Getting started with Maven

Setup Maven

If not already done, make sure to setup Maven as explained in this document

Getting Started

Complete this short tutorial "Maven in 5 Minutes". Don't try to do it in 5 minutes ;-)

Do this exercise WITHOUT NetBeans

a) Complete all steps, and make sure you understand everything you do. If you don't ASK.

b) Make the test fail, and run `mvn test` to see the result

c) Still with the failing test, run: `mvn clean verify`
Observe the target folder, and explain the result

Project from Archetype in NetBeans

1. In NetBeans use the new wizard to create a new Maven project like this:
   a. `Maven → Project from Archetype → Press Next`
   b. In the Search box type: `maven-archetype-quickstart`
   c. From the Known Archetypes select the one with the Group ID: `org.apache.maven.archetypes`
   d. Change the Version into 1.4. This should give you a project similar to what you did in the previous step.
2. Open a terminal and navigate in the root of this project. Repeat most of steps to see that, obviously, we are in no way tied to NetBeans, even if the project was created and as you will see, developed from here. (Explanation required here)
3. In the package created by the project, create a new class `SayHi` and paste in the code below:

```
public class SayHi {
  /**
   * Prints out a great Hi message
   * @param name
   * The name to use for the message
   */
  public String sayHi(String name){
      return "Hi "+name;
   }
  public String sayHiHi(String name){
      return "Hi Hi  "+name;
   }
}
```

4. Add the missing JavaDoc for the Class and the `sayHiHi` method (hint: type `/**` followed by return will give you the template for JavaDoc

Test the new Class

Add a new test class called testSayHi and write JUnit tests to verify the behaviour of the class.

a) Run the test, from within NetBeans, similar to what you did last semester. Make sure that all the tests are green.

b) Now, from the terminal, re-run the tests using Maven like this: `mvn test`
c) Verify that the tests still pass.

d) Change the code, so the test will fail. Re-run the test from the terminal via maven

e) Observe the output generated by Maven. Open the file sure-fire report file to see the test report for your test.

f) Change the code back, to get a green test

Creating Documentation for the Project

The Site Plugin is used to generate a site for the project. The generated site also includes the project's reports that were configured in the POM. Don't spend to much time on this part, but your CA's and the semester project should include a documentation site

a) Add this section to your pom-file, just above the properties section.

```
<reporting>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-javadoc-plugin</artifactId>
        <version>3.0.1</version>
        <configuration>
          <show>public</show>
        </configuration>
      </plugin>
    </plugins>
</reporting>
```

b) Generate the site info using Maven like this: `mvn site`

c) Navigate into target→ site→ and double-click the project-info.html file to see the generated information. Make sure that you can find the Javadoc API documentation.

d) Use this link and add the necessary fields to your pom-file to include, as a minimum, the following in the report: name, description, list of developers (name+email), the ciManagement system used (Travis)This could benefit from a more detailed explanantion of where in the very long document to look for the relevant information.