Kader Kawsar

12 August 2012

Telegraph Media Group Technical Test

Objective:
The object was to build an API to be used internally and externally to manage users. This is the API the “web” and “mobile” teams will call from the “register” and “my details” pages on the website and the internal tools will use for house keeping.

Introduction:

The API was build using Jersey RESTful Web Service framework in Java. The build process was carried out with Maven version 3.0.4, The JUnit framework was used for Test Driven Development unit testing and Cucumber-JVM was used for Behaviour Driven Development. The final integration testing was on the Tomcat server via maven-failsafe-plugin.

The project is a web app and the source code is divided into main, test, bdd feature, doc and maven pom.xml file. The eclipse project file are also included for easy importing into eclipse IDE.

Main:
The src/main/java directory contains deployable code, arranged in the MVC directory structure arrangement. src/main/java/com/tmg/ws/resource/UserResource.java is the REST web service class and the other classes are supporting this class to fulfil the desired behaviour.

src/main/java/com/tmg/repository/impl/UserDaoImpl.java class has a singleton map, acting as the in-memory database, which can easily be swapped with relational database implementation.

src/main/java/com/tmg/controller/impl/UserManagerImpl.java implementation class acts a mediator between the web service and database. Its also controls the delegation to validation class for validating data.

Test:
There are three JUnit test classes i.e. src/test/java/com/tmg/controller/UserManagerTest.java, src/test/java/com/tmg/repository/UserDoaTest.java and
src/test/java/com/tmg/service/UserValidatorTest.java
Those classes were developed using TDD techniques and perform the unit tests.

BDD:
The src/test/resources/user_manager.feature is the BDD story file, describing each scenario being tested. The src/test/java/com/tmg/bdd/RunCukesTest.java class is an empty class with

annotation and it is the entry point for running the steps described in user_manager.feature via the src/test/java/com/tmg/bdd/steps/UserManagerStepdefs.java step definition class.

The src/test/java/com/tmg/bdd/support/UserManagerSupport.java class uses the RestAssured framework to run the RESTful service tests.

Docs:
The JavaDocs are placed in the src/main/webapp/doc directory and can be accessed via {hostname}/tmg/doc/index.html. The JavaDoc only contains the REST API usage details as it is the only client facing class.

Maven:
The following Maven command can be used for building and running the unit and integration test on a standalone tomcat server. Maven will download all the required dependencies and run the tests, including the RESTful service test. pom.xml file is in the root directory. This command will also create a war file in the target directory for deploying in a web container such as tomcat.

```
mvn clean verify package
```

The following Maven command could be used to run the web service in a self-contained tomcat server.

```
mvn clean tomcat:run
```

Once the server is running, the API doc can be viewed via
http://localhost:8080/tmg/doc/index.html

cURL:
The following curl commands can be used testing the service.

```
$ curl -i -X PUT -H "Content-Type: application/json" -H "Accept: application/json" --data
'{"user":{"type":"Subscriber","firstName":"Normal","lastName":"Su
bscriber","title":"Mr","dateOfBirth":"2008-02-01","email":"newsubscriber@tmg.com","password":"
password","home":{"houseNumber":"101", "address1":"Home", "cit
y":"London"},"billing":{"houseNumber":"101", "address1":"Home", "city":"London"}}}'
"http://localhost:8080/tmg/api/user/register"
```

```
$ curl -i -X POST -H "Content-Type: application/json" -H "Accept: application/json" --data
'username=user@tmg.com&password=user' "http://localhost:8080/tmg/api/user/mydetails"
```

```
$ curl -i -X DELETE -H "Content-Type: application/json" -H "Accept: application/json" --data
'userToDelete=usesr@tmg.com&username=superuser@tmg.com&password=super'
```

"http://localhost:8080/tmg/api/user/delete"

Security:

The http GET is never used and POST was used instead for security reason.

A self-signed certificate was created using the following command

```
keytool -genkey -alias kader -keyalg RSA -keystore c:\kader\tmg.crt
```

and the tomcat config/server.xml file was configured with the following so that the service could be run under the https protocol. However, in order to keep the security as a separate layer, it is not included in this build.

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
           maxThreads="150" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS"
              keystoreFile="c:\kader\tmg.crt"
              keystorePass="changeit" />
```

Conclusion:
The web app is a RESTful web service API developed using BDD, TDD, Eclipse IDE with Jersey framework. The RestAssured framework was used to test the RESTful services and the overall project was build with Maven.