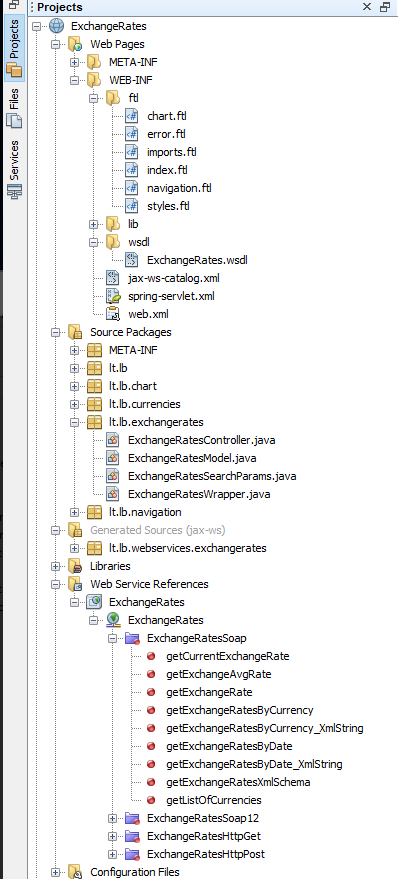
**Exchange rates web application**

For current assignment these tools and technologies were introduced:

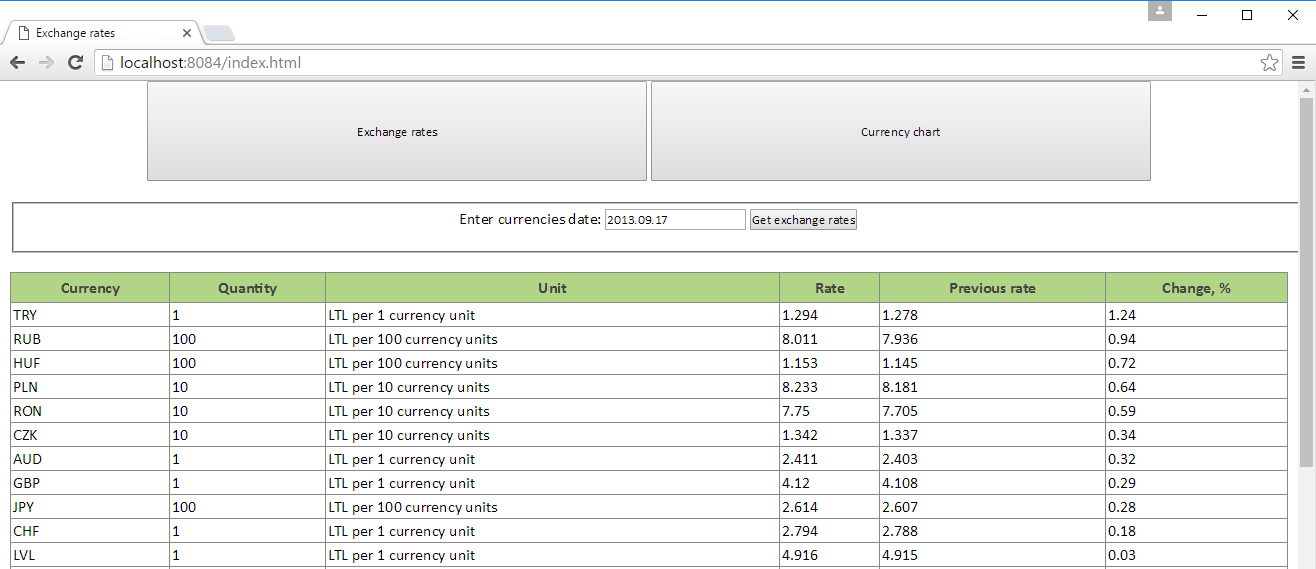
1. Netbeans 8.0.2 IDE.
2. JDK 1.7.
3. Apache Tomcat 8.0.15.0 server.
4. Spring 3.1.2. framework for handling web structure.
5. JAX-WS implementation for calling webservices.
6. Freemarker 2.2.8 for displaying viewers.
7. JQuery for using datepicker component.
8. Google charting tools for displaying area chart (<https://developers.google.com/chart/>).

To run web application you have to open/import project into your favorite IDE (in my case I was using netbeans) and set Tomcat 8 as a running server. You should see this project structure:

