SEBA Master: Web Application Engineering

Introduction to the play-framework

Felix Lachenmaier, 11.05.2015, MI HS 2

Software Engineering for Business Information Systems (sebis)
Department of Informatics
Technische Universität München, Germany

wwwmatthes.in.tum.de

Felix Lachenmaier

- Email: felix.lachenmaier@tum.de
- Office hours: Every Thurstday 8.00 – 16.00, Room MI 01.12.034
- Support for all technical issues

Outline

1. Prerequisites
   a. Version control
   b. Motivation to use a MVC-Framework
   c. Handling user requests with play
   d. Model – View – Controller pattern
2. Hands on!
   a. Installation and configuration
   b. Live Demo

Version Control

sebis

1. Always use version control
2. Always use version control
3. Always use version control

Version Control

Have you ever:

Made a change to code, realized it was a mistake and wanted to revert back?

Lost code or had a backup that was too old?

Had to maintain multiple versions of a product?

Wanted to see the difference between two (or more) versions of your code?

Wanted to prove that a particular change broke or fixed a piece of code?

Wanted to review the history of some code?

Wanted to submit a change to someone else's code?

Wanted to share your code, or let other people work on your code?

Wanted to see how much work is being done, and where, when and by whom?

Wanted to experiment with a new feature without interfering with working code?

