
https://docs.jboss.org/author/display/WFLY/Documentation

Project Description

In this project, we will build a tiered application, using Java EE technologies. The goal is to simulate a scenario where we need to query data and integrate services. We will use a **data persistence tool** (JPA/Hibernate) to manage data about **researchers**. Next, we will build **EJBs** to provide access to the JPA managed information and **web services** (provided by the application server) to provide operations to a **web service client** application. Finally, we will create a **web front-end** to provide information to users using browsers. Overall, we are going to build the following set of applications:

a) An application that loads data about **researchers, publications, and institutions** into a database (a single database to simplify the configuration).
b) A SOAP web service that uses an EJB that can access the database and provides basic information about **researchers**.
c) A SOAP web service that uses an EJB that can access the database and provides basic information about **publications**.
d) A REST web service that provides information about **institutions**.
e) A thin client application that can query the web services.
f) A web front-end to display basic information regarding **publications**.

The following paragraphs describe the minimum requirements, but you are free to create other operations, as needed, as long as they do not simplify the basic operations. Note that

some parts of the exercise are slightly different than what they would be in a real scenario, the goal is to have more problem diversity and gain broader knowledge on enterprise application development. You may assume authors' names are unique.

a) Loader

This standalone application will load the following information into a database. We will be simulating the presence of three databases:

i) **Researchers database:** Stores information about researchers (use only the information present in the 'about' box of ResearchGate);

ii) **Publications database:** Stores details about publications and respective author names;

iii) **Institutions database:** Stores institution details (name, location, department) and respective researcher names.

(create also an sql with the necessary commands to load 5 researchers and 5 publications for each and they should come from 3 institutions.

b) SOAP Web Service – Researchers

This is a SOAP service which has access to the Researchers database (via its own EJB) and provides the following functionality:

- Obtain all data about researchers.
- Given a researcher name, obtain all the associated data.
- Given a skill, obtain all researcher data associated with that skill.

c) SOAP Web Service – Publications

This is a SOAP service which has access to the Publications database (via its own EJB) and provides the following functionality:

- Obtain all data about publications.
- Given a publication title, obtain all the associated data.
- Given a researcher name, obtain all the publications s/he authored.

d) REST Web Service – Institutions

This is a REST service which has access to the Institutions database (via its own EJB) and provides the following functionality (responses must be JSON):

- Obtain all data about institutions.
- Given an institution name, obtain all the associated data.
- Given a researcher name, obtain the information about her/his institution.

e) Client application

A command-line client application capable of using all functionality provided by the web services.

f) Web front-end

Finally, we will create a web front-end so that we can use the system in a browser. The functionality to be supported by the web front-end is the following:

- Show the list of publications present in the database.

Good Work!