Introduction to Ruby on Rails

Ralf Teusner
ralf.teusner@hpi.de

Software Engineering II
WS 2018/19

Prof. Plattner, Dr. Uflacker
Enterprise Platform and Integration Concepts group

Introduction to Ruby on Rails

1. **Ruby & Ruby on Rails**
   - What is Ruby on Rails?
   - A few words about Ruby
   - Rails' core components
   - RESTful architecture
2. Your first Rails application
3. Your introductory Rails exercise

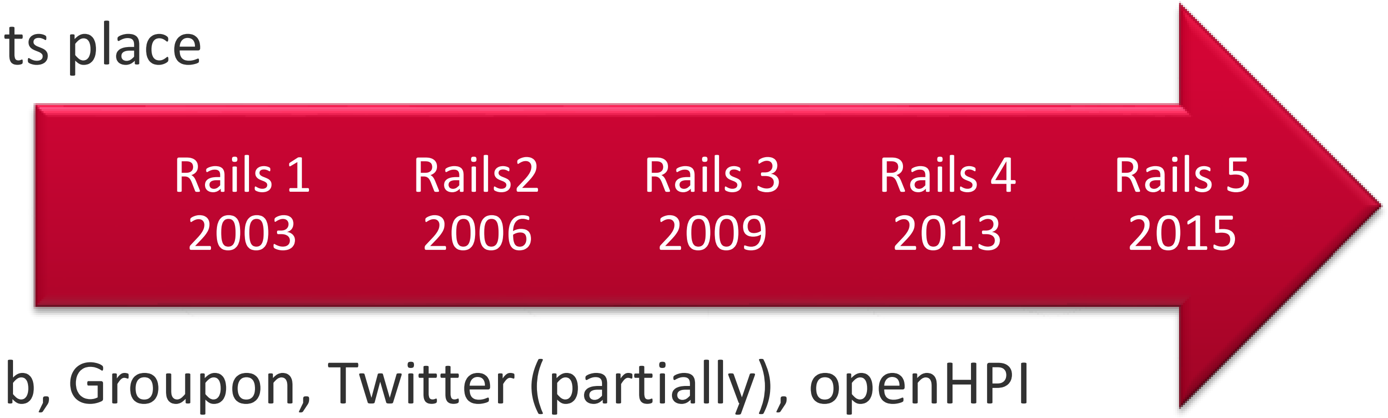
What is Ruby on Rails?

Web application development framework written in Ruby

- http://rubyonrails.org/

Philosophy

- "Don't repeat yourself" – DRY
- Convention over Configuration – there is "the Rails way"
- RESTful architecture
- Everything in its place



Rails 1
2003

Rails2
2006

Rails 3
2009

Rails 4
2013

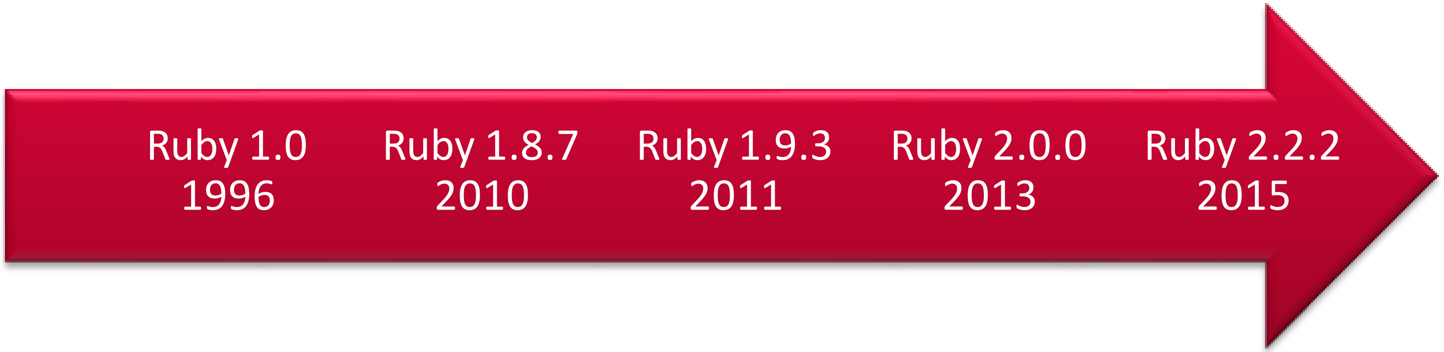
Rails 5
2015

- Used by Github, Groupon, Twitter (partially), openHPI

A few words about Ruby

http://www.ruby-lang.org/

- Dynamic, reflective, general-purpose, object-oriented
- Influenced by Perl, Smalltalk, Eiffel, and Lisp
- Open-source, mature software
- Matz’s Ruby Interpreter (MRI) versions:



