Introduction to Ruby on Rails

Keven Richly
keven.richly@hpi.de

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

Software Engineering II
WS 2017/18
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/](http://rubyonrails.org/)

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

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)
Rails Core Components

View
- Action Pack
  - Action View (renders template)
- Action Dispatch (parses HTTP, sessions, cookies, etc.)

Controller
- Action Controller (make data available, application flow)

Model
- Active Model (e.g. validations)

Data storage
- 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/

Introduction to Ruby on Rails — Software Engineering II — WS 2017/18

RESTful Architecture

- **Representational State Transfer (REST)** is a software architecture style for distributed systems.

- **Principles**
  - Uniform Interface
  - Stateless Interactions
  - Cacheable
  - Clients and servers
  - Layered System
  - Code on Demand (optional)

- 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

<table>
<thead>
<tr>
<th>Resource</th>
<th>GET</th>
<th>PUT</th>
<th>POST</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single element <a href="http://localhost:3000/authors/1">http://localhost:3000/authors/1</a></td>
<td>Retrieve</td>
<td>Update or create</td>
<td>Create</td>
<td>Delete</td>
</tr>
<tr>
<td>Collection <a href="http://localhost:3000/authors">http://localhost:3000/authors</a></td>
<td>List</td>
<td>Replace</td>
<td>Create</td>
<td>Delete</td>
</tr>
</tbody>
</table>

Oktober 23, 2017
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
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](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/](https://www.vagrantup.com/))
  □ Uses VirtualBox in the backend (free on all platforms) ([https://www.virtualbox.org/](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
Comprehensive RoR tutorial

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

Tip:
Before you start coding, make sure, the correct versions are installed.
$ ruby --version
$ rails --version

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.

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](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

Manually install a gem (Ruby package)

$ gem install

Install all gems listed as dependencies in Gemfile

$ bundle install

---

# 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

---

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

Setup the database and run all migrations

$ rake db:setup db:migrate

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

$ rake db:schema:load

Run Rspec (testing framework for RoR) tests

$ rake spec

or

$ rspec

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 - committing 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)
  - Visit [https://classroom.github.com/g/vWYU5y08](https://classroom.github.com/g/vWYU5y08)
  - Follow the instructions in the readme

- Deadline
  - Nov 10, 11:59 pm CET (firm)
  - POs are exempt from completing this task.
Exercise – Your First Rails Project

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

<table>
<thead>
<tr>
<th>test</th>
<th>1 student</th>
<th>Join</th>
</tr>
</thead>
</table>

OR Create a new team

Create a new team

+ Create team
Exercise – Your First Rails Project
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
   - rails, rake, git

3. Your introductory Rails exercise
   - On Github