Composer packages

Installing and Using Composer, Packagist, Packaging your code

SOFTWARE UNIVERSITY FOUNDATION

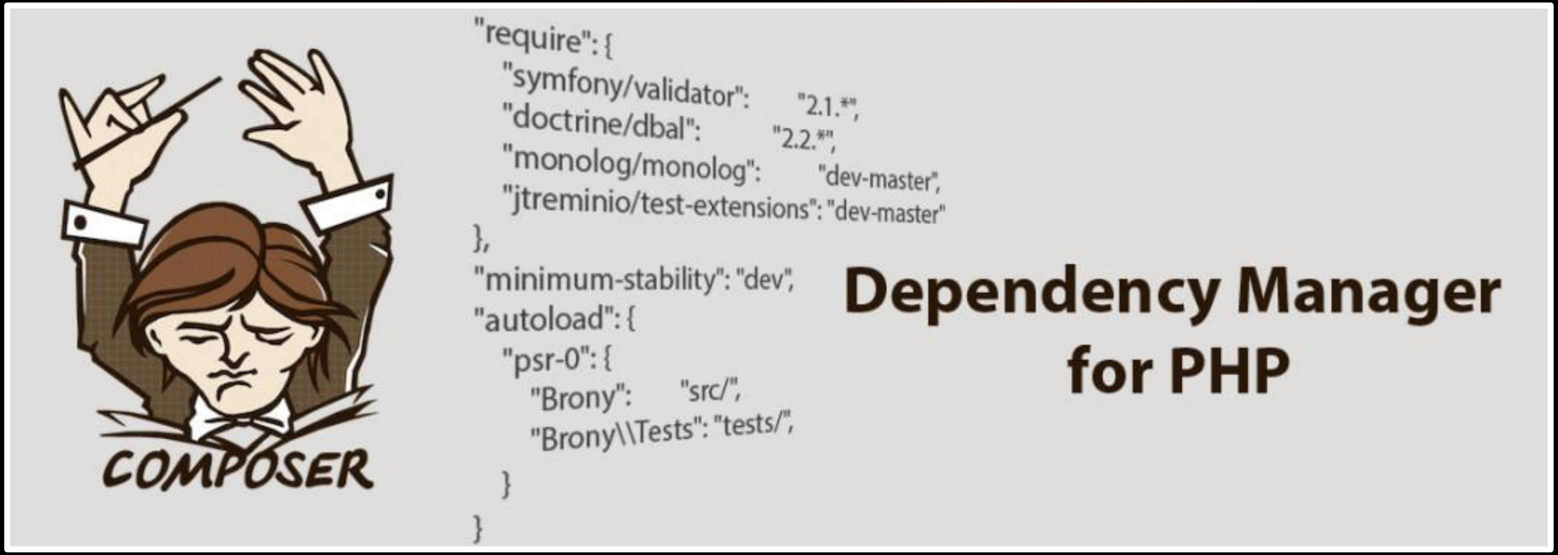
Mario Peshev

Technical Trainer

http://peshev.net

Software University

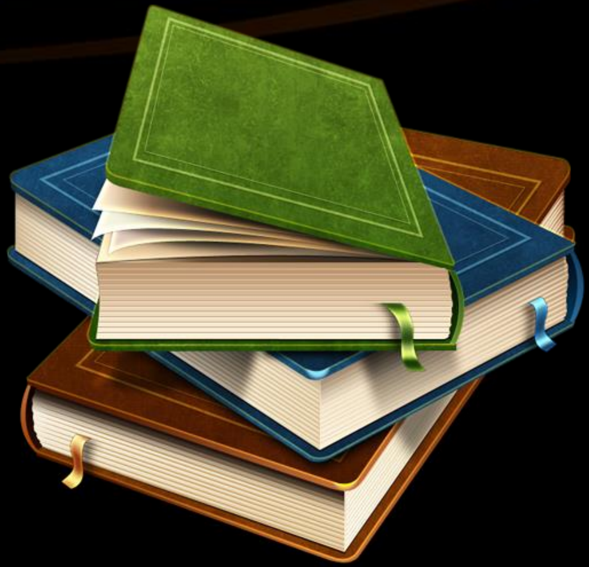
http://softuni.bg



```
"require": {
    "symfony/validator":        "2.1.*",
    "doctrine/dbal":            "2.2.*",
    "monolog/monolog":          "dev-master",
    "jtreminio/test-extensions": "dev-master"
},
"minimum-stability": "dev",
"autoload": {
    "psr-0": {
        "Brony":        "src/",
        "Brony\\Tests": "tests/",
    }
}
```

Table of Contents

1. PHP package
2. Composer overview
3. Advantages
4. Download and install
5. Autoloading
6. Packagist
7. Using Composer
8. Packaging your own code
9. Famous Libraries Examples



PHP package

```
import(‘weatherForecast.arras.*’);
include(‘weather.arras.*’);
```

What is PHP package?

