



composer.json

What's in there and why?

Frank Giesecke

DevOp



v!vamera[®]

Enjoy your life!

Composer

- Building packages / libraries
- Use third party packages / libraries
- Use semantic versioning



```

{
  "name": "final-gene/talk-composer-json",
  "type": "project",
  "description": "My talk about the composer.json file",
  "keywords": [
    "talk",
    "composer"
  ],
  "homepage": "https://github.com/final-gene/talk-composer-json",
  "license": "MIT",
  "authors": [
    {
      "name": "Frank Giesecke",
      "email": "frank.giesecke@final-gene.de",
      "homepage": "https://github.com/FrankGiesecke",
      "role": "Developer"
    }
  ],
  "support": {
    "email": "support@final-gene.de",
    "issues": "https://github.com/final-gene/talk-composer-json/issues",
    "wiki": "https://github.com/final-gene/talk-composer-json/wiki"
  },
  "repositories": [...],
  "require": {...},
  "require-dev": {...},
  "suggest": {...},
  "autoload": {...},
  "autoload-dev": {...},
  "scripts": {...},
  "config": {...},
  "bin": [...],
  "archive": {...}
}

```

Package Name

- vendor-name/package

Package Description

- License Information
- Author Information
- Support Information

```

{
  ...
  "repositories": [
    {
      "type": "vcs",
      "url": "git@github.com:final-gene/pho.git"
    },
    {
      "type": "composer",
      "url": "https://packages.example.com"
    },
    {
      "type": "package",
      "package": {
        "name": "smarty/smarty",
        "version": "3.1.7",
        "dist": {
          "url": "http://www.smarty.net/files/Smarty-3.1.7.zip",
          "type": "zip"
        },
        "source": {
          "url": "https://smarty-php.googlecode.com/svn/",
          "type": "svn",
          "reference": "tags/Smarty_3_1_7/distribution/"
        }
      }
    }
  ],
  ...
}

```

Source definitions

- VCS like
GitHub, GitLab, Bitbucket
- Package Server
- Archive Source


```

{
    "require": {
        "php": "^7.1",
        "roave/security-advisories": "dev-master",
        "snipe/banbuilder": "dev-master"
    },
    "require-dev": {
        "composer/composer": "^1.5",
        "danielstjules/pho": "dev-master",
        "squizlabs/php_codesniffer": "^3.1"
    },
    "suggest": {
        "final-gene/some-optional-dependency": "You can suggest for ..."
    },
    ...
}

```

Dependencies

- „require“ for packages used by the application/library itself
- „require-dev“ for packages which are only required for development
- „suggest“ information for the consumer of the package about optional dependencies

```

{
    ...
    "autoload": {
        "psr-4": {
            "FinalGene\\": "src"
        }
    },
    "autoload-dev": {
        "psr-4": {
            "FinalGene\\": "tests"
        }
    },
    "scripts": {
        "post-install-cmd": [
            "echo 'command that will be executed after composer install'",
            "echo 'another command that will be executed'"
        ],
        "check-style": "vendor/bin/phpcs -p --standard=PSR2 ...",
        "fix-style": "vendor/bin/phpcbf -p --standard=PSR2 ..."
    },
    ...
}

```

Autoloading

- Configuration separated for operation and development

Scripts

- Definition of commands which can be called using the Composer CLI

```

{
    "config": {
        "preferred-install": {
            "*": "dist"
        },
        "platform": {
            "php": "7.2",
            "ext-something": "4.0"
        },
        "sort-packages": true
    },
    "bin": [
        "bin/some-binary",
        "bin/another-binary"
    ],
    "archive": {
        "exclude": [
            "/composer.*",
            "/phpunit.*",
            "/tests"
        ]
    }
}

```

Configurations

- Configuration of the Composer CLI
- Definition of binaries
- Configuration of the archive handling

Is there more?



Packagist

The PHP package archivist.



Yes!



Packagist

The PHP package archivist.



The post install command

```
{  
  ...  
  "scripts": {  
    "post-install-cmd": [  
      "echo 'command that will be executed after composer install'",  
      "echo 'another command that will be executed'"  
    ],  
  },  
  ...  
}
```

This command called by the Composer CLI after install gets done and may be used for:

- Loading/installing external tools
- Loading of external data like schemata

The post update command

```
{
  ...
  "scripts": {
    "post-update-cmd": [
      "echo 'command that will be executed after composer update'",
      "echo 'another command that will be executed'"
    ],
  },
  ...
}
```

This command called by the Composer CLI after update gets done and may be used for:

- Run migration
- Generate documentation

The pre archive command

```
{  
  ...  
  "scripts": {  
    "pre-archive-cmd": [  
      "echo 'command that will be executed after composer update'",  
      "echo 'another command that will be executed'"  
    ],  
  },  
  ...  
}
```

This command called by the Composer CLI before archive gets started and may be used for:

- Preparing files to get stored in the archive

Coding for Events?



Packagist

The PHP package archivist.



Yes!



Packagist

The PHP package archivist.



Callbacks

```
{
    ...
    "scripts": {
        "post-install-cmd": [
            "FinalGene\\ComposerCommand::soSomethingAfterInstall"
        ]
    },
    ...
}
```

Build an Object which contains static callback methods.

```
<?php declare(strict_types=1);

namespace FinalGene;

use Composer\\Script\\Event;

class ComposerCommand
{
    public static function soSomethingAfterInstall(Event $event)
    {
        // make some fancy action
    }
}
```

Inline PHP

```
{  
  ...  
  "scripts": {  
    "post-install-cmd": [  
      "php -r \"copy('config/local.dist.php', 'config/local.php');\""br/>    ],  
  },  
  ...  
}
```

Because we can 🧐

Shell command

```
{  
  ...  
  "scripts": {  
    "post-install-cmd": [  
      "cp 'config/local.dist.php' 'config/local.php'"  
    ],  
  },  
  ...  
}
```

Shell command

```
{  
  ...  
  "scripts": {  
    "post-install-cmd": [  
      "cp 'config/local.dist.php' 'config/local.php'"  
    ],  
  },  
  ...  
}
```

But be careful...

Shell command

```
{  
  ...  
  "scripts": {  
    "post-install-cmd": [  
      "cp 'config/local.dist.php' 'config/local.php'"  
    ],  
  },  
  ...  
}
```

But be careful...

These commands depends on the operating system we're on 🤔

Even more?



Yes!



Packagist

The PHP package archivist.



Referencing Composer scripts

```
{
    ...
    "scripts": {
        "post-install-cmd": [
            "@some-script",
            "@another-script"
        ],
        "some-script": [
            ...
        ],
        "another-script": [
            ...
        ]
    },
    ...
}
```


Calling the Composer

```
{  
  ...  
  "scripts": {  
    "post-install-cmd": [  
      "@composer some-command"  
    ],  
  },  
  ...  
}
```

Calling PHP scripts

```
{  
  ...  
  "scripts": {  
    "post-install-cmd": [  
      "@php some.php"  
    ],  
  },  
  ...  
}
```

Are there helpful
packages?



Yes!



Packagist

The PHP package archivist.



hirak/prestissimon

- Plugin to install packages parallel which saves time
- Should be global installed



roave/security-advisories

- Prevents installation of composer packages with known security vulnerabilities: no API, simply require it
- Must be installed as „no-dev“



Anything else?



Packagist

The PHP package archivist.



Maybe ... 🙇



Any remarks?



Packagist

The PHP package archivist.



Thanks for your
listening 🙌

