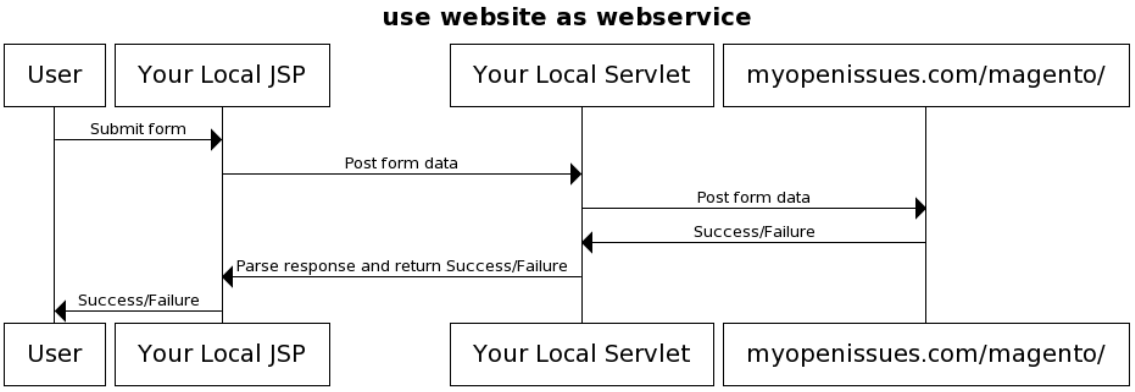
Task: Use a website form as a web service

Please take a look at following registration form:
http://myopenissues.com/magento/index.php/customer/account/create/

You can enter data on this form and setup the account. That is easy. What we want you to do is use the above registration form as webservice with following steps --

a) Please write a webapp (http://localhost:8080/tys2345/register.jsp). Please note that you have been assigned : cdetsd123 id. So your project shall use this id as application name.
b) register.jsp shall be your own local form. This form shall collect the data from user.
c) register.jsp shall then post the data to your local servlet.
d) Your local servlet shall post the data to suitable end point on http://myopenissues.com/magento/index.php and then parse the response and display proper error message or success on your displayed form. That means, your servlet will be sitting in between the user and http://myopenissues.com/magento/ and processing the input and return. Please make sure you use apache HttpClient library to perform the web interactions. Please do not use Selenium webdriver
e) The action above shall setup a new account on http://myopenissues.com/magento/
f) Please provide mvn 2 scripts that shall compile, build and run the webapp locally in Jetty. So as a end result, I can open up jetty url and confirm that application is running properly.



Please use following setup - mvn (2) scripts that shall build, compile and run the app in local Jetty (please be aware of adding proper compile instructions in pom.xml as your development environment can be completely different that our servers) - App shall run in Jetty and Tomcat 6 - Java 6

Bonus tasks:

1. Create JUNIT tests and hook up with cobertura code coverage so that it will show coverage report of your unit testing.
2. put the code in github free public repo
3. Hook up github code with Travis-CI for builds and show it compiles, builds. Deploy the code only if it passes unit tests, integration tests. Please remember to refer to your project as tys2345.
4. Selenium scripts to demonstrate working of your app.
5. Deploy the instance on the AWS free account through build scripts (you can setup a free account with limited use AWS, S3 -- if you have access to other free hosting that you have used before you can use that as well)

Solutions

a) The whole application is packaged into a zip file and sent by email. Please use following commands to run the application
    a. To verify the application: mvn verify
    b. To package the application: mvn package. The packaged war file can be deployed in any server like Tomcat
    c. To run the application: mvn jetty:run
        i. You can open the application at http://localhost:8080/register.jsp
        ii. There was an issue configuring context path along with selenium and hence I could not set the path to cdetsd123. Without selenium, I could set it but due to lack of time, I could not debug further. Since the requirement for the product is to run on 1.6, I had some restrictions in using maven versions of plugins and I think that is causing the issue.
        iii. The register.jsp page will show success or error message depending on input.

Bonus Tasks

JUnit and Cobertura

1. Basic JUnits to test the service class function processRegistration have been written. Since the Servlet class has very trivial code, I did not implement any JUnits on it.
2. Cobertura has been integrated with the project. Please use following command to run cobertura
    a. mvn cobertura:cobertura

    b. It will generate reports in target/site/cobertura folder. The coverage rate for Service class is 68%. Only defensive programming like null checks and exceptions are not covered which are in the rest 32% of the program.
3. An integration test case has been implemented with JUnit in selenium script.

GitHub

1. Code is available in GitHub at https://github.com/santoshkt/cdetsd123
2. Selenium standalone code to test Heroku app is available at https://github.com/santoshkt/cdetsd123_selenium

Travis

1. Travis-CI is hookedup with GitHub. It can be accessed at https://travis-ci.org/santoshkt/cdetsd123
2. Travis shows compile and build passes. It will deploy only if the com test cases pass.
3. Travis will automatically deploy to Heroku if the build passes.

Selenium

1. A selenium script with 5 test cases have been written to test basic functionality of the app.
2. The scripts execution can be seen on mvn verify command.
3. To run script on standalone java application, please refer to code at https://github.com/santoshkt/cdetsd123_selenium

Deploy on Cloud

1. I have used Heroku instead of AWS as I already have a Heroku account. Please access the app at http://cdetsd123.herokuapp.com/register.jsp