Front End Workflow Automation for Theme and Plugin Development

a.k.a.
How to Make the Computer do the Boring Stuff
Founder & Designer
@ Citrus Mist
citrus-mist.com
@citrusmist

Milos Soskic
This is a DevOps talk
WTH
Designer talking DevOps!
or why should you believe anything I say...
Small outfit
I wear many hats

1 Created by Alex Berkowitz from the Noun Project
2 Created by Nicolas Molès from the Noun Project
3 Created by Evert Jan Boon from the Noun Project
4 Created by Norbert de Graaf from the Noun Project
12 total WP installs

8 new

6 on shared hosting
... and I have a CompSci degree
WP projects mainly brochure type sites for small and medium businesses.

Have an opinion on design
Less so on build quality
Even less on best practices
Small changes end up taking time due to process – minify, concat, upload etc.
Even I think my time is better spent doing something else

I want it, but I don't want to think about it too much
Do Some Work
› Optmise Imgs, JS & CSS
› Upload via FTP
› Export DB › Upload DB
› Search & Replace

CodeKit ()
Transmit
phpMyAdmin
Search Replace
Different pieces of software excluding text editor & graphics
Cognitively expensive

Error prone

Boring & cumbersome

FACEPALM!
There must be another way!
WP scaffolding
Theme scaffolding
LiveReload
Preprocessing

Code linting
Theme asset optimising
Plugin scaffolding

Plugin asset optimising
Deployment
Conditional asset loading

REQUIREMENTS
Yeoman, Grunt, Bower & WP-CLI
WP scaffolding
Theme scaffolding
LiveReload
Preprocessing

Code linting
Theme asset optimisation
Plugin scaffolding

Plugin asset optimisation
Deployment
Conditional asset loading

• Solved
• WIP

REQUIREMENTS
Am I just swapping one toolbox for another?
YES BUT

Achieves much more

Streamlined interface, albeit CLI
Yeoman:
scaffolding
– smart templates

Grunt:
task runner – automator

Bower:
front-end pkg manager

WP-CLI:
mange WP in Terminal

Requires:
node.js, PHP 5.3.2+, Cygwin on Win
STEP 1

WP scaffolding,
Theme scaffolding,
LiveReload,
Preprocessing,
Code linting,
Asset optimisation
In The Terminal

```bash
npm install -g yo generator-wordpress
mkdir my-wp-site
cd my-wp-site
yo wordpress
```

Requires:
node.js with npm
What Just Happened?

DL-ed WordPress

Created wp-config with nice defaults

Initiated a git repo

DL-ed a starter theme

Made a first commit

http://yourchosenurl.dev
There Is More

Vagrant w/ puppet

Custom dirs

WP as submodule
In The Terminal:

cd wp-content/themes/chosen_theme

grunt

{  
Compile SASS
LiveReload
JSHint
Uglify JS
}

Also Supports:
Compass, Stylus, RequireJS
Watching for changes
When a file is updated
grunt build

} Compile SASS
JSHint
Uglify JS
ImageMin
(JPG, GIF, PNG, SVG)
STEP 2

Conditional asset loading
wp-config.php

```php
if (file_exists(dirname($__FILE__) . '/wp-config-local.php')) {
    // Local Environment
    define('WP_ENV', 'local');
    define('WP_DEBUG', true);
    define('WP_DEBUG_LOG', true);
    include(dirname($__FILE__) . '/wp-config-local.php');
}
elseif (file_exists(dirname($__FILE__) . '/wp-config-staging.php')) {
    // Staging Environment
    define('WP_ENV', 'staging');
    include(dirname($__FILE__) . '/wp-config-staging.php');
}
elseif (file_exists(dirname($__FILE__) . '/wp-config-production.php')) {
    // Production Environment
    define('WP_ENV', 'production');
    include(dirname($__FILE__) . '/wp-config-production.php');
}
```
wp-config-{env}.php

