

# Clean Software Repositories

Best practices for software repositories and bundled platforms.

Ole Reglitzki

May 20, 2017

- 1 Struktur
- 2 Where should it be?
- 3 What is it for?
- 4 Management practices

# 1 Stuktur

2 Where should it be?

3 What is it for?

4 Management practices

- Software repository

- Software repository
- Issue tracker

- Software repository
- Issue tracker
- Merge Requests

- Software repository
- Issue tracker
- Merge Requests
- Continuous Integration (Continuous Delivery)

- Software repository
- Issue tracker
- Merge Requests
- Continuous Integration (Continuous Delivery)
- Continuous Deployment



- Software repository
- Issue tracker
- Merge Requests
- Continuous Integration (Continuous Delivery)
- Continuous Deployment
- Wiki

- Software repository
- Issue tracker
- Merge Requests
- Continuous Integration (Continuous Delivery)
- Continuous Deployment
- Wiki
- "Pages"

- Software repository
- Issue tracker
- Merge Requests
- Continuous Integration (Continuous Delivery)
- Continuous Deployment
- Wiki
- "Pages"
- Snippets

1 Struktur

2 Where should it be?

3 What is it for?

4 Management practices

# Documentation

## Documentation

- ⇒ API documentation: Wiki
- ⇒ Code documentation: Repository (within source code)

## Pending features (TODOs)

## Pending features (TODOs)

⇒ Issue tracker



## Software preview/presentation

## Software preview/presentation

- ⇒ Short: Repository (README.md)
- ⇒ Long/Demo: "Pages"
- ⇒ Demo/Custom Domain: (Dedicated) Server

License text

License text

⇒ Repository

Screenshot (1080p)

## Screenshot (1080p)

- ⇒ Static distribution
- ⇒ `git lfs`

# Changelog

## Changelog

⇒ Repository



## CI definition

## CI definition

⇒ Repository

## Sample applications

## Sample applications

- ⇒ Short: Snippets
- ⇒ Medium: Multi-file snippet platforms
- ⇒ Long: Separate repositories

## Benchmarks

## Benchmarks

⇒ Wiki

⇒ Repository (README.md) if it's your entire marketing :D

- 1 Struktur
- 2 Where should it be?
- 3 What is it for?
- 4 Management practices

## Environment Variables

Private data, you may document them and set defaults thought.

- Runtime specific stuff (PORT, SECRET,...)


















## Repository
















- Source code
- Test code
- Files with special meaning

**NOT** Dependency source code or binaries (if possible to resolve via buildfile)

**NOT** Large files (`git lfs` is great!)

**NOT** Passwords (if still in use)

Name	Last commit > <a href="#">94954fee</a>  a week ago - Release 0.3.1   <a href="#">History</a>	Last Update
 assets	Add readme preview	5 months ago
 bench	Add benchmark limited-queue	6 months ago
 examples	Add readme example	5 months ago
 legacy	Update legacy README.md	5 months ago
 lib	Merge branches '#27' and '#28' into development	4 months ago
 test	Rename BacktrackTransport to ThresholdTransport	5 months ago
 .eslintrc	Remove indent rules from eslint	8 months ago
 .gitignore	Update legacy build process	5 months ago
 .gitlab-ci.yml	Add linter to CI	8 months ago
 CHANGELOG.md	Release 0.3.1	a week ago
 CONTRIBUTING.md	Add issue template	7 months ago
 LICENSE	Update LICENSE	8 months ago
 README.md	Add cheeky flag	5 months ago
 package.json	Release 0.3.1	a week ago

Name	Last commit > <a href="#">94954fee</a>  a week ago - Release 0.3.1   <a href="#">History</a>	Last Update
 assets	Add readme preview	5 months ago
 bench	Add benchmark limited-queue	6 months ago
 examples	Add readme example	5 months ago
 legacy	Update legacy README.md	5 months ago
 lib	Merge branches '#27' and '#28' into development	4 months ago
 test	Rename BacktrackTransport to ThresholdTransport	5 months ago
 .eslintrc	Remove indent rules from eslint	8 months ago
 .gitignore	Update legacy build process	5 months ago
 .gitlab-ci.yml	Add linter to CI	8 months ago
 CHANGELOG.md	Release 0.3.1	a week ago
 CONTRIBUTING.md	Add issue template	7 months ago
 LICENSE	Update LICENSE	8 months ago
 README.md	Add cheeky flag	5 months ago
 package.json	Release 0.3.1	a week ago

## README.md

Presentate of your software!

- Badges are great (e.g. <https://img.shields.io/>)
- KISS
- Some people advise to spend as much time on your README.md as on your code.
- I wouldn't go so far. Impress with great code, but don't scare away with ugly README.md

README.md

## oddlog - Object driven data logging

license MIT npm v0.3.1 build success coverage 66% downloads 250/month

oddlog is a fast JSON object based logging library for node.js.

