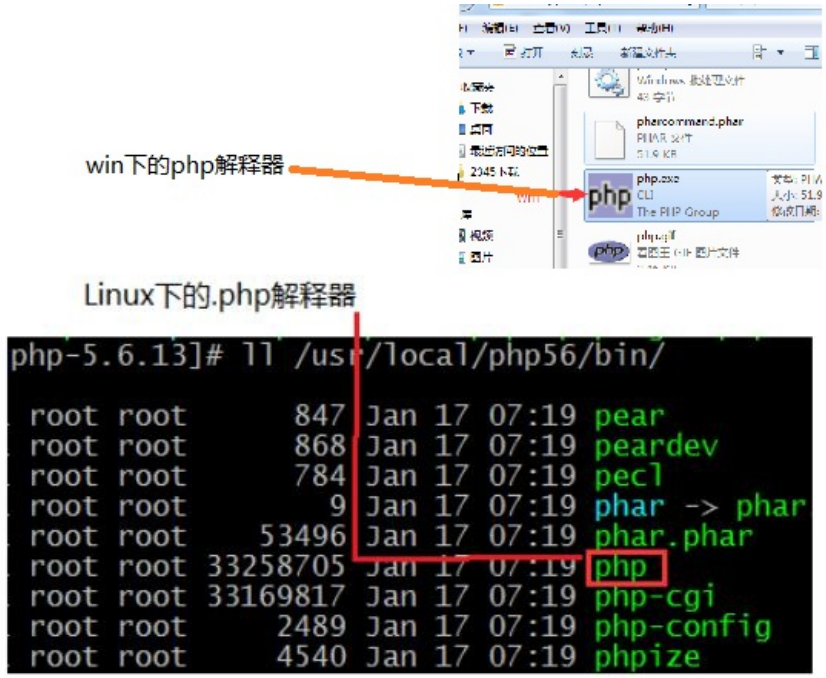
PHP之Composer类库管理

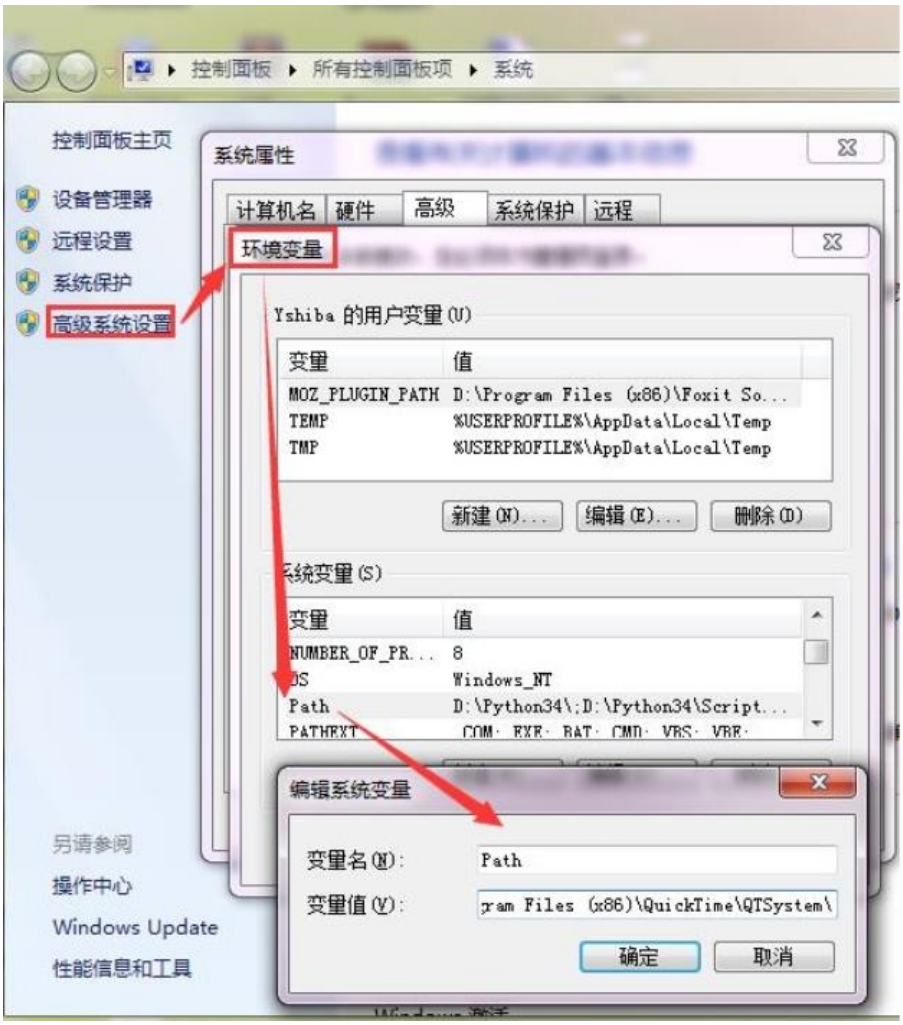
> 布尔教育 http://www.itbool.com

php命令行的调用

php是一种脚本语言,语言通过"PHP解释器"来解释执行.
我们经常在web开发中通过浏览器运行xx.php.
其实是nginx/apache 通知php解释器来执行xx.php.
我们也可以直接调用php解释器,让它来执行xx.php.
更通俗的说:命令行方式来调用.



`path/to/php path/to/php-file`
命令行调用实例:
win例: `D:\xampp\php.exe`
解释: 让D盘`xampp\php`目录下的`php.exe`,去解释执行`D:\blog\a.php`
lin例: `/usr/local/php/bin/php` `/root/a.php`
解释:让`/usr/local/php/bin/php` 程序,去解释执行 `/root/a.php`
让命令行更简短
linux操作系统会到/usr/bin下找相关的命令.
我们在此目录下建立php的软链接:
ln -s /usr/local/php56/bin/php /usr/bin/php
