# 目录

[目录 1](#_Toc494290956)

[1. 说明 4](#_Toc494290957)

[2. 开发环境安装（Windows） 4](#_Toc494290958)

[2.1. Visual Studio 2015运行库（VC14） 4](#_Toc494290959)

[2.1.1. 下载地址 4](#_Toc494290960)

[2.1.2. 安装 4](#_Toc494290961)

[2.2. Apache 5](#_Toc494290962)

[2.2.1. 下载地址 5](#_Toc494290963)

[2.2.2. 安装 5](#_Toc494290964)

[2.2.2.1. 解压缩 5](#_Toc494290965)

[2.2.2.2. 增加windows系统服务 5](#_Toc494290966)

[2.2.2.3. 卸载windows系统服务 5](#_Toc494290967)

[2.2.3. 配置 6](#_Toc494290968)

[2.2.3.1. 修改配置文件 6](#_Toc494290969)

[2.2.3.2. 配置虚拟主机路径 6](#_Toc494290970)

[2.3. php 7](#_Toc494290971)

[2.3.1. 下载地址 7](#_Toc494290972)

[2.3.2. 安装 7](#_Toc494290973)

[2.3.2.1. 解压缩 7](#_Toc494290974)

[2.3.2.2. 修改配置文件 7](#_Toc494290975)

[2.3.2.3. 为Apache服务添加php支持 7](#_Toc494290976)

[2.3.2.4. 修改系统环境变量 8](#_Toc494290977)

[2.3.3. 配置 8](#_Toc494290978)

[2.3.3.1. 扩展模块 8](#_Toc494290979)

[2.3.3.2. 增加MySQL支持（选配） 8](#_Toc494290980)

[2.3.3.3. 增加SQLServer支持（选配） 9](#_Toc494290981)

[2.3.3.4. 增加Oracle支持（选配） 9](#_Toc494290982)

[2.3.3.5. 增加PostgreSQL支持（选配） 9](#_Toc494290983)

[2.3.3.6. 其他配置 9](#_Toc494290984)

[2.3.4. 注意 9](#_Toc494290985)

[2.3.4.1. upload\_tmp\_dir 9](#_Toc494290986)

[2.4. PhpStorm开发工具安装 10](#_Toc494290987)

[2.4.1. 下载地址 10](#_Toc494290988)

[2.4.2. 安装PhpStorm 10](#_Toc494290989)

[2.4.3. 破解PhpStorm 10](#_Toc494290990)

[2.4.4. 配置PHP环境 10](#_Toc494290991)

[2.4.5. 配置PhpStorm调试环境 11](#_Toc494290992)

[3. 生产环境安装（Linux） 13](#_Toc494290993)

[3.1. Apache（编译安装） 13](#_Toc494290994)

[3.1.1. 下载地址 13](#_Toc494290995)

[3.1.2. 安装 13](#_Toc494290996)

[3.1.3. 开机启动 14](#_Toc494290997)

[3.1.4. 配置 15](#_Toc494290998)

[3.2. PHP（编译安装） 16](#_Toc494290999)

[3.2.1. 下载地址 16](#_Toc494291000)

[3.2.2. 安装 16](#_Toc494291001)

[3.2.3. 安装扩展 16](#_Toc494291002)

[3.2.3.1. curl 16](#_Toc494291003)

[3.2.3.2. gd 17](#_Toc494291004)

[3.2.3.3. openssl 17](#_Toc494291005)

[3.2.3.4. mbstring 17](#_Toc494291006)

[3.2.3.5. mysqli 17](#_Toc494291007)

[3.2.3.6. mcrypt 17](#_Toc494291008)

[3.2.3.7. pdo\_mysql 18](#_Toc494291009)

[3.2.3.8. pdo\_pgsql 18](#_Toc494291010)

[3.2.4. 配置 18](#_Toc494291011)

[3.3. 部署运行 18](#_Toc494291012)

[4. 附录 20](#_Toc494291013)

[4.1. php.ini部分配置项说明 20](#_Toc494291014)

[4.1.1. 文件上传配置 20](#_Toc494291015)

[4.1.1.1. 文件上传功能及大小限制 20](#_Toc494291016)

[4.1.1.2. 上传进度配置 20](#_Toc494291017)

[4.1.2. 运行配置 20](#_Toc494291018)

[4.1.3. 上传配置 21](#_Toc494291019)

[4.1.3.1. session.upload\_progress.enabled 21](#_Toc494291020)

[4.1.3.2. session.upload\_progress.cleanup 21](#_Toc494291021)

[4.1.3.3. session.upload\_progress.prefix 21](#_Toc494291022)

[4.1.3.4. session.upload\_progress.name 21](#_Toc494291023)

[4.1.3.5. session.upload\_progress.freq 21](#_Toc494291024)

