

# Magento 2: From Setup to Deployment

## Automate Everything

Juan Alonso  
@Jalogut

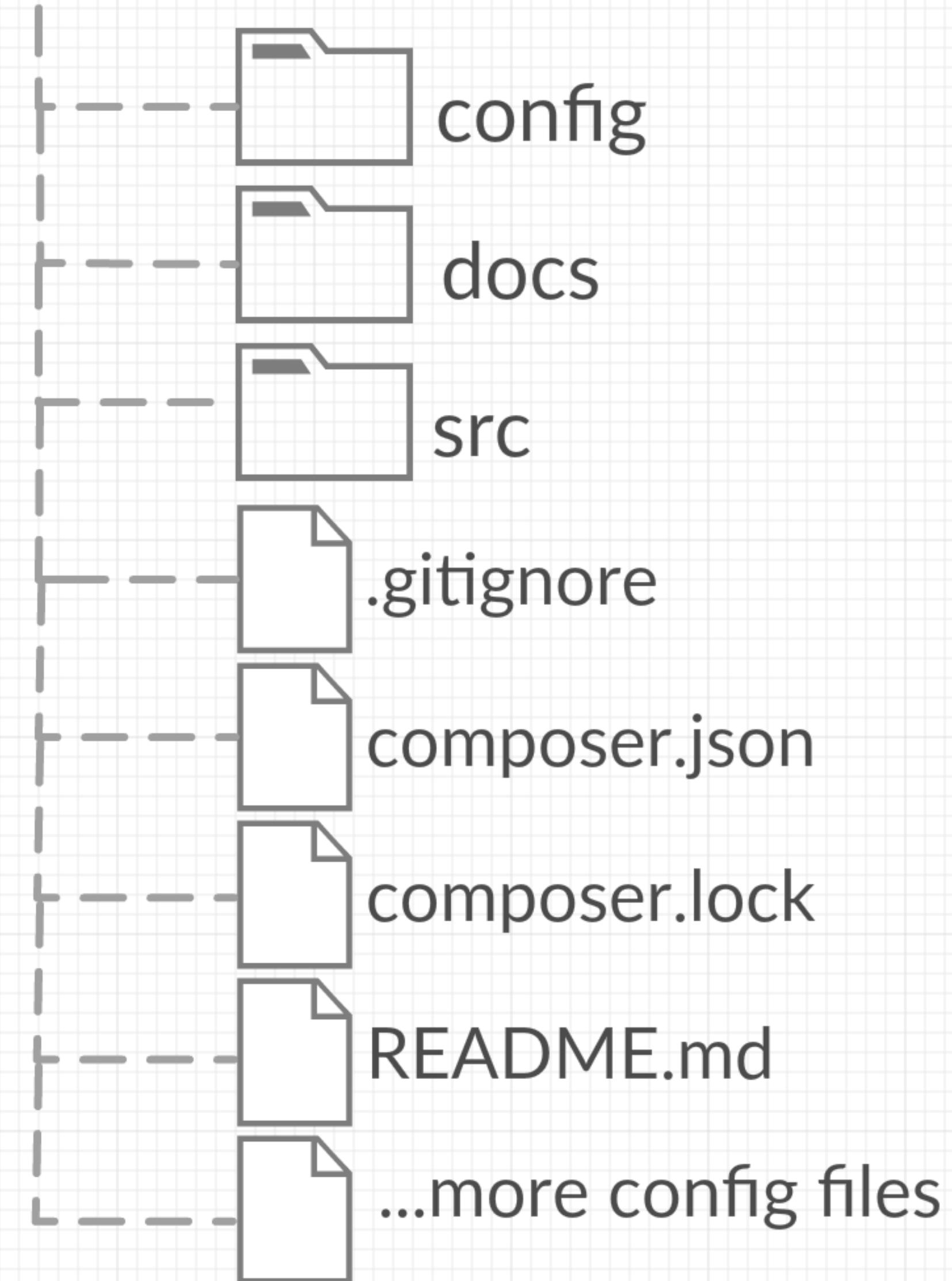


# Setup





# Project



▼ mmpl /Volumes/workspace/projects/mmpl

► app  
► bin  
► config  
► dev  
► docs  
► lib  
► phpserver  
► pub  
► setup  
► update  
► var  
► vendor  
    ❖ .gitignore  
    ❖ auth.json  
    ❖ CHANGELOG.md  
    ❖ composer.json  
    ❖ composer.lock  
    ❖ CONTRIBUTING.md  
    ❖ COPYING.txt  
    ❖ grump.php.yml  
    ❖ Gruntfile.js.sample  
    ❖ Index.php  
    ❖ ISSUE\_TEMPLATE.md  
    ❖ Jenkinsfile  
    ❖ LICENSE.txt  
    ❖ LICENSE\_AFL.txt  
    ❖ nginx.conf.sample  
    ❖ package.json.sample  
    ❖ php.ini.sample  
    ❖ README.md

Magento files into “magento” folder

```
"extra": {
    "magento-root-dir": "magento/"
},
"config": {
    "vendor-dir": "magento/vendor"
},
"autoload": {
    "psr-4": {
        "Magento\\Setup\\": "magento/setup/src/Magento/Setup/"
    },
    "psr-0": {
        "": "magento/app/code/"
    },
    "files": [
        "magento/app/etc/NonComposerComponentRegistration.php"
    ]
},
"autoload-dev": {
    "psr-4": {
        "Magento\\Sniffs\\": "magento/dev/tests/static/framework/Magento/Sniffs/",
        "Magento\\Tools\\": "magento/dev/tools/Magento/Tools/",
        "Magento\\Tools\\Sanity\\": "magento/dev/build/publication/sanity/Magento/Tools/Sanity/",
        "Magento\\TestFramework\\Inspection\\": "magento/dev/tests/static/framework/Magento/TestFramework/Inspection/",
        "Magento\\TestFramework\\Utility\\": "magento/dev/tests/static/framework/Magento/TestFramework/Utility/"
    }
}
```

done?

