HPI Hasso Plattner Institut

Your Project

Software Engineering II
WS 2019/20

Enterprise Platform and Integration Concepts

SWT2 Schedule

Project

- Sprint 1 Planning: 11.-15.11.
- Sprint 1 Review & Retrospective: 18.-22.11.
- **Weekly Stand-ups** in weeks when there are no other Scrum meetings
  - Timeboxed. 2 minutes per person. Over lunch?

Meetings

- Exact dates for meetings should be **negotiated** with your tutor
- Sprint Review and Planning for next sprint can be colocated
- You are a team of software professionals

- On demand: User Research with Customer. Clarify questions!

Let’s get started

POs

- Extract requirements + create user stories (GitHub tickets)
- Get an idea of the interaction workflows in the system (**mockups**?)
- Prepare Sprint Planning, inform yourselves on what the team is doing
- PO should roughly know what the team has done **before the review**

Developers

- Clone repository, get application working, **understand architecture and dependencies**
- **If you were lead architect, how would you construct this?**
  - *What are some challenges with the problem domain?*
- **Play around** and try things out.
  - *Where does the system have problems? What makes no sense to you?*

Let’s get started

SMs

- Part of your job is research and retrospection
  - Observe the meta-level of a meeting
  - Equal participation? Focus on problems? Communication strategies? Agenda?
  - This is a hard job, **focus on it**
  - What is working well in the team? What isn't?
  - What are good ways of how a team meeting can be structured?
- Every team is different. **Experiment!**

Meeting spaces

- Regular meeting + work timeslot
- We can reserve spaces in the Villa at the EPIC chair, if this is needed

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Project Infrastructure

"Utility pole in Curitiba" by Leonardo.stabile - Own work. Licensed under Public domain via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Utility_pole_in_Curitiba.JPG#mediaviewer/File:Utility_pole_in_Curitiba.JPG

Communication Infrastructure

- Email lists
  - If you want: separate lists for each team (*lists.myhpi.de*)
  - Keep your teammates in the loop, add teaching team
  - Rules and filters help organizing your inbox
- **swtii2019.slack.com**
- Ticket system for overview and feedback about current tasks and progress
- Wiki for lean and globally **accessible documentation**
- Telephone and personal contact for direct communication
- … be **creative**!

  (but let us know, we are interested in learning what might be useful in the future)

Project Management Tools

The Swiss army knife of software development

- Integrating tools for most common activities in **one place**
- Wiki, bug tracking, time management, project analytics, discussions, …

- Examples:
  - Microsoft Team Foundation Server
  - Redmine, Plan.io (SaaS based on Redmine)
  - Gitlab
  - **GitHub**

Version Control System

Repository to store development artifacts

Features:

- Versioning
- Dealing with variants: **branches**
- Access control
  - Authentication, authorization
  - Locking
  - **Concurrent** development
- Reporting and communication
  - How many versions, variants, changes, persons
  - History of changes