这样,我们可以更简单的调用php解释器了.
在win下,我们用"环境变量"来告诉操作系统到哪儿找php.exe
如下图,把你的php.exe所在的目录路径,加入环境变量.
以我的电脑为例,我在path变量末尾加上;D:\xampp\php
注意:前面有个;分号



加入环境变量后,重新打开cmd命令行窗口
运行 php -v,看到下例:

```
D:\www>php -v
PHP 5.6.20 (cli) (built: Mar 31 2016 14:55:52)
Copyright (c) 1997-2016 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
```

1章 PHP之前的类库管理

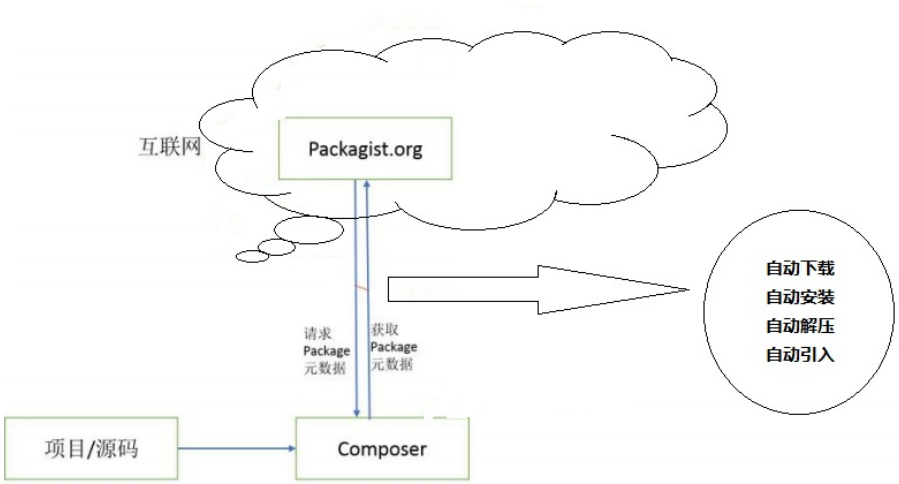
php开发者很多,并且在web开发领域占据绝对统治地位.
在20年的发展过程中,无数开发者开发了无数的类库.
但是,当你想用某个库时,是怎么做的呢?
比如:phpmailer,一个发邮件的库,我们往往这样做:

1. 打开搜索引擎,搜索phpmailer.
2. 从phpmailer官网或不知名的网站,下载源码.
3. 解压然后放到自己的项目中,在看手册调用.
4. 假如phpmailer需要smtp类才能正常运行,又要继续从开始下载.

