PHP7扩展开发之hello word

2016-01-18　[信海龙](https://www.bo56.com/author/dragonsea/)

[12条评论](https://www.bo56.com/php7%e6%89%a9%e5%b1%95%e5%bc%80%e5%8f%91%e4%b9%8bhello-word/#comments)

本文是以PHP7作为基础，讲解如何从零开始创建一个PHP扩展。本文主要讲解创建一个扩展的基本步骤都有哪些。示例中，我们将实现如下功能：

|  |  |
| --- | --- |
| 1  2  3 | <?php  echo say();  ?> |

输出内容：

|  |  |
| --- | --- |
| 1  2 | $ php ./test.php  $ hello word |

在扩展中实现一个say方法，调用say方法后，输出 hello word。

**第一步：生成代码**

PHP为我们提供了生成基本代码的工具 ext\_skel。这个工具在PHP源代码的./ext目录下。

|  |  |
| --- | --- |
| 1  2 | $ cd php\_src/ext/  $ ./ext\_skel --extname=say |

extname参数的值就是扩展名称。执行ext\_skel命令后，这样在当前目录下会生成一个与扩展名一样的目录。

**第二步，修改config.m4配置文件**

config.m4的作用就是配合phpize工具生成configure文件。configure文件是用于环境检测的。检测扩展编译运行所需的环境是否满足。现在我们开始修改config.m4文件。

|  |  |
| --- | --- |
| 1  2 | $ cd ./say  $ vim ./config.m4 |

打开，config.m4文件后，你会发现这样一段文字。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | dnl If your extension references something external, use with:    dnl PHP\_ARG\_WITH(say, for say support,  dnl Make sure that the comment is aligned:  dnl [  --with-say             Include say support])    dnl Otherwise use enable:    dnl PHP\_ARG\_ENABLE(say, whether to enable say support,  dnl Make sure that the comment is aligned:  dnl [  --enable-say           Enable say support]) |

其中，dnl 是注释符号。上面的代码说，如果你所编写的扩展如果依赖其它的扩展或者lib库，需要去掉PHP\_ARG\_WITH相关代码的注释。否则，去掉 PHP\_ARG\_ENABLE 相关代码段的注释。我们编写的扩展不需要依赖其他的扩展和lib库。因此，我们去掉PHP\_ARG\_ENABLE前面的注释。去掉注释后的代码如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | dnl If your extension references something external, use with:     dnl PHP\_ARG\_WITH(say, for say support,   dnl Make sure that the comment is aligned:   dnl [  --with-say             Include say support])     dnl Otherwise use enable:     PHP\_ARG\_ENABLE(say, whether to enable say support,   Make sure that the comment is aligned:   [  --enable-say           Enable say support]) |

**第三步，代码实现**

修改say.c文件。实现say方法。  
找到PHP\_FUNCTION(confirm\_say\_compiled)，在其上面增加如下代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | PHP\_FUNCTION(say)  {          zend\_string \*strg;          strg = strpprintf(0, "hello word");          RETURN\_STR(strg);  } |

找到 PHP\_FE(confirm\_say\_compiled, 在上面增加如下代码：

|  |  |
| --- | --- |
| 1 | PHP\_FE(say, NULL) |

修改后的代码如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | const zend\_function\_entry say\_functions[] = {       PHP\_FE(say, NULL)       /\* For testing, remove later. \*/       PHP\_FE(confirm\_say\_compiled,    NULL)       /\* For testing, remove later. \*/       PHP\_FE\_END  /\* Must be the last line in say\_functions[] \*/   };   /\* }}} \*/ |

**第四步，编译安装**

编译扩展的步骤如下：

|  |  |
| --- | --- |
| 1  2  3 | $ phpize  $ ./configure  $ make && make install |

修改php.ini文件，增加如下代码：

|  |  |
| --- | --- |
| 1  2 | [say]  extension = say.so |

然后执行，php -m 命令。在输出的内容中，你会看到say字样。

**第五步，调用测试**

自己写一个脚本，调用say方法。看输出的内容是否符合预期。