

# PHP QA TOOLS

PHP Code Quality



Sergej Kurakin

# Why?

- New project
- Clean modern code base
- Many developers
- New challenge

# What?

- PSR Code Style
- Minimize Copy/Paste
- Reduce Code Complexity
- Tests...

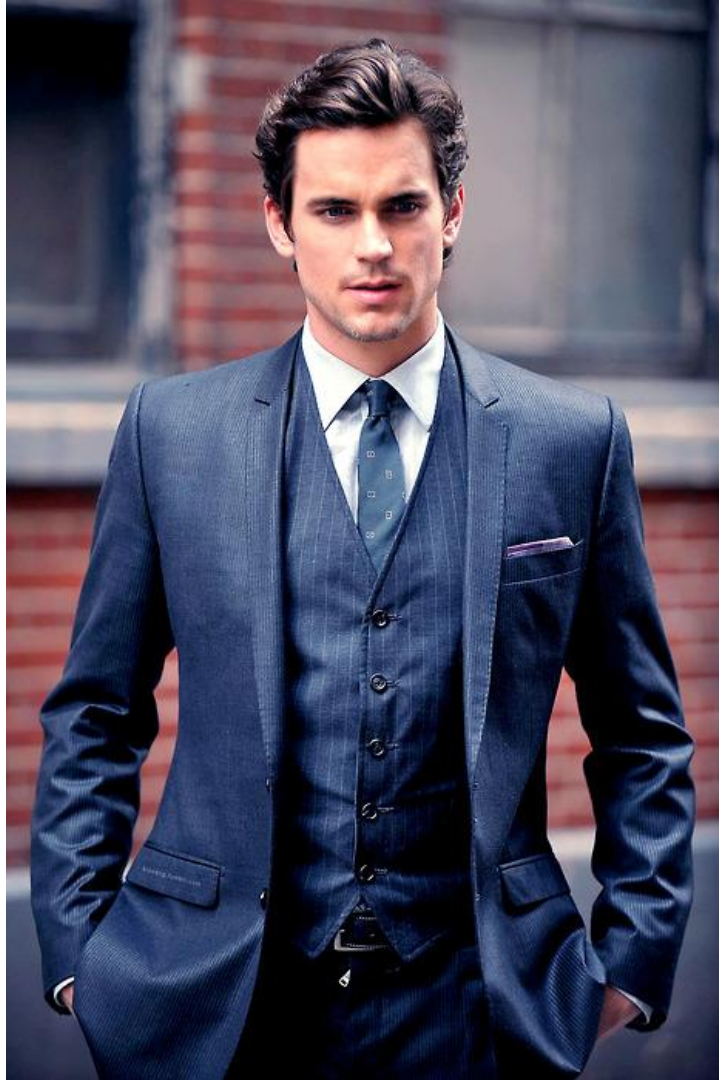
T\_PAAMAYIM\_NEKUDOTAYIM

# PHP Lint - syntax check

- Checks PHP code for syntax errors
- Built-in into PHP binary

```
$ php -l path/to/file.php
```

Code Style







# PHP Code Sniffer

- Detects coding standard violations
- [https://github.com/squizlabs/PHP\\_CodeSniffer](https://github.com/squizlabs/PHP_CodeSniffer)
- Composer/PEAR

Interesting side project: [http://squizlabs.github.io/PHP\\_CodeSniffer/analysis/](http://squizlabs.github.io/PHP_CodeSniffer/analysis/)

# PHP Code Sniffer: Run

```
$ phpcs path/to/file.php
```

```
$ phpcs path/to/dir
```

```
$ phpcs -h
```

```
$ phpcs -i
```

# PHP Code Sniffer: Keys

- `--sniffs=<sniffs>`
- `--encoding=<encoding>`
- `--extensions=<extensions>`
- `--ignore=<patterns>`
- `--standard=<standard>`

# PHP Code Sniffer: Standards

- PEAR
- PHPCS
- PSR1
- PSR2
- Squiz
- Zend

# PHP Code Sniffer: Reports

- full
- xml
- checkstyle
- csv
- json
- emacs
- source
- summary
- svnblame
- gitblame
- hgblame
- notifysend

Copy & Paste

**COPY PASTE**



**Programmer...**





# PHP Copy/Paste Detector

- CPD - Copy/Paste (code) Detector
- <https://github.com/sebastianbergmann/phpcpd>
- Composer/PHAR

# PHP Copy/Paste Detector: Run

```
$ phpcpd path/to/file.php path/to/other/file.php
```

```
$ phpcpd path/to/dir
```

```
$phpcpd -h
```

# PHP Copy/Paste Detector: Keys

`--names=<names>`

`--names-exclude=<names-exclude>`

`--exclude=<exclude>`

`--log-pmd`

`--min-lines=<5>`

`--min-tokens=<70>`

Mess



# PHP Mess Detector

- Source Code Analyzer
- <http://phpmd.org/>
- Composer

# PHP Mess Detector: Run

```
$ phpmd path/to/dir text cleancode,  
codesize,controversial,design,naming,  
unusedcode
```

```
$ phpmd path/to/dir text  
path/to/ruleset.xml
```

# PHP Mess Detector: Reports

- xml
- text
- html



# PHP Mess Detector: Rulesets

- `cleancode`
- `codesize`
- `controversial`
- `design`
- `naming`
- `unusedcode`

Unused Code



# PHP Dead Code Detector

- Unused/Dead Code Detector
- <https://github.com/sebastianbergmann/phpdcd>
- Composer/PHAR

# PHP Dead Code Detector

```
$ phpdcdd path/to/file.php
```

```
$ phpdcdd path/to/dir
```

```
$phpdcdd -h
```

# Code Statistics



# phploc

- Size and Analytics
- <https://github.com/sebastianbergmann/phploc>
- Composer/PHAR



# phploc: Run

```
$ phploc path/to/dir
```

```
$ phploc -h
```

# phploc: Keys

- names
- names-exclude
- git-repository
- log-csv=<path/to/file.csv>
- log-xml=<path/to/file.xml>



# PHP Depend

- Software Metrics
- <http://pdepend.org/>
- <http://pdepend.org/documentation/software-metrics/index.html>
- Composer/PHAR/PEAR

# PHP Depend: Run

```
$ pdepend \  
  --summary-xml=/tmp/summary.xml \  
  --jdepend-chart=/tmp/jdepend.svg \  
  --overview-pyramid=/tmp/pyramid.svg \  
path/to/dir
```

Tests...

**TESTING IS FOR WIMPS**



**REAL MEN TEST IN  
PRODUCTION**

# Testing Frameworks

- PHPUnit - <http://phpunit.de/>
- Behat - <http://behat.org/>
- Spec - <http://www.phpspec.net/>
- [JFGI](#)



Weapon of choice

Other stuff

# PhpStorm

Preferences > PHP >

Code Sniffer

Mess Detector

Preferences > Inspections

Code > Inspect Code

# php-cs-fixer

- Fixes coding standards
- <http://cs.sensiolabs.org/>
- Composer/PHAR
- PSR1 and PSR2 only

```
$ php-cs-fixer fix path/to/dir --  
level=all
```

You can find more at <http://phpqatools.org/>

Big big thanks to Sebastian Bergmann for the work over PHP QA tools collection.

# Questions?



Sergej Kurakin

Email: [sergej.kurakin@nfq.it](mailto:sergej.kurakin@nfq.it)

Twitter: [@zaza\\_it](https://twitter.com/zaza_it)

Special thanks to authors of all pictures used in this presentation.