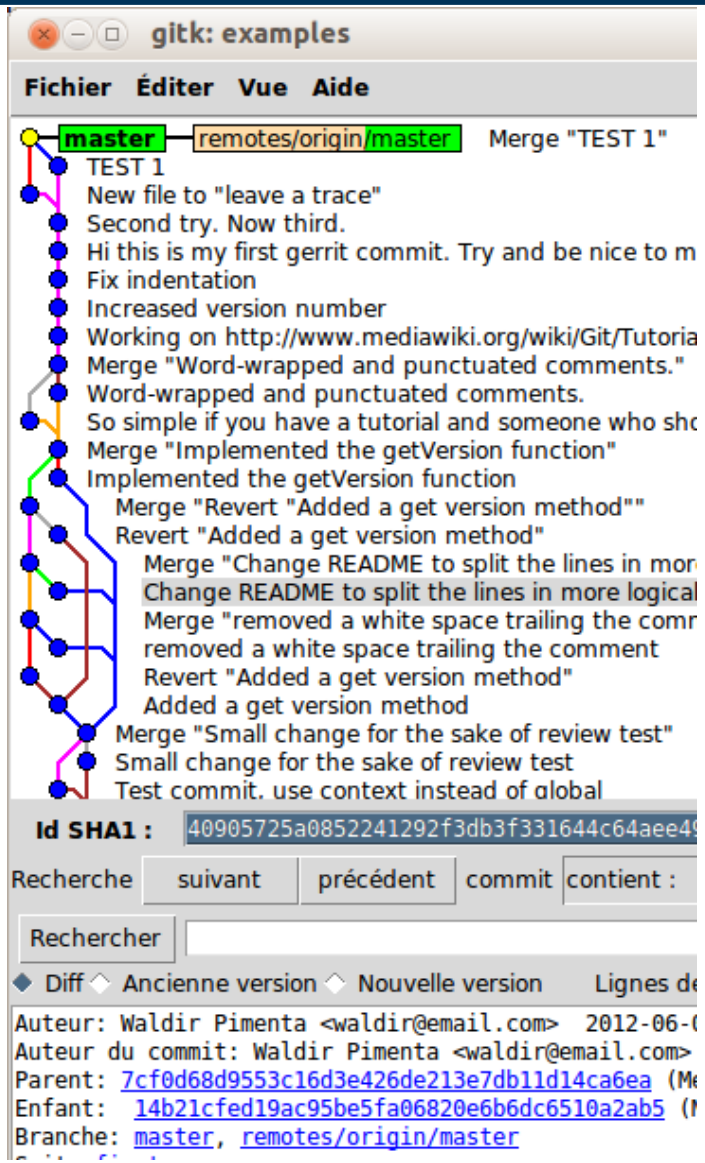
Branching

→ (Slightly) different versions of one project

→ Makes a lot of things easier

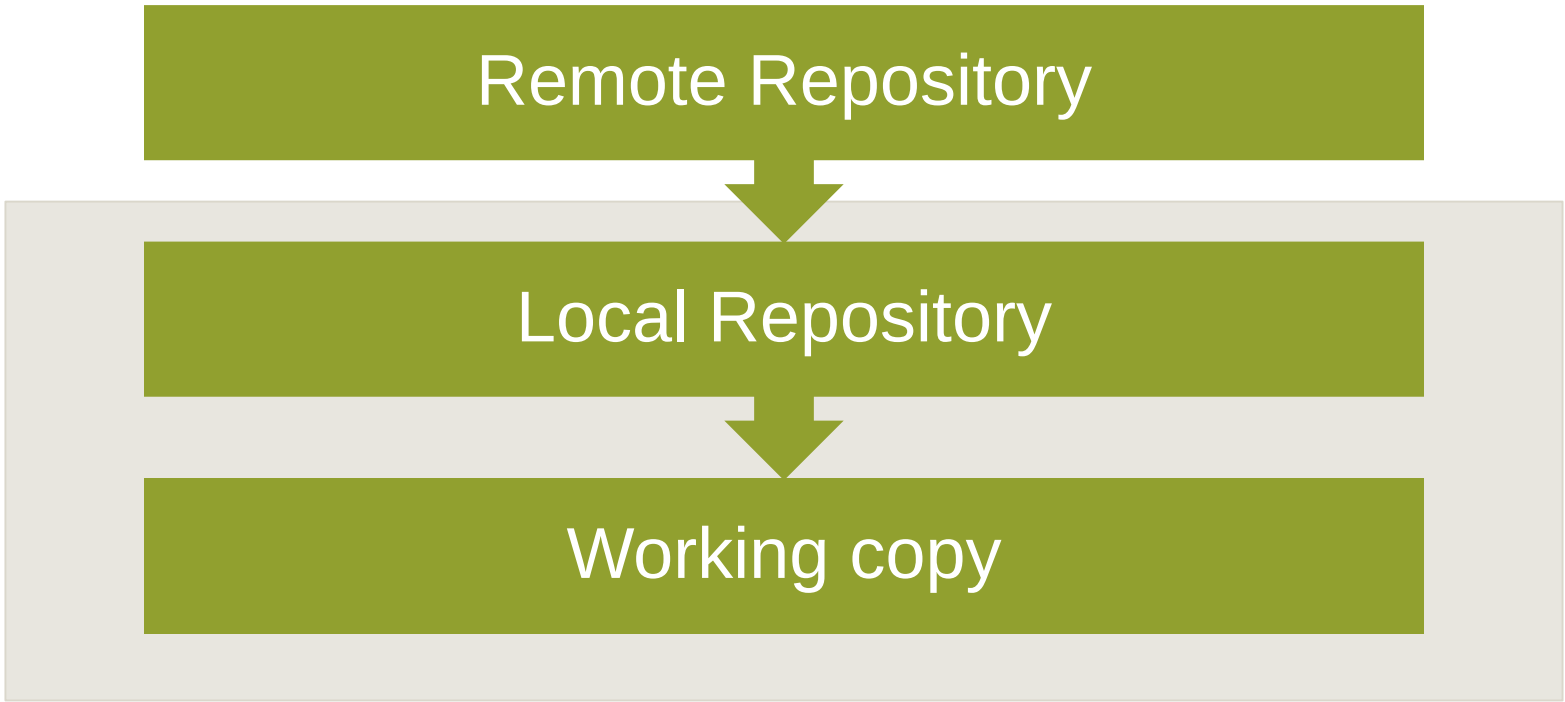
→ You should create new branches for each user and each feature

→ Use as few as possible, but as many as necessary



Remote and Local Repositories

git is distributed, not centralized



Ignoring Files

.gitignore vs. assume-unchanged

.gitignore

```
# Extracted from https://github.com/ul
# Ignore all dotfiles...
.*
# except for .gitignore
!.gitignore

# Ignore Play! working directory #
db
eclipse
lib
log
logs
modules
precompiled
project/project
project/target
target
tmp
test-result
server.pid
*.iml
*.eml
activator-*.sbt
```

angular-seed-play — bash — 80×24

```
Felixs-MacBook-Pro-2:angular-seed-play felix$ git commit -m 'server configuratio
n'
[master ebec59e] server configuration
 1 file changed, 68 insertions(+)
 create mode 100644 conf/application.conf
Felixs-MacBook-Pro-2:angular-seed-play felix$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   conf/application.conf

no changes added to commit (use "git add" and/or "git commit -a")
Felixs-MacBook-Pro-2:angular-seed-play felix$ git update-index --assume-unchange
d conf/application.conf
Felixs-MacBook-Pro-2:angular-seed-play felix$ git status
On branch master
nothing to commit, working directory clean
Felixs-MacBook-Pro-2:angular-seed-play felix$
```

