最详细的phpstorm+xdebug调试详细教程，没有之一

花了一周多时间最详细的整理了PHPSTORM+XDEBUG的调试教程，实现了php程序的远程调试、单步调试、跟踪等高级功能。工欲善其事必先利其器，希望对需要的小伙伴有所帮助。

## 0 Xdebug调试的原理（选看）

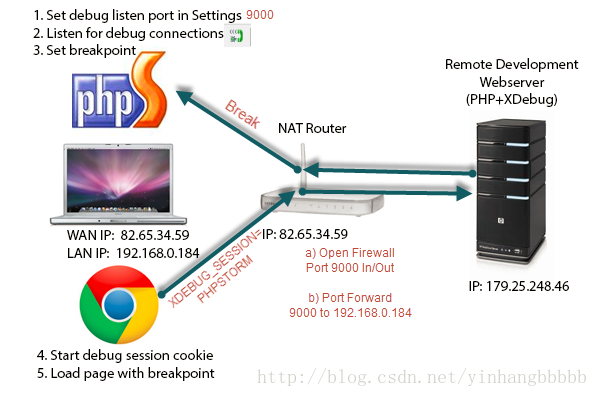


图0-1 单机调试原理示意图

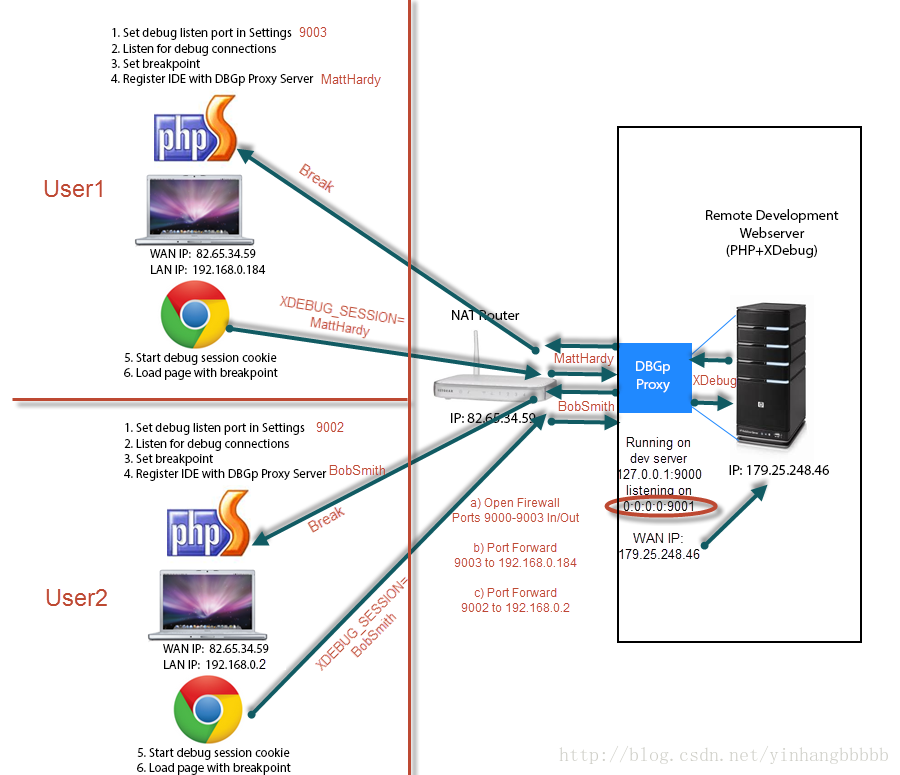


图0-2 多机调试原理示意图

对于PHP开发，初来咋到，开发环境的搭建和理解感觉是最烦人的一件事了。不像JAVA，打开一个Eclipse就可以开搞，Php的Debug都要几个插件来配合工作。这些都是死的，好说。但是对于Xdebug的工作原理我一直是耿耿于怀，后来看了一篇神文Xdebug断点调试的工作原理详解。