Ruby 1.0
1996

Ruby 1.8.7
2010

Ruby 1.9.3
2011

Ruby 2.0.0
2013

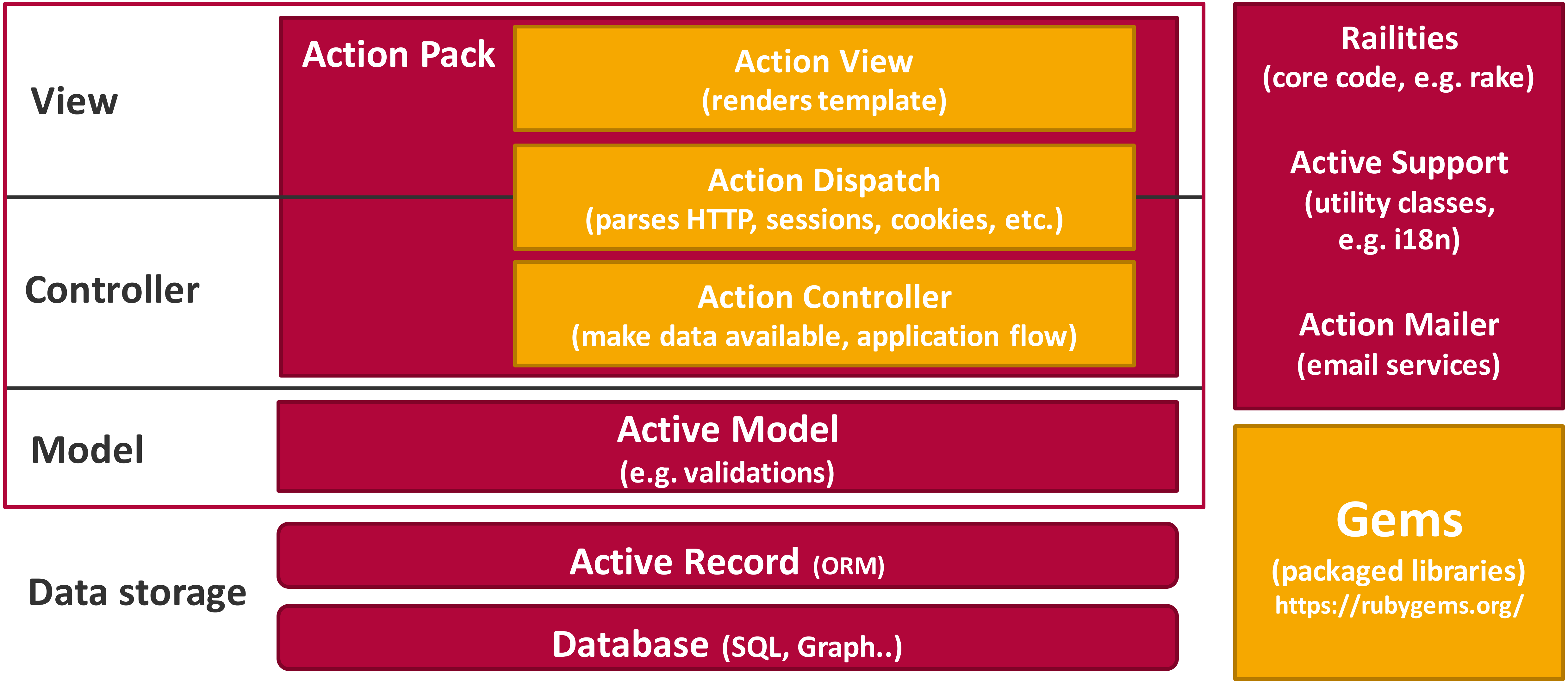
Ruby 2.2.2
2015

  - □ Additionally different VMs available (JRuby, Rubinius, IronRuby, Maglev)

Yukihiro "Matz" Matsumoto with R. Stallman

Image by Rubén Rodríguez (CC BY 3.0) - http://commons.wikimedia.org/wiki/File:Matz.jpg

Rails Core Components



View

Controller

Model

Data storage

Action Pack

Action View
(renders template)

Action Dispatch
(parses HTTP, sessions, cookies, etc.)

Action Controller
(make data available, application flow)

Active Model
(e.g. validations)

Active Record (ORM)

Database (SQL, Graph..)

Railities
(core code, e.g. rake)

