PhpZone Shell Documentation

Release 0.2

Jakub Zapletal

April 29, 2015

Contents

Getting Started

1.1 Requirements

PhpZone requires PHP 5.3 or higher.

1.2 Installation

Installation is provided via Composer, if you don't have it, do install:

```
$ curl -s https://getcomposer.org/installer | php
```

then PhpZone Shell can be added into your dependencies by:

```
$ composer require --dev phpzone/shell 0.2.*
```

or add it manually into your `composer.json`:

```
{
    "required-dev": {
        "phpzone/shell": "0.2.*"
    }
}
```

1.3 Configuration file

If the configuration file doesn't exist yet, you can find more information in PhpZone - Getting Started

1.3.1 Registration of the extension

Registration is provided by a simple definition of full name of the class (namespace included):

```
extensions:
    PhpZone\Shell\Shell: ~
```

Note: This extension is a command builder with definitions within its values. It means that only the registration without any values would have no effect.

1.3.2 Creating of commands

As mentioned in the PhpZone documentation, each extension gets its configuration via values which are defined during its registration. PhpZone Shell expects to get an array of required commands. Each command has defined its name as a key and its values are definitions for exact command:

```
extensions:
    PhpZone\Shell\Shell:
        'command:one':
            script:
                - echo Command 1 is working
        'command:two':
            script:
                - echo Command 2 is working
```

Now, when we would run:

```
$ vendor/bin/phpzone command:one
```

we would see an output: `Command 1 is working`

Definitions For Command

Commands can contain following definitions:

Name	Type	Default	Required
script	array		Yes
description	string	null	No
enable	boolean	true	No
help	string	null	No
stop_on_error	boolean	false	No
tty	boolean	true	No

Example:

```
extensions:
    PhpZone\Shell\Shell:
        command:
            script:
                - echo Foo
                - echo Bar
            description:   Short descriptive text
            enable:        true
            help:          Long helpful text
            stop_on_error: false
            tty:           true
```

Note: The order of definitions can be random.

Note: Not required definitions don't need to be set.

2.1 script

Type	Default	Required
array		Yes

A simple array of commands/scripts which should be executed. They are executed in exact order as defined.

2.2 description

Type	Default	Required
string	null	No

The description of a command will be displayed when a developer would run the command `list` or without any command.

2.3 enable

Type	Default	Required
boolean	true	No

All defined commands are enabled by default. Sometimes can be useful to disable a command without its removal.

2.4 help

Type	Default	Required
string	null	No

The help of a command will be displayed when a developer would run the command `help`.

2.5 stop_on_error

Type	Default	Required
boolean	false	No

When some of defined commands in the script fails, remaining commands are still executed. If set to `true`, this definition forces to stop in case of any error and not attempt to execute remaining commands. A list of the remaining commands will be displayed.

2.6 tty

Type	Default	Required
boolean	true	No

It is a definition of how commands are executed. By default all commands are executed in TTY mode.

Options For Command

All built commands have their definitions of how they are executed, but sometimes it's useful to have an option to rewrite a defined parameter or extend a functionality. This is provided by options below.

Options are extended attributes which can be set either before command name or after command name, so both following examples are valid:

```
$ vendor/bin/phpzone <OPTION> <COMMAND>
```

```
$ vendor/bin/phpzone <COMMAND> <OPTION>
```

Tip: All available options can be displayed by:

```
$ vendor/bin/phpzone <COMMAND> --help
```

3.1 –stop-on-error

When some of defined commands in the script fails, remaining commands are still executed. If it is used, this option forces to stop in case of any error and not attempt to execute remaining commands. A list of the remaining commands will be displayed. This option overwrites the `stop_on_error` definition.

3.2 –no-tty

By default all commands are executed in TTY mode. If this option is used, commands are not executed in TTY mode. This option overwrites the `tty` definition.

A command/script builder configured by YAML, based on PhpZone. Its primary purpose is to provide an easy way to define multiple scripts used in daily workflow of every developer.

Basic Usage

An example speaks a hundred words so let's go through one.

The configuration file below is used for the development of this extension:

```
extensions:
    PhpZone\Shell\Shell:
        tests:
            description: Run all tests
            script:
                - bin/behat -f progress
                - bin/phpunit
                - bin/phpspec run -f progress
                - bin/phpcs -p --colors --standard=PSR2 src/ features/bootstrap/
                - bin/phpcs -p --colors --standard=vendor/jakubzapletal/php_codesniffer-rules/psr2-w
```

Now we can just run following command and all tests would be executed:

```
$ vendor/bin/phpzone tests
```