

# COMPOSER

THERE'S A MODULE (OR LIBRARY) FOR THAT!



Presented by [Rob Loach](#) and [Larry Garfield](#)

**DEPENDENCY MANAGEMENT SUCKS**

# PHP SUCKS AT SHARING

- How do I get a 3rd party library?
- How do I load its code?
- Does it depend on anything?
- Where do I put the files?

**SHARING IS HOW OPEN SOURCE WORKS**

**SUCKING AT SHARING IS HOW  
OPEN SOURCE DIES**

# BUT WE'RE DRUPAL!

- Drupal.org: Great for modules, nothing else
- Duplicate work elsewhere
- Code is not reusable

# 3RD PARTY CODE

- Manually download into your module and pray
- Manually download into /libraries/ and pray
- Manually setup extra hook

(... and pray)

**'DO IT MANUALLY AND PRAY'  
IS FOR LOSERS**

*WE'RE BETTER THAN THAT, RIGHT?*





# A LITTLE HISTORY

# PEAR

- PHP Extension and Application Repository
- Founded 2000 on the "new" PHP 4
- Required root to use properly (WTF?)
- Complex process for adding packages
- Easy to get wrong

# SYMFONY 2.0 & PHPBB

- Initial alpha, Symfony Live Paris, February 2010
- PHPBB: We're rebuilding on it!
- Symfony2 uses 3rd party libs (Doctrine, Monolog)

# DEPENDENCY HELL



# WE NEED A DEPENDENCY SYSTEM!

*"It's dumb to make it Symfony specific"*  
—Jordi Boggiano

# APRIL 2011

Nils Aderman (PHPBB): libzypp -> PHP

Jordi Boggiano (Monolog): Packagist, less sucky front-end for  
PEAR

# WITH THEIR POWERS COMBINED...

**Composer** is a tool for *dependency management* in PHP. It allows you to declare the dependent libraries your project needs and it will install them in your project for you.

# UHH, WHAT?

- No more Copy/Paste of libraries
- Resolves dependencies
- Each project defines its own requirements
- Performs build tasks
- Like `module.info` files, but for any library



# SEPTEMBER 2011

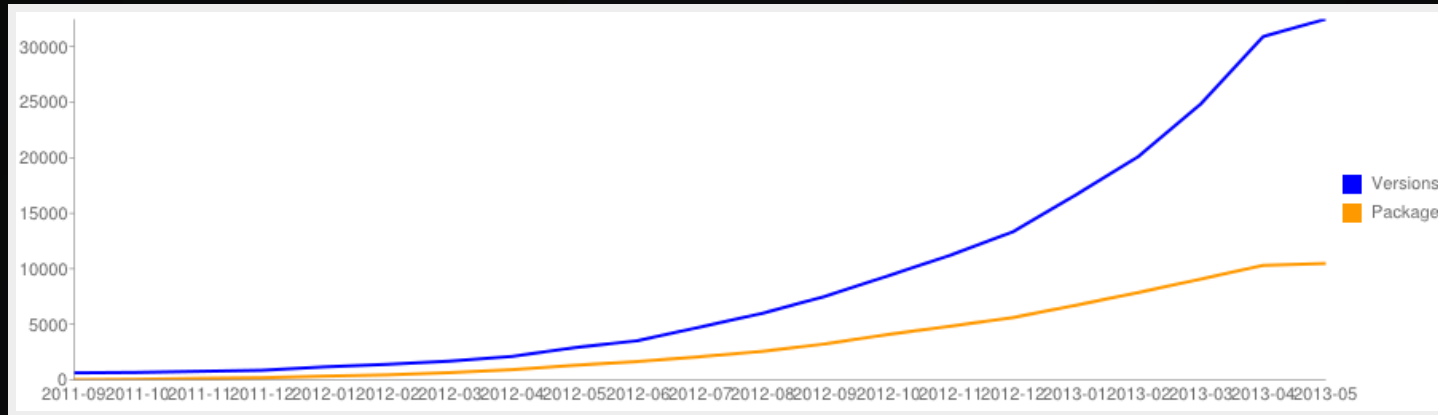
## Packagist.org

1. **Monolog**
2. **Twig**
3. **Pirum**
4. **Swiftmailer**
5. **Pimple**
6. **Symfony**

*“I'd say until early 2012 it was pretty unusable, yet people used it. I guess that's how badly it was needed.”*

— Jordi Boggiano

# PACKAGE COUNT



April 2013: 10,000 packages  
(Not just Symfony!)