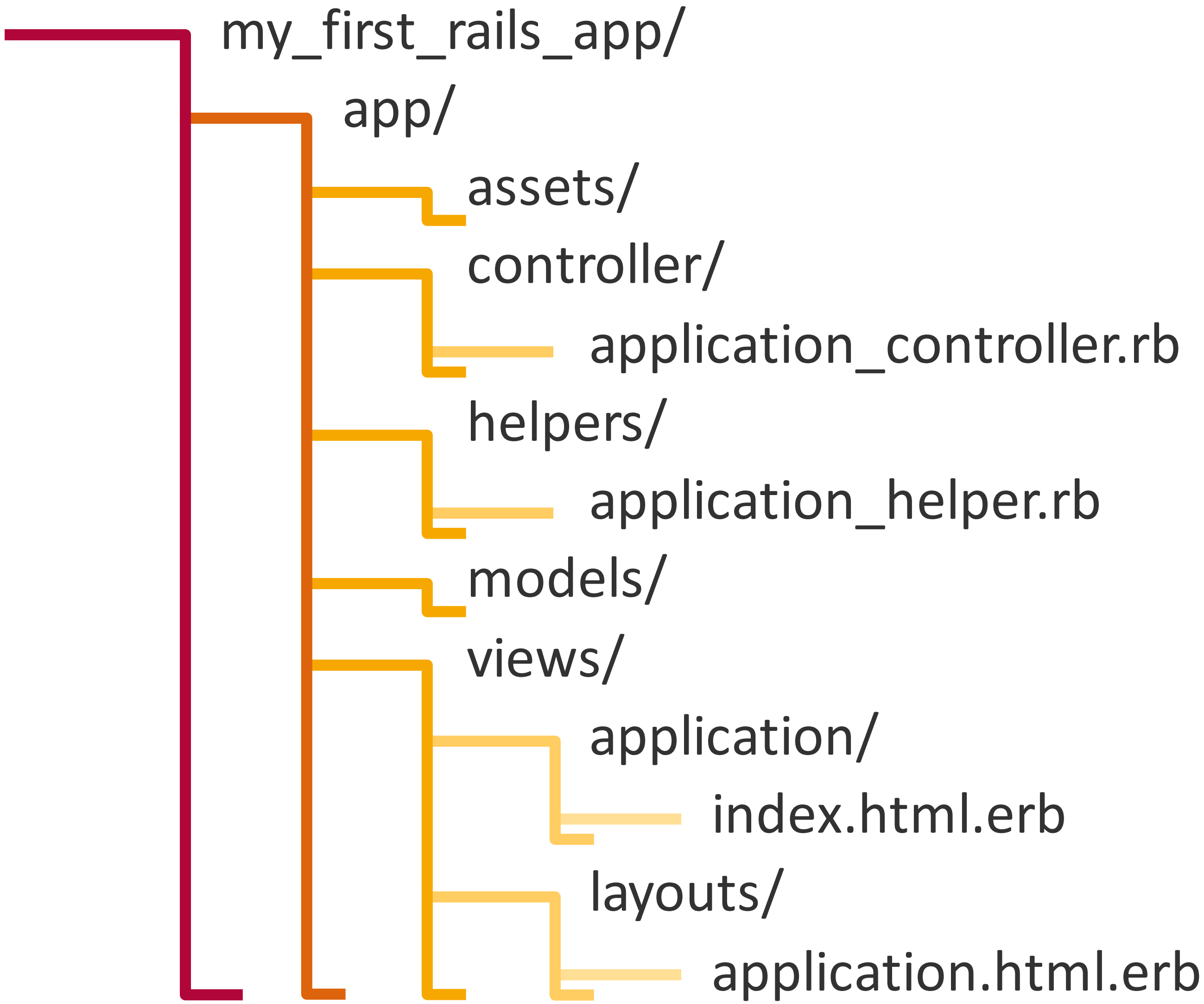
Active Support
(utility classes, e.g. i18n)

Action Mailer
(email services)

Gems
(packaged libraries)
https://rubygems.org/

http://adrianmejia.com/blog/2011/08/11/ruby-on-rails-architectural-design/

Rails Application Layout



```
my_first_rails_app/
    app/
        assets/
        controller/
            application_controller.rb
        helpers/
            application_helper.rb
        models/
        views/
            application/
                index.html.erb
            layouts/
                application.html.erb
```

RESTful Architecture

- **Re**presentational **S**tate **T**ransfer (REST) is a software architecture style for distributed systems
- Principles
  - Uniform Interface
  - Stateless Interactions
  - Cacheable
  - Clients and servers
  - Layered System
- Largest RESTful implementation: World Wide Web

RESTful Architecture – HTTP verbs

- REST supports all 4 HTTP 1.1 verbs: GET, PUT, POST, DELETE
- Differentiation of collections and individual elements

Resource	GET	PUT	POST	DELETE
Single element http://localhost:3000/authors/1	Retrieve	Update or create	Create	Delete
Collection http://localhost:3000/authors	List	Replace	Create	Delete

Examples of Routes

- GET / # invoke “home” controller
- GET /authors # retrieve a list of all authors
- GET /authors/new # get the form to enter a new author
- POST /authors # create a new author
- GET /authors/1 # show details of the first author
- GET /authors/1/edit # get the form to edit the first author
- PUT /authors/1 # update the first author
- DELETE /authors/1 # delete the first author