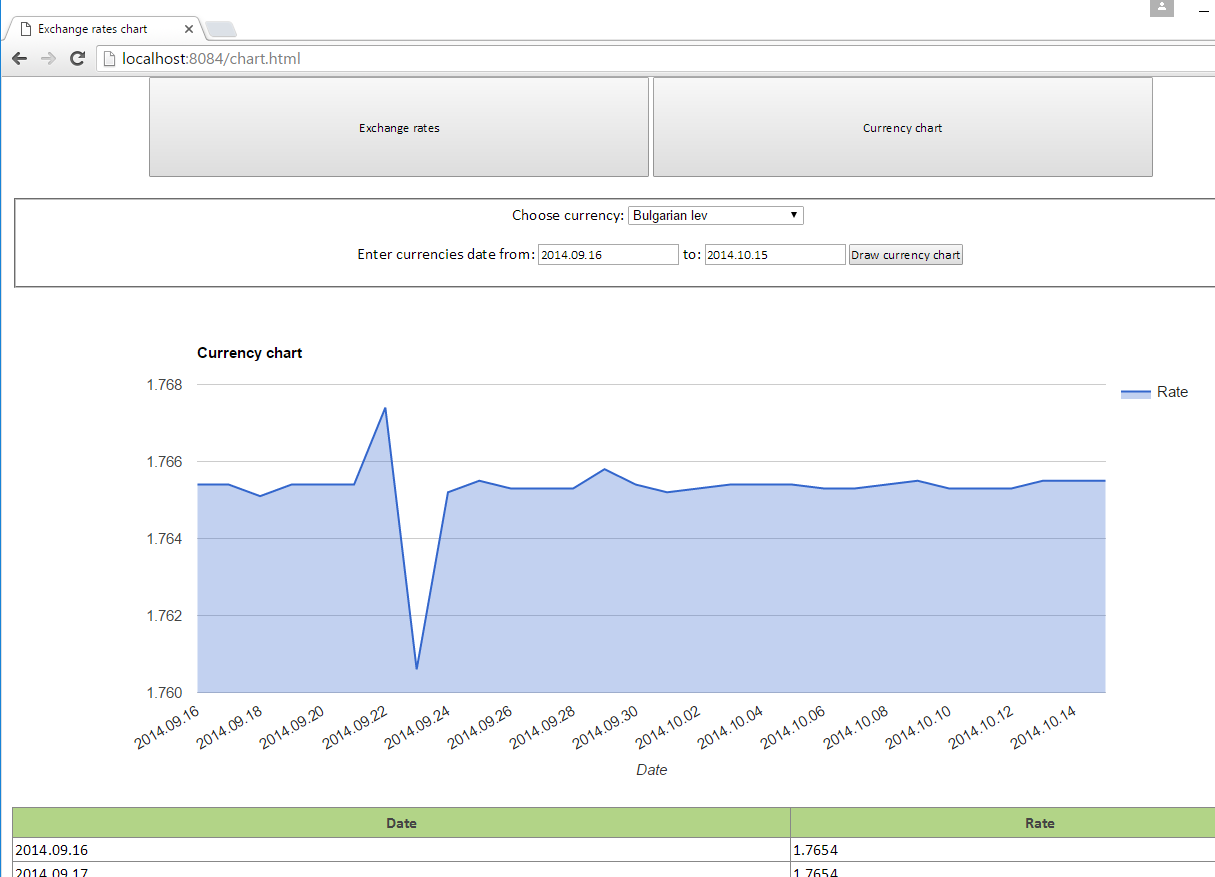


JAX-WS handles creation of objects of ExchageRates.wsdl which then are used to call <http://www.lb.lt/webservices/ExchangeRates/ExchangeRates.asmx> webservices. There are three calls made to webservices:

1. getExchangeRatesByDateXmlString – for displaying exchange rates by entering date into input field (ExchangeRatesModel.getExchangeRatesByDateXmlString):



1. getExchangeRatesByCurrencyXmlString – for displaying chart by choosing period and currency (ChartModel.getExchangeRatesByCurrencyXmlString):

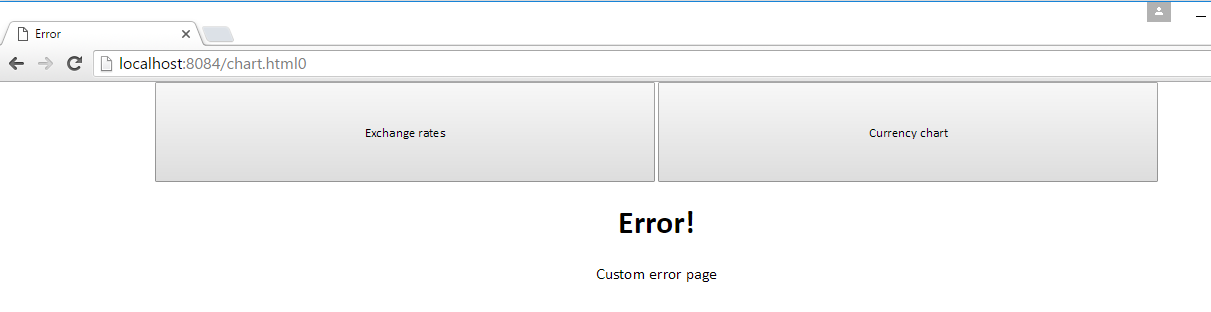


1. Separate call to currency list webservice by parsing its xml file – for populating combobox component and displaying list of currencies (CurrenciesModel.parseXml).

All results from calls are returned as string values which are parsed and set into wrapper objects. Model-View-Controller pattern is used by this structure:

1. \*.ftl viewers (freemarkers files);
2. \*Controller.class for handling requests between viewer and model;
3. \*Model.class for getting and parsing data into wrappers.

Errors from input data like dates or chosen currency are handled in models. Navigation is made by clicking two navigation buttons “Exchange rates” and “Currency chart”. If unexpected error occur (like page not found) current page is opened:



MVC pattern with freemarker as its viewers template are best for reusability, simple logic handling. JAX-WS makes it easy to call webservices by providing necessary objects. Google charts are easily used because you don’t have to over think it.