



# COMPOSER IN DRUPAL WORLD

---

*Johannes Haseitl - @derhasi*



 #drupaldevdays

wunderkraut 

• **Smile** •  
OPEN SOURCE SOLUTIONS

• adyax •

• **Kaliop** •  
INTERACTIVE MEDIA

• mSYSTEMS •

•  **actency** •  
Stratégies & Solutions Web

•  **TRAINED  
PEOPLE** •

• **Amazee**  
labs

# ME

- Johannes Haseitl
- [@derhasi](#) everywhere
- Maintainer of [Master](#), [Search API Override](#), ...
- CEO of [undpaul GmbH](#)



WHAT IS COMPOSER?



*Dependency Manager for PHP*

<https://getcomposer.org>

# DECLARING DEPENDENCIES

Add a `composer.json` to the root:

```
{
  "name": "derhasi/toggl2redmine",
  "description": "Command line tool to sync toggl time entries to redmine",
  "require": {
    "php": ">=5.4.0",
    "ajl/guzzle-toggl": ">=0.9",
    "kbsali/redmine-api": "1.5.*",
    "symfony/console": "2.6.*",
    "symfony/config": "2.6.*",
    "symfony/yaml": "2.6.*"
  },
  "bin": [
    "toggl2redmine"
  ],
  "autoload": {
    "psr-0": {
      "": "src/"
    }
  }
},
```

[getcomposer.org/doc/04-schema.md](http://getcomposer.org/doc/04-schema.md)

# INSTALL PROJECT

- `composer install`
- Fetches information from [Packagist.org](https://packagist.org)
- Downloads all packages to the `vendor` directory.

```
toggli2redmine | master => 
```

# COMPOSER.LOCK

- Keeps track of exact version/commit for every package installed
- Installs exact version/commit on `composer install`
- Improves performance on `composer install`



# PACKAGIST



# Packagist

The PHP package archivist.

[Submit Package](#)

Packagist is the main **Composer** repository. It aggregates all sorts of PHP packages that are installable with Composer.  
[Browse packages](#) or [submit your own](#).

## Getting Started

### Define Your Dependencies

Put a file named `composer.json` at the root of your project, containing your project dependencies:

```
{
  "require": {
    "vendor/package": "1.3.2",
    "vendor/package2": "1.*",
    "vendor/package3": ">=2.0.3"
  }
}
```

### Install Composer In Your Project

Run this in your command line:

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

Or [download composer.phar](#) into your project root.

## Publishing Packages

### Define Your Package

Put a file named `composer.json` at the root of your package, containing this information:

```
{
  "name": "your-vendor-name/package-name",
  "description": "A short description of
what your package does",
  "require": {
    "php": ">=5.3.0",
    "another-vendor/package": "1.*"
  }
}
```

This is the strictly minimal information you have to give.

For more details about package naming and the fields you can use to document your package better, see the [about](#) page.

### Commit The File

# A LITTLE BIT MORE

- Dependency manager
- [Autoloader](#) for PHP dependencies  
(supports *PSR-4*, *PSR-0*, *classmap* and *files*)  
at `vendor/autoload.php`
- Binary shortcuts (`vendor/bin/...`)
- [Script events and custom commands](#)

```
"scripts": {  
    "post-install-cmd": ...,  
    "dependency-cleanup": ...  
}
```

- [Composer plugins](#)

# WHY USE COMPOSER?

- "Getting off the island!"
- Architecture allows custom workflows
- Drupal 8 already uses it

# CURRENT USE IN DRUPAL 8



# DEPENDENCIES

core/composer.json:

```
{
  "name": "drupal/core",
  "description": "Drupal is an open source content management platform powering m",
  "type": "drupal-core",
  "license": "GPL-2.0+",
  "require": {
    "php": ">=5.4.5",
    "sdboyer/glyph": "0.1.*",
    "symfony/class-loader": "2.6.*",
    "symfony/css-selector": "2.6.*",
    "symfony/dependency-injection": "2.6.*",
    "symfony/event-dispatcher": "2.6.*",
    "symfony/http-foundation": "2.6.*",
    "symfony/http-kernel": "2.6.*",
    "symfony/routing": "2.6.*",
    "symfony/serializer": "2.6.*",
    "symfony/validator": "2.6.*",
    "symfony/process": "2.6.*",
    "symfony/yaml": "2.6.*",
  }
}
```

# AUTOLOADER

core/autoload.php

```
<?php
```

```
/**
```

```
 * @file
```

```
 * ...
```

```
 */
```

```
return require __DIR__ . '/core/vendor/autoload.php';
```

