

Assignment 4 - Web services

Network Programming, ID1212

Bernardo Gonzalez Riede, begr@kth.se

December 17, 2017

1 Introduction

Goals:

- Develop a three-tier web-based application using frameworks
- Deploy and run a web-based application on an application server.

2 Literature Study

As in previous assignments, the main source were the video lectures on the topic and the example code.

3 Method

As in previous assignments, the mvc approach has been used again to layer the program.

The choice for a database fell on MySQL because of past works with it. The implementation was realized in Netbeans in Java 8 EE using Pyara as the application server.

4 Result

Link to public Github repository with code: <https://github.com/MemBernd/ID1212-Web-services>

4.1 Layering

The layering of the application is divided into four layers:

- The *view* holds the JSF page.
- The *controller* is used as a EJB.

- The *model* holds the JPA entities.
- The *storage* is responsible for connecting to the MySQL database.

4.2 Database

The database in 1 consists of two tables. The first table acts as a catalog, holding the different currencies, while the second one is the result of a many to many association of the catalog to itself.

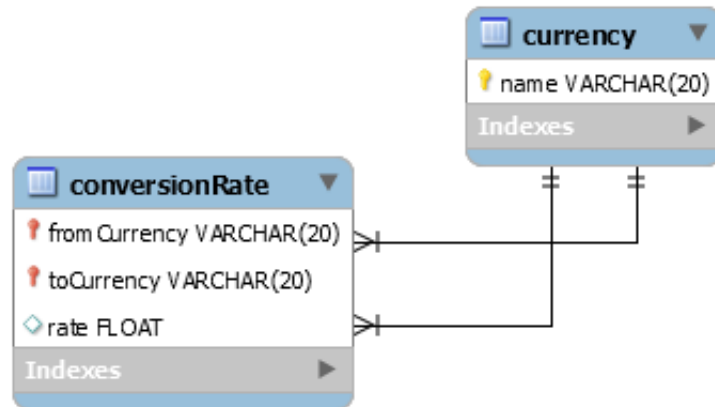


Figure 1: Table relations in database.

4.3 Frameworks

4.3.1 JSF

Being a JSF, `view.CurrencyConverter` has mostly getters and setters (l. 52 - 118) for several private variables. The JSF annotations are in line 22 & 23, namely the name and the scope.

Some of the variables from the JSF are injected dynamically in `converter.xhtml`. A few use a conditional (e. g. `converter.xhtml` l. 32) to be rendered only if their value is different from 0, as to not show these values at the first loading of the page.

4.3.2 EJB

EJB related annotations in `controller.Controller` are found in line 22 & 23. Here, the EJB runs every method in a new transaction and is *stateless*. It contains two methods:

1. `listCurrencies()` (l. 31) to list all available currencies, e.g. to populate the comboboxes.
2. `convert()` (l. 39) to do the actual conversion.

4.3.3 JPA

After the database had been created, the automated process in netbeans to create entities from an existing database has been used to create the three entities in the model layer.

The entities corresponding to a table, i.e. *Currency* and *Conversionrate*, include named queries in the beginning of their files (between lines 25 to 30). One of them is used in storage.CurrencyDAO line 33 to find all available currencies. In the same file on line 40, the model is used directly to find a certain primary key, i.e. a combination of *from* and *to* currency.

The model is accessed through an EntityManager (l. 28 & 29). Line 28 references the persistence unit defined in *persistence.xml*.

4.4 Client side

The user can access the application through a browser, which presents him with the jsf (see 2). The user is able to select between 4 currencies to convert between them. In case he selects the same currency, the application will return, hence show, the same amount as entered.

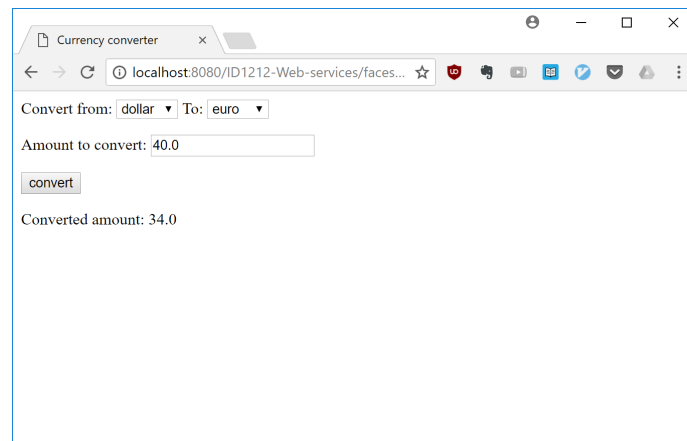


Figure 2: Browser interface

5 Discussion

Through the use of comboboxes the incorrect input is limited to the use of non digits in the currency amount. Through the use of Beans, this behaviour is captured instantly and the user is shown a message similar to:

'XX' must be a number consisting of one or more digits.

6 Comments About the Course

It took almost 9 hours to complete this assignment.

- 1 hour 37 min for coding.
- 4 hours 6 min for the report.
- 3 hours 13 min watching videos, taking notes and thinking about how to tackle the problem.