- A directory containing something. In this case it is PHP code, but in theory it could be anything.
- Used to help you logically group related elements just like you write classes to group related functions and data together.

Composer overview

requires PHP 5.3.2+ to run

What is Composer?

- Dependency manager for PHP.
- List your project’s dependencies in a composer.json file and with a few simple commands - Composer will automatically download your project’s dependencies and setup autoloading for you.

What is Composer?

- It is not a package manager.
- It deals with "packages" or libraries, but it manages them on a per-project basis, installing them in a directory (e.g. vendor) inside your project.
- By default it will never install anything globally.

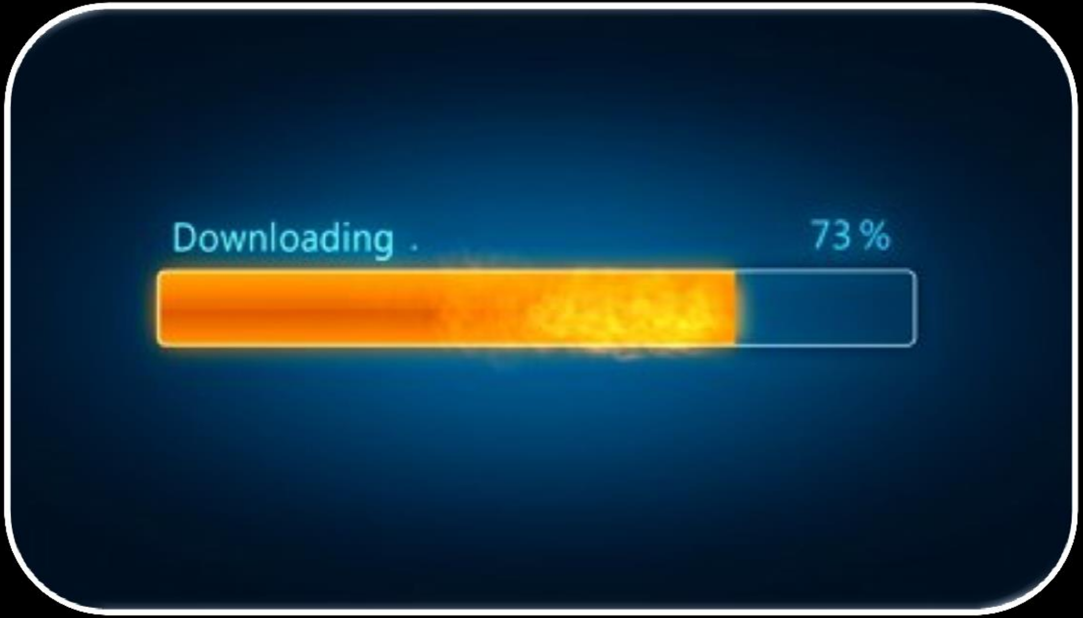
Advantages

Why use Composer?

- lot of PHP libraries that are compatible with Composer
- Already use PHP framework
- ready to be used in your project
- these “packages” are listed on

Packagist *is the official repository for Composer-compatible PHP libraries!*

Download and Install



Download and Install for NON-Windows users

- Download: $ curl -sS https://getcomposer.org/installer | php

- If you don't have curl:

```
$ php -r "readfile('https://getcomposer.org/installer');" | php
```

Download and Install for NON-Windows users

- recommended directory is /usr/local/bin

- install composer to a specific directory:

```
$ cd /path/to/my/project
$ curl -sS https://getcomposer.org/installer | php -- --install-
dir=bin
```

Download and Install on Windows

- Download https://getcomposer.org/Composer-Setup.exe

```
php -r "readfile('https://getcomposer.org/installer');" | php
```

- If you don’t use the Composer-Setup installer - don’t forget to set up your $PATH so that you can just call composer from any directory in your command line.

Packagist

Packagist

The PHP package archivist.