# LIBRARIES

Some of the libraries available via Composer include...



PHP framework for web applications

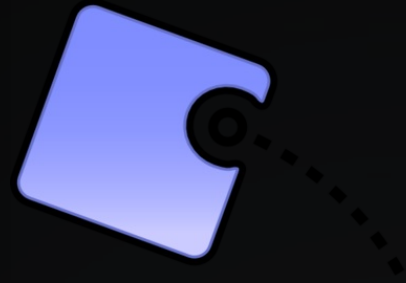
<http://symfony.com>

# ZEND FRAMEWORK



PHP framework for web applications

<http://framework.zend.com>



# PHPUNIT

Unit testing software framework for PHP

<http://phpunit.de>



# GUZZLE

HTTP client & framework for building RESTful web service clients

<http://guzzlephp.org>

# SOLARIUM



Solr client library for PHP

<http://solarium-project.org>

# ELASTICA

Elasticsearch search engine/database for PHP

<http://elastica.io>

# MONOLOG

Log to files, sockets, inboxes, databases and various web services

<https://github.com/seldaek/monolog>

# ASSETIC

JavaScript, stylesheet, and image asset management

<http://github.com/kriswallsmith/assetic>





**LET'S DO THIS**



# COMPOSER.JSON

Base manifest file for your project

```
{
  "name": "crell/mypackage",
  "description": "This package provides some awesomeness.",
  "require": {
    "guzzle/guzzle": "3.4.*"
  },
  "autoload": {
    "psr-0": {
      "MyName\\MyPackage": "src/"
    }
  }
}
```

# COMPOSER INSTALL

Installs all dependencies for your project

```
$ curl -sS https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer successfully installed to: composer.phar
Use it: php composer.phar
$ php composer.phar install
```

# FILE SYSTEM

-rw-rw-r-- composer.json (your composer file)

-rw-rw-r-- composer.lock (generated)

drwxrwxr-x src/ (your code)

drwxrwxr-x vendor/ (everyone else's code)

# INDEX.PHP

```
require_once __DIR__ . '/vendor/autoload.php';

// Every class is now yours to command!

// Autoload on demand! Your work: zero.
$client = new Guzzle\Http\Client('https://api.github.com');
$request = $client->get('/user')->setAuth('user', 'pass');
$response = $request->send();
echo $response->getBody();
```

# COMPOSER UPDATE

Updates all installed dependencies to the latest version

```
$ php composer.phar update  
Loading composer repositories with package information  
Updating dependencies
```

# MORE SCHEMA OPTIONS

## Schema Documentation

```
{
  "license": "MIT",
  "require": {
    "php": ">=5.3.10",
    "guzzle/guzzle": "3.4.*"
  },
  "require-dev": {
    "phpunit/phpunit": "3.7.*"
  },
  "suggest": {
    "monolog/monolog": "Advanced logging package"
  }
}
```

# PRIVATE REPOSITORIES

*Satis* allows you to host your own Composer repositories

```
{
  "repositories": [
    {
      "type": "composer",
      "url": "http://packages.example.org/"
    }
  ],
  "require": {
    "company/package": "1.2.0",
    "company/package2": "1.5.2",
    "company/package3": "dev-master"
  }
}
```

[github.com/composer/satis](https://github.com/composer/satis)





# COMPOSER FOR DRUPAL

# DRUSH COMPOSER

Allows running Composer commands through **Drush**

```
$ drush dl composer-8.x  
$ drush composer
```

[drupal.org/project/composer](https://drupal.org/project/composer)

# COMPOSER AUTOLOAD

Automatically loads any generated Composer files

[Jimmy Berry- drupal.org/project/composer\\_autoload](https://drupal.org/project/composer_autoload)

## Composer autoload

[View](#) [Version control](#)

Posted by [boombatower](#) on August 11, 2012 at 8:45am

Caches a list of autoload.php files generated by [composer](#) and automatically includes them during [hook\\_init\(\)](#).

