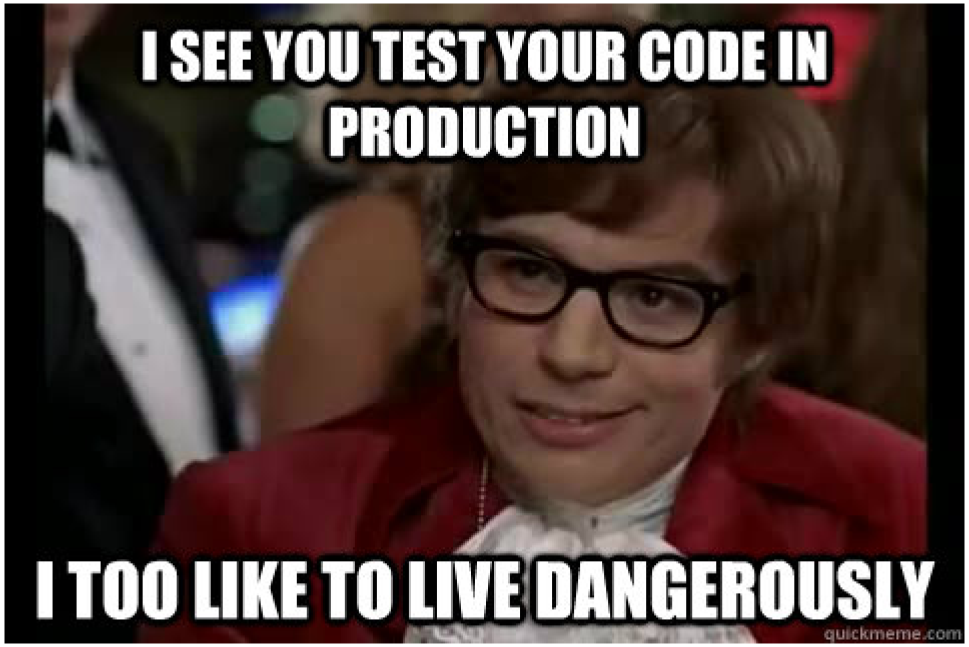
Different levels of Testing WordPress Plugins with Travis CI


I SEE YOU TEST YOUR CODE IN PRODUCTION
I TOO LIKE TO LIVE DANGEROUSLY
quickmeme.com

Who am I?

Devops Lead in Geniem Oy, WordPress enthusiast, Docker wizard, Automation lover, Lazy tester

Find me in github.com/onnimonni.

geniem

What we do?

Geniem is specialized in enterprise level WordPress sites and scalable nodeJS applications.

http://github.com/devgeniem/

http://geniem.fi

Why would you automate your testing?