→ **mmp1** magento/bin/magento

```
Composer file not found#0 /Volumes/workspace/projects/mmpl/magento/vendor/magento/framework/Composer/ComposerInformation.php(104): Magento\Framework\Composer\ComposerInformation->getComposerFile()
#1 /Volumes/workspace/projects/mmpl/magento/vendor/magento/framework/Composer/ComposerInformation.php(65): Magento\Framework\Composer\ComposerInformation->getComposerFile()
#2 /Volumes/workspace/projects/mmpl/magento/vendor/magento/framework/Composer/ComposerInformation.php(81): Magento\Framework\Composer\ComposerInformation->__construct()
#3 /Volumes/workspace/projects/mmpl/magento/bin/magento(22): Magento\Framework\Console\Cli->__construct()
#4 {main}
```

PHP Fatal error: Uncaught Error: Class 'Cli' not found in /Volumes/workspace/projects/mmpl/magento/bin/magento

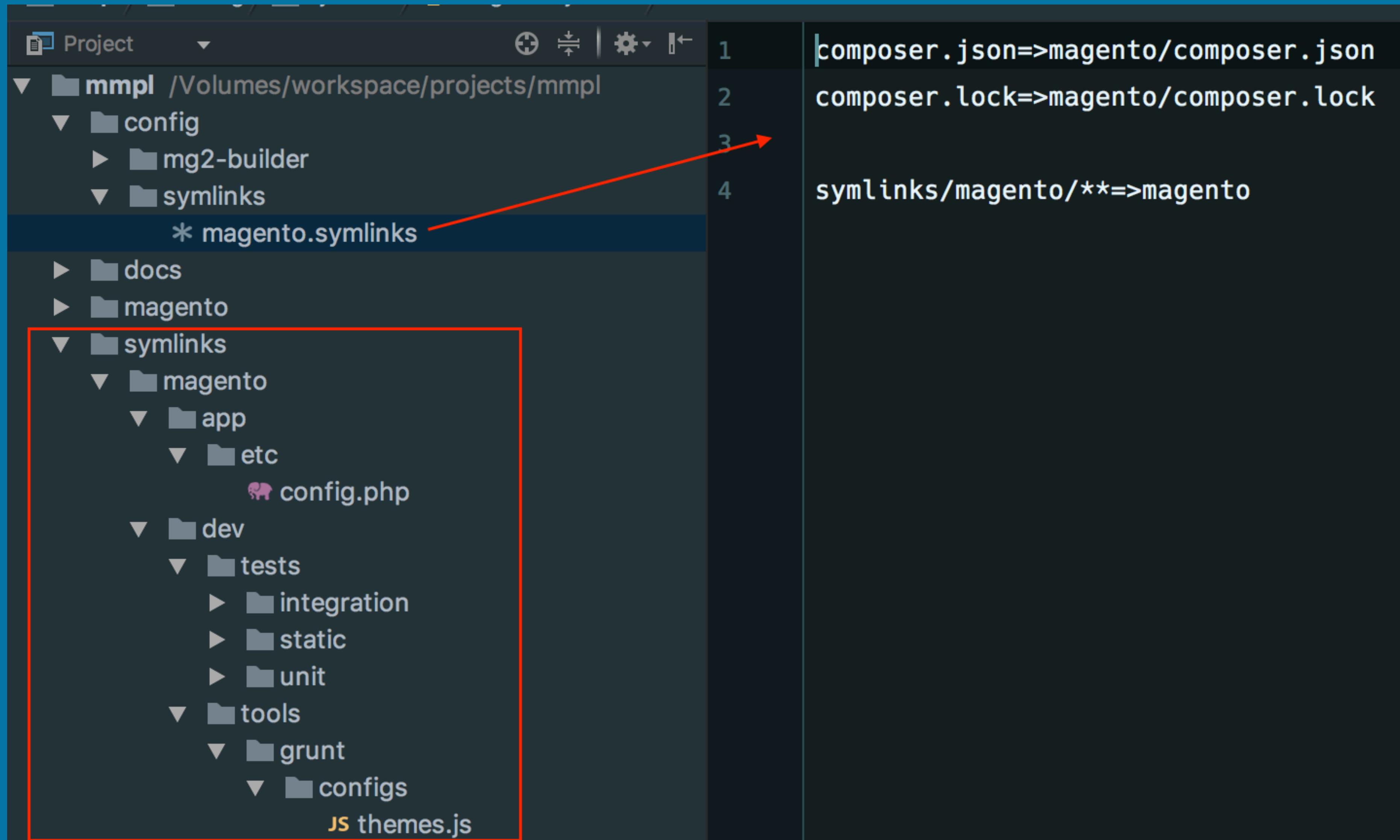
Stack trace:

```
#0 {main}
    thrown in /Volumes/workspace/projects/mmpl/magento/bin/magento on line 31
```

“*cd magento && ln -sf ./composer.\* .*”

# Symlinker-pro

<https://github.com/staempfli/symlinker-pro>



The screenshot shows the Symlinker-pro interface with a project structure on the left and configuration on the right.