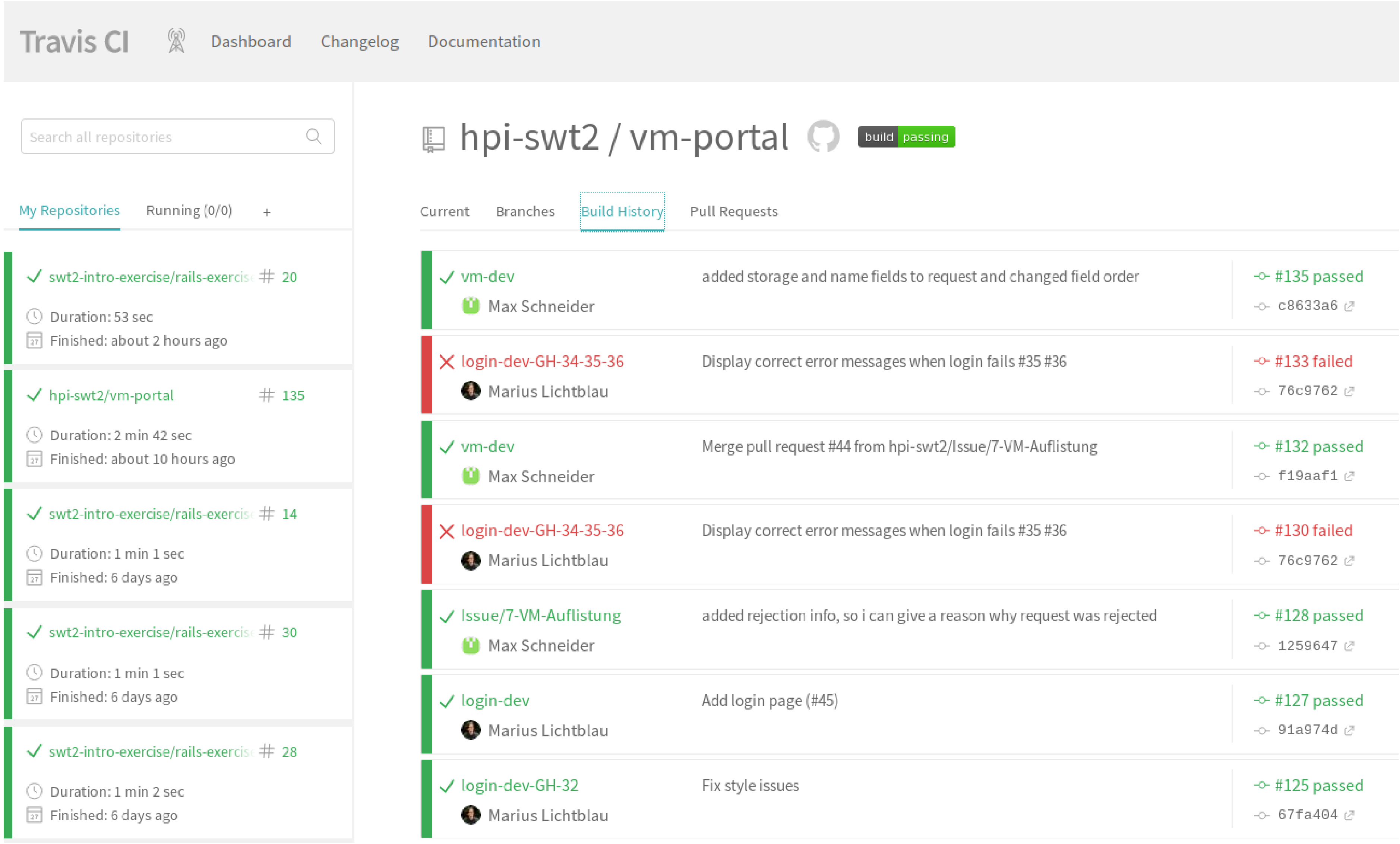
Continuous Integration

How do you make sure your software **always works**?

Continuous Integration!

- **Connected** to version control
- Customizable run scripts
- Ideally covering all development branches
- Examples:
  - CruiseControl
  - Anthill
  - Jenkins/Hudson
  - **Travis CI**

Travis CI



Travis CI Dashboard Changelog Documentation

Search all repositories

My Repositories Running (0/0) +

✓ swt2-intro-exercise/rails-exercis # 20
Duration: 53 sec
Finished: about 2 hours ago

✓ hpi-swt2/vm-portal # 135
Duration: 2 min 42 sec
Finished: about 10 hours ago

✓ swt2-intro-exercise/rails-exercis # 14
Duration: 1 min 1 sec
Finished: 6 days ago

✓ swt2-intro-exercise/rails-exercis # 30
Duration: 1 min 1 sec
Finished: 6 days ago

✓ swt2-intro-exercise/rails-exercis # 28
Duration: 1 min 2 sec
Finished: 6 days ago

hpi-swt2 / vm-portal

build passing

Current Branches Build History Pull Requests

✓ vm-dev Max Schneider	added storage and name fields to request and changed field order	#135 passed c8633a6
✗ login-dev-GH-34-35-36 Marius Lichtblau	Display correct error messages when login fails #35 #36	#133 failed 76c9762
✓ vm-dev Max Schneider	Merge pull request #44 from hpi-swt2/Issue/7-VM-Auflistung	#132 passed f19aaf1
✗ login-dev-GH-34-35-36 Marius Lichtblau	Display correct error messages when login fails #35 #36	#130 failed 76c9762
✓ Issue/7-VM-Auflistung Max Schneider	added rejection info, so i can give a reason why request was rejected	#128 passed 1259647
✓ login-dev Marius Lichtblau	Add login page (#45)	#127 passed 91a974d
✓ login-dev-GH-32 Marius Lichtblau	Fix style issues	#125 passed 67fa404

Application Deployment

How can you always have a **running version** available?
(why would you want to?)

Deploy your application!

- Simple solution: test deployment on local machine
- Deployment on separate machine:
  - Dedicated Servers
  - Infrastructure-as-a-Service
  - Platform-as-a-Service, e.g. Heroku
- **Continuous Deployment:**
  Deployment **automatically** triggered by successful CI build
  - Deployment config is part of the project
  - No extra effort