如上,我们可以看出,php的开发者虽多,类库虽多,但存在以下几个问题.

1. 没有统一的资源仓库,到处乱找.
2. 没有统一的安装方式.rar,zip,tar各种包都有.下载后自己得整理.
3. 遇到库的依赖关系,得自己再次下载解决.

可见,php的库虽多,但都是散落在互联网的各个角落,不成系统,且没有统一的规范.
java有maven , python有pip, node.js有npm,前端有bower
自从有了composer,这些库就被组织起来了.



2章 安装composer

我们安装composer需要一定的要求,既然我们来到php这个目录下,就把需要的扩展打开;
php版本要求:>=5.5.9;
.OpenSSL扩展
.PDO扩展
.Mbstring扩展
如不满足以上条件,请修改`php.ini` 配置或重新编译PH

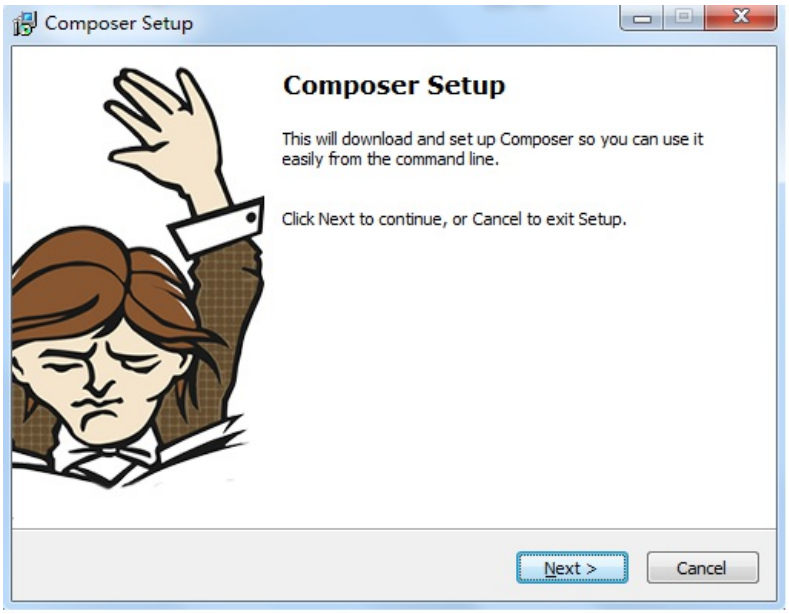
2.1 liu下安装composer

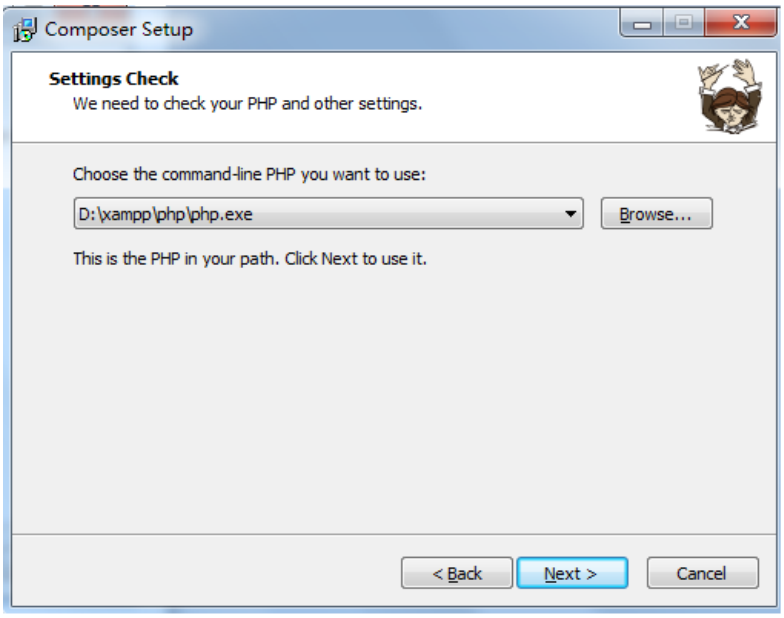
cd
curl -sS https://getcomposer.org/installer | php
mv composer.phar /usr/bin/composer