**Project Structure:**

- Project: mmpl /Volumes/workspace/projects/mmpl
- config
- mg2-builder
- symlinks
- \* magento.symlinks
- docs
- magento
- symlinks
- magento
- app
- etc
- config.php
- dev
- tests
- integration
- static
- unit
- tools
- grunt
- configs
- themes.js

A red box highlights the "magento" folder under "symlinks". A red arrow points from the "magento" folder in the project structure to the "magento" line in the configuration file.

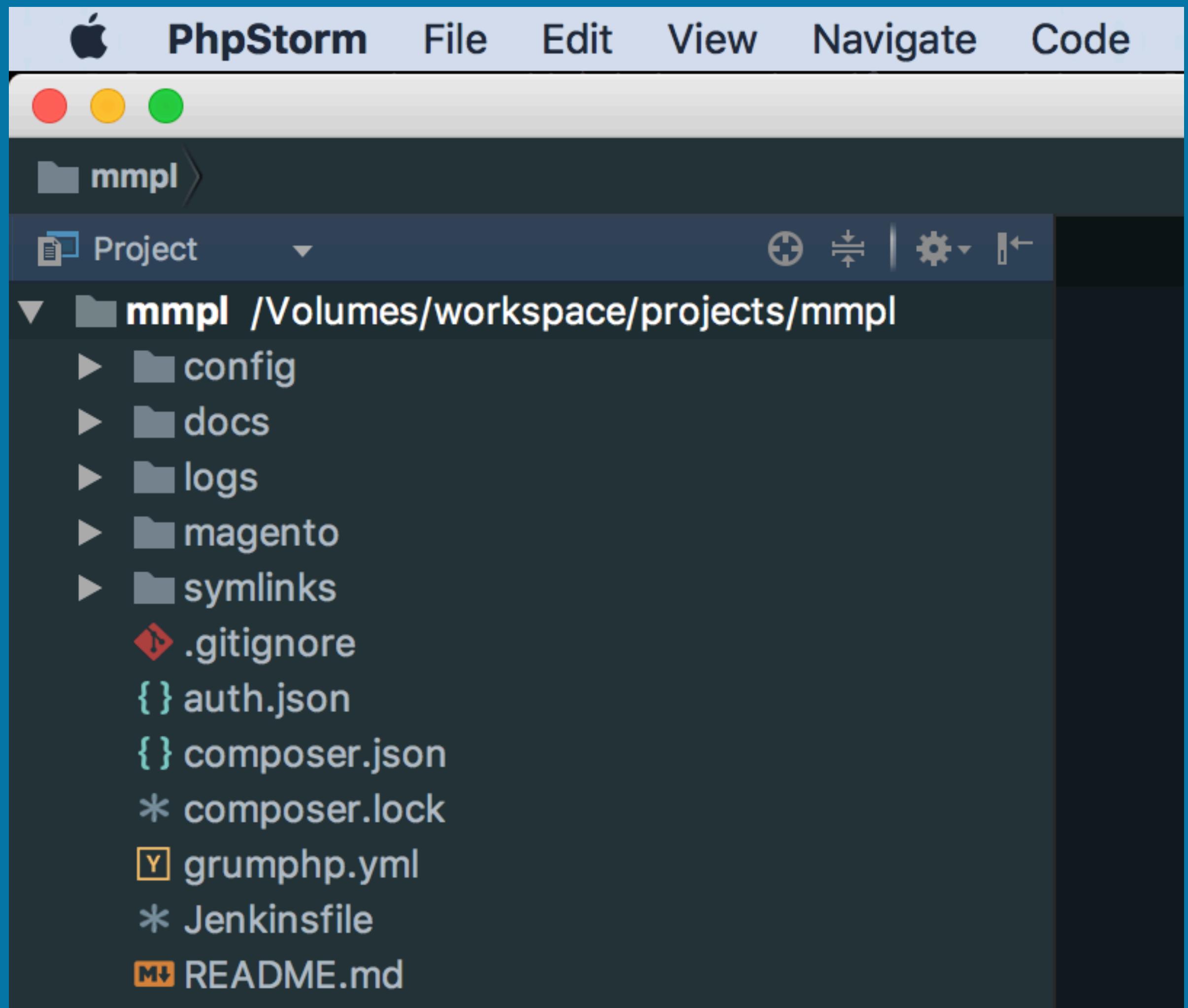
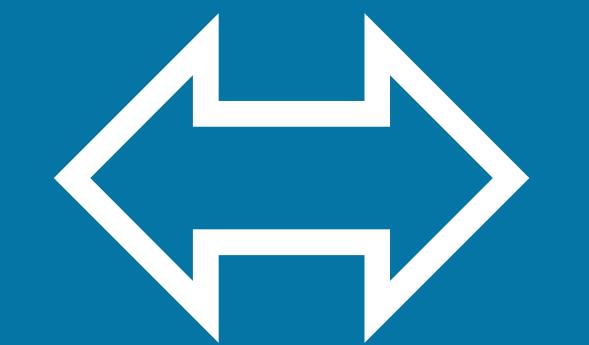
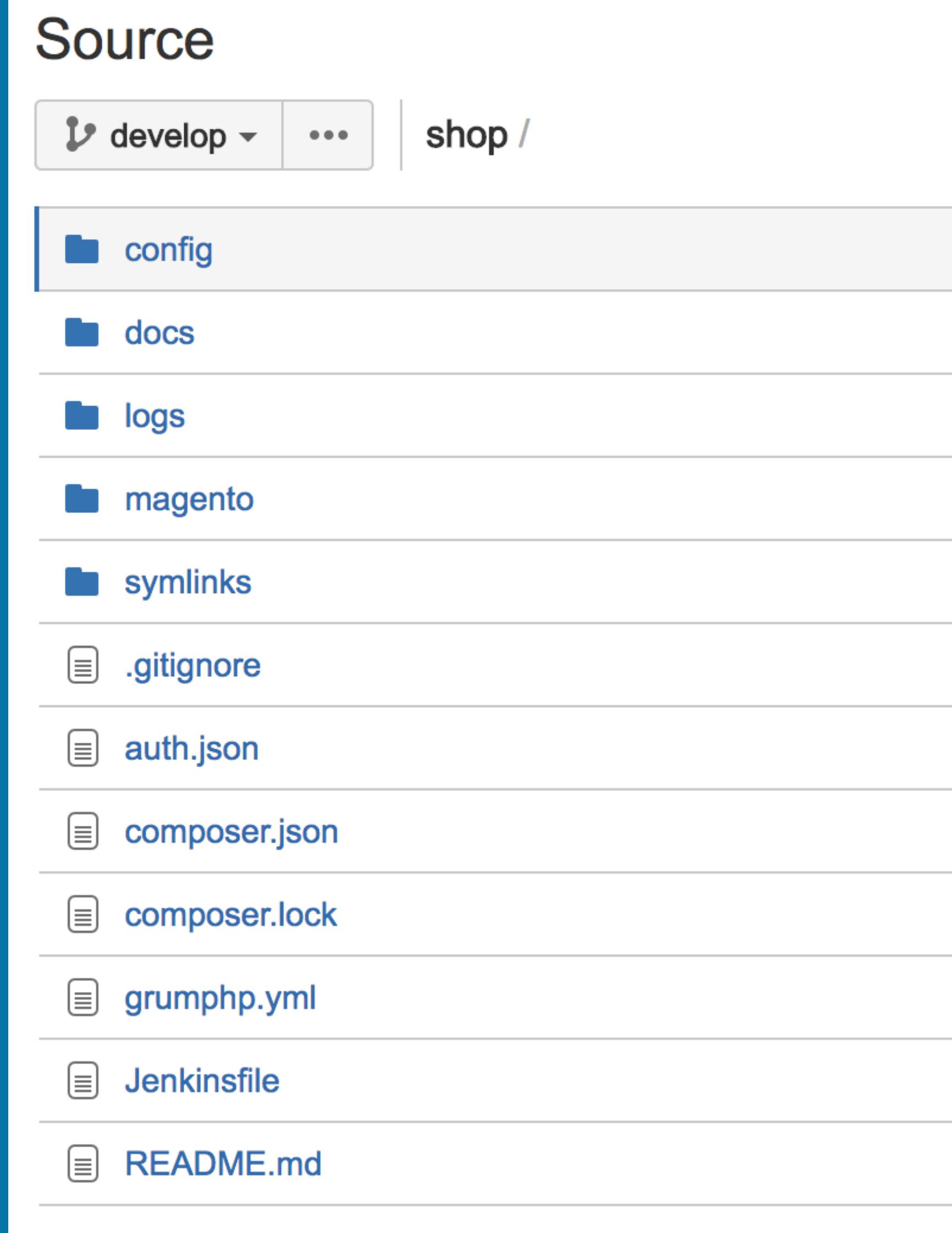
**Configuration File (Right Panel):**

```
1 composer.json=>magento/composer.json
2 composer.lock=>magento/composer.lock
3
4 symlinks/magento/**=>magento
```