Ignoring Files

.gitignore vs. assume-unchanged

```
modules
precompiled
project/project
project/target
target
tmp
```

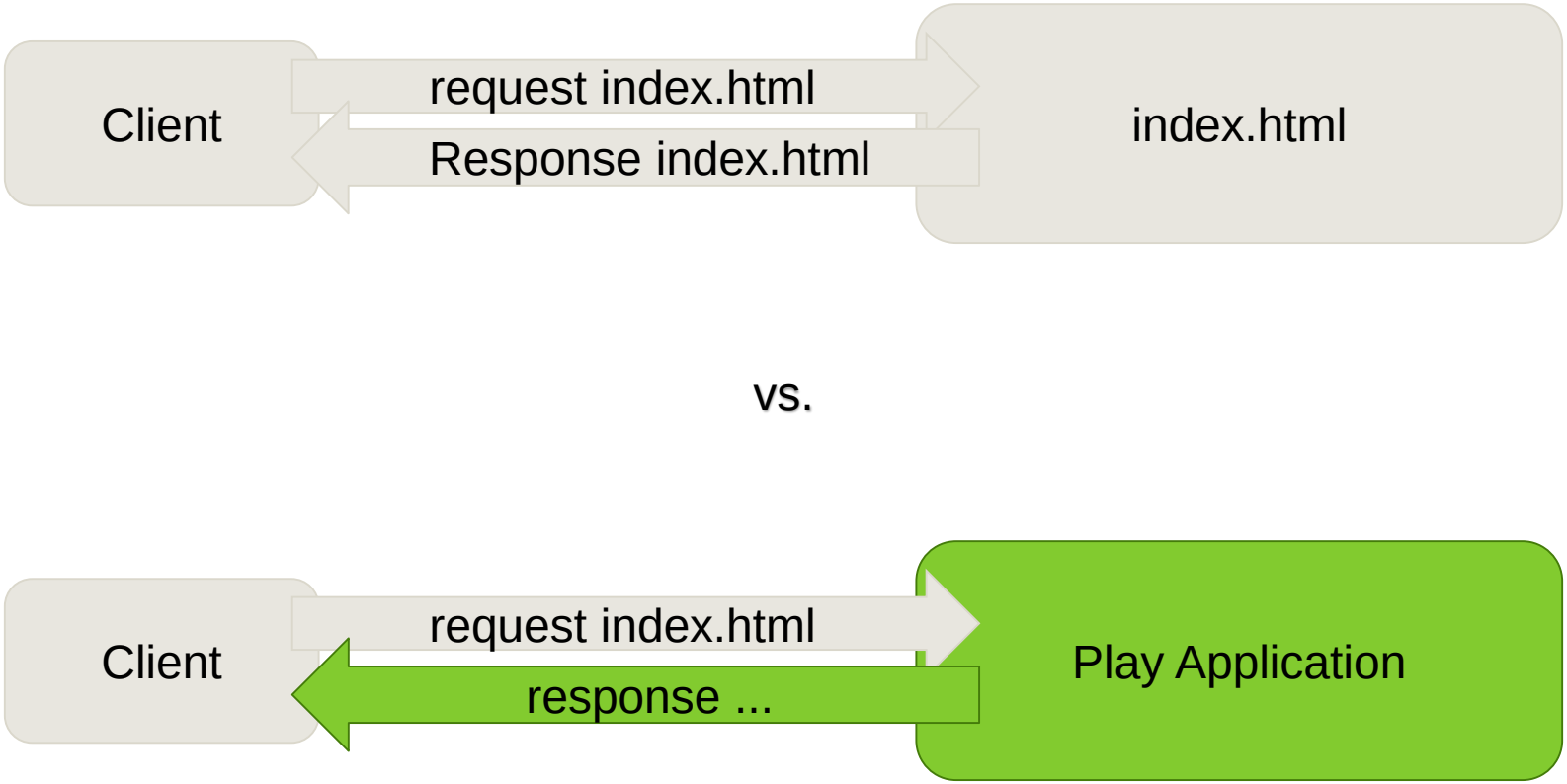
```
        modified:   conf/application.conf

no changes added to commit (use "git add" and/or "git commit -a")
Felixs-MacBook-Pro-2:angular-seed-play felix$ git update-index --assume-unchange
d conf/application.conf
Felixs-MacBook-Pro-2:angular-seed-play felix$ git status
On branch master
nothing to commit, working directory clean
```