## Xdebug的工作原理可以总结为下面几个步骤

1）IDE（比如PhpStorm，下文所述的客户端）中已经集成了一个遵循BGDp协议（一个专门用来调试的协议）的Xdebug插件。当要debug的时候，点击一些IDE的某个按钮，启动这个插件。该插件会启动一个9000的端口监听远程服务器发过来的debug信息。

2）浏览器向Httpd服务器发送一个带有XDEBUG\_SESSION\_START参数的请求，Httpd收到这个请求之后交给后端的PHP进行处理（下面就忽略Httpd，直接把Php叫做Server）。

3）Php看到这个请求是带了XDEBUG\_SESSION\_START 参数，就告诉Xdebug，“嘿，我要debug喔，你准备一下”。这时，Xdebug这时会向来源ip客户端的9000端口（即客户端，也即是IDE）发送一个debug请求，然后客户端的9000端口响应这个请求，那么debug就开始了。

4）Php知道Xdebug已经准备好了，那么就开始开始一行一行的执行代码，但是每执行一行都会让Xdebug过滤一下。

5）Xdebug开始过滤代码，Xdebug在过滤每一行代码的时候，都会暂停代码的执行，然后向客户端的9000端口发送该行代码的执行情况，等待客户端的决策（是一句代码还是下一个断点待）。