# Symlinker-pro

<https://github.com/staempfli/symlinker-pro>

```
"require": {  
    "staempfli/symlinker-pro": "~1.0"  
},  
"scripts": {  
    "setSymlinks": "magento/vendor/bin/symlinker-pro create:from:file config/symlinks/magento.symlinks -f --enable-wildcards",  
    "post-install-cmd": [  
        "@setSymlinks"  
    ],  
    "post-update-cmd": [  
        "@setSymlinks"  
    ]  
}
```



# Proper Magento 2 Composer Setup

*<https://blog.hauri.me>*

# Build



# Install

*database* → *DB user* → *integr test DB*

→ *setup:install* → *Apache/Nginx*

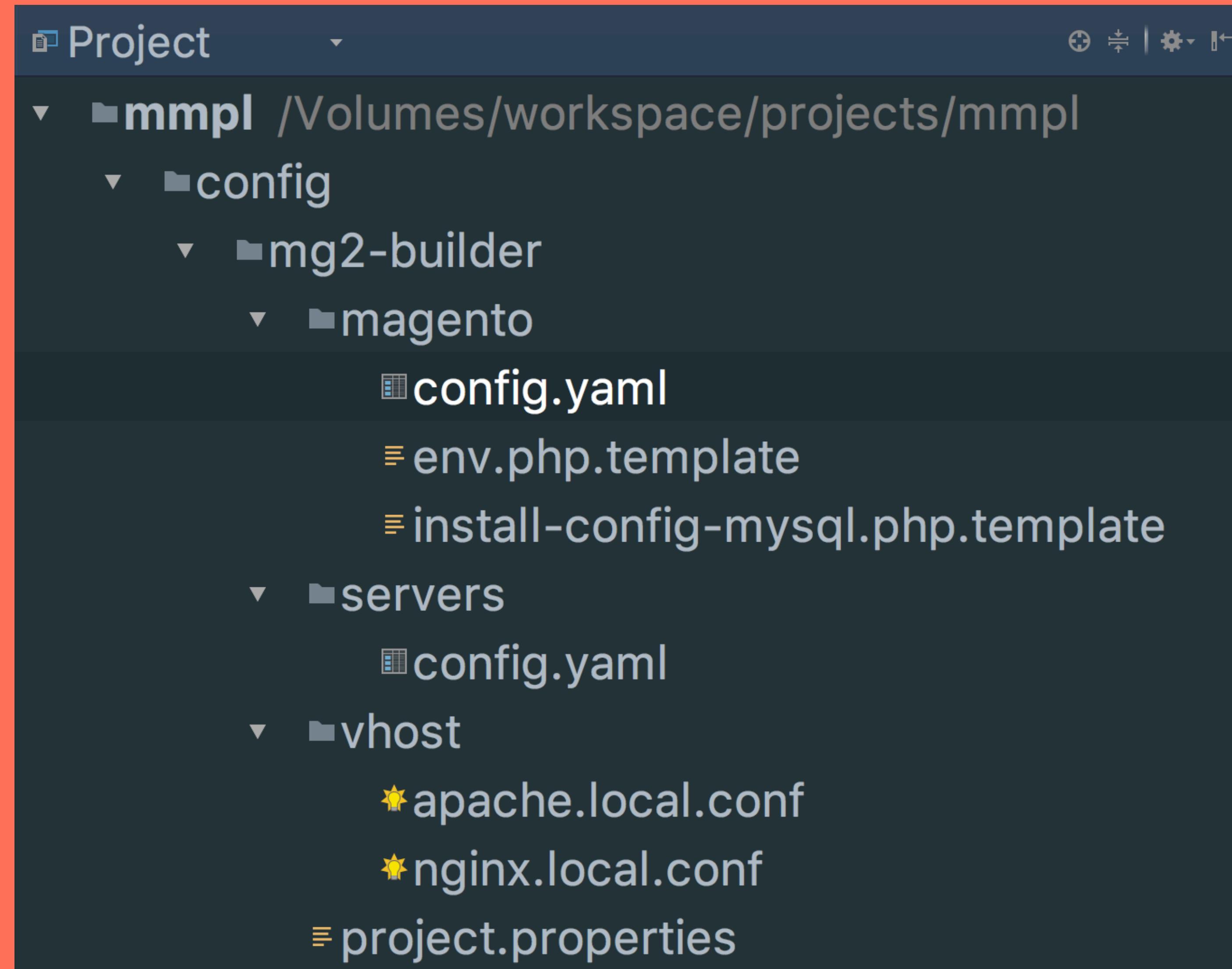
# Sync Data

*ssh server* → *DB dump* → *download dump* →  
→ *import dump* → *setup:upgrade* →  
→ *core\_config\_data* → *rsync media* →  
→ *cache:clean*

How to automate that?

# *Magento2-builder-tool*

<https://github.com/staempfli/magento2-builder-tool>



# *magento2-builder-tool*

*<https://github.com/staempfli/magento2-builder-tool>*

Install:

*\$ mg2-builder install*

Sync Data:

*\$ mg2-builder sync*

# *magento2-builder-tool*

*<https://github.com/staempfli/magento2-builder-tool>*

From scratch:

```
$ git clone  
$ composer install  
$ mg2-builder install -Dinstall.type=sync
```

*<https://youtu.be/KFFkmqX1-5M>*

# Development



# Generate boilerplate code

Stop copy-paste

Focus on writing new code

# Live Templates

- Standalone files
- Blocks of code



# magento2-phpstorm-templates

<https://github.com/staempfli/magento2-phpstorm-templates>

- 2magento sequence
- 2magento system section
- 2magento admin menu active
- 2magento grid text column
- 2magento grid select column
- 2magento grid date column
- 2magento grid store column
- 2magentofieldset
- 2magento field text
- 2magento field select
- 2magento field date
- 2magento widget param select
- 2magento widget param text
- 2magento javascript
- 2magento phpcs ignore block
- 2magento phpcs ignore line

- 2magento config save
- 2magento table new
- 2magento table column
- 2magento table foreignKey
- 2magento field file
- 2magento field hidden
- 2magento field dynamic
- 2magento field wysiwyg
- 2magento field wysiwyg simple
- 2magento field wysiwyg simple + widgets
- 2magento widget xml
- 2magento block widget
- 2magento source option
- 2magento layout
- 2magento requirejs-config
- 2magento js class

- 2magento config xml
- 2magento Controller Admin
- 2magento Controller Front
- 2magento di xml
- 2magento events xml
- 2magento helper
- 2magento InstallSchema
- 2magento menu xml
- 2magento module xml
- 2magento routes xml
- 2magento system xml
- 2magento InstallData
- 2magento eav text attribute
- 2magento eav textarea attribute
- 2magento eav boolean attribute
- 2magento eav date attribute

