# JPA, REST, Test, CI and Travis with our start code

Both of the exercises use the start code linked to under the *Repos* tab and the section *Start Code for Flow1+2 projects* on the course page.

You must clone the start code for each of the two exercises as described in the project's documentation.

**Preconditions:** You must have completed the exercise “[Gettings Started with Travis and CI](https://docs.google.com/document/d/1yN8fiKHI5oz4TfZrH_zHla51eiQ42F8tognZ4vefcVw/edit?usp=sharing)” before you start.

**1)** Getting Started with the start code

If not done as part of the lecture, complete the guidelines given in projects readme-file in the section *First time users - getting started*.

Make sure to go “all the way” so you get a project on your droplet, deployed by Travis.

**2)** Using the start code for your own projects

Clone the start code again, into a **new** project:

clone <https://github.com/dat3startcode/rest-jpa-devops-startcode.git> PICK-A-NAME

Complete the steps given in the readme documentation [*How to use this start project for future rest/jpa project with CI → Initial Setup*](https://github.com/dat3startcode/rest-jpa-devops-startcode/blob/master/README_how_to_use.md)*.* Before you create the databases see the next step, to provide database names that makes sense.

Add the following changes to the project (described in the section *Add your own features* in the documentation). Feel free to twist everything below as you like, as long as you come through all the steps:

* Change the start codes Entity Class, into a class **Movie**, with a number of relevant fields to represent movies in a movie-database.
* Change the Facade Class to be a **MovieFacade**, with a few relevant methods to add and get information about movies.
* Change the REST part to implement a number of relevant endpoints, according to what your facade provides. For yellow, and definitely red students you should come up with relevant DTO-classes as mappers for your JSON-conversion.
* Change the *test classes* to reflect your changes (see hints below)

Complete the steps given in the documentation *Deploy the project (manually) via Maven* and *ADD CI-control to your project and let Maven deploy when the project BUILDS and all tests are GREEN*

**3) I want more ;-)**

Wait until Friday, then you will be asked to add a simple JavaScript-driven frontend to this exercise

**Hints, related to testing**

When you start making changes to the code, you will notice that they break the existing test-cases. The effect of this (thats a maven thing) is that you cannot build your code.

We suggest that you remove/uncomment the existing test, and then do test-driven design like this:

Before you add a method to the facade, implement a test for the method, run the test an see it fails. Then implement the method and run your test so it turns green... and so on.

Alternatively, if you want to focus on only one thing at the time, while you are learning, you can disable the tests via the @Disabled annotation (see the top of both test classes)

If you complete part-2, day-1, you will probably notice that there are “things” in the REST-test code you don’t understand.

***This will be covered day 2***