e.g.
- build files
- log-files
- temp-files
- local (private) IDE configuration

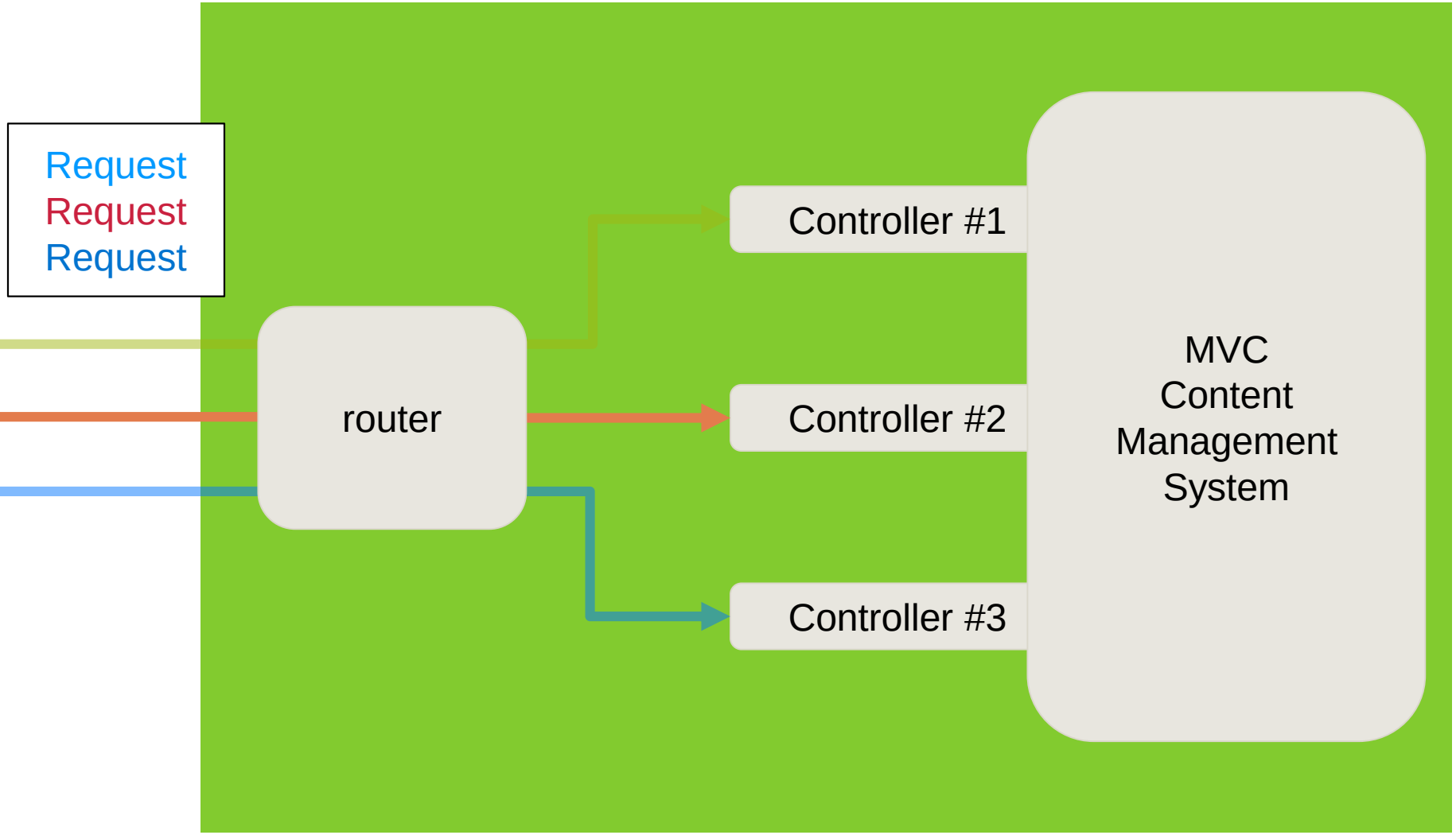
e.g.
- Database configuration
- Support Mail