# COMPOSER IN YOUR DRUPAL PROJECT



# SETTING UP CORE

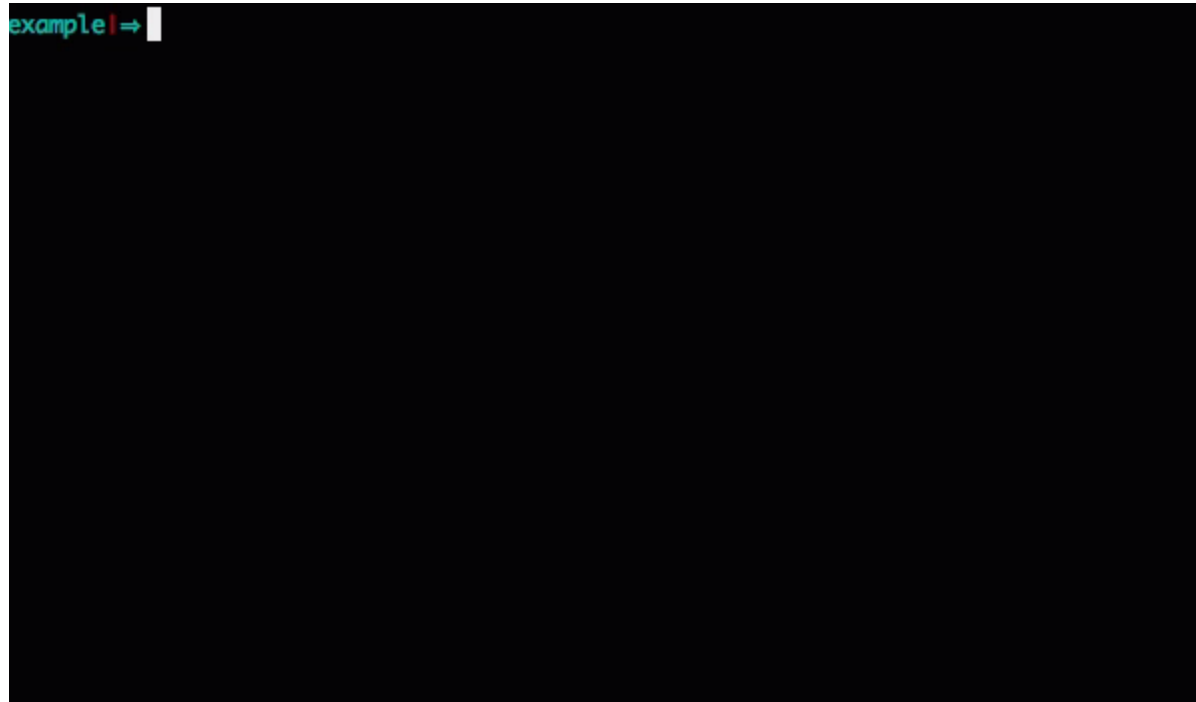
Set up a composer.json with:

```
{
  "name": "yourname/yourprojectname",
  "type": "project",
  "require": {
    "composer/installers": "^1.0.20",
    "drupal/core": "8.0.*"
  },
  "minimum-stability": "dev",
  "prefer-stable": true
}
```

- `drupal/core` subtree split by tstoeckler
- Component dependencies put in `./vendor`
- `Composer Installers plugin` puts Core in `./core` by default.
- (!) Manually add: index.php, updated autoload.php, .htaccess, ...

# ADDING MODULES / PROJECTS

```
composer require drupal/micro:^8.2.0
```



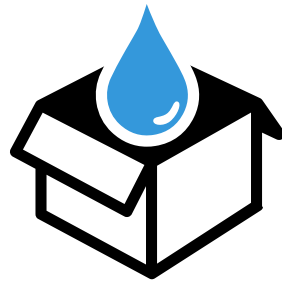
- type `drupal-module`
- placed in `modules/{name}` by *composer-installers*
- Version constraints `^1.2.0` vs `>=1.2.0`

# ADDING CUSTOM REPOSITORIES AND PACKAGES

```
"repositories": [  
  {  
    "type": "vcs",  
    "url": "ssh://gitolite@yourgitserver.com/up_p_download.git"  
  },  
  {  
    "type": "package",  
    "package": {  
      "type": "drupal-module",  
      "name": "drupal-sandbox/disable_defaults",  
      "version": "dev-7.x-1.x",  
      "source": {  
        "url": "http://git.drupal.org/sandbox/derhasi/2004516.git",  
        "type": "git",  
        "reference": "7.x-1.x"  
      }  
    }  
  },  
  {  
    "type": "composer",  
    "url": "https://packagist.org"  }  
]
```

- `repositories` section in `composer.json`
- Use `package server` (e.g. `satis`) instead when possible

# DRUPAL PACKAGIST



Packagist server with package information for every project on  
drupal.org: <http://packagist.drupal-composer.org/>

# IMPLEMENTATION

```
"repositories": [{  
  "type": "composer",  
  "url": "http://packagist.drupal-composer.org"  
}],
```

Issues on github: [drupal-composer/drupal-packagist](#) and [drupal-composer/drupal-parse-composer](#)).