define('DB_NAME', 'db_user');
define('DB_USER', 'db_pass');
define('DB_PASSWORD', 'yourSafePass');
define('DB_HOST', 'localhost');
$table_prefix = 'wp_';
if (WP_ENV == 'local') {

    wp_register_script('livereload',
        'http://highrise.dev:35729/livereload.js?snipver=1',
        null, false, true
    );
    wp_register_script('myplugin',
        get_bloginfo('template_url') . '/js/bespoke/myplugin.js',
        null, array('jquery'), true
    );
    wp_register_script('main',
        get_bloginfo('template_url') . '/js/main.js',
        null, array('jquery', 'myplugin'), true
    );
    wp_enqueue_script('livereload');
} elseif (WP_ENV == 'staging' || WP_ENV == 'production') {

    wp_register_script('main',
        get_bloginfo('template_url') . '/js/main.min.js',
        null, array('jquery'), true
    );
}

wp_enqueue_script('main');
vendor
bower package store
bespoke
your plugins / modules
main.js
main / bootstrap file

main.min.js
above contents optimised
Exclude `wp-config-*` from version control!
Deployment

STEP 3
Passwordless SSH access from local machine

```
ssh user@staging.url
cd path/to/siteroot
git clone -b develop
git@github.com:user/repo
```

---

git-flow

```
master: latest stable
dev: stable dev
feature/*/: feature specific
```

Staging  ➔ develop
Production ➔ master

---

SERVER SETUP
In The Terminal

cd path/to/wp

git clone https://github.com/c10b10/wp-cli-deploy
touch wp-cli.local.yml

wp-cli.local.yml

require:
  - wp-cli-deploy/deploy.php
define('STAGING_URL', 'your.staging.url');
define('STAGING_WP_PATH', '/path/to/wp/root');
define('STAGING_HOST', 'your.staginghost.com');
define('STAGING_USER', 'staging_user');
define('STAGING_PORT', '22');
define('STAGING_PATH', '/path/to/site/root');
define('STAGING_UPLOADED_PATH', '/path/to/uploads/folder');
define('STAGING_THEMES_PATH', '/path/to/themes/folder');
define('STAGING_PLUGINS_PATH', '/path/to/plugins/folder');
define('STAGING_DB_HOST', 'localhost');
define('STAGING_DB_NAME', 'stage_db_name');
define('STAGING_DB_USER', 'stage_db_user');
define('STAGING_DB_PASSWORD', 'stage_db_pass');
define('STAGING_EXCLUDES', 'node_modules/');
Include `wp-config-deploy.php` from `wp-config.php`
In The Terminal

upload

wp deploy push staging --what=themes
wp deploy push staging --what=plugins
wp deploy push staging --what=uploads

download

wp deploy pull staging --what=themes
wp deploy pull staging --what=plugins
wp deploy pull staging --what=uploads
... and my favourites
wp deploy push staging --what=db

Export local DB
Search & replace
Upload to server
Substitute remote DB

wp deploy pull staging --what=db

Export remote DB
DL remote DB
Search & replace
Substitute local DB
Mega useful for

Checking your design over a real network (without having to commit)

Prototyping

Testing on handhelds

Working on site w/ content strategist

Deployment on shared hosting
WPZest

Wrapper around YeoPress
Automates steps 1-3 excl. server setup

Plugin scaffolding based on WP Plugin Boilerplate
Plugin asset optimisation

https://github.com/artificer/generator-wpzest
Would love to hear your ideas about what should go in it
Resources

YeoPress
github.com/wesleytodd/
YeoPress

WP-CLI
wp-cli.org

WP-CLI-Deploy
github.com/c10b10/
wp-cli-deploy

Bower
bower.io

Grunt
gruntjs.com

Yeoman
yeoman.io

WP Plugin Boilerplate
github.com/tommcfarlin/
WordPress-Plugin-Boilerplate
THANK YOU!