Put everything in .gitignore that is not needed on a fresh git clone

Motivation to use a MVC-Framework

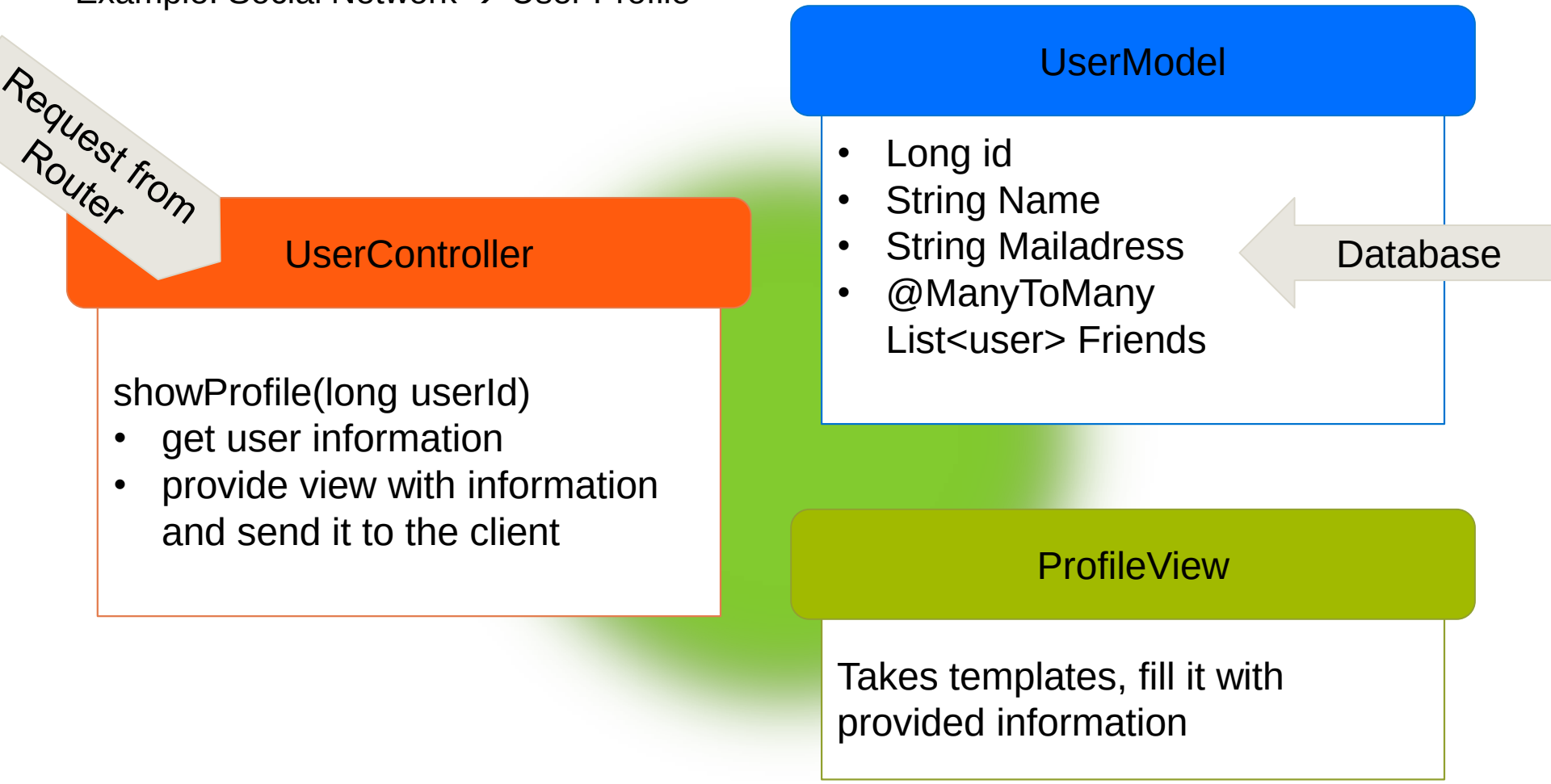


Handling user requests with play



Model View Controller Pattern

Example: Social Network → User Profile



Hands on! Let's play

- Install Intellij from Jetbrains
- Create Play Project with Intellij

  Create from existing template e.g. *Scala → Activator → hello-play-java*

  (you will need internet connection)

  or create an empty project *Java → Play 2.0*
- Run Play project

  right-click on any .java file → Run Play 2 App

All you need and where you can get it - part 1

sebis

Play Documentation

https://www.playframework.com/documentation/2.3.x/JavaHome

You need the documentation for Java developers (not Scala), also pay attention to the version numbers of the documentation and your app. They should match.

Intellij IDE

https://www.jetbrains.com/student/

Create a Student Account, Download the Ultimate Edition, activate the License with your Account data

Git Version Control