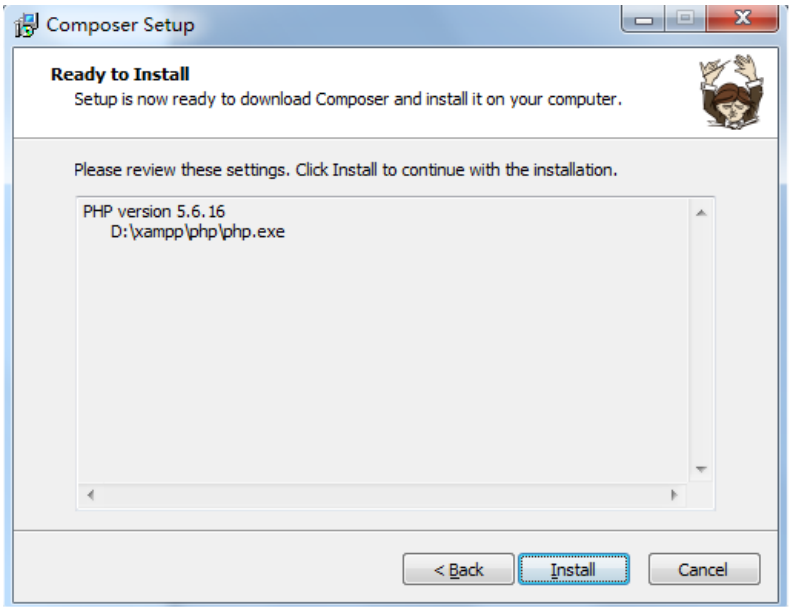
2.2 win下安装composer

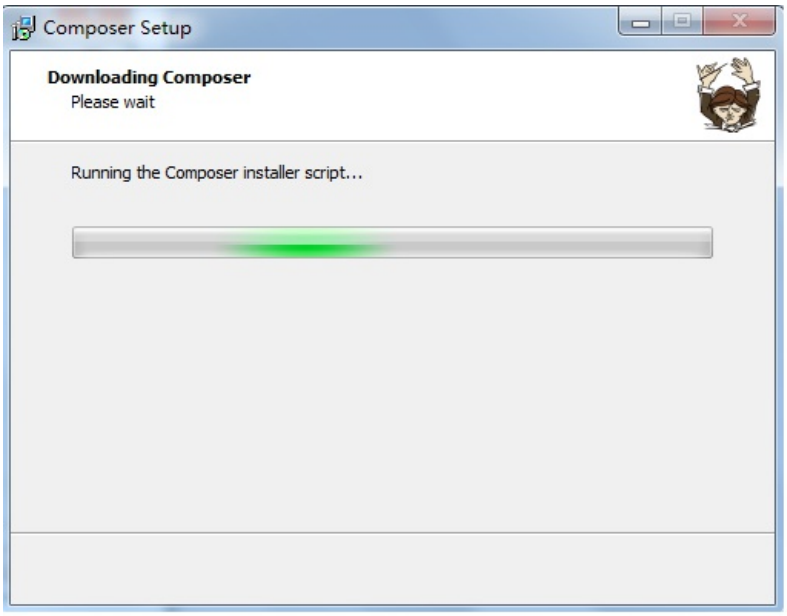
有两种办法:::

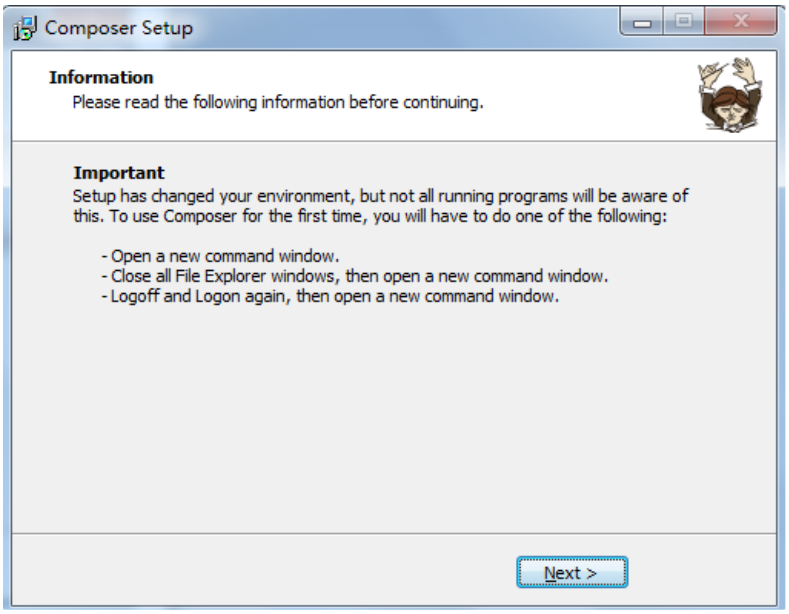
1,从官网上下载,然后解压,一路狂飙(next)