# EXAMPLES

```
composer require drupal/drupal:7.*  
composer require drupal/ctools:7.*  
composer require drupal/radix:7.*  
composer require drupal/drush_language:7.*
```

# SUBMODULES AND METAPACKAGES

- Prevent packages from being installed: `replace`
- Drupal packagist will likely provide metapackages

```
"replace": {
  "drupal/action": "self.version",
  "drupal/aggregator": "self.version",
  "drupal/ban": "self.version",
  "drupal/bartik": "self.version",
  "drupal/basic_auth": "self.version",
  "drupal/block": "self.version",
  "drupal/block_content": "self.version",
  "drupal/book": "self.version",
  "drupal/breakpoint": "self.version",
  "drupal/ckeditor": "self.version",
  "drupal/classy": "self.version",
  "drupal/color": "self.version",
  "drupal/comment": "self.version",
  "drupal/config": "self.version",
  "drupal/config_translation": "self.version",
  "drupal/contact": "self.version",
  "drupal/content_translation": "self.version",
  "drupal/contextual": "self.version",

```

# NAMING CONVENTION

Packages in Drupal context have to follow the [Drupal composer package naming convention \(WIP\)](#).

Current state:

```
drupal/drupal  
drupal/ctools  
drupal/views  
drupal/datetime  
drupal/core-datetime
```



# PATCHES

- Composer patches plugin (Netresearch) or composer-patcher by jp-stacey/webflo
- Work needed for better experience
- Alternative: create (temporary) forks

```
{
  "type": "package",
  "package": {
    "name": "yourname/yourproject-patches",
    "version": "1.0.10",
    "type": "patches",
    "require": {
      "netresearch/composer-patches-plugin": "~1.0"
    },
    "extra": {
      "patches": {
        "drupal/file_entity": {
          "dev-7.x-2.x": [
            {
              "title": "Panelizer settings aren't saved (Issue: https://www.drupal.org/issue/2411111)",
              "url": "https://www.drupal.org/files/issues/panelizer_settings-2411111.patch"
            }
          ]
        }
      }
    }
  }
}
```

# FRONTEND LIBRARIES

- some Components on Packagist (see <https://github.com/components/components>)
- [composer-asset-plugins](#) fetches info from bower and npm
- Alternative: adapt [bower](#) or [npm](#) natively

# EXAMPLE FOR D7

```
{
  "name": "undpaul/some-project",
  "description": "Some drupal project",
  "authors": [
    {
      "name": "Johannes Haseitl",
      "email": "johannes@undpaul.de"
    }
  ],
  "repositories": [
    {
      "type": "composer",
      "url": "http://packagist.drupal-composer.org"
    },
    {
      "type": "vcs",
      "url": "https://github.com/derhasi/slick.git"
    },
    {
```

# RISK OF OVERRIDING PATHS

- Composer preserve paths plugin
- Drupal tangler

```
"require": {  
  "composer/installers": "~1.0",  
  "derhasi/composer-preserve-paths": "0.1.*",  
  "drupal/drupal": "7.*"  
},  
"extra": {  
  "preserve-paths": [  
    "sites/all/modules/contrib",  
    "sites/all/themes/contrib",  
    "sites/all/libraries",  
    "sites/all/drush",  
    "sites/default/settings.php",  
    "sites/default/files",  
    ".htaccess",  
    "robots.txt",  
  ]  
}
```

# PROJECT TEMPLATES

```
composer create-project drupal-composer/drupal-project:8.*
```

- Separate templates for D7 and D8
- <https://github.com/drupal-composer/drupal-project>
- You can create your own.

# COMPOSER VS. DRUSH

# CAN COMPOSER REPLACE DRUSH?

- For most parts: *NO!*
- But `drush dl`, `drush make`

# DOWNLOADING DRUPAL PROJECTS

`composer require` can replace `drush dl`

- bound to a predefined directory pattern  
(*composer installer plugins*)
- Dependencies are resolved automatically
- keeps track of exact versions with `composer.lock`
- Do not use a mix of `drush dl` and `composer require`



# DRUSH MAKE

- Composer in relation to Drush make  
([drupal.org/node/2471553](https://drupal.org/node/2471553))
- Composer Generate  
([drupal.org/project/composer\\_generate](https://drupal.org/project/composer_generate))  
like `drush make-generate`

**COMMUNITY**

# THANKS

- webflo
- winmillwill
- tstoeckler
- davidbarrat
- kasperg
- everyone I forgot

# DRUPAL PROJECTS

- [Composer Manager](#)
- [Composer](#)
- [Composer Generate](#)
- [Composer Autoload](#)
- [Composer Vendor Autoload](#)

# GOALS

- Make Drupal 8 core work with composer
- Easy adoption of Composer for Drupal
- Figuring out best practices:  
[drupal-composer/drupal-project](#)

# TASKS

- **Test, test, test** Drupal Packagist
- Update `packagist.drupal-composer.org`
- Provide metapackages
- More flexible installer: `composer/installers` vs `davidbarratt/custom-installer`
- Missing `version=...` in `.info`-files => [Composify by bforchhammer\(sandbox\)](#)
- Improve patch workflow
- Handling different versions of same package
- Handling frontend-libraries and assets
- **a lot lot more**

# META DOCUMENT

## ON GROUPS.DRUPAL.ORG/COMPOSER

<https://groups.drupal.org/node/450258#Issues>

# DRUPAL-COMPOSER.ORG



Join us!



**THANK YOU!**

Q & A