[4.1.3.6. session.upload\_progress.min\_freq 21](#_Toc494291025)

[4.1.4. session配置 22](#_Toc494291026)

[4.1.4.1. session.save\_path 22](#_Toc494291027)

[4.1.4.2. session.use\_strict\_mode 22](#_Toc494291028)

[4.1.4.3. session.use\_cookies 22](#_Toc494291029)

[4.1.4.4. session.cookie\_secure 22](#_Toc494291030)

[4.1.4.5. session.use\_only\_cookies 22](#_Toc494291031)

[4.1.4.6. session.name 22](#_Toc494291032)

[4.1.4.7. session.auto\_start 22](#_Toc494291033)

[4.1.4.8. session.cookie\_lifetime 23](#_Toc494291034)

[4.1.4.9. session.cookie\_path 23](#_Toc494291035)

[4.1.4.10. session.cookie\_domain 23](#_Toc494291036)

[4.1.4.11. session.cookie\_httponly 23](#_Toc494291037)

[4.1.4.12. session.serialize\_handler 23](#_Toc494291038)

[4.1.4.13. session.gc\_probability 23](#_Toc494291039)

[4.1.4.14. session.gc\_divisor 24](#_Toc494291040)

[4.1.4.15. session.gc\_maxlifetime 24](#_Toc494291041)

[4.1.4.16. session.referer\_check 24](#_Toc494291042)

[4.1.4.17. session.entropy\_file 24](#_Toc494291043)

[4.1.4.18. session.entropy\_length 24](#_Toc494291044)

[4.1.4.19. session.cache\_limiter 24](#_Toc494291045)

[4.1.4.20. session.cache\_expire 25](#_Toc494291046)

[4.1.4.21. session.use\_trans\_sid 25](#_Toc494291047)

[4.1.4.22. session.hash\_function 25](#_Toc494291048)

[4.1.4.23. session.hash\_bits\_per\_character 25](#_Toc494291049)

[4.1.4.24. url\_rewriter.tags 25](#_Toc494291050)

[4.2. RSA加密解密配置 25](#_Toc494291051)

[4.2.1. 生成公私钥对 25](#_Toc494291052)

[4.3. Apache代理tomcat 26](#_Toc494291053)

[4.3.1. http\_proxy方式连接tomcat 26](#_Toc494291054)

[连接配置 26](#_Toc494291055)

[4.3.2. ajp\_proxy方式连接tomcat 26](#_Toc494291056)

[连接配置 26](#_Toc494291057)

# 说明

1. **所有安装包均使用线程安全版（Thread Safe）**
2. **写文时间所涉及的所有安装包均是当时最新版本，并且本文中的版本为最低要求，如有更新后的版本也可参考本文进行安装**
3. **Apache httpd 2.4.20**

**PHP 7.0.8**

# 开发环境安装（Windows）

## Visual Studio 2015运行库（VC14）

注：已安装Visual Studio 2015可忽略此步骤

### 下载地址

<https://www.microsoft.com/zh-CN/download/details.aspx?id=48145>

### 安装

运行vc\_redist.x64.exe

## Apache

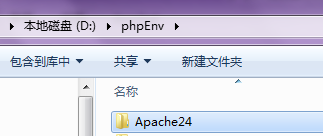
### 下载地址

<http://www.apachelounge.com/download/>

### 安装

#### 解压缩

解压缩httpd-2.4.20-win64-VC14.zip到D:\phpEnv 下



#### 增加windows系统服务

1. cmd 到 D:\phpEnv\Apache24\bin 目录下；
2. 执行命令：httpd.exe -k install -n "Apache24"；

#### 卸载windows系统服务

1. cmd 到 D:\phpEnv\Apache24\bin 目录下；
2. 执行命令：httpd.exe -k uninstall -n "Apache24"；

### 配置

#### 修改配置文件

1. 打开配置文件 D:\phpEnv\Apache24\conf\httpd.conf；
2. 修改apache目录ServerRoot "D:/phpEnv/Apache24"
3. 修改端口号，默认是80端口；

Listen 8888

1. 修改主机名，默认是localhost:80；

ServerName localhost:8888

1. 修改文档主目录

DocumentRoot "D:/"

<Directory "D:/">

注：今后部署的文件所在路径的顶层必须包含此目录

1. 修改入口文件，默认是 index.html；

DirectoryIndex index.html index.php

1. LoadModule vhost\_alias\_module modules/mod\_vhost\_alias.so
2. 注释 LoadModule ssl\_module modules/mod\_ssl.so
3. Include conf/extra/httpd-vhosts.conf
4. LoadModule proxy\_module modules/mod\_proxy.so
5. LoadModule proxy\_http\_module modules/mod\_proxy\_http.so
6. LoadModule proxy\_ajp\_module modules/mod\_proxy\_ajp.so
7. LoadModule rewrite\_module modules/mod\_rewrite.so
8. LoadModule vhost\_alias\_module modules/mod\_vhost\_alias.so
9. 所有AllowOverride None修改为：AllowOverride All