Introduction to Ruby on Rails

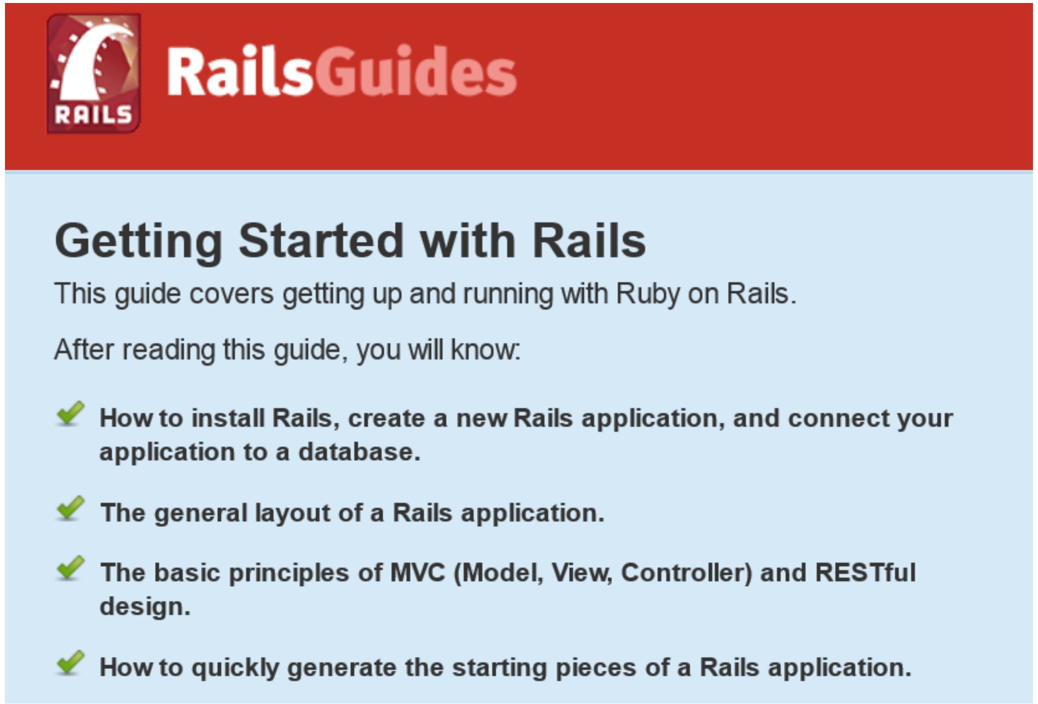
1. Ruby & Ruby on Rails
2. **Your first Rails application**
3. Your introductory Rails exercise

How to Start?

- **Option 1:** You use *Mac* or *Linux*
  - Install and use Ruby on Rails directly on your OS
  - Ruby version manager (e.g. RVM, rbenv) if older versions of Ruby should be kept
  - http://guides.rubyonrails.org/getting_started.html#installing-rails
  - Or use option 2
- **Option 2:** You have *Windows* or want to use a VM (recommended)
  - We prepared one for you via Vagrant (https://www.vagrantup.com/)
  - Uses VirtualBox in the backend (free on all platforms) (https://www.virtualbox.org/)
  - Use your own tools & editors, run the project in a headless VM
  - See project README for setup instructions
- **Option 3:** You have *Windows* and install Ruby on Rails directly on your OS
  - Tends to consume some time, might cause problems with certain dependencies
  - http://railsinstaller.org/en

Comprehensive RoR tutorial

Recommended to work through / read this hands-on tutorial. Seriously.
http://guides.rubyonrails.org/getting_started.html



Getting Started with Rails

This guide covers getting up and running with Ruby on Rails.

After reading this guide, you will know:

- ✓ How to install Rails, create a new Rails application, and connect your application to a database.
- ✓ The general layout of a Rails application.
- ✓ The basic principles of MVC (Model, View, Controller) and RESTful design.
- ✓ How to quickly generate the starting pieces of a Rails application.

Tip:

Before you start coding, make sure, the correct versions are installed.

```
$ ruby --version
$ rails --version
```