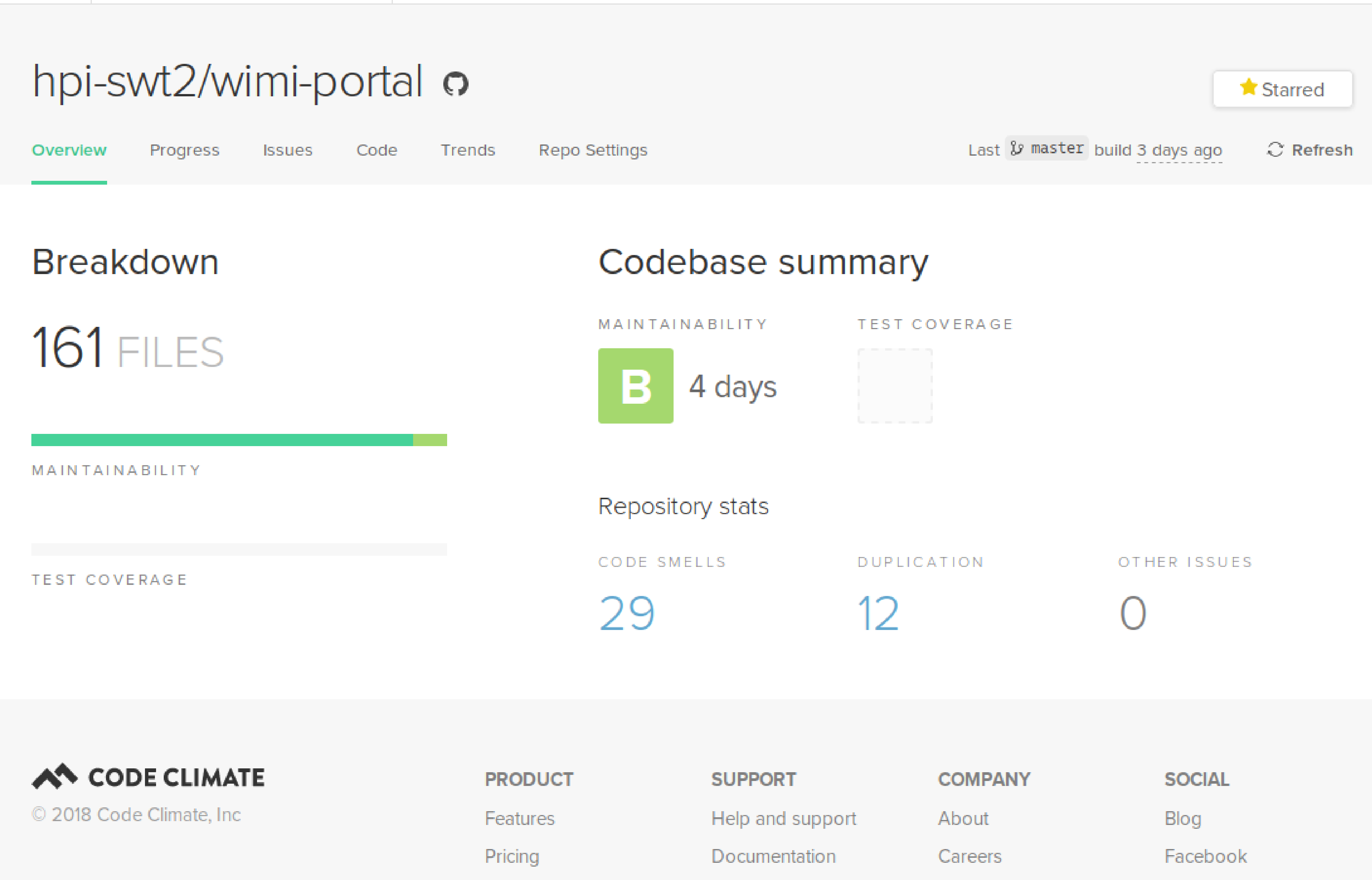
Code Quality

How can you ensure that the software adheres to certain **quality standards** (complexity, test coverage, etc.)?

Check for compliance

- Self-control, code reviews
- **Automatic** checks
  - □ Hosted tools: e.g. CodeClimate, Codefactor, Codebeat
  - □ Local code coverage: SimpleCov (http://www.simplecov.org/)
    - Can run automatically during each test run
    - coverage/index.html in your application folder
  - □ Local code smells: RuboCop (https://www.rubocop.org)