# Code generators

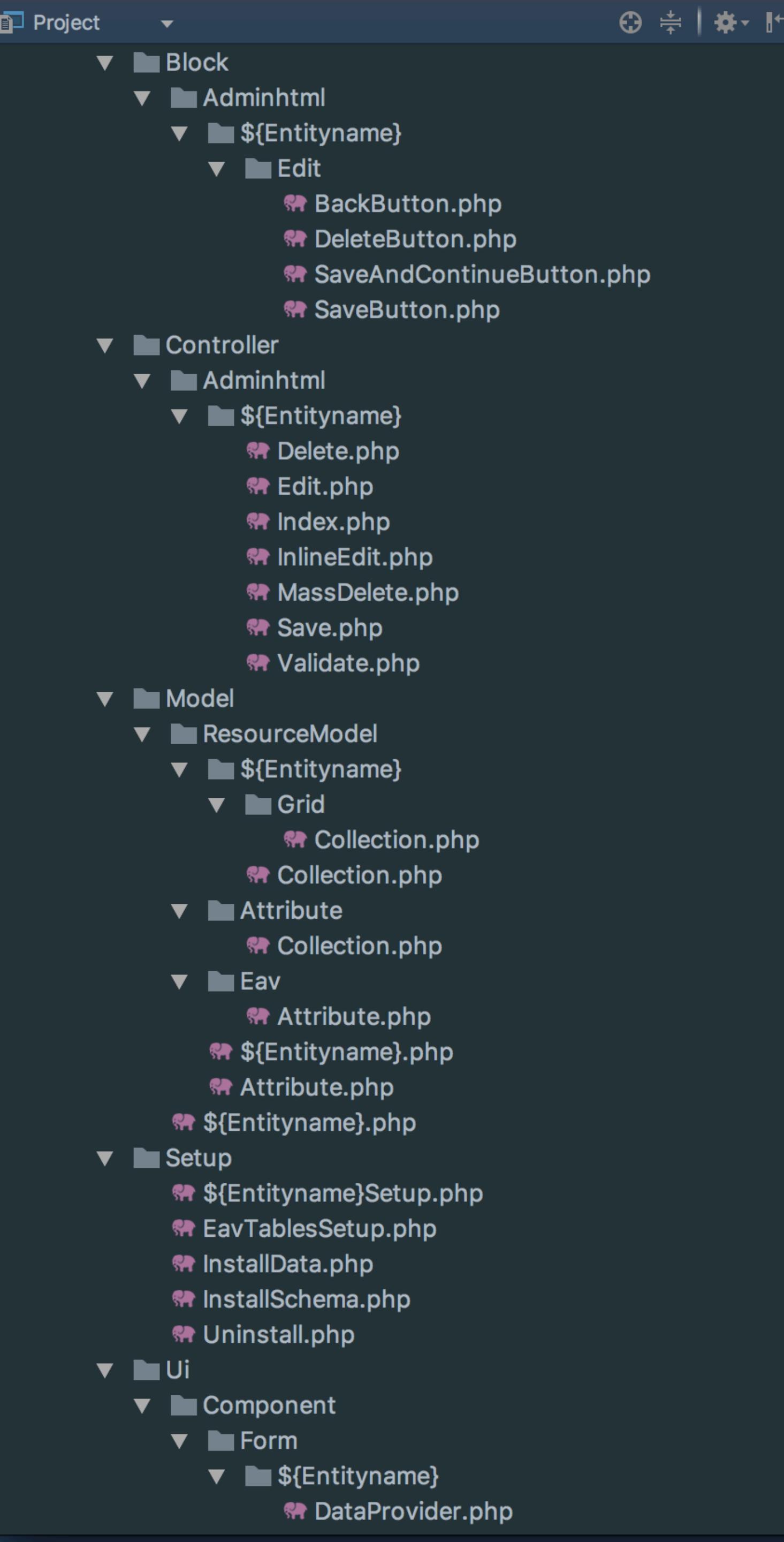
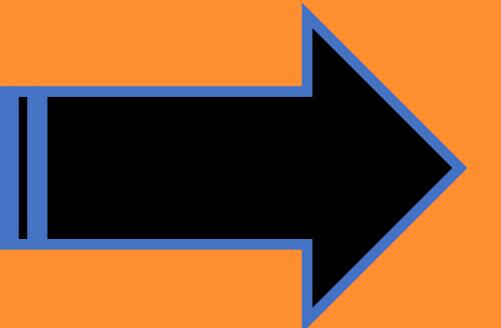
- Bunch of files
- Complete functionality