Packagist

- Packagist is the main Composer repository
- A Composer repository is a place where you can get packages from
- You can automatically require any package that is available there
- https://packagist.org – You can browse all available packages on the official website

Using Composer

Autoloading

- To use it, just add the following line to your code's bootstrap process:

```
require ‘vendor/autoload.php’;
```

Adding a library

- To add a library to your project, create a file named composer.json with content that resembles this example:

```
{
       "require": {
              "illuminate/foundation": "1.0.*“
      },
      "minimum-stability": "dev“
}
```

- You must run composer to have your dependencies installed:

```
$ php composer.phar install
```

Using Composer

Live Demo

Packaging Your Own Code

Packaging Your Own Code

- Using custom repositories key:

```
"repositories": [
    {
        "type": "package",
        "package": {
            "name": "weaverryan/derp-dangerzone",
            "version": "dev-master",
            "source": {
                "url": "git://github.com/weaverryan/derp-
dangerzone.git",
                "type": "git",
                "reference": "master"
            },
```

Packaging Your Own Code (2)

```
"autoload": {
                "psr-0" : {
                    "Weaverryan\\DangerZone" : "src"
                }
            }
        }
    }
],
```

Packaging Your Own Code (3)

- You can later require it

```
"require": {
    "weaverryan/derp-dangerzone": "dev-master"
},
```

Packaging Your Own Code (4)

- You can simplify the packaging by creating a composer.json file in the library itself with a simillar content

```
{
    "name": "weaverryan/derp-dangerzone",
    "autoload": {
        "psr-0" : {
            "Weaverryan\\DangerZone" : "src"
        }
    }
}
```

- Version and Source keys are removed. Composer will look at branches and tags to get information

Packaging Your Own Code (5)

- Update your composer.json to have the following:

```
"repositories": [
    {
        "type": "vcs",
        "url": "https://github.com/weaverryan/derp-
dangerzone"
    }
],
```

- Now, instead of a packages key, you have a simpler vcs key, which consumes the repository's composer.json file

Registering with Packagist

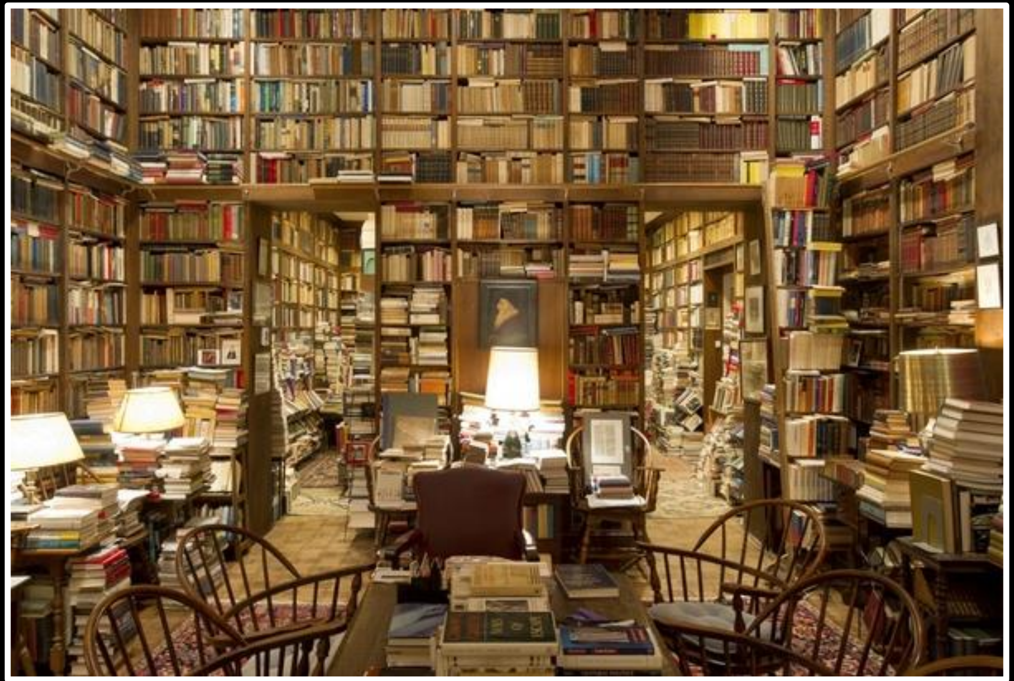
- In order to register you package with packagist you have to fill a form in the official website
- Once you have registered you library with Packagist and it has been crawled – you can use it by just requiring it

```
"require": {
        "weaverryan/derp-dangerzone": "dev-master"
},
```

Packaging Your Own Code

Live Demo

Famous Libraries Examples



Facebook SDK

- The Facebook SDK is added to Packagist

```
{
    "require": {
        "facebook/php-sdk": "dev-master"
    }
}
```

- Run composer and you will have a file named composer.lock and a directory named vendor
- The lock file sets the dependency to one specific revision
- In the vendor directory you will find the Facebook SDK

PHP Unit

- The Unit Testing library PHP Unit is also added to Packagist

```
{
    "require": {
        "phpunit/phpunit": "4.3.*@dev"
    }
}
```

