Project Set-up

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

ADAP A03

Project Set-up Steps

1. Decide on topic for course project
2. Choose and setup development environment
3. Register on GitHub and fork Wahlzeit
4. Set-up Dockerhub
5. Set-up Travis CI
6. Get Wahlzeit running on your machine
7. Set-up debugging
8. Run unit tests

1. Course Project Decision

- Decide on your project idea for Wahlzeit
  - Give it a name, for example, “flowers”

2. Development Environment Set-up

- Install and set-up Java IDE of your choice, e.g.
  - IntelliJ IDEA: https://www.jetbrains.com/idea
  - Eclipse: https://www.eclipse.org/ide
  - Netbeans: https://netbeans.org
- Install Java JDK 8 (or higher, up to 11)
  - As of 2019-10-09, only up to Java 11; **Java 12 does not work**
  - If you need to manage multiple JDK versions: https://sdkman.io/
- Install Git (command-line)
  - If you need a GUI: https://git-scm.com/downloads/guis
- Install Docker
  - Follow the instructions for your OS: https://docs.docker.com/install/

3. GitHub Set-up

- Register on github.com
  - Go to https://github.com/dirkriehle/wahlzeit and fork the repository

4. DockerHub Set-up

- Create DockerHub repository
  - Register on https://hub.docker.com
  - Create a new repository
    - Name it “wahlzeit”
    - Set visibility to public

5. Travis CI Set-up

- Set-up continuous Integration with Travis CI
  - Sign in with your GitHub account on https://travis-ci.org/
    - Add your forked repository to the list of repositories Travis CI should check
    - Set environment variables DOCKERHUB_USER and DOCKERHUB_PW
    - Ensure Travis CI is able to build your project

6. Wahlzeit Set-up

- Clone your repository to create a local repository
  - Ensure master branch is selected
- Import the local repository into your IDE
- Build and run Wahlzeit, try it at http://localhost:8080/wahlzeit
  - On the command line, call “./gradlew appengineRun”
  - A first build will take a while and may have intermediate steps

7. Debugging Set-up

- Create remote Java debugging configuration
  - Connection string is “localhost:8000”

8. Unit Test Try-out

- Use Gradle tasks to run unit tests
  - “./gradlew test” runs all unit tests and provide a summary
  - “./gradlew appengineRun” will also run test before starting the application

```
org.wahlzeit.services.mailing.EmailServiceTest > testSendValidEmail PASSED

org.wahlzeit.handlers.TellFriendTest > testTellFriendMakeWebPart PASSED

org.wahlzeit.handlers.TellFriendTest > testTellFriendPost PASSED
Tests run: 83, Failures: 0, Skipped: 0
```

Thank you! Questions?

dirk.riehle@fau.de – http://osr.cs.fau.de

dirk@riehle.org – http://dirkriehle.com – @dirkriehle

Credits and License

- Original version
  - © 2012-2019 Dirk Riehle, some rights reserved
  - Licensed under Creative Commons Attribution 4.0 International License
- Contributions
  - Andreas Bauer (2018)