The following slides give a general overview

rails – Main executable

Start interactive shell to test out ideas

```
$ rails console
```

Start new rails application

```
$ rails new
```

Generate boilerplate for models, controllers & views

```
$ rails generate
```

Start the development server

```
$ rails server
```

Start a direct database shell

```
$ rails dbconsole
```

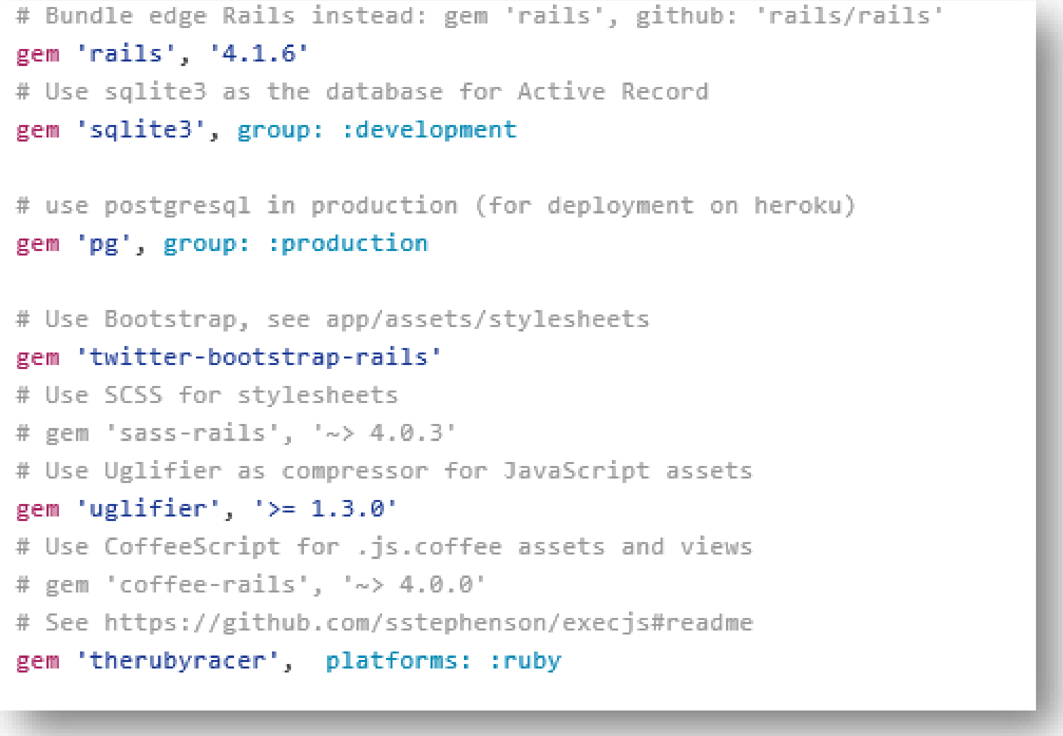
- Example: generate model, controller and view without controller specs

```
$ rails g scaffold author last_name:string
homepage:string --controller-specs false
```

http://guides.rubyonrails.org/command_line.html

Bundler – Ruby package manager

- Ruby libraries are packaged as "gems"
- Online repository at https://rubygems.org/
- Bundler resolves dependencies of gems
- `Gemfile` holds a list of required gems
  - □ Specify versions, e.g. gem 'rails' >= '4.1.6'
  - □ Alt. sources, e.g. :github => "tkowark/sawyer"
- `Gemfile.lock` is populated with resolved dependencies
  - □ Should be under version control



```
# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '4.1.6'
# Use sqlite3 as the database for Active Record
gem 'sqlite3', group: :development

# use postgresql in production (for deployment on heroku)
gem 'pg', group: :production

# Use Bootstrap, see app/assets/stylesheets
gem 'twitter-bootstrap-rails'
# Use SCSS for stylesheets
# gem 'sass-rails', '~> 4.0.3'
# Use Uglifier as compressor for JavaScript assets
gem 'uglifier', '>= 1.3.0'
# Use CoffeeScript for .js.coffee assets and views
# gem 'coffee-rails', '~> 4.0.0'
# See https://github.com/sstephenson/execjs#readme
gem 'therubyracer',  platforms: :ruby
```

Manually install a gem (Ruby package)

```
$ gem install
```

Install all gems listed as depencies in Gemfile

```
$ bundle install
```

Info:
`Gemfile.lock` contains all the actually installed versions of gems.

rake – Ruby make

List all available rake commands

```
$ rake -T
```

List all configured routes

```
$ rake routes # Rails 4
```

```
$ rails routes # Rails 5
```

Setup the database and run all migrations

```
$ rails db:setup db:migrate
```

Replace database with db layout from db/schema.rb
Do not run migrations.

```
$ rails db:schema:load
```

Run Rspec (testing framework for RoR) tests

```
$ bundle exec rspec
```

or

```
$ rspec
```

Tip:

Rails 5 changed some responsibilities. Tasks might be with 'rails' instead of 'rake' now

Info:

Running `schema:load` is advisable when setting up a completely new project. It is not intended to work around bad migrations.

Git – distributed version control system

- **Install Git:**
  - □ sudo apt-get install git
  - □ http://git-scm.com/ (Installers for all systems)

- **Setting up user name and email:**
  - □ Mandatory to commit changes
  - □ Use your github credentials!

```
$ git config --global user.email “vorname.nachname@student.hpi.de”
$ git config --global user.name “Max Mustermann”
```

- Alternative: setting parameters only for one project:

```
$ cd /path/to/your/project
$ git config user.email “vorname.nachname@student.hpi.de”
$ git config user.name “Max Mustermann”
```

Git workflow – commiting a change

Checkout remote repository to local copy

```
$ git clone https://github.com/hpi-swt2/sport-portal
```

Change main layout template app/views/layouts/application.html.erb

Stage changes (add files from working copy to repository index)

```
$ git add app/views/layouts
```

List changes to be committed

```
$ git status
```

Commit with commit messages. Reference Github issue #25

```
$ git commit –m "Fixed issue #25"
```