It's completely free for open source

Travis CI

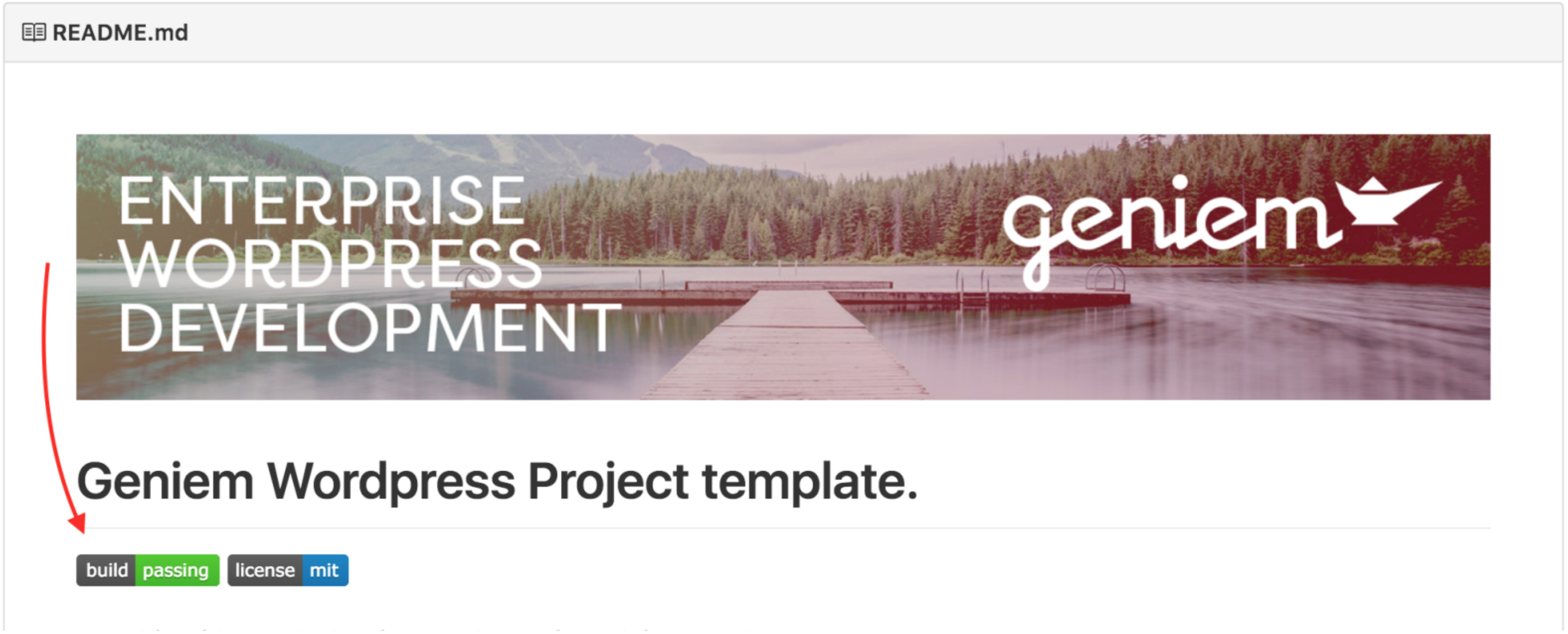
From this moment on Travis and you are going to be the best buddies ever.

It helps you maintain your code

It's much easier to fix problems when you notice them immediately.

Doesn't seem important?

Try to push a breaking change into plugin into popular plugin and ask how your users feel after that!



Geniem Wordpress Project template.

It makes your project README look great

It's now simpler to fool people into using your code.

How to get started

1. Create repository in Github

2. Register into https://travis-ci.org/

3. Add .travis.yml into your project

4. Enable travis for your repository

5. Watch the results

Different levels of testing

Let's start with baby steps in testing and then progress to more higher level tests later.

Check syntax from your code

The easiest way to start is with automatic syntax checking

```
# 32767 (E_ALL) shows all errrors
# 4095 (E_STRICT) is easier to start with
$ php -d error_reporting=32767 -l plugin.php
```

Example can be found in devgeniem/better-wp-install-dropin.

Check syntax from your other files

You can syntax check other files too.

For example composer:

```
# Checks that composer.json is okay
$ composer validate

# Or try to install it and see if you have errors?
$ composer install -o --prefer-dist --no-interaction
```

Example can be found in devgeniem/wp-project.

Check code style with php codesniffer

If you are using codesniffer it's easy to use in travis too.

You can also use WordPress-Coding-Standards.

```
# Check that code is compliant with phpcs.xml rules
$ phpcs --standard=./phpcs.xml -n -p .
```

Example config: devgeniem/wp-project.

Example failing output: devgeniem/wp-project travis

PHP unittests

This is what people usually think when writing tests for WordPress.

This is ultimately the best way to test your code but it takes more time. Unit tests can pinpoint the exact parts which are broken.

```
# Run unit tests defined in phpunit.xml
$ phpunit
```

Example travis config: devgeniem/wp-sanitize-accented-uploads.

Example failing output: devgeniem/wp-sanitize-accented-uploads

Use headless browser for integration tests and also JS errors

You can actually install fully configured WordPress instance into travis and test it with real browsers using phantomjs or selenium.

PhantomJS and ruby rspec example can be found in: Koodimonni/wordpress-test-template.

Example travis config: k1sul1/custom-admin-notices

Example travis output: k1sul1/custom-admin-notices travis.

Thanks for listening!

Let's add more test together shall we?

If you have questions you can reach me in twitter: @koodimonni.

If you do please use hashtag #testingmakesmyheadhurt.