#### 配置虚拟主机路径

1. 启用Include conf/extra/httpd-vhosts.conf
2. 打开配置文件 D:\phpEnv\Apache24\conf\extra\httpd-vhosts.conf；
3. 增加：

<VirtualHost \*:8888>

DocumentRoot "D:/phpEnv/document/cms"

ServerName 127.0.0.1

ErrorLog "logs/cms-error.log"

CustomLog "logs/cms-access.log" common

</VirtualHost>

## php

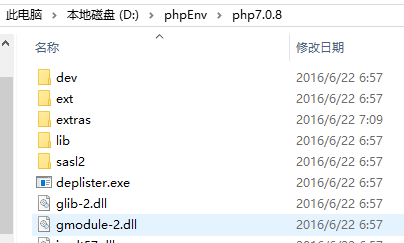
### 下载地址

<http://windows.php.net/download#php-7.0>

### 安装

#### 解压缩

解压缩php-7.0.8-Win32-VC14-x64.zip到D:\phpEnv\php7.0.8下



#### 修改配置文件

1. php目录下：

php.ini-development 适合开发程式使用（测试用）

php.ini-production 拥有较高的安全性设定，则适合上线当产品使用

一般将php.ini-production修改为php.ini，确保测试环境（本地）与正式环境（线上）统一

php.ini-production文件复制一份并改名为php.ini；

1. date.timezone = PRC；
2. extension\_dir = " D:\phpEnv\php7.0.8\ext"

#### 为Apache服务添加php支持

打开Apache的配置文件http.conf在最后加上

# php7 support

LoadModule php7\_module "D:/phpEnv/php7.0.8/php7apache2\_4.dll"

AddType application/x-httpd-php .php .html .htm

# configure the path to php.ini

PHPIniDir "D:/phpEnv/php7.0.8"

#### 修改系统环境变量

1. 新增变量PHP\_HOME=D:\phpEnv\php7.0.8
2. Path中增加 %PHP\_HOME%； %PHP\_HOME%\ext
3. PATHEXT中增加 .PHP；

### 配置

#### 扩展模块

1. 打开配置文件 D:\phpEnv\php7.0.8\php.ini
2. 添加扩展：

extension=php\_curl.dll

extension=php\_gd2.dll

extension=php\_openssl.dll

extension=php\_mbstring.dll

#### 增加MySQL支持（选配）

1. 打开配置文件 D:\phpEnv\php7.0.8\php.ini
2. 添加扩展：

extension=php\_mysqli.dll

extension=php\_pdo\_mysql.dll

1. url：

mysql:host=127.0.0.1:3306;dbname=acp;charset=utf8

#### 增加SQLServer支持（选配）

1. 复制sqlsrv\_非官方\_3.0.2.2%28支持32、64位php%29\x64 下的 php\_pdo\_sqlsrv\_56\_ts.dll 和 php\_sqlsrv\_56\_ts.dll 到 D:\phpEnv\php7.0.8\ext 目录下；
2. php.ini中增加：

extension=php\_sqlsrv\_56\_ts.dll

extension=php\_pdo\_sqlsrv\_56\_ts.dll

1. url：sqlsrv:Server=127.0.0.1,1433;Database=reallink

#### 增加Oracle支持（选配）

增加：

extension=php\_oci8\_12c.dll

extension=php\_pdo\_oci.dll

extension=php\_pdo\_odbc.dll

url：oci:dbname=//127.0.0.1:1521/ORCL;charset=utf8

#### 增加PostgreSQL支持（选配）

增加：

extension=php\_pdo\_pgsql.dll

url：pgsql:host=127.0.0.1;port=5432;dbname=acp

#### 其他配置

修改：

extension\_dir = "D:\phpEnv\php7.0.8\ext"