Fetch and merge changes from remote repository

```
$ git pull
```

Publish local commits

```
$ git push
```

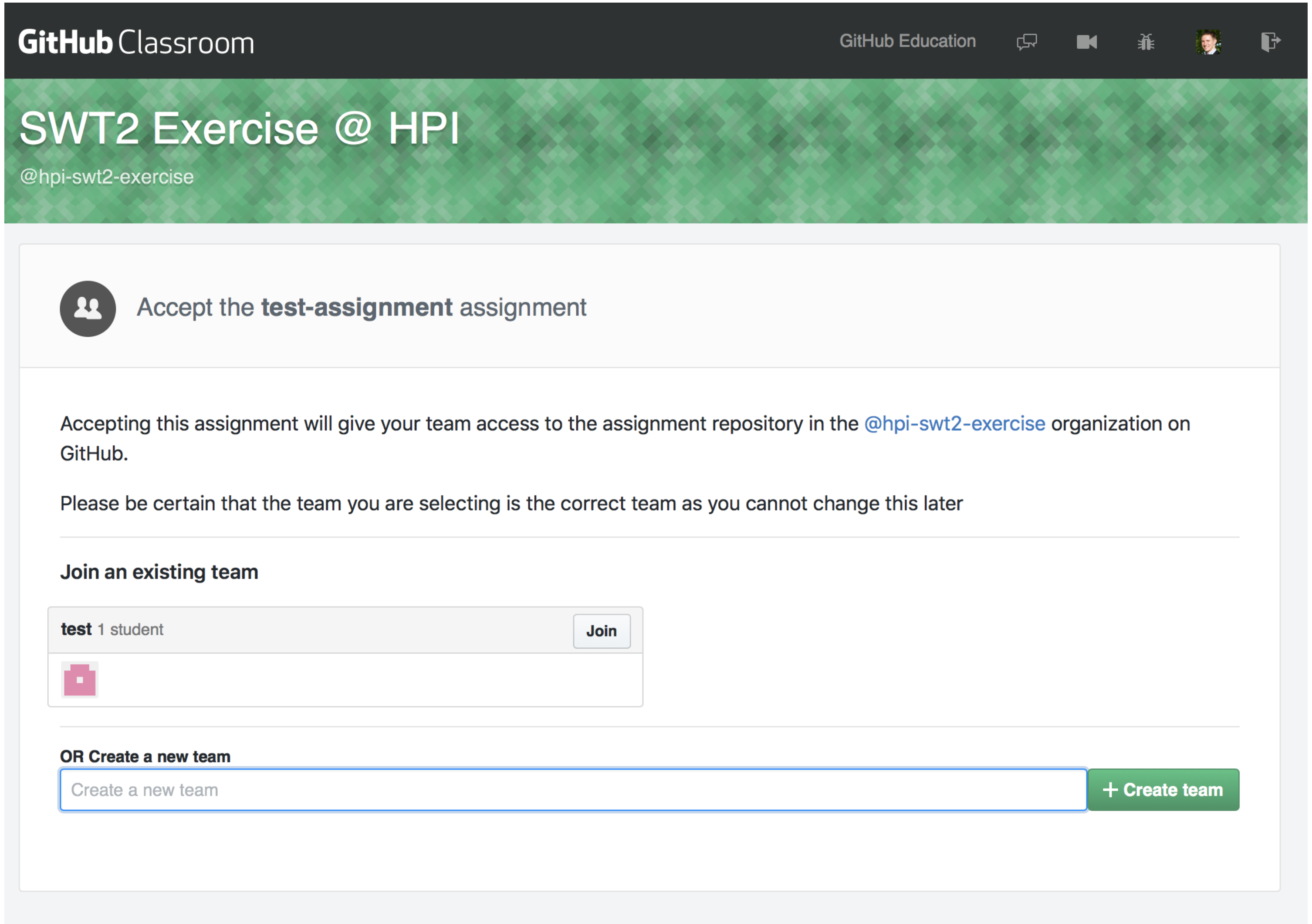
Introduction to Ruby on Rails

1. Ruby & Ruby on Rails
2. Your first Rails application
3. **Your introductory Rails exercise**
4. Additional Literature

Exercise – Your First Rails Project

- **Goals**
  - □ Get familiar with Ruby (on Rails)
  - □ Create necessary accounts for the project
- **Tasks**
  - □ (Create a Github account and get familiar with it)
  - □ Visit https://classroom.github.com/a/d1BMYkfA
  - □ Accept the assignment
  - □ Follow the instructions in the README file
- **Deadline**
  - □ Nov 9, 15:00 pm CET
  - □ POs are not required to complete this task

Exercise – Rails Project



SWT2 Exercise @ HPI

@hpi-swt2-exercise

Accept the **test-assignment** assignment

Accepting this assignment will give your team access to the assignment repository in the @hpi-swt2-exercise organization on GitHub.