# CRUD EAV

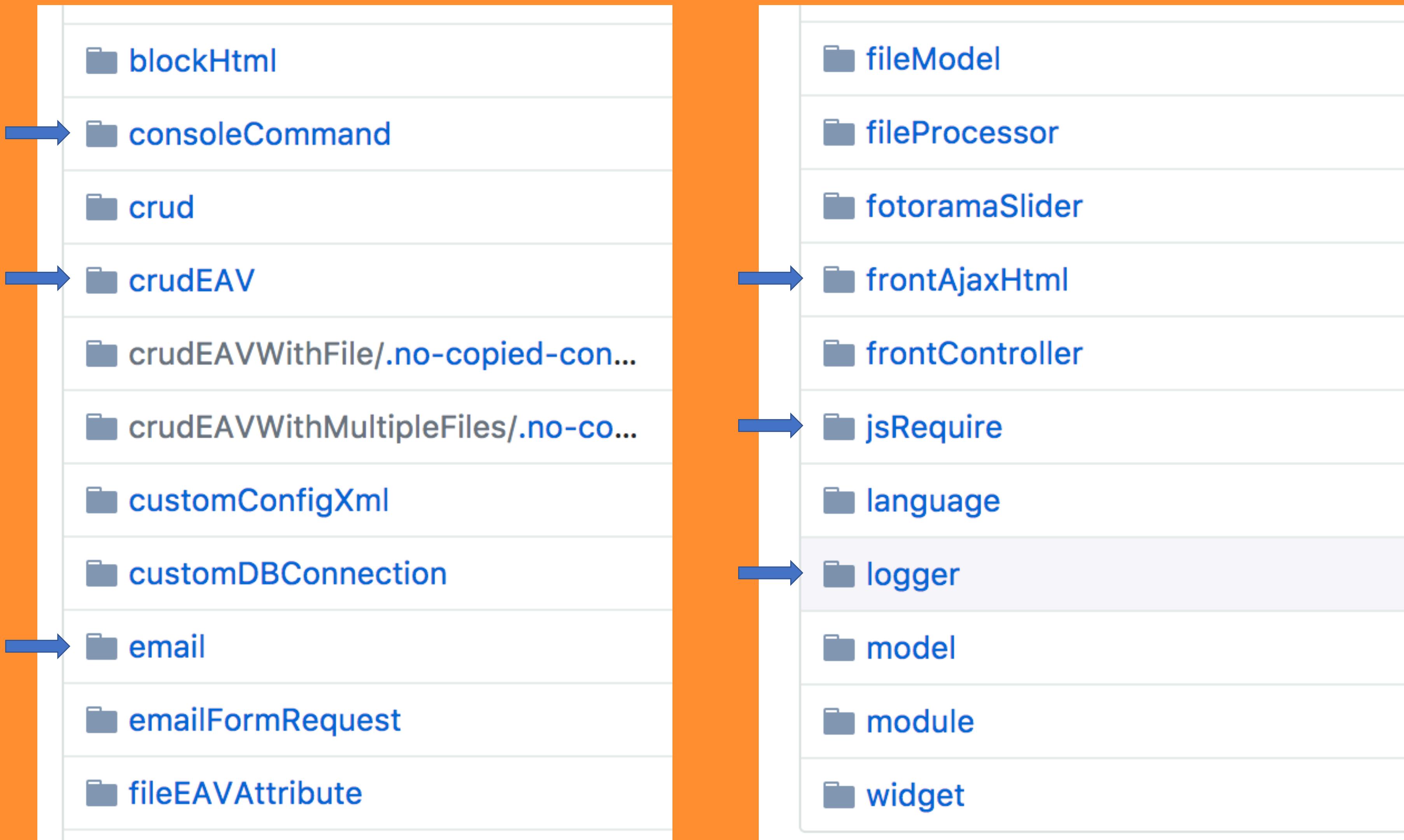
- More than 30 files

```
mg2-codegen template:generate crudEAV
```



# Magento2-code-generator

<https://github.com/staempfli/magento2-code-generator>



# Code generators

## Magento2-code-generator

- *<https://github.com/staempfli/magento2-code-generator>*

## Mage2Gen

- *<https://mage2gen.com>*

# Deployment





KEEP  
CALM  
AND  
**CLEAR CACHE  
& REINDEX**

**COMPOSER UPDATE**

`rm -r var/di`

`rm -r var/cache/*`

`rm -r var/generation/*`

`rm -r var/log/*`

`rm -r var/view_preprocessed/*`

`rm -r pub/static/_requirejs/*`

`rm -r pub/static/adminhtml/*`

`rm -r pub/static/frontend/*`

`bin/magento setup:upgrade`

`bin/magento setup:static-content:deploy`

`bin/magento setup:di:compile-multi-tenant`

**COMPILE LESS  
CLEAR LOCAL STORAGE**

*git clone/pull* ➔ *composer install*

➔ *setup:di:compile* ➔ *static-content:deploy*

➔ *permissions* ➔ *set maintenance*

➔ *setup:upgrade* ➔ *cache:clean*

➔ *unset maintenance*

# New on version 2.2

→ *app:config:import -n*

→ *cron:install --force*

→ *setup:db:status*











KEEP  
CALM  
AND

mg2-deployer release



# Deployment Tools

## Magento2-deployment-tool

- [\*https://github.com/staempfli/magento2-deployment-tool\*](https://github.com/staempfli/magento2-deployment-tool)