file\_uploads = On

### 注意

#### upload\_tmp\_dir

请修改此配置项，手动指定一个有读写权限的路径

## PhpStorm开发工具安装

### 下载地址

<http://www.jetbrains.com/phpstorm/download/#section=windows-version>

### 安装PhpStorm

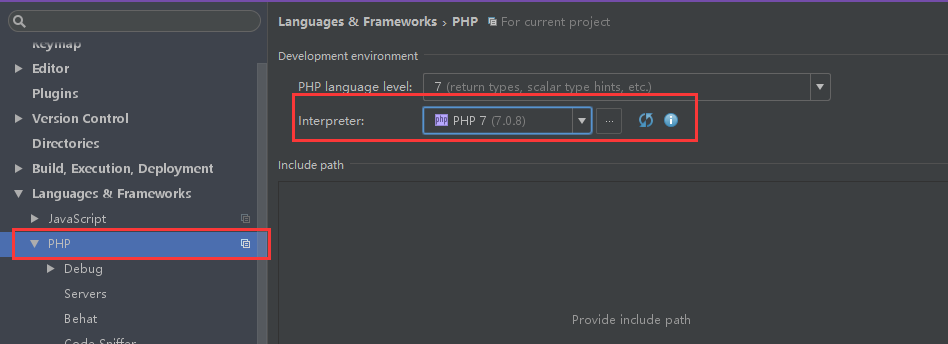
运行PhpStorm-2016.1.exe 进行安装。

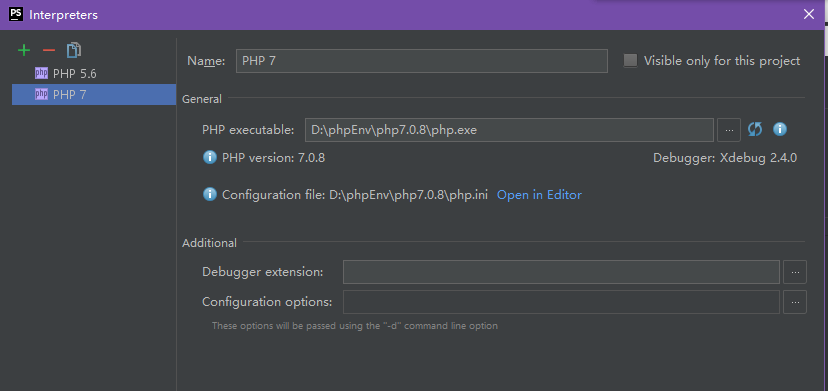
### 破解PhpStorm

1. 在线激活，地址：<http://idea.qinxi1992.cn/>

### 配置PHP环境

1. File→Settings→PHP，配置interpreter





### 配置PhpStorm调试环境

1. 下载

地址：https://xdebug.org/download.php

php\_xdebug-2.3.1-5.6-vc11-x86\_64.dll；

1. 将php\_xdebug-2.3.1-5.6-vc11-x86\_64.dll 复制到php7.0.8\ext；
2. 在php.ini文件中增加：

[Xdebug]

zend\_extension = "d:/phpEnv/php7.0.8/ext/php\_xdebug-2.3.1-5.6-vc11-x86\_64.dll"

;开启自动跟踪

xdebug.auto\_trace = On

;开启异常跟踪

xdebug.show\_exception\_trace = On

;开启远程调试自动启动

xdebug.remote\_autostart = On

;开启远程调试

xdebug.remote\_enable = On

;收集变量

xdebug.collect\_vars = On

;收集返回值

xdebug.collect\_return = On

;收集参数

xdebug.collect\_params = On

xdebug.trace\_output\_dir="D:/phpEnv/php7.0.8/debuginfo"

xdebug.profiler\_enable=On

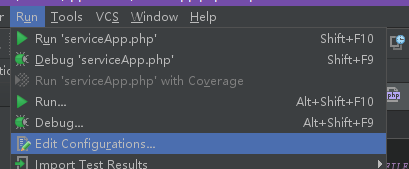
xdebug.profiler\_output\_dir="D:/phpEnv/php7.0.8/phpdebuginfo"