Composer Setup
Composer Setup
This will download and set up Composer so you can use it easily from the command line.
Click Next to continue, or Cancel to exit Setup.
Next >
Cancel


Composer Setup
Settings Check
We need to check your PHP and other settings.
Choose the command-line PHP you want to use:
D:\xampp\php\php.exe
Browse...
This is the PHP in your path. Click Next to use it.
< Back
Next >
Cancel


Composer Setup
Ready to Install
Setup is now ready to download Composer and install it on your computer.
Please review these settings. Click Install to continue with the installation.
PHP version 5.6.16
D:\xampp\php\php.exe
< Back
Install
Cancel


Composer Setup
Downloading Composer
Please wait
Running the Composer installer script...



::完成以上步骤后从新打开cmd命令行窗口,输入 `composer about`

2,离线安装:

```
由于国内的高墙,win下安装composer十分吃力
我做了一个comoposer离线安装包.

用法:
1. 把php.exe所在目录的路径,加入环境变量,保证随处可以cmd下调用php命令.
2. 把本压缩包下的`composer.bat`,`composer.phar`解压放到php.exe相同的目录下.
3. (xampp下php目录下)

打开cmd窗口 , composer -v ,看到如下类似信息,即宣告成功.
composer -v
   ______
  / ____/___  ____ ___  ____  ____  ________  _____
 / /   / __ \/ __ `__ \/ __ \/ __ \/ ___/ _ \/ ___/
/ /___/ /_/ / / / / / / /_/ / /_/ (__  )  __/ /
\____/\____/_/ /_/ /_/ .___/\____/____/\___/_/
                    /_/
Composer version 1.2.1 2016-09-12 11:27:19
```

2.3 osx安装composer

```
1.直接输入命令安装,在命令行执行
curl -sS https://getcomposer.org/installer | php
如果没安装 curl 执行以下代码
php -r "readfile('https://getcomposer.org/installer');" | php
然后执行
sudo mv composer.phar /usr/local/bin/composer
然后在终端用管理员权限运行composer命令,有可能报错
-bash: /usr/local/bin/composer: Permission denied
这是权限错误,所以修改composer权限
sudo chmod a+x /usr/local/bin/composer
2,官网上手动下载composer.phar,然后直接放到/usr/local/bin/composer目录下,直接去执行

注意:sudo mv composer.phar /usr/local/bin/composer 不用去创建composer目录
```

2.4 配置composer 修改国内镜像

因为composer的软件仓库位置在国外,所以我们修改国内镜像,提高速度

`composer config -g repositories.packagist composer https://packagist.phpcomposer.com`

3章 初试composer

3.1 为项目引入某个库

假如我们blog项目中需要smarty这个组件,我们来到https://packagist.org/ 搜索smarty,并注意看项目的目录和版本信息.
我们需要在项目根目录下写入composer.json文件,注意必须是json格式
内容如下:

```
{
   "require" : {
         "smarty/smarty":"3.1.30"
    }
}
```

然后,我们到项目的根目录下输入composer install

```
D:\xampp\htdocs\myphp\blog>composer install
Loading composer repositories with package information
Updating dependencies (including require-dev)
  - Installing smarty/smarty (v3.1.30)
    Downloading: 100%

Writing lock file
Generating autoload files
```