6）相应，客户端（IDE）收到Xdebug发送过来的执行情况，就可以把这些信息展示给开发者看了，包括一些变量的值等。同时向Xdebug发送下一步应该什么。

1下载对应版本的xdebug

xdebug官网下载地址：https://xdebug.org/download.php

你需要仔细分析和选择要下载的对应版本，否则无法调试。由于非常容易出错，建议采用下面这种简单方法：

xdebug网站提供一个自动分析你系统对应的xdebug版本的页面，网址是  https://xdebug.org/wizard.php

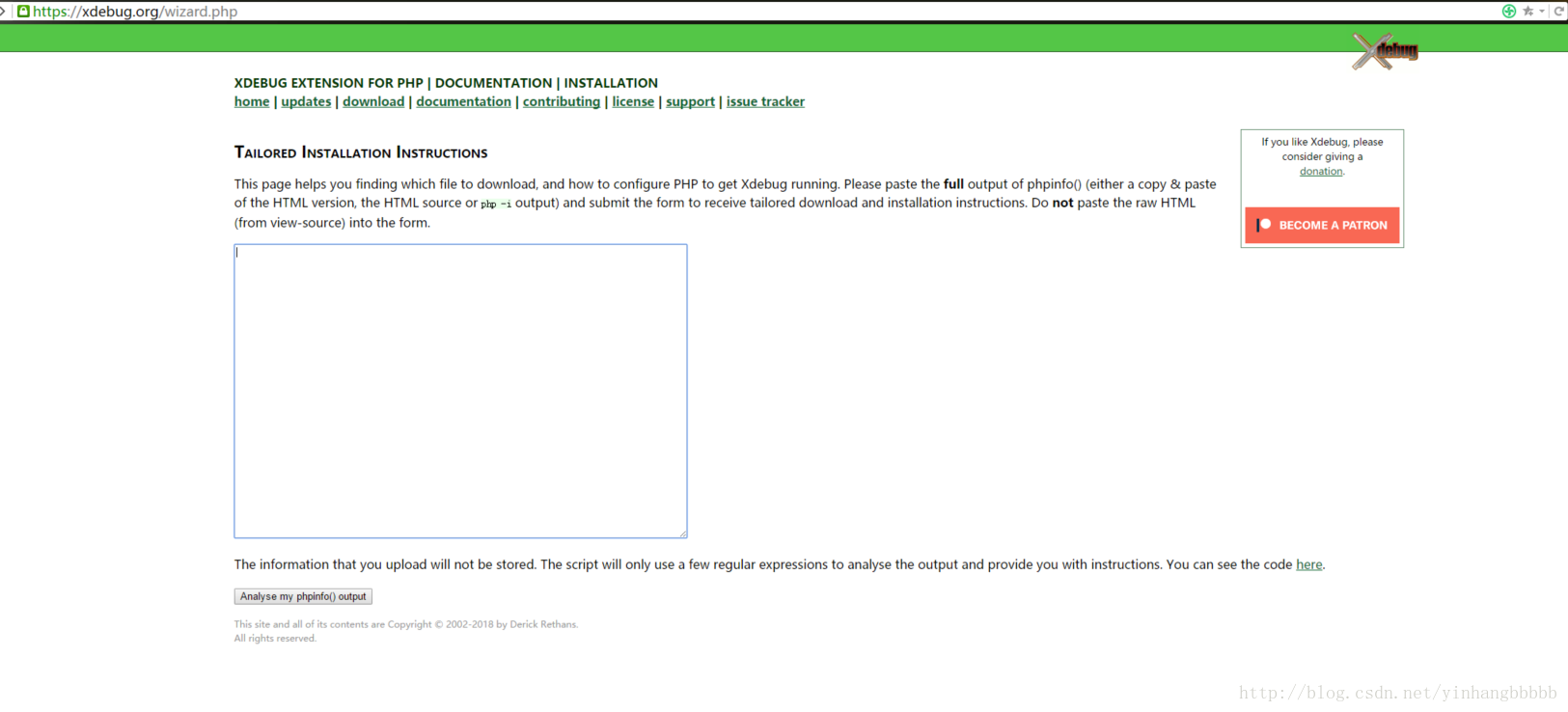