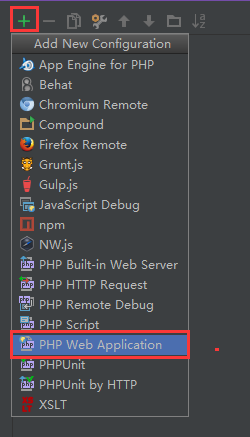
xdebug.remote\_host=127.0.0.1

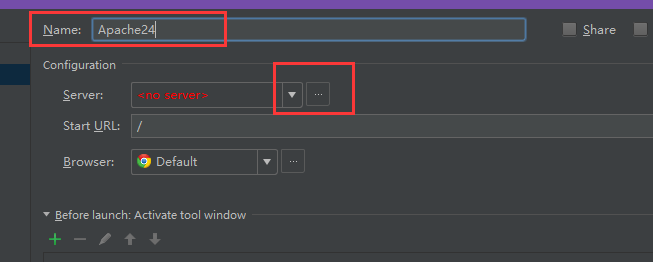
xdebug.remote\_port=29000

xdebug.remote\_handler=dbgp

1. PhpStorm File→Settings→Languages & Frameworks→PHP→Debug，Debug port：29000
2. 设置debug







# 生产环境安装（Linux）

本文默认已安装MySQL5.5以上版本

## Apache（编译安装）

### 下载地址

<http://httpd.apache.org/download.cgi#apache24>

### 安装

1. 安装 apr

下载地址：<http://apr.apache.org/download.cgi>

cd /usr/local

tar jxvf apr-1.5.2.tar.bz2

cd apr-1.5.2

./configure --prefix=/usr/local/apr

make

make install

1. 安装 apr-util

下载地址：<http://apr.apache.org/download.cgi>

cd /usr/local

tar jxvf apr-util-1.5.4.tar.bz2

cd apr-util-1.5.4

./configure --prefix=/usr/local/apr --with-apr=/usr/local/apr

make

make install

1. 安装 pcre

下载地址：<https://sourceforge.net/projects/pcre/>

cd /usr/local

tar -zxvf pcre-8.38.tar.gz

cd pcre-8.38

./configure --prefix=/usr/local/pcre

make

make install

注意：如果在安装 pcre 时，遇到问题：configure: error: You need a C++ compiler for C++ support.

解决方法：yum install -y gcc gcc-c++

1. 安装epel

yum install epel-release

1. 安装ssl模块

yum install -y openssl

yum install -y openssl-devel

yum install -y mod\_ssl

1. 安装certbot

cd ~

wget https://dl.eff.org/certbot-auto --no-check-certificate

chmod +x ./certbot-auto

如果python安装太慢，可以执行以下指令后，再安装

mkdir ~/.pip

cat > ~/.pip/pip.conf <<EOF

[global]

index-url = https://pypi.doubanio.com/simple/

[install]

trusted-host=pypi.doubanio.com

EOF

1. 安装httpd

上传httpd-2.4.20.tar.gz至/usr/local

tar -zxvf httpd-2.4.20.tar.gz

cd httpd-2.4.20

./configure --prefix=/usr/local/apache2420 --with-apr=/usr/local/apr --with-apr-util=/usr/local/apr --with-pcre=/usr/local/pcre --with-mpm=worker --enable-mpms-shared=all --enable-rewrite --enable-ssl=shared --with-ssl=/usr/lib64/openssl

**或直接指定 --with-mpm=worker**

make

make install

### 开机启动

1、cp /usr/local/apache2420/bin/apachectl /etc/rc.d/init.d/apache2420

2、编辑apache2420文件，#!/bin/sh下面增加：

#chkconfig: 2345 10 90

#description: Activates/Deactivates Apache Web Server

3、cd /etc/rc.d/init.d

chkconfig --add apache2420

chkconfig --levels 345 apache2420 on

4、启动、停止、重启

service apache2420 start|stop|restart

### 配置

参考[2.2.2](#_安装)和[2.2.3](#_扩展配置)

httpd.conf

1. 打开配置文件/usr/local/apache2420/conf/httpd.conf
2. 修改端口号，默认是80端口；

Listen 8888

1. 修改主机名

ServerName localhost:8888

1. 修改文档主目录

DocumentRoot "/usr"

<Directory "/usr">

注：今后部署的文件所在路径的顶层必须包含此目录

1. 修改入口文件，默认是 index.html；

DirectoryIndex index.html index.php

1. LoadModule mpm\_worker\_module modules/mod\_mpm\_worker.so
2. LoadModule vhost\_alias\_module modules/mod\_vhost\_alias.so
3. Include conf/extra/httpd-vhosts.conf
4. LoadModule proxy\_module modules/mod\_proxy.so
5. LoadModule proxy\_http\_module modules/mod\_proxy\_http.so
6. LoadModule proxy\_ajp\_module modules/mod\_proxy\_ajp.so
7. LoadModule rewrite\_module modules/mod\_rewrite.so
8. 所有AllowOverride None修改为：AllowOverride All
9. 增加：

AddType application/x-httpd-php .php .html .htm

PHPIniDir "/usr/local/php-7.0.8"

httpd-vhosts.conf

1. 注释原有配置
2. 增加配置：

<VirtualHost \*:8888>

DocumentRoot "/usr/phpweb/OLink-Accounts"

ServerName localhost

ErrorLog "logs/accounts-error.log"

CustomLog "logs/accounts-access.log" common

</VirtualHost>

## PHP（编译安装）

### 下载地址

[http://php.net/get/php-7.0.8.tar.bz2/from/a/mirror](http://php.net/get/php-7.0.7.tar.bz2/from/a/mirror)

### 安装

准备工作：

yum -y install php-mcrypt libmcrypt libmcrypt-devel libxml2-devel openssl-devel libcurl-devel libjpeg.x86\_64 libpng.x86\_64 freetype.x86\_64 libjpeg-devel.x86\_64 libpng-devel.x86\_64 freetype-devel.x86\_64 libjpeg-turbo-devel mhash re2c curl curl-devel m4 autoconf

1. 上传php-7.0.8.tar.bz2至/usr/local
2. cd /usr/local
3. tar jxvf php-7.0.8.tar.bz2
4. cd php-7.0.8
5. 依次执行

