Test and CI, with Maven and test Travis-CI

control and the second algorithms and the second algorithms and the second algorithms and the second algorithms are requirements defects the second and the

This exercise will demonstrate how to use Travis to execute our Unit- and Integration tests, and, if you complete it to the end, also how to let Travis deploy your tested WAR-file to an online Tomcat instance.

Preconditions for this exercise

- 1. This exercise is a continuation of the exercise <u>Testing and Deployment with Maven</u>, so you must have started with this exercise since all the following steps should be made in this project.
- 2. Before you start, you should have completed the exercise "Continuous integration with Travis-CI", which ensures you have created an account on Travis and allowed Travis access to your repository.

Prepare your project for Travis)

Remember, from the Travis getting started exercise? that all it takes is to integrate with Travis is:

- 1. Host your project on GitHub
- 2. Sign in to Travis CI
- 3. Flip the Switch
- 4. Add a .travis.yml file, commit, and push this file to Github.
- a) Complete step 1-3 above
- b) Add a .travis.yml file in the root of the project (remember the leading dot), with the following content.

```
language: java
jdk: oraclejdk8

cache:
    directories:
        - $HOME/.m2

script:
    .....
```

The script section above is where we tell Travis how to execute our tests. What you write here, depends on how far you came with the initial exercise (Testing and Deployment with Maven). Replace XXX with the content given below (exactly as given, including the "-"):

- mvn test	If you only have implemented the "Getting started" part of the exercise
- mvn verify	If you have implemented the WEB-API, and it's integration tests
- mvn verify -Ptest	If you have implemented the requested profiles in your pom-file

c) Commit and push to GitHub, and watch your project being built on Travis.ci.org.

Travis with a failing test

Change one of the tests to fail deliberately. Commit and push (shame on you ;-)

Verify that your build server "turns red", and just as important, that you receive a mail with a notification about the failing build.

See here for details about email notifications:

https://docs.travis-ci.com/user/notifications/#How-is-the-build-email-receiver-determined%3F

Let Travis deploy your WAR-file if all tests passed

If you have completed the last section of the test-exercise "<u>Deployment with Maven</u>" using a hosted Tomcat Server, you can let Travis deploy your war-file to this server, after a successful build.

a) Add this section to the end of your .travis.yml file:

```
after_success:
```

- mvn tomcat7:deploy -Premote
- **b)** Make a visible change to your web-project, for example in the index.html file to make it easy to see that a new version was deployed.
- c) Run all your tests locally (mvn verify or mvn verify -Ptest), and if green, commit and push to Github.
- **d)** Go to Travis-ci.org and navigate to the status page for your project. In the Job log, almost all the way down to the bottom, locatate this line and press the "after_success" button to see the status of your deployment.



e) If the previous step reported a success (don't expect a single line, like "Everything was fine";-) you should navigate to your web-app, and see that the change actually was deployed (you probably need to refresh your page to see the change).