## MageDeploy2

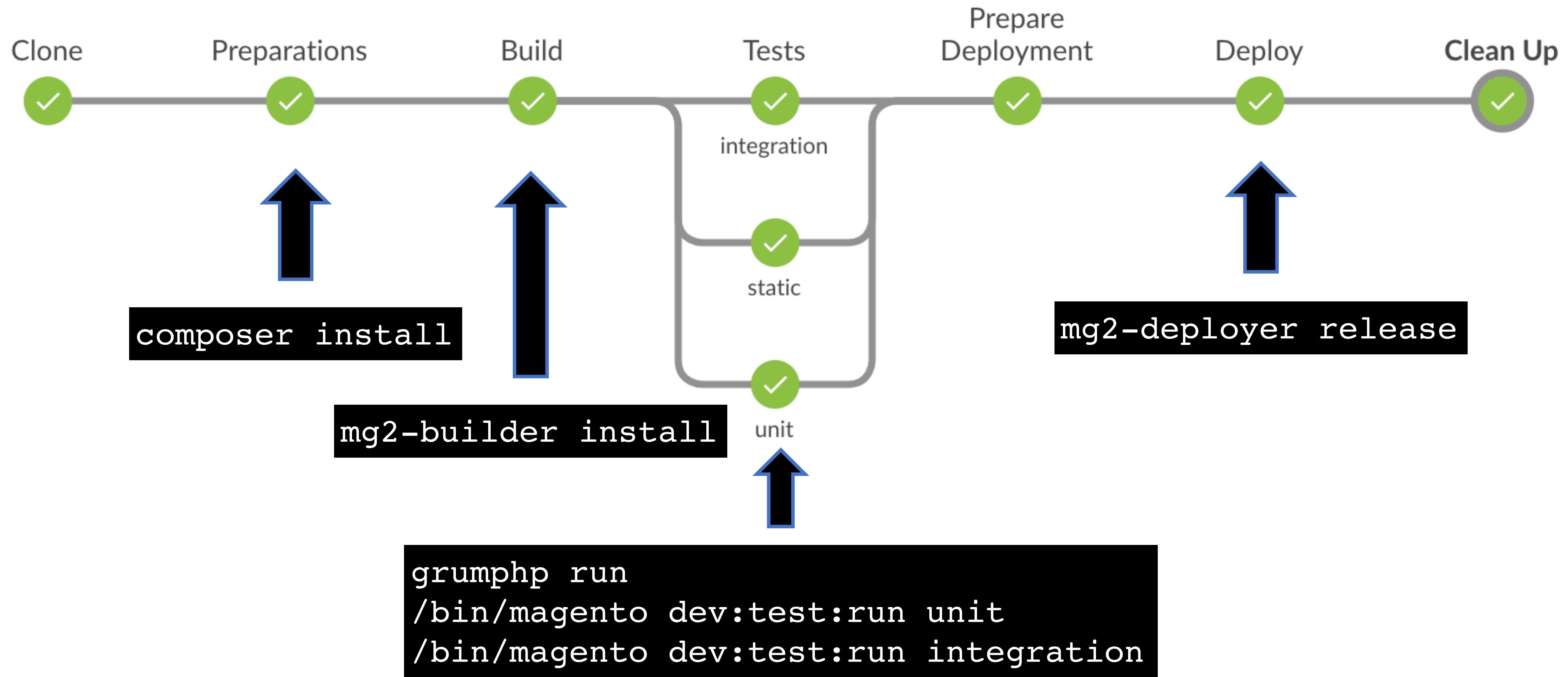
- [\*https://dev98.de\*](https://dev98.de) -> *introducing-magedeploy2*

## Deployer recipe

- [\*https://github.com/deployphp/deployer\*](https://github.com/deployphp/deployer) -> *recipe/magento2.php*

# All Together





# Jenkins Tutorial

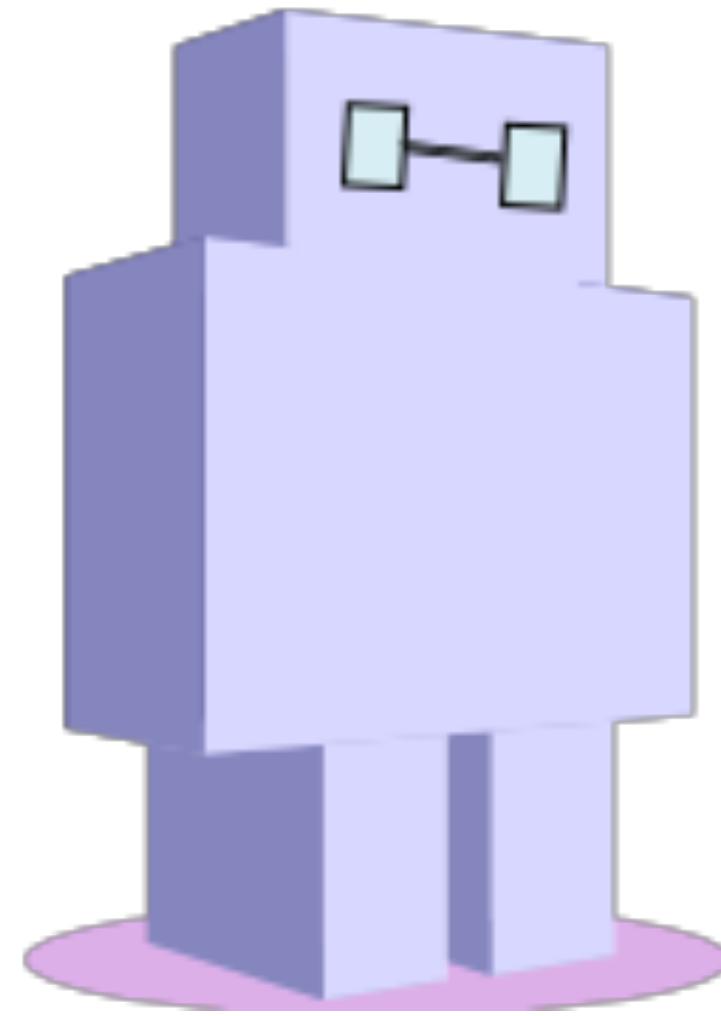
<https://dev.to/jalogut> -> setup continuous integration

# Task runner tools



**<Phing>**

<https://phing.info>



<https://robo.li>

# TAKE AWAYS

---

- Automate the standard
- Less boilerplate & more focus on new code
- Create tools for yourself
- Use existing open source tools



- @Jalogut
- Senior developer at Stämpfli AG
- +6 years working on Magento
- 2x certified