./configure --prefix=/usr/local/php-7.0.8 --with-apxs2=/usr/local/apache2420/bin/apxs --with-iconv-dir=/usr --with-freetype-dir=/usr/include/freetype2/freetype --with-jpeg-dir --with-png-dir --with-zlib --enable-xml --disable-rpath --enable-bcmath --enable-shmop --enable-sysvsem --enable-inline-optimization --with-curl --enable-mbregex --enable-mbstring --with-gd --enable-gd-native-ttf --with-openssl --with-mhash --enable-sockets --with-xmlrpc --enable-zip --enable-soap --enable-exif --enable-calendar --with-mysqli --with-pdo-mysql --with-mcrypt=/usr/include --enable-maintainer-zts --with-libmcrypt

make

make install

### 安装扩展

#### curl

1、cd /usr/local/php-7.0.8/ext/curl

2、/usr/local/php-7.0.8/bin/phpize

3、./configure --with-php-config=/usr/local/php-7.0.8/bin/php-config

4、make

5、make install

#### gd

1、cd /usr/local/php-7.0.8/ext/gd

2、/usr/local/php-7.0.8/bin/phpize

3、./configure --with-php-config=/usr/local/php-7.0.8/bin/php-config

4、make

5、make install

#### openssl

1、cd /usr/local/php-7.0.8/ext/openssl

2、cp config0.m4 config.m4

3、/usr/local/php-7.0.8/bin/phpize

4、./configure --with-php-config=/usr/local/php-7.0.8/bin/php-config

5、make

6、make install

#### mbstring

1、cd /usr/local/php-7.0.8/ext/mbstring

2、/usr/local/php-7.0.8/bin/phpize

3、./configure --with-php-config=/usr/local/php-7.0.8/bin/php-config

4、make

5、make install

#### mysqli

1、cd /usr/local/php-7.0.8/ext/mysqli

2、/usr/local/php-7.0.8/bin/phpize

3、./configure --with-php-config=/usr/local/php-7.0.8/bin/php-config

4、make

5、make install

#### mcrypt

1、cd /usr/local/php-7.0.8/ext/mcrypt

2、/usr/local/php-7.0.8/bin/phpize

3、./configure --with-php-config=/usr/local/php-7.0.8/bin/php-config

4、make

5、make install

#### pdo\_mysql

1、cd /usr/local/php-7.0.8/ext/pdo\_mysql

2、/usr/local/php-7.0.8/bin/phpize

3、./configure --with-php-config=/usr/local/php-7.0.8/bin/php-config

4、make

5、make install

#### pdo\_pgsql

1、cd /usr/local/php-7.0.8/ext/pdo\_pgsql

2、/usr/local/php-7.0.8/bin/phpize

3、./configure --with-php-config=/usr/local/php-7.0.8/bin/php-config

4、make

5、make install

### 配置

参考[2.3.3](#_配置)

1. 确保apache下的httpd.conf中加载了php7模块

LoadModule php7\_module modules/libphp7.so

1. 打开 /usr/local/php-7.0.8 下的php.ini-production文件（复制到本地），进行配置
2. 增加：

extension=curl.so

extension=gd.so

extension=mbstring.so

extension=mysqli.so

extension=openssl.so

extension=pdo\_mysql.so

extension=mcrypt.so

1. 修改：

extension\_dir = "/usr/local/php-7.0.8/lib/php/extensions/no-debug-zts-20151012"

file\_uploads = On

upload\_tmp\_dir = /tmp

1. 重命名php.ini-production为php.ini，上传至/usr/local/php-7.0.8/lib

## 部署运行

1. 将PHP工程OLink-Accounts上传至/usr/phpweb
2. chmod o+rwx /usr/phpweb/OLink-Accounts
3. 修改/usr/phpweb/OLink-Accounts/config/DataBaseConfig.php数据源配置
4. 执行 service apache2420 start启动
5. 打开测试页面，查看安装信息，<http://host:port/portal/test>

# 附录

## php.ini部分配置项说明

### 文件上传配置

#### 文件上传功能及大小限制

1. 打开配置文件php.ini；
2. 开启上传文件功能

file\_uploads = On；

1. 设置上传文件的最大体积

upload\_max\_filesize = 200M；

1. 设置post请求所接收的最大值

post\_max\_size = 200M；

#### 上传进度配置

1. 打开配置文件php.ini；
2. session.upload\_progress.enabled = On
3. session.upload\_progress.cleanup = On
4. session.upload\_progress.prefix = "upload\_progress\_"
5. session.upload\_progress.name = "PHP\_SESSION\_UPLOAD\_PROGRESS"
6. session.upload\_progress.freq = "1%"
7. session.upload\_progress.min\_freq = "1"

