**已弃用** - 截止到2014年10月21日，PSR-0已被弃用。 推荐替代使用[PSR-4](http://www.php-fig.org/psr/psr-4/)。

下文描述了若要使用一个通用的自动加载器(autoloader)，你所需要遵守的规范：

**规范**

* 一个完全标准的命名空间(namespace)和类(class)的结构是这样的：\<Vendor Name>\(<Namespace>\)\*<Class Name>
* 每个命名空间(namespace)都必须有一个顶级的空间名(namespace)("组织名(Vendor Name)")。
* 每个命名空间(namespace)中可以根据需要使用任意数量的子命名空间(sub-namespace)。
* 从文件系统中加载源文件时，空间名(namespace)中的分隔符将被转换为 DIRECTORY\_SEPARATOR。
* 类名(class name)中的每个下划线\_都将被转换为一个DIRECTORY\_SEPARATOR。下划线\_在空间名(namespace)中没有什么特殊的意义。
* 完全标准的命名空间(namespace)和类(class)从文件系统加载源文件时将会加上.php后缀。
* 组织名(vendor name)，空间名(namespace)，类名(class name)都由大小写字母组合而成。

**示例**

* \Doctrine\Common\IsolatedClassLoader => /path/to/project/lib/vendor/Doctrine/Common/IsolatedClassLoader.php
* \Symfony\Core\Request => /path/to/project/lib/vendor/Symfony/Core/Request.php
* \Zend\Acl => /path/to/project/lib/vendor/Zend/Acl.php
* \Zend\Mail\Message => /path/to/project/lib/vendor/Zend/Mail/Message.php

**空间名(namespace)和类名(class name)中的下划线**

* \namespace\package\Class\_Name => /path/to/project/lib/vendor/namespace/package/Class/Name.php
* \namespace\package\_name\Class\_Name => /path/to/project/lib/vendor/namespace/package\_name/Class/Name.php

以上是我们为实现通用的自动加载而制定的最低标准。你可以利用能够自动加载PHP 5.3类的SplClassLoader来测试你的代码是否符合这些标准。

**实例**

下面是一个怎样利用上述标准来实现自动加载的示例函数。

<?php

function autoload($className)

{

$className = ltrim($className, '\\');

$fileName = '';

$namespace = '';

if ($lastNsPos = strrpos($className, '\\')) {

$namespace = substr($className, 0, $lastNsPos);

$className = substr($className, $lastNsPos + 1);

$fileName = str\_replace('\\', DIRECTORY\_SEPARATOR, $namespace) . DIRECTORY\_SEPARATOR;

}

$fileName .= str\_replace('\_', DIRECTORY\_SEPARATOR, $className) . '.php';

require $fileName;

}

**SplClassLoader实现**

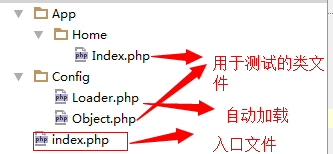
下面的gist是一个按照上面建议的标准来自动加载类的SplClassLoader实例。这是依据这些标准来加载PHP 5.3类的推荐方案。

* <http://gist.github.com/221634>

1、PSR-0规范  
   [1]命名空间必须与绝对路径一致  
   [2]类名首字母必须大写  
   [3]除去入口文件外，其他“.php”必须只有一个类  
   [4]php类文件必须自动载入，不采用include等  
   [5]单一入口

2、案例

  [1]目录结构



  [2]源码

  index.php

**[php]** [view plaincopy](http://blog.csdn.net/zhao1234567890123456/article/details/41628163?utm_source=tuicool&utm_medium=referral)[在CODE上查看代码片](https://code.csdn.net/snippets/537634)

1. <?php
2. define('BASEDIE',\_\_DIR\_\_\_\_);
3. **require\_once**('/Config/Loader.php');
4. spl\_autoload('\\Config\\Loader.php::autoload');
5. Config\Object::test();
6. App\Home\Index::test();

Config/Object.php

**[php]** [view plaincopy](http://blog.csdn.net/zhao1234567890123456/article/details/41628163?utm_source=tuicool&utm_medium=referral)[在CODE上查看代码片](https://code.csdn.net/snippets/537634)

1. <?php
2. namespace Config;
4. **class** Object{
5. **static** **function** test(){
6. echo "nihao";
7. }
8. }

Config/Loader.php

**[php]** [view plaincopy](http://blog.csdn.net/zhao1234567890123456/article/details/41628163?utm_source=tuicool&utm_medium=referral)[在CODE上查看代码片](https://code.csdn.net/snippets/537634)

1. <?php
2. namespace Config;

5. **class** Loader{
6. **static** **function**   autoload($class)
7. {
8. **require\_once**(BASEDIE.'/Config/'.str\_replace('\\','/',$class).'.php');
9. }
10. }

App/Home/Index.php

**[php]** [view plaincopy](http://blog.csdn.net/zhao1234567890123456/article/details/41628163?utm_source=tuicool&utm_medium=referral)[在CODE上查看代码片](https://code.csdn.net/snippets/537634)

1. <?php
2. namespace App\Home;
4. **class** Index{
5. **static**  **function** test(){
6. echo "ceshixinxi";
7. }
8. }