Symfony3 is coming soon and with it a handful of wonderful changes and restructuring to the core directory structure of a symfony app.

The new directory structure has a number of benefits, all of which are minor and may require minimal changes to your workflow.

PHPUnit

`phpunit` can now be run from the project root without having to explicitly specify the path of the configuration file.

```
# Symfony2
phpunit -c app/phpunit.xml

# Symfony3 (no need to specify the configuration file location)
phpunit
```

Binary Executables

All binary executable files are now all located in a single location – the `bin` directory.

This simplifies things also so on a *NIX machine you can configure your $PATH to simplify calling repeatable commands.

```
# you can update your `PATH` to include the `bin` directory
PATH="./bin:$PATH"

# From your project root you can now run executables like so:
console
symfony_requirements
doctrine
```

```
# else with no `PATH` update
bin/console
bin/symfony_requirements
bin/doctrine
```

A new `/var` directory

The new `/var` directory contains the files to which the system writes data to during the course of its operation; this replaces/moves cache and logs from the app directory.

This also makes it easier to add permissions, the entire `/var` directory should be writable by your webserver (apache, nginx, etc).

```
# default symfony3 `var` directory
var/bootstrap.php.cache
var/cache
var/logs
```

Symfony requirements check

Running `symfony_requirements` will output mandatory & optional environment configurations.

```
cd /path/to/my_app
symfony_requirements
```

Credit

Most of the credit for this post belongs to justAnil @ StackOverflow for his amazing write-up.