Being restricted to strings for logging sucks. So we use arbitrary JSON objects instead.

```
const oddlog = require("oddlog");
const log = oddlog.createLogger("my-app"), child = log.child({user: "max"});
log
  .handleUncaughtExceptions()
  .info({random: 42}, "Application startup");
child
  .warn({method: "terminate"}, "Procedure initiated.");

throw new Error("Nah, nothing failed!");
```

```
readme ▶ node lib/preview.js | oddlog -vvv
[07:20:45.572] INFO: my-app Application startup {random:42}
[07:20:45.575] WARN: my-app Procedure initiated. {method:"terminate", user:"max"}
[07:20:45.577] ERROR: my-app Uncaught exception
Error: Nah, nothing failed!
    at Object.<anonymous> (/home/ole/src/gitlab.com/frisssidiegurke/oddlog/examples/readme/lib/preview.js:12:7)
    at Module._compile (module.js:571:32)
    at Object.Module._extensions..js (module.js:588:10)
    at Module.load (module.js:488:32)
    at tryModuleLoad (module.js:447:12)
    at Function.Module._load (module.js:439:3)
    at Module.runMain (module.js:685:10)
    at run (bootstrap_node.js:420:7)
    at startup (bootstrap_node.js:139:9)
    at bootstrap_node.js:535:3
```

### Table of Contents

- [Status Quo](#)
- [Installation](#)

## linter files

Ensure code quality!

- Use linters (depends on language)
- Continuous Delivery

`.*ignore`

Prevent inappropriate files!

- Trailing / for directories
- Leading / for root-only data
- /local/ directory

CHANGELOG.md

Your users will thank you.

- <http://keepachangelog.com/en/0.3.0/>



## CONTRIBUTING.md

Encourage contribution!



[2]

- Ways of contribution
  - Spread the word
  - Report issues
  - Feature requests
- ...
- Development setup
- Code (quality) standards
  - E.g. [chris.beams.io](#) commit message recommendations or [my variant](#)
  - Branching practice
- Recognition of contributors
- Contact (e.g. IRC channel)

## LICENSE

Permit according to your ethics!

- No, I will not give examples; I'm not suicidal!
- List of commonly found licenses: <https://spdx.org/licenses/>
- TL;DR Legal

## Issue tracker

- Report bugs
- Pending/Requested features
- Define Milestones
- Use the Assignee feature
- Swimlanes, Story points, etc.
- Be friendly!

## Merge Requests

- Check changed files before merge
- Clone the fork for testing
- Be friendly, but consequent!

## Continuous Integration/Delivery

- Run your linter
- Run your tests
- Keep a badge within your `README.md`

## Continuous Deployment

- Use webhooks
  - Use verification token/secret
- **Git-Auto-Deploy**
  - Simple
  - Full power of your shell

## Wiki

- Installation and setup
- API documentation
- Usage guides
- General suggestions

## Wiki

- Installation and setup
- API documentation
- Usage guides
- General suggestions

If wiki features are not sufficient:

- ⇒ Setup external wiki
- ⇒ Put a link into `README.md` and the bundled wiki (don't show anything else in there).



## "Pages"

- Static content
- *Better* presentation of your software
- Demo of your product (for client-side web software)

## "Pages"

- Static content
- *Better* presentation of your software
- Demo of your product (for client-side web software)

Prefer a plain README.md for presentation:

- KISS
- Always cloned aside your source code (and distributions)

## Snippets

- Simple usage samples
- Related samples (e.g. `nginx/systemd/whatever` files)

- 1 Struktur
- 2 Where should it be?
- 3 What is it for?
- 4 Management practices**

## Checklist:

- Use proper versioning.
  - E.g. <http://semver.org/>
  - Avoid private repositories; don't be ashamed.

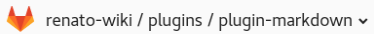
## Checklist:

- Use proper versioning.
  - E.g. <http://semver.org/>
  - Avoid private repositories; don't be ashamed.
- Name properly
  - Same name for repository and package.
  - Avoid case sensitivity  $\Rightarrow$  use lower-case.
  - Avoid spaces.
  - Consider the best name scope.

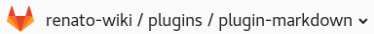


renato-wiki / plugins / plugin-markdown  $\downarrow$

## Checklist:

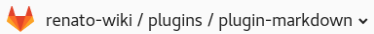
- Use proper versioning.
  - E.g. <http://semver.org/>
  - Avoid private repositories; don't be ashamed.
- Name properly
  - Same name for repository and package.
  - Avoid case sensitivity  $\Rightarrow$  use lower-case.
  - Avoid spaces.
  - Consider the best name scope. 
- Disable unused features.
  - E.g. if your project has been superseded, make this clear within the README.md and disable issue reports.