http://git-scm.com/

Already included in Intellij, but you might need it for git tools or command line usage

Bitbucket hosting

https://bitbucket.org/

Free online hosting of your git projects

All you need and where you can get it - part 2

Webjars

http://www.webjars.org/classic

easily manage client-side dependencies

SQLite DB Browser

http://sqlitebrowser.org/

You need the documentation for Java developers (not Scala)

All you need and where you can get it - part 3

sebis

HTML, javascript, CSS tutorials and refernces

http://www.w3schools.com/

jQuery

https://jquery.com/

Makes javascript coding more efficient (not always in a good way)

Sass Module for play

https://www.playframework.com/modules/sass-0.1/home

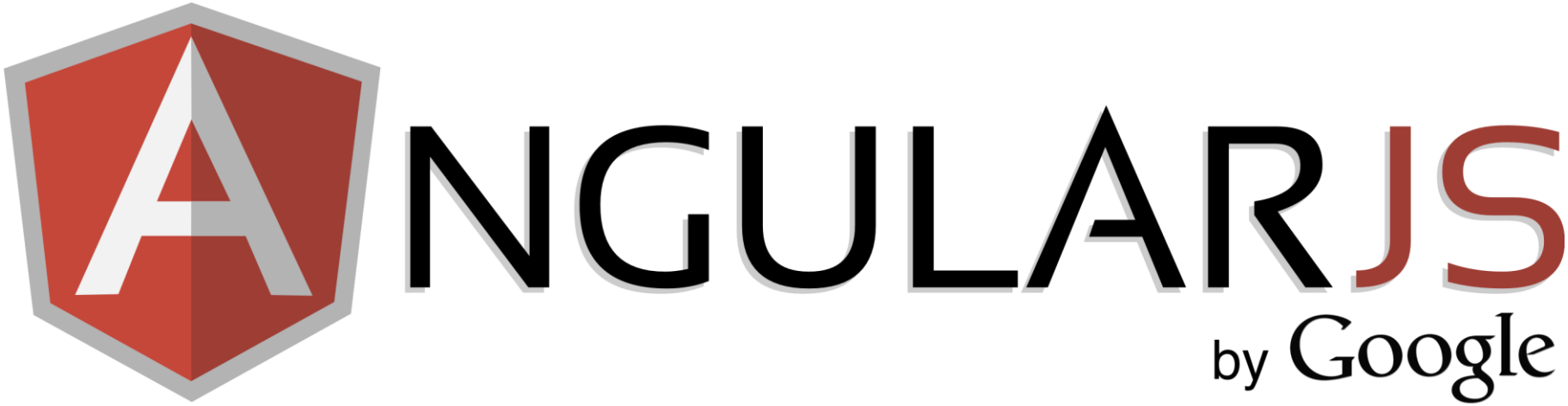
Makes CSS coding more efficient (in a good way). Also works great with bootstrap

Bootstrap

http://getbootstrap.com/

Makes layouting more efficient

Thank you for your attention!



Next week, same place, same time

How to build a real-time Web 2.0 Application with Angular JS