查看项目根目录下,有一个vendor的文件夹,文件夹内有一个smarty的文件夹

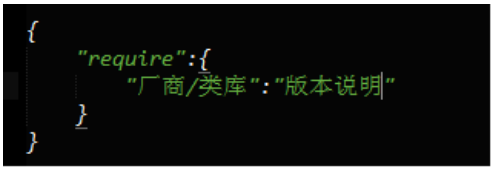
3.2 如何加载引入的库

.如果我们通过composer引入上百个类库,我们如何引入呢,手工require?
.不用的,composer的类都满足一定的标准(psr-4标准);
.Composer中生成了一个vendor/autoload.php的文件(我们都只到autoload是自动加载)
.利用它你可以很容易的就引入这个文件,会得到完善的自动加载支持.
.下面我们自己在项目下写一个文件看如何引入smarty这个类

```
require(__DIR__ . '/vendor/autoload.php');

print_r( new Smarty() );
```

3.3 配置文件格式



3.4 添加某个新库

假如项目后期需要引入某个新库,比如phpmailer处理库
只需要在require后添加就可以,例:

```
{
   "require" : {
         "smarty/smarty":"3.1.30",
         "phpmailer/phpmailer": "5.2.16"
    }
}
```

然后composer install,可能会出现composer.json较新,composer.lock较旧这样的错误

```
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Warning: The lock file is not up to date with the latest changes in composer.jso
```

```
n. You may be getting outdated dependencies. Run update to update them.
Nothing to install or update
Generating autoload files
```

解决办法::(warning 提示的就是答案)
`composer update`

3.5 卸载某个库

以卸载phpmailer为例
`composer remove phpmailer/phpmailer` ,不必加版本号.

卸载成功后,`composer.json`自动变为:

```
{
   "require" : {
         "smarty/smarty":"3.1.30"
    }
}
```

3.6不配置json文件新增某库

`composer require 厂商/类库=版本说明`
比如:我们新增另外一个叫`phpmailer`的库,先到`https://packagist.org/`搜索`phpmailer`
`composer require phpmailer/phpmailer=5.2.16`
运行后会发现vendor下多出来一个`phpmailer'`目录.
且`composer.json`文件和`comopose.lock`文件也已自动更新

如下::

```
{
   "require" : {
         "smarty/smarty":"3.1.30",
         "phpmailer/phpmailer": "5.2.16"
    }
}
```

4章 库的版本说明

版本约束可以由以下几个方法来指定

名称	实例	描述
确切的版本号	`1.0.2`	你可以指定包的确切版本。
范围	`>=1.0` `>=1.0,<2.0` `>=1.0,<1.1|>=1.2`	
通配符	`1.0.*`	你可以使用通配符* 来指定一种模式。`1.0.*` 与 `>=1.0,<1.1`是等效的。
赋值运算符	`~1.2`	这对于遵循语义化版本号的项目非常有用。`~1.2`相当于`>=1.2,<2.0`。想要了解更多，请阅读下一小节。

波浪号运算

`~`最好用例子来解释：`~1.2`相当于`>=1.2,<2.0`， 而`~1.2.3` 相当于 `>=1.2.3,<1.3`。 即， 版本号最后一位数字可且只可提升.

5章 composer 创建项目

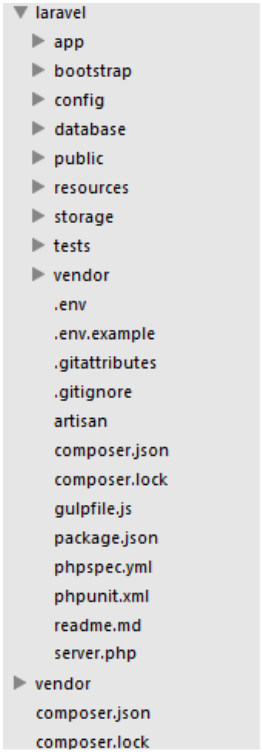
刚才我们是用composer引入某个库,库放在`vendor`目录下,供项目使用

那对于项目,我们需要怎么去做,显示这种效果呢?
我们需要下载项目的源码,然后放在自定义的文件夹下,而不是vendor目录下

然后执行cmd命令

```
composer create-project laravel/laravel=5.1.33
```

执行后,在当前目录下,我们会发下多出一个laravel的目录
代码结构如下::



你可以看到， laravel目录下， 有个composer.json文件， 说明他需要依赖很多库。 当然， 这些都可以自动下载。