Please be certain that the team you are selecting is the correct team as you cannot change this later

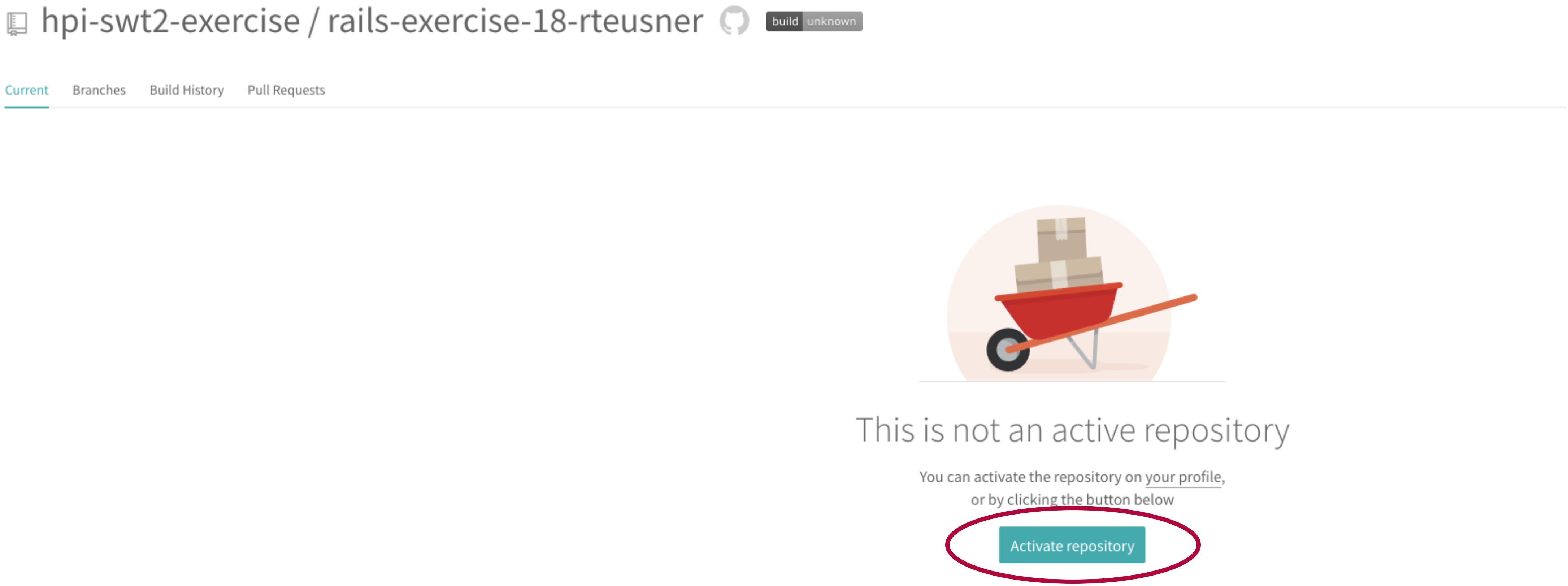
Join an existing team

test 1 student

OR Create a new team

Exercise – Travis CI

- **Travis CI**
  - Continuous Integration Server
  - *Experience?*

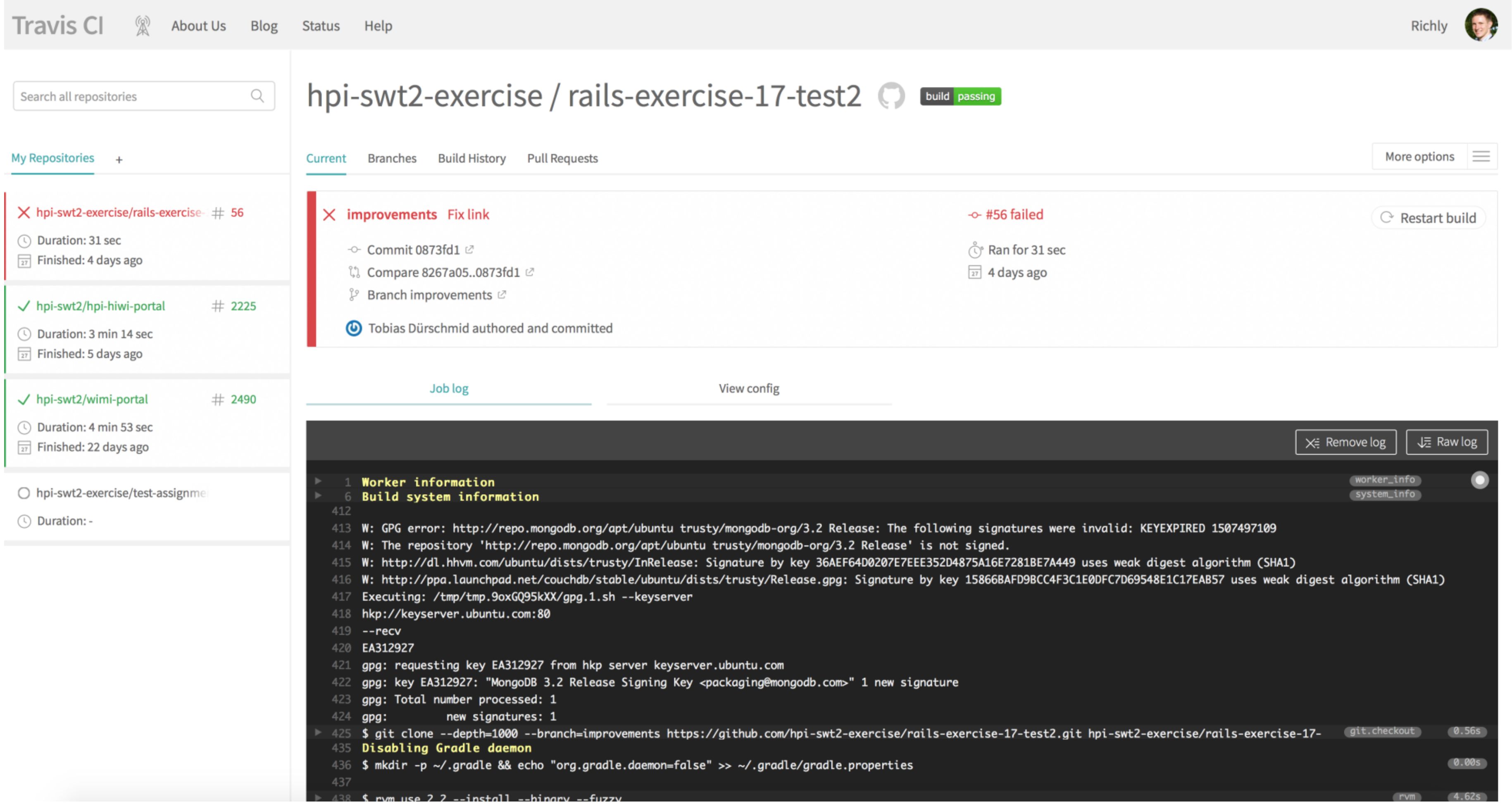


hpi-swt2-exercise / rails-exercise-18-rteusner

This is not an active repository

You can activate the repository on your profile, or by clicking the button below

Exercise – Travis CI



hpi-swt2-exercise/rails-exercise- # 56
Duration: 31 sec
Finished: 4 days ago

hpi-swt2/hpi-hiwi-portal # 2225
Duration: 3 min 14 sec
Finished: 5 days ago

hpi-swt2/wimi-portal # 2490
Duration: 4 min 53 sec
Finished: 22 days ago

hpi-swt2-exercise/test-assignme
Duration: -

hpi-swt2-exercise / rails-exercise-17-test2

build passing

improvements Fix link

Commit 0873fd1
Compare 8267a05..0873fd1
Branch improvements
Tobias Dürschmid authored and committed

#56 failed
Ran for 31 sec
4 days ago

```
Worker information
Build system information

W: GPG error: http://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.2 Release: The following signatures were invalid: KEYEXPIRED 1507497109
W: The repository 'http://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.2 Release' is not signed.
W: http://dl.hhvm.com/ubuntu/dists/trusty/InRelease: Signature by key 36AEF64D0207E7EEE352D4875A16E7281BE7A449 uses weak digest algorithm (SHA1)
W: http://ppa.launchpad.net/couchdb/stable/ubuntu/dists/trusty/Release.gpg: Signature by key 15866BAFD9BCC4F3C1E0DFC7D69548E1C17EAB57 uses weak digest algorithm (SHA1)