图1-1 xdebug网站提供的一个自动分析系统对应xdebug版本的页面

在页面中需要粘贴进去php版本信息，也就是phpinfo()函数的信息，如下图：

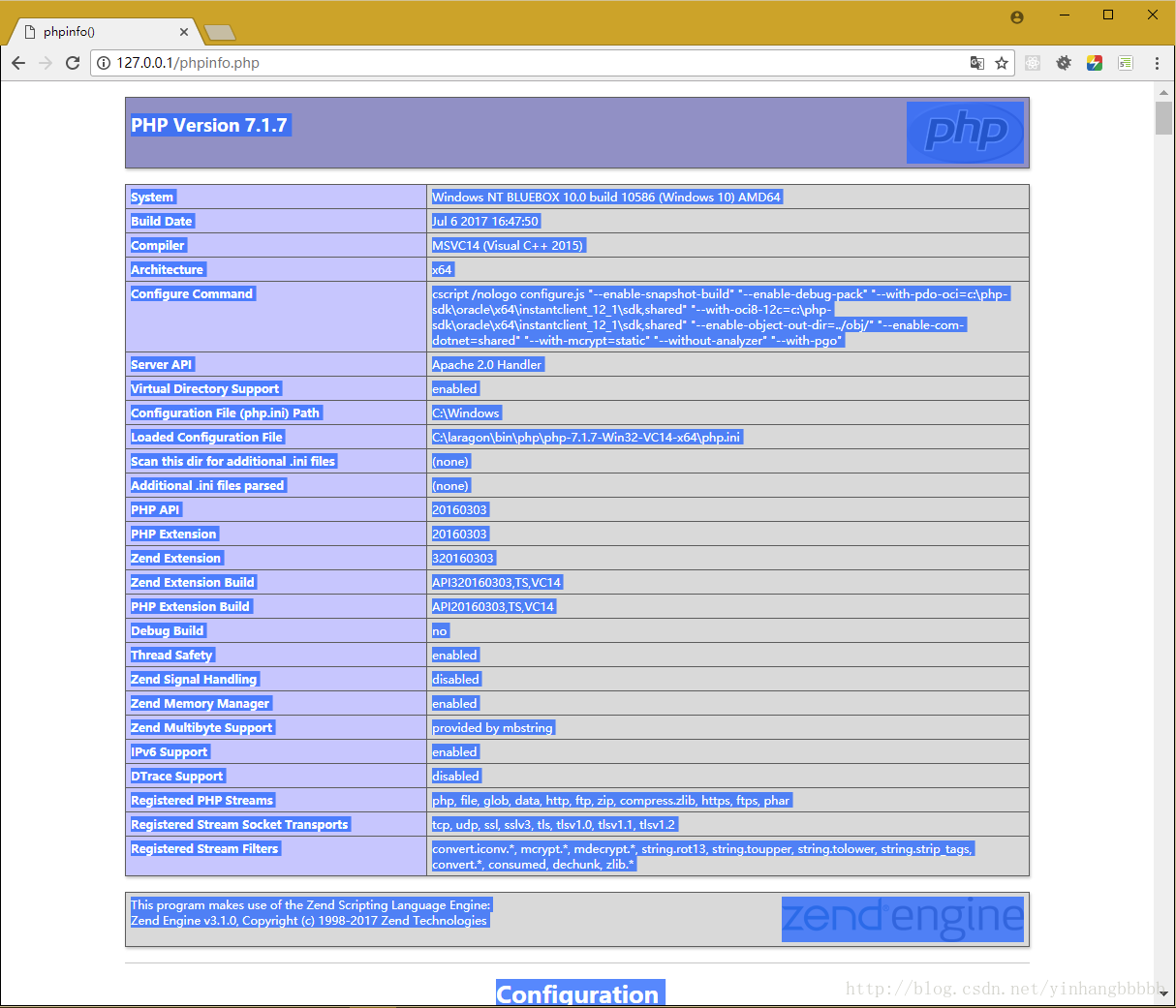


图1-2 phpinfo();函数返回的信息

ctrl+A全选这个页面的信息，然后粘贴到第一个图片的页面中。

不知道这个页面如何出现的同学在你的php服务器程序目录下 创建一个phpinfo.php文件，只需一行代码

<?php  phpinfo(); ?>