Hosted Code Quality Checks



hpi-swt2/wimi-portal

☆ Starred

Overview Progress Issues Code Trends Repo Settings

Last master build 3 days ago

Refresh

Breakdown

161 FILES

MAINTAINABILITY

TEST COVERAGE

Codebase summary

MAINTAINABILITY

B 4 days

TEST COVERAGE

Repository stats

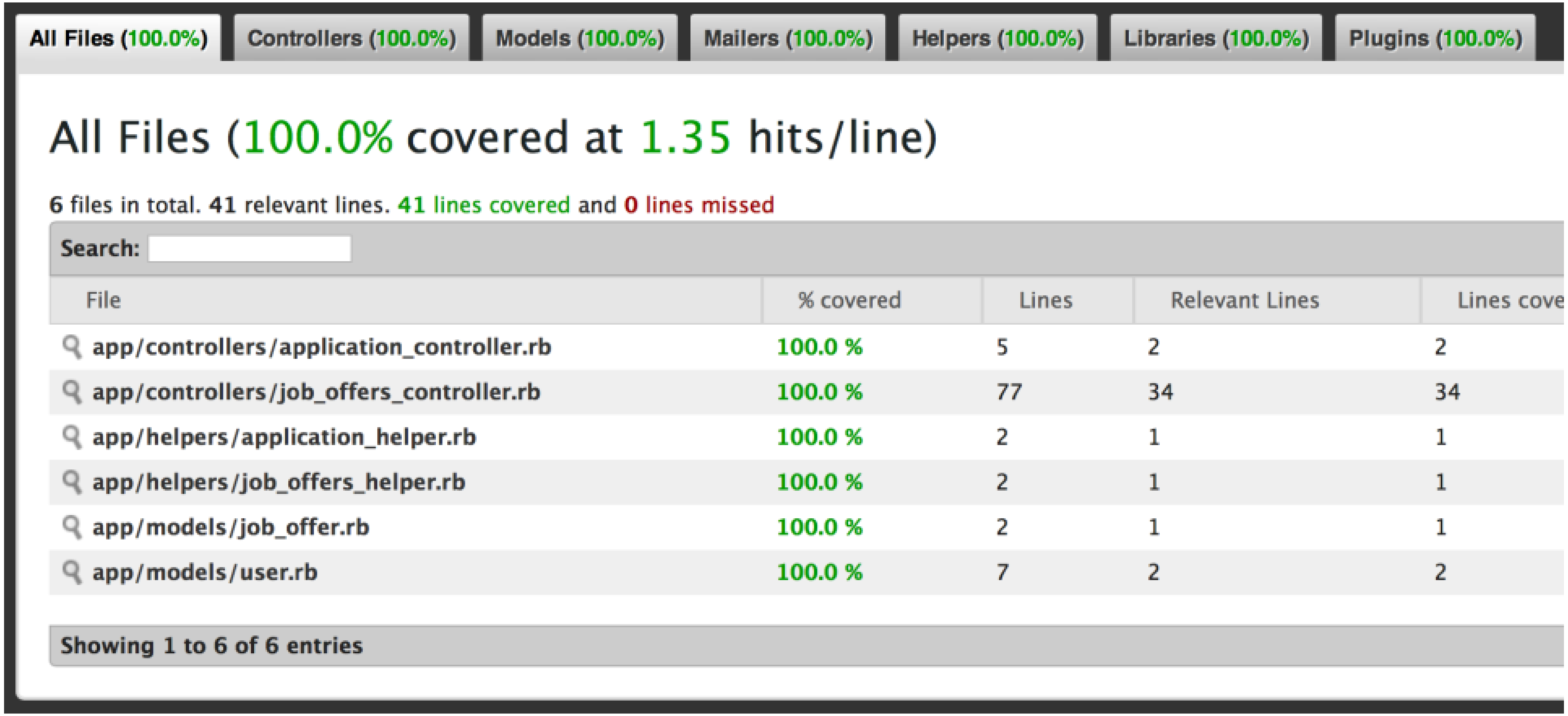
CODE SMELLS	DUPLICATION	OTHER ISSUES
29	12	0

CODE CLIMATE

© 2018 Code Climate, Inc

PRODUCT	SUPPORT	COMPANY	SOCIAL
Features	Help and support	About	Blog
Pricing	Documentation	Careers	Facebook

SimpleCov Code Coverage



All Files (100.0% covered at 1.35 hits/line)

6 files in total. 41 relevant lines. 41 lines covered and 0 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines cove
app/controllers/application_controller.rb	100.0 %	5	2	2
app/controllers/job_offers_controller.rb	100.0 %	77	34	34
app/helpers/application_helper.rb	100.0 %	2	1	1
app/helpers/job_offers_helper.rb	100.0 %	2	1	1
app/models/job_offer.rb	100.0 %	2	1	1
app/models/user.rb	100.0 %	7	2	2

Showing 1 to 6 of 6 entries

Dependencies

I want to solve a specific (web-dev) problem.

More than likely someone else has had this problem and has solved it.

Libraries (Ruby gems) & external dependencies

- Most likely more mature and bug-free than your custom solution
- Someone else has checked the code in your application
- You are maximizing development time
- **But:** dependencies introduce complexity
  - Mostly very powerful, generic solutions
  - Require extensive configuration
  - Need to be learned by every developer (effort x 26)
  - Consensus among developers on employment?

Your Project

1. Communication infrastructure
2. Continuous Integration
3. Continuous Deployment
4. Code Quality
5. Dependencies

6. Any other tools you might want to use!

Something you have had good experiences with in the past?

Your favorite development tool?

But, your team is not the only one using it, communicate.