Laravel Framework

- The Famous MVC Framework – Laravel can be obtained from Packagist too

```
{
       "require": {
             "laravel/laravel": "dev-master"
      }
}
```

And a thousand more libraries on Packagist…

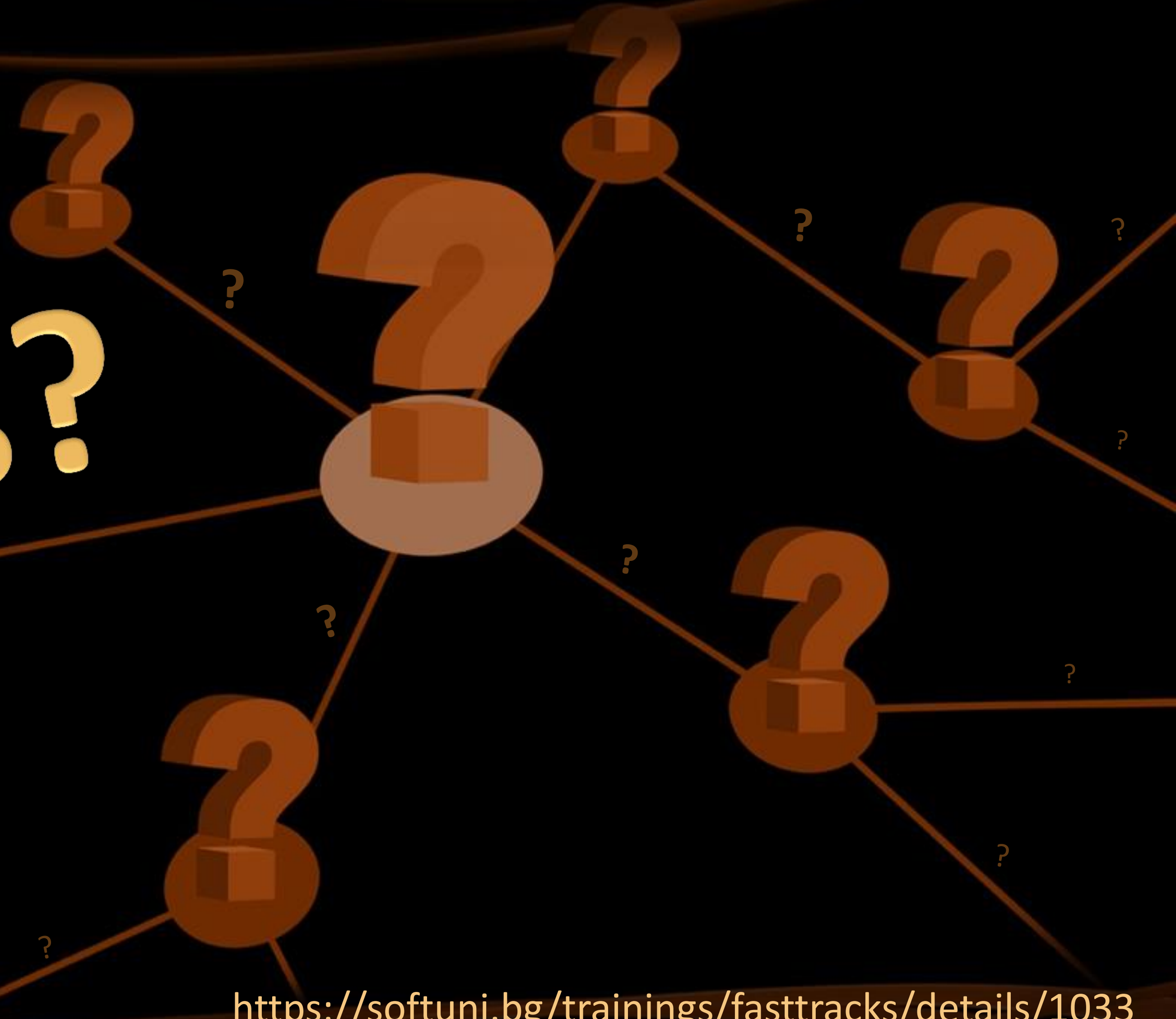
Famous Libraries Examples

Live Demo

Summary

- Easy to use – just list your dependencies in a composer.json file
- Packagist is the default package repository
- You can package your own code
- You can submit your package to Packagist

Questions?



License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

- Attribution: this work may contain portions from
  - "PHP Manual" by The PHP Group under CC-BY license
  - "PHP and MySQL Web Development" course by Telerik Academy under CC-BY-NC-SA license

Free Trainings @ Software University

- Software University Foundation – softuni.org
- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University @ YouTube
  - youtube.com/SoftwareUniversity
- Software University Forums – forum.softuni.bg