参考：<http://blog.csdn.net/baixiaoshi/article/details/50993696>

[composer详解](http://blog.csdn.net/baixiaoshi/article/details/50993696)

标签： [管理](http://www.csdn.net/tag/%e7%ae%a1%e7%90%86)

2016-03-27 20:43 1887人阅读 [评论](http://blog.csdn.net/baixiaoshi/article/details/50993696#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/baixiaoshi/article/details/50993696#report)

 分类：

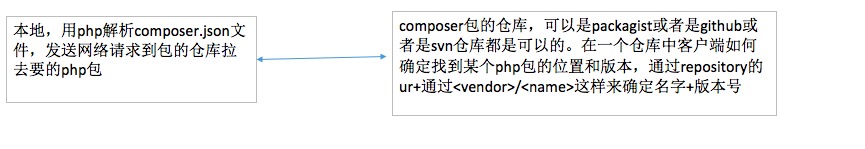
php基础语法（40）

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录[(?)[+]](http://blog.csdn.net/baixiaoshi/article/details/50993696)

[**PHP**](http://lib.csdn.net/base/php)要什么包管理工具，虽然我至今觉得这个东西有点多余，但是经常用的到，还是做一下笔记的，因为记性不好啦。

composer架构原理



composer的原理和其他的包管理工具都是一样的，只是实现的细节有些不同，例如yum，例如brew，例如apt-get还有packets。

composer管理的原理上图已经表达的非常的清楚了，通过上图我们应该知道如何正确的使用composer了吧

使用自己的composer库

1.首先我们要搞定本地，也就是图上本地的composer.json

可以使用命令或者自己创建

composer init #创建一个composer.json文件

composer install #第一次从远处仓库拉取php包的时候执行，该命令会根据composer.json的配置下载指定的库和依赖

composer update #修改本地的composer.json后执行，修改composer.json后还会更新composer.lock文件，composer install 后由composer.lock来控制库的版本和依赖，是不会自动更新的，除非执行了composer update

composer about #查看composer的信 息

composer info

composer self-update#composer.phar自己更新

composer –-help#查看帮助

composer require #包含一个php包进来

remove 移除一个包

本地的必要配置如下

{

"repositories": [

{//自定义仓库

"type": "vcs",

"url": "git@github.com:baixiaoshi/composer\_test.git"

},

{//官方仓库，false表示禁用官方仓库

"packagist": false

}

],

"require": {

"hello/hello": "dev-master"

}

}

#这是一个最简短的配置,现在要拉取github上的一个名字叫做hello/hello的php包

repositories : 就是php远端仓库的地址啦，这里我用的类型是vcs(me：version control)也就是版本控制github上自己的测试包

packagist : 这里是禁用(false)了官方的包仓库地址，

我们可以配置官方的地址来拉取php包

{

"type": "composer",

"url": "http://packagist.phpcomposer.com"

}

require选项包含远端php包，其中hello/hello是包的名字,所以待会我们自己创建一个包的名字也就是这个了,dev-master是版本的意思，我们经常看到这样的吧

{

"monolog/monolog": "~0.1" #~号表示大于的意思

}

2.下面就来搞定远端了,客户端的配置明显是要拉取一个名字叫做hello/hello,版本是dev-master的包,那就直接来了

{

"name": "hello/hello",#包的名字，客户端要找这个的啦

"autoload": {

"psr-4": {

"Hello\\": ""#这里设置包拉下来自动加载包里面的那些文件，是通过命令空间来自动加载的哦

}

},

"require": {

"php": ">=5.4.0"#如果有对其他的一些包的依赖还是require在这里就好了

}

}

<?php

namespace Hello;#这个命名空间就是上面autoload配置中的啦

class Hello {

public function say\_hello() {

exit('hello world');

}

}

3.回到客户端执行composer install 当然就拉下来了这个包，当然只有一个[**测试**](http://lib.csdn.net/base/softwaretest)文件而已.此时的你是不是轻松+愉快了呢顶