Project 2

Enterprise Application Integration (MEB)
Department of Informatics Engineering
Delivery date: 21 November 2022

1 2 90
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

Objectives

Gain familiarity with the development of tiered enterprise applications using the **Jakarta Enterprise Edition (Jakarta EE)** model. This includes the development of applications based on **Enterprise JavaBeans (EJB)**, the development of **SOAP** and **REST Web Services,** the use of a persistence engine compliant with the **Java Persistence API (JPA),** 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:
  - All source code of the requested applications ready to compile and execute.
- Make sure you can download and quickly execute the submitted assignment (for demonstration purposes).
- After submitting, you are required to register the (extra-class) effort spent solving the assignment. This step is mandatory. Please fill the effort form at:
  https://docs.google.com/spreadsheet/viewform?formkey=dHFha3NuWE1pYWJDbTJMTWcxTWRMS1E6MA

References

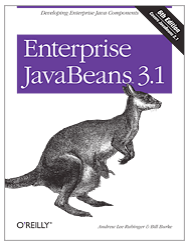
Jakarta EE in Practice by Filipe Araújo and Nuno Laranjeiro

This book is provided along with the course materials and provides a practical view of several important Jakarta EE technologies. In *Chapter 1* you can find information about the required software (in Section 1.1) and conceptual information about Maven, the tool we will be using to compile and build the projects (in **Section 1.3**).

After reading the first chapter, you can find the information about the technologies associated with this project in ***Chapter 2*** (Java Persistence API), ***Chapter 3*** (Enterprise Java Beans), ***Chapter 4*** (SOAP web services), ***Chapter 5*** (REST web services), *Chapter 6* (Web applications, **Section 6.1**). In ***Chapter 8*** you can find information associated with a full setup, allowing you to make use of all previous technologies integrated into a single application, which is packaged as an Enterprise Archive in an Application Server.

Other resources

There is extensive bibliography online about Jakarta EE technologies. A good reference is:



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

O'Reilly Media
ISBN 0596158025
September 24, 2010

Other useful references include:

- The **Java EE Tutorial**, available at https://eclipse-ee4j.github.io/jakartaee-tutorial/
- **WildFly** documentation available at https://docs.wildfly.org/

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 **television series**. 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 **series** into a database (a single database only to simplify the configuration).
b) A SOAP web service that uses an EJB that can access the database and provides basic information about **series**.
c) A REST web service that uses an EJB that can access the database and provides advanced information about **series**.
d) A thin client application that can query the web services.
e) A web front-end to display basic information regarding **series**.

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 required 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 that actor names are unique.

a) Loader

This standalone application will load the initial XML information about series, used in the previous project, into a database management system.

b) SOAP Web Service

This service has access to the database (via EJB) and provides the following functionality:

- Obtain all data about series
- Obtain all series titles
- Given a series name, obtain all its data
- Given a series name, obtain its IMDB rating

c) REST Web Service

This service has access to the database (via EJB) and provides the following functionality:

- Obtain all actors names
- Given a set of keywords, obtain the series titles that include these keywords in their description. It should be possible to select between returning series that match *all* keywords or series that match *at least one* of the keywords.
- Given an IMDB rating and number of voters, obtain all series titles above both parameters.
- Given a Genre, obtain all series that match the Genre.
- Given a Genre, obtain a list of actors and their number of appearances in series belonging that Genre. The list should be ordered by the number of appearances.
- **[EXTRA POINTS]** Obtain a list of Genres present in the database along with the following statistics. For each Genre the average IMDB rating of the corresponding set of series should be calculated and also returned. It should be possible to specify the number of maximum results (e.g., calculate the statistics for 3 Genres only).

d) Client application

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

e) 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 actors present in the database.
- Your own custom functionality. Use your time wisely!

Good Work!