## Checklist:

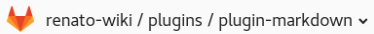
- Use proper versioning.
  - E.g. <http://semver.org/>
  - Avoid private repositories; don't be ashamed.
- Name properly
  - Same name for repository and package.
  - Avoid case sensitivity  $\Rightarrow$  use lower-case.
  - Avoid spaces.
  - Consider the best name scope. 
- Disable unused features.
  - E.g. if your project has been superseded, make this clear within the README.md and disable issue reports.
- Utilize the project description, icon, etc.



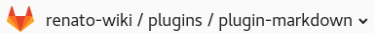
## Checklist:

- Use proper versioning.
  - E.g. <http://semver.org/>
  - Avoid private repositories; don't be ashamed.
- Name properly
  - Same name for repository and package.
  - Avoid case sensitivity  $\Rightarrow$  use lower-case.
  - Avoid spaces.
  - Consider the best name scope. 
- Disable unused features.
  - E.g. if your project has been superseded, make this clear within the README.md and disable issue reports.
- Utilize the project description, icon, etc.
- You may or may not want to create templates for new issues and merge requests.

## Checklist:

- Use proper versioning.
  - E.g. <http://semver.org/>
  - Avoid private repositories; don't be ashamed.
- Name properly
  - Same name for repository and package.
  - Avoid case sensitivity  $\Rightarrow$  use lower-case.
  - Avoid spaces.
  - Consider the best name scope. 
- Disable unused features.
  - E.g. if your project has been superseded, make this clear within the README.md and disable issue reports.
- Utilize the project description, icon, etc.
- You may or may not want to create templates for new issues and merge requests.
- Care for your repository size.
  - Remove unused big files from repository entirely!

## Checklist:

- Use proper versioning.
  - E.g. <http://semver.org/>
  - Avoid private repositories; don't be ashamed.
- Name properly
  - Same name for repository and package.
  - Avoid case sensitivity ⇒ use lower-case.
  - Avoid spaces.
  - Consider the best name scope. 
- Disable unused features.
  - E.g. if your project has been superseded, make this clear within the README.md and disable issue reports.
- Utilize the project description, icon, etc.
- You may or may not want to create templates for new issues and merge requests.
- Care for your repository size.
  - Remove unused big files from repository entirely!
- Validate feature requests ASAP.

- C** **renato-wiki / core**  
the core of renato-wiki

---

- E** **renato-wiki / example-basic**  
a basic example to set-up a renato-wiki instance for static files

---

- E** **renato-wiki / example-self**  
a basic example of set-up providing wiki content for renato-wiki itself

---

- M** **renato-wiki / module-showcase**  
a showcase for all known plugins, themes, etc.

---

- T** **renato-wiki / themes / theme-nerissa**  
an artless and lightweight theme with blue colors

---

- P** **renato-wiki / plugins / plugin-markdown**  
markdown language support

---

- U** **renato-wiki / ui-plugins / ui-plugin-file-upload**  
file upload and management UI

---

- U** **renato-wiki / ui-plugins / ui-plugin-git**  
page edit UI for git support

---

- P** **renato-wiki / plugins / plugin-git**  
multi-user git management

Avoid:

- removing repositories without good reasons.

Avoid:

- removing repositories without good reasons.
- being impolite.

## Avoid:

- removing repositories without good reasons.
- being impolite.
- releasing half-baked Software. (e.g. version 1.0.0)
  - Make it open-sourced anyways :D

## Avoid:

- removing repositories without good reasons.
- being impolite.
- releasing half-baked Software. (e.g. version 1.0.0)
  - Make it open-sourced anyways :D
- violating licenses.
  - Inter-platform copying may very well be license violation. Read the ToS :-)



## Avoid:

- removing repositories without good reasons.
- being impolite.
- releasing half-baked Software. (e.g. version 1.0.0)
  - Make it open-sourced anyways :D
- violating licenses.
  - Inter-platform copying may very well be license violation. Read the ToS :-)
- starting a new competing project
  - Issue feature requests
  - Wait for some sort of validation
  - Create merge requests

## Avoid:

- removing repositories without good reasons.
- being impolite.
- releasing half-baked Software. (e.g. version 1.0.0)
  - Make it open-sourced anyways :D
- violating licenses.
  - Inter-platform copying may very well be license violation. Read the ToS :-)
- starting a new competing project
  - Issue feature requests
  - Wait for some sort of validation
  - Create merge requests
  - Unless you have good reasons to (e.g. bad management/code quality)

Thanks for your attention!





<https://gitlab.com/>



<https://www.emojione.com/developers/free-license>