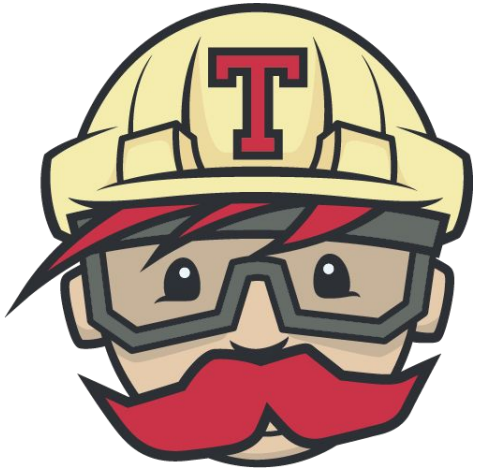


Continuous Integration

Continuous Integration

- Automatisches *testing*, *building* und *deploying* für jeden *commit*

git push VS **git push**

Build

Execute Tests

Package and Upload

Test Again

Other annoying steps

Anbieter

Travis CI: aus Berlin, gratis für Open Source Projekte, sehr gute Github Integration, gratis für Studenten

AppVeyor: gratis für Open Source Projekte, unterstützt Windows Builds

und weitere...

Badges: *So much winning!*

README.md

These are "javascript bindings" to liblouis created by cross compiling liblouis using emscripten. The Liblouis API written in C can be directly called using the `ccall` and `cwrap` functions provided by emscripten. As directly calling the C API is cumbersome, an additional API — called Easy API — is provided for most functions. This package supports NodeJS and browser environments.

Easy-API @npm v0.3.0 | Easy-API @bower v0.3.0 | Latest C-API Build @npm v0.0.0-test.4 | Latest C-API Build @bower v0.0.0-test.4

Build of Easy-API passing | Build of C-API passing

Table of Contents

Travis CI Einrichten

- Mit Github verknüpftes Konto erstellen
  https://travis-ci.org/
- *Travis CI* für das gewünschte *Repository* aktivieren
- *Travis CI* Konfigurationsdatei **.travis.yml** im *Root* des *Repositories* anlegen
- Distribution, Virtualisierungstyp und Programmiersprache auswählen

```
dist: trusty # Ubuntu 14.04
sudo:  false # in Docker
language:  node_js # mit NodeJS Tools vorinstalliert
# da keine Befehle spezifiziert sind, wird der Standardbefehl für Java Programme ausgeführt:
# npm test
```

Travis CI Befehl anpassen

- Durchläuft mehrere Phasen, in welchen Befehle ausgeführt werden können:

apt addons
cache components
before_install
install
before_script

script
before_cache
after_success
after_failure
before_deploy

deploy
after_deploy
after_script

```
# Syntax für .travis.yml

phase:
- Befehl
andere_phase:
- Befehl
- noch ein Befehl
```

- *Exit Code* der einzelnen Phasen entscheidet über *success*, *failure* oder *error*

Travis CI Pitfalls

- Skripte im Repository mit `chmod +x` ausführbar machen
  Befehle mit `set -x` anzeigen, `$?` nur in Skripten verwenden
- Gefährliche Schritte (wie *Deploy*) für *Pull Requests* deaktivieren

```
if [ "$TRAVIS_PULL_REQUEST" != "false" -o "$TRAVIS_BRANCH" != "master" ]; then
        echo "Not publishing. Is pull request or non-master branch."
        exit 0
fi
```

- Passwörter und andere sensitive Informationen immer verschlüsselt anlegen
- Zeitnahe *Commits* können einen *Race Condition* im *Deployment* erzeugen!
- *Travis CI* führt *merged pull requests* aus, also wird *Travis CI* nicht bei *Merge Conflicten* ausgeführt!

Thesis Automatisch Kompilieren

Ziel: Jede Änderung des Latexquelltextes soll automatisch eine öffentlich zugängliche PDF erzeugen.

```
$ pdflatex ausarbeitung.tex
$ bibtex ausarbeitung
$ pdflatex ausarbeitung.tex
$ pdflatex ausarbeitung.tex
$ # irgendwie hochladen
```

Idee: In *Travis CI* kompilieren und auf *Github Pages* veröffentlichen. (Analog auch für eine Webseite einsetzbar…)

Thesis Automatisch Kompilieren Cont'd

```
sudo: required
dist: trusty
before_install: # get latex packages required by your thesis
- sudo apt-get -qq update && sudo apt-get install -y --no-install-recommends
texlive-latex-recommended
script: # make pdf, allow bibtex to contain errors
- mkdir build
- pdflatex -interaction=nonstopmode -halt-on-error -output-directory build ausarbeitung.tex &&
  (bibtex build/ausarbeitung || true) &&
  pdflatex -interaction=nonstopmode -halt-on-error -output-directory build ausarbeitung.tex &&
  pdflatex -interaction=nonstopmode -halt-on-error -output-directory build ausarbeitung.tex
after_success: # collect everything we want to publish/deploy
- mkdir deploy
- cp build/*.pdf deploy/
after_failure:
- cat build/ausarbeitung.log
deploy: # abuse deploy for github pages
  provider: pages
  github_token: $GITHUB_TOKEN
  local_dir: deploy
  target_branch: pdfs
  skip_cleanup: true
  on:
    branch: master
```

Linting & Testing der Konfiguration

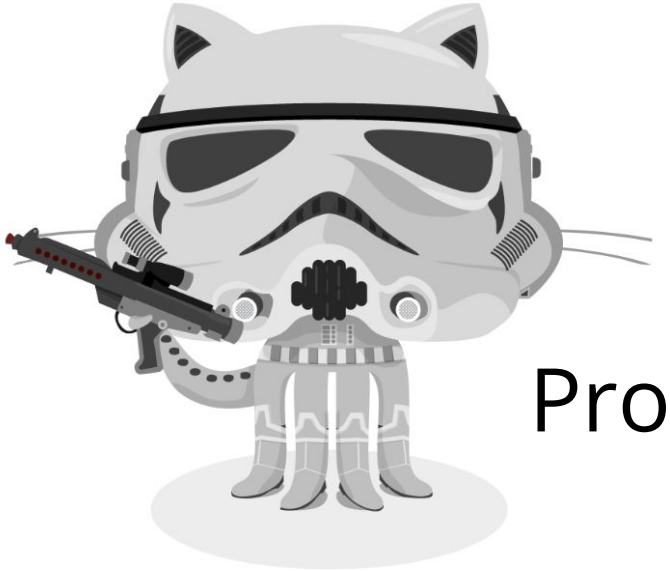
Häufiges Problem: dutzende *Commits* beim Aufsetzen von *Travis CI*

- Einfache syntaktische Fehler durch *Linting* finden
  - *http://lint.travis-ci.org/*

- Für sudo: false verwendet *Travis CI* ein *Open Source Docker Image*
  - *https://quay.io/organization/travisci* für precise (Ubuntu 12.04)
  - *https://quay.io/organization/travisci* für trusty (Ubuntu 14.04)

```
docker run --name travis-debug -dit quay.io/travisci/travis-ruby /sbin/init  docker  exec   -it travis-debug  bash  -l
su  - travis
```

- Später mehr zu *Docker*...



Protect and PR!