### 运行配置

1. 打开配置文件php.ini；
2. 每个PHP页面运行的最大时间值(秒)，默认30秒

max\_execution\_time = 43200

1. 每个PHP页面接收数据所需的最大时间，默认60秒

max\_input\_time = 43200

1. 每个PHP页面所吃掉的最大内存，默认128M

memory\_limit = -1

### 上传配置

#### session.upload\_progress.enabled

1. 说明：允许上传进度跟踪，填充$\_SESSION变量。默认值为启用。
2. 配置：第1586行，默认即可；

#### session.upload\_progress.cleanup

1. 说明：清理进度信息，只要所有的POST数据已读（即上传完成）。默认值为启用。
2. 配置：第1594行，默认即可；

#### session.upload\_progress.prefix

1. 说明：用于在$\_SESSION上传进度的关键前缀。这个密钥将与$\_ POST价值被连接[ini\_get（“session.upload\_progress.name”），以提供一个独特的索引。默认为“upload\_progress\_”。
2. 配置：第1601行，默认即可；

#### session.upload\_progress.name

1. 说明：在$ \_SESSION存储进度的信息所使用的密钥的名称。又见session.upload\_progress.prefix。如果$\_ POST[ini\_get（“session.upload\_progress.name”）]不通过或可用，上传进展将不会被记录下来。默认为“PHP\_SESSION\_UPLOAD\_PROGRESS”。
2. 配置：第1609行，默认即可；

#### session.upload\_progress.freq

1. 说明：定义被上传进度信息应该被更新。这可以以字节为单位（“接收整个文件的大小每1％后，更新的进度信息”，即）来定义（即“每100个字节后更新进度信息”），或百分比。默认为“1％”。
2. 配置：第1617行，默认即可；

#### session.upload\_progress.min\_freq

1. 说明：日期之间的最小延迟，以秒为单位。默认为“1”（1秒）。
2. 配置：第1624行，默认即可；

### session配置

#### session.save\_path

1. 说明：服务器用于保存session文件的位置；
2. 配置：第1417行session.save\_path = "/tmp"；可自定义位置，也可默认；

#### session.use\_strict\_mode

1. 说明：是否启用严格的会话模式，如果浏览器未初始化会话ID，就不接受未初始化会话ID和再生会话ID，默认是0（禁用）；
2. 配置：第1425行，默认即可；

#### session.use\_cookies

1. 说明：指定是否在客户端用cookie来存放会话ID，默认为1（启用）；
2. 配置：第1429行，默认即可；

#### session.cookie\_secure

1. 说明：指定是否仅通过安全连接发送 cookie，默认为 off；
2. 配置：第1432行，默认即可；

#### session.use\_only\_cookies

1. 说明：指定是否在客户端仅仅使用 cookie 来存放会话 ID，启用此设定可以防止有关通过 URL 传递会话 ID 的攻击，默认为1（启用）；
2. 配置：第1439行，默认即可；

#### session.name

1. 说明：指定会话名以用做 cookie 的名字。只能由字母数字组成，默认为“PHPSESSID”；
2. 配置：第1443行，默认即可；

#### session.auto\_start

1. 说明：指定会话模块是否在请求开始时自动启动一个会话。默认为 0（不启动）；
2. 配置：第1447行，默认即可；

#### session.cookie\_lifetime

1. 说明：以秒数指定了发送到浏览器的 cookie 的生命周期。值为 0 表示“直到关闭浏览器”，默认为 0；
2. 配置：第1451行，默认即可；

#### session.cookie\_path

1. 说明：此 cookie 的有效路径，设置为“/”表示对于本域上所有的路径此 cookie 都可用，默认为“/”；
2. 配置：第1455行，默认即可；

#### session.cookie\_domain

1. 说明：指定了要设定会话 cookie 的域名。默认为无，表示根据 cookie 规范产生 cookie 的主机名；
2. 配置：第1459行，默认即可；

#### session.cookie\_httponly

1. 说明：标志着作为cookie的访问只能通过HTTP协议。这意味着，该cookie也不会通过脚本语言，如JavaScript访问。此设置可以有效地帮助减少通过XSS攻击身份盗窃（尽管它不支持所有的浏览器），默认为不启用；
2. 配置：第1463行，默认即可；

#### session.serialize\_handler

1. 说明：定义用来序列化/解序列化的处理器名字，默使用PHP；
2. 配置：第1467行，默认即可；

#### session.gc\_probability

1. 说明：session.gc\_probability 与 session.gc\_divisor 合起来用来管理 gc（garbage collection 垃圾回收）进程启动的概率。默认为 1；
2. 配置：第1479行，默认即可；