Executing: /tmp/tmp.9oxGQ95kXX/gpg.1.sh --keyserver
hkp://keyserver.ubuntu.com:80
--recv
EA312927
gpg: requesting key EA312927 from hkp server keyserver.ubuntu.com
gpg: key EA312927: "MongoDB 3.2 Release Signing Key <packaging@mongodb.com>" 1 new signature
gpg: Total number processed: 1
gpg:         new signatures: 1
$ git clone --depth=1000 --branch=improvements https://github.com/hpi-swt2-exercise/rails-exercise-17-test2.git hpi-swt2-exercise/rails-exercise-17-
Disabling Gradle daemon
$ mkdir -p ~/.gradle && echo "org.gradle.daemon=false" >> ~/.gradle/gradle.properties

$ rvm use 2.2 --install --binary --fuzzy
```

Next Steps

- **Get to know GitHub**
  - □ You probably already do
  - □ All the cool projects are there!
- **Familiarize yourselves with Ruby on Rails and web dev**
  - □ Read the Ruby on Rails guide (https://guides.rubyonrails.org/getting_started.html)
  - □ Understand the core Ruby language constructs (coming from other languages) (https://www.ruby-lang.org/en/documentation/ruby-from-other-languages/)
- **Start the exercise once released**

Introduction to Ruby on Rails

1. Ruby & Ruby on Rails
   - What is Ruby on Rails?
   - Rails' core components
   - RESTful architecture
2. Your first Rails application
   - Folder structure
   - Databases
   - `rails`, `git`
3. Your introductory Rails exercise
   - Using Travis CI

Questions?

Ralf Teusner
ralf.teusner@hpi.de

Software Engineering II
WS 2018/19

Prof. Plattner, Dr. Uflacker
Enterprise Platform and Integration Concepts group