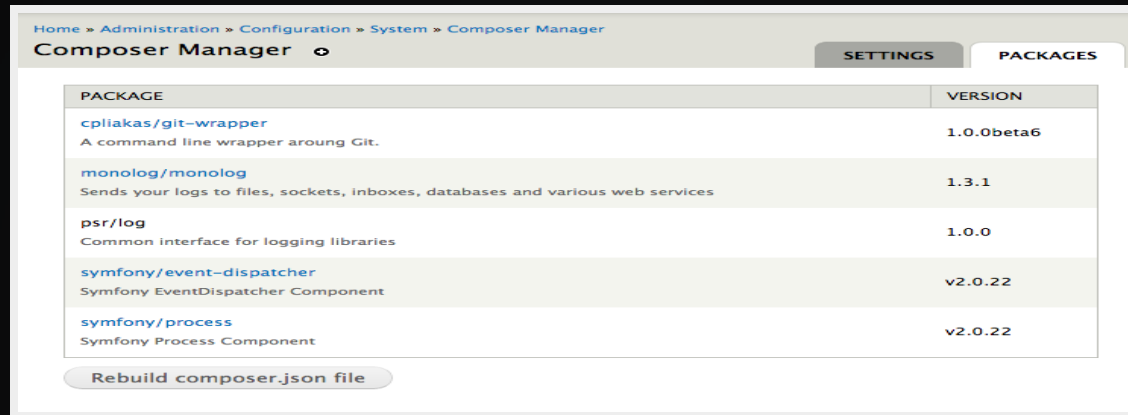
The [composer Drupal project](#) in conjunction with [drush](#) make generating autoload.php files very simple.

### Downloads

Version	Downloads	Date	Links
<a href="#">7.x-1.0</a>	<a href="#">tar.gz (7.74 KB)</a>   <a href="#">zip (8.49 KB)</a>	2012-Aug-11	<a href="#">Notes</a>
<a href="#">6.x-1.0</a>	<a href="#">tar.gz (7.74 KB)</a>   <a href="#">zip (8.49 KB)</a>	2012-Oct-24	<a href="#">Notes</a>

# COMPOSER MANAGER

Builds Composer files from module $\oplus$ composer.json files



The screenshot shows the 'Composer Manager' interface with a 'PACKAGES' tab selected. It displays a table of installed packages with their names and versions. Below the table is a button to 'Rebuild composer.json file'.

PACKAGE	VERSION
<a href="#">cpliakas/git-wrapper</a> A command line wrapper around Git.	1.0.0beta6
<a href="#">monolog/monolog</a> Sends your logs to files, sockets, inboxes, databases and various web services	1.3.1
<a href="#">psr/log</a> Common interface for logging libraries	1.0.0
<a href="#">symfony/event-dispatcher</a> Symfony EventDispatcher Component	v2.0.22
<a href="#">symfony/process</a> Symfony Process Component	v2.0.22

Rebuild composer.json file

Chris Pliakas- [drupal.org/project/composer\\_manager](https://drupal.org/project/composer_manager)

# DRUSH COMPOSER MANAGER

Downloads and installs dependencies for Drupal modules

```
$ drush dl composer_manager monolog
$ drush en composer_manager monolog
$ drush composer-manager
Loading composer repositories with package information
Installing dependencies
- Installing psr/log (1.0.0)
  Downloading: 100%

- Installing monolog/monolog (1.5.0)
  Downloading: 100%
```

[drupal.org/project/composer\\_manager](https://drupal.org/project/composer_manager)

# COMPOSER INSTALLERS

Facilitates installation of Drupal modules/themes/profiles



[github.com/composer/installers](https://github.com/composer/installers)

# COMPOSER INSTALLERS

Add a *composer.json* file in your module/theme directory

```
{  
  "name": "dries/mymodule",  
  "type": "drupal-module",  
  "require": {  
    "composer/installers": "*"   
  }  
}
```

[github.com/composer/installers](https://github.com/composer/installers)

Adding these to each module is cumbersome though, so there is...

# DRUPAL PACKAGIST

Indexes Drupal themes/modules for installation using Satis

```
{
  "repositories": [
    {
      "type": "composer",
      "url": "http://drugist.espend.de"
    }
  ],
  "require": {
    "drupal/views": "*"
  }
}
```

**Haehnchen- [github.com/haehnchen/drupal-packagist](https://github.com/haehnchen/drupal-packagist)**

Still very early





# SPRINT: FRIDAY

Sprint with the community on Friday. Tasks for every skill set.  
Mentors are available for new contributors.

Optional Friday morning workshop will help you set up  
community tools.

Drupal tag: [#composer](#) Follow [@drupalmentoring](#)