#### session.gc\_divisor

1. 说明：session.gc\_divisor 与 session.gc\_probability 合起来定义了在每个会话初始化时启动 gc（garbage collection 垃圾回收）进程的概率。此概率用 gc\_probability/gc\_divisor 计算得来。例如 1/100 意味着在每个请求中有 1% 的概率启动 gc 进程。session.gc\_divisor 默认为 1000；
2. 配置：第1493行，默认即可；

#### session.gc\_maxlifetime

1. 说明：指定过了多少秒之后数据就会被视为“垃圾”并被清除，默认为1440秒；
2. 配置：第1498行，默认即可；

#### session.referer\_check

1. 说明：包含有用来检查每个 HTTP Referer 的子串。如果客户端发送了 Referer 信息但是在其中并未找到该子串，则嵌入的会话 ID 会被标记为无效。默认为空字符串；
2. 配置：第1512行，默认即可；

#### session.entropy\_file

1. 说明：给出了一个到外部资源（文件）的路径，该资源将在会话 ID 创建进程中被用作附加的熵值资源。例如在许多 Unix 系统下都可以用 /dev/random 或 /dev/urandom。在 Windows 上自 PHP 5.3.3 起加入了此功能。 设置 session.entropy\_length 为非零的值将使 PHP 使用 Windows Random API 作为熵值源；
2. 配置：第1525行，默认即可；

#### session.entropy\_length

1. 说明：指定了从上面的文件中读取的字节数。默认为禁用。
2. 配置：第1516行，默认即可；

#### session.cache\_limiter

1. 说明：指定会话页面所使用的缓冲控制方法（none/nocache/private/private\_no\_expire/public）。默认为 nocache；
2. 配置：第1530行，默认即可；

#### session.cache\_expire

1. 说明：以分钟数指定缓冲的会话页面的存活期，此设定对 nocache 缓冲控制方法无效，默认为 180；
2. 配置：第1534行，默认即可；

#### session.use\_trans\_sid

1. 说明：指定是否启用透明 SID 支持。默认为 0（禁用）；
2. 配置：第1546行，默认即可；

#### session.hash\_function

1. 说明：允许用户指定生成会话 ID 的散列算法。'0' 表示 MD5（128 位），'1' 表示 SHA-1（160 位）。默认是0；
2. 配置：第1556行，默认即可；

#### session.hash\_bits\_per\_character

1. 说明：允许用户定义将二进制散列数据转换为可读的格式时每个字符存放多少个比特。可能值为 '4'（0-9，a-f），'5'（0-9，a-v），以及 '6'（0-9，a-z，A-Z，"-"，","）。默认为5；
2. 配置：第1568行，默认即可；

#### url\_rewriter.tags

1. 说明：指定在使用透明 SID 支持时哪些 HTML 标记会被修改以加入会话 ID。默认为 a=href,area=href,frame=src,input=src,form=fakeentry
2. 配置：第1579行，默认即可；

## RSA加密解密配置

### 生成公私钥对

1、安装openssl运行环境：

2、Windows下请下载并安装Win64OpenSSL-1\_0\_2h.exe

3、Cmd下执行命令，生成私钥文件：

openssl genrsa -out rsa\_private\_key.pem 2048

1. Cmd下执行命令，生成公钥文件：

openssl rsa -in rsa\_private\_key.pem -pubout -out rsa\_public\_key.pem

1. 执行命令，获取公钥值：

openssl asn1parse -out temp.ans -i -inform PEM < rsa\_private\_key.pem



1. 拷贝下公钥及e值前端保留

## Apache代理tomcat

### http\_proxy方式连接tomcat

#### 连接配置

1. 打开配置文件 D:\phpEnv\Apache24\conf\httpd.conf；
2. 加载proxy\_module、proxy\_ajp\_module、proxy\_http\_module三个模块；
3. 打开配置文件 D:\phpEnv\Apache24\conf\extra\httpd-vhosts.conf；
4. 增加

<VirtualHost \*:8888>

ProxyPass / http://127.0.0.1:8080/

ProxyPassReverse / http://127.0.0.1:8080/

ServerName localhost

</VirtualHost>

以localhost访问时代理连接到tomcat的8080端口，ServerName和端口号可根据生产部署调整；

### ajp\_proxy方式连接tomcat

#### 连接配置

1. 打开配置文件 D:\phpEnv\Apache24\conf\httpd.conf；
2. 加载proxy\_module、proxy\_ajp\_module、proxy\_http\_module三个模块；
3. 打开配置文件 D:\phpEnv\Apache24\conf\extra\httpd-vhosts.conf；
4. 增加

<VirtualHost \*:8888>

ProxyPass / ajp://127.0.0.1:8080/

ProxyPassReverse / ajp://127.0.0.1:8080/

ServerName localhost

</VirtualHost>

以localhost访问时代理连接到tomcat的ajp端口，ServerName和端口号可根据生产部署调整；