浏览器中输入 http://127.0.0.1/phpinfo.php 就出现图2 中的页面了。

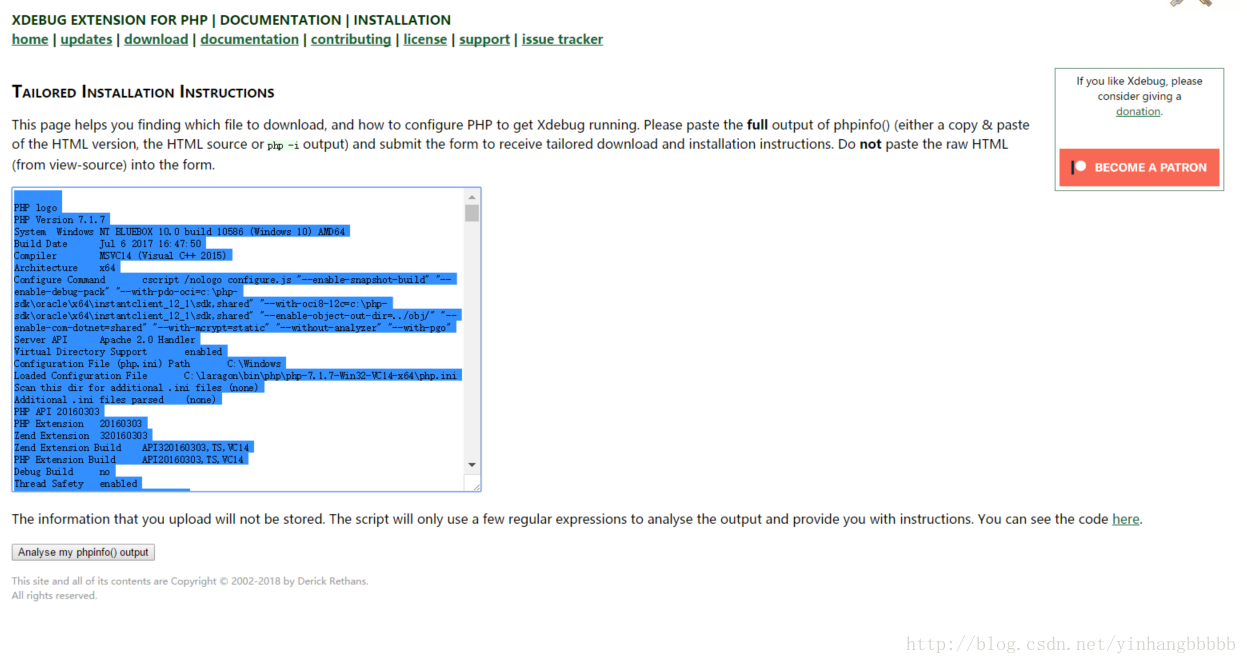


图1-3 向页面中拷贝phpinfo()获取的信息

点击 analyse my phpinfo() output 按钮

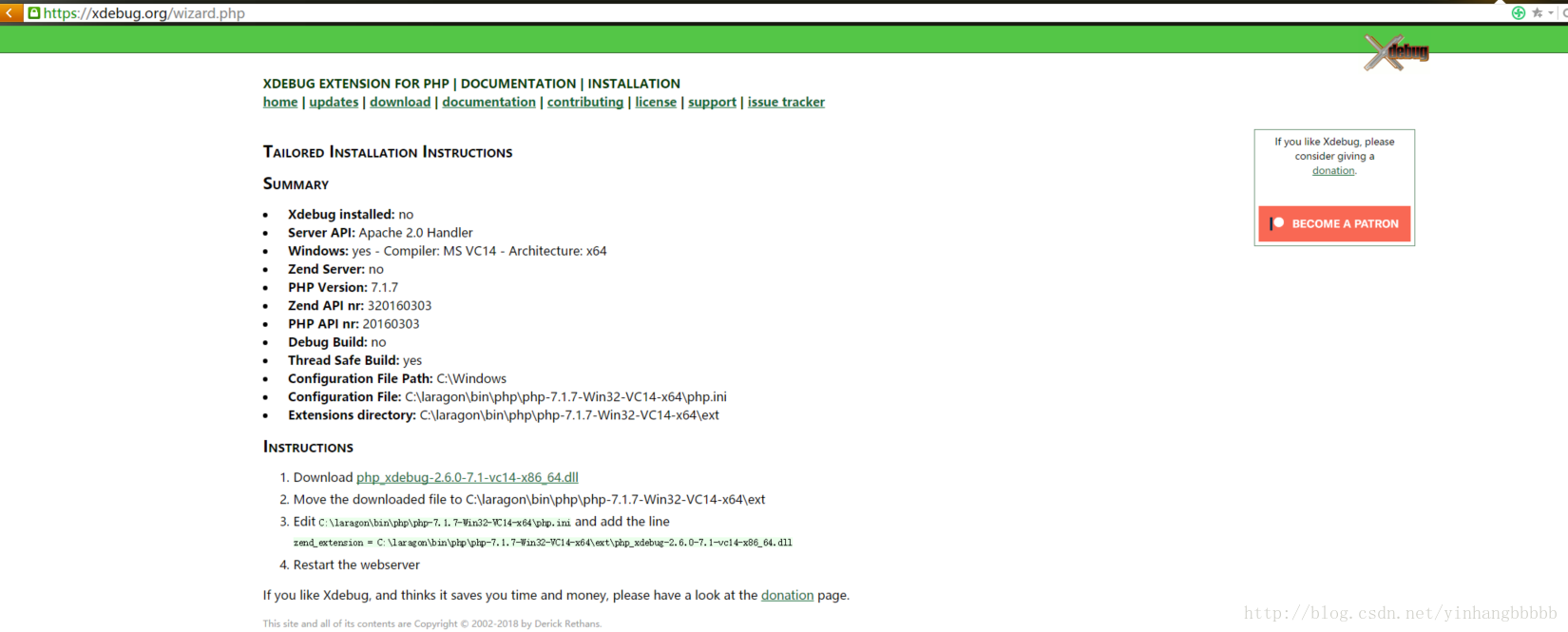


图1-4 找到你需要下载的xdebug版本

将下载的DLL文件拷贝到指定目录，按照页面上的提示即可

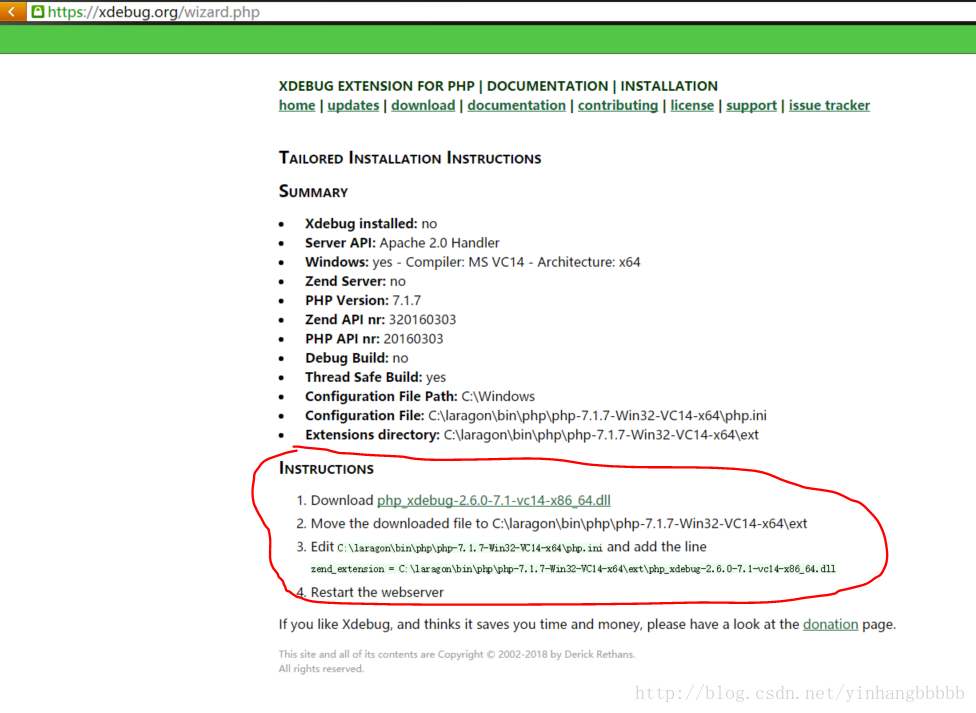
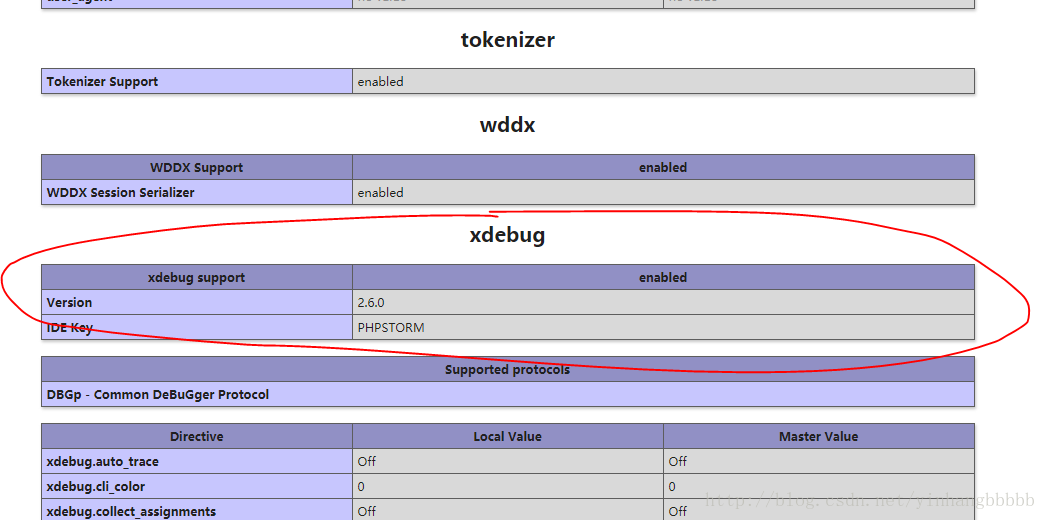


图1-5 下载后拷贝到指定目录，并在对应的php.ini中末尾处添加图中第3小点的内容

到此为止，xdebug的下载和启用就完成了，重新运行 phpinfo.php 文件，在打开的页面中出现图1-6则说明安装正确：



注意问题

1phpinfo.php需要启动服务器后，通过浏览器访问。

2 php.ini文件中增加的行，等号前后有空格，字符串值可以有引号也可以没有引号。

zend\_extension = C:\laragon\bin\php\php-7.1.7-Win32-VC14-x64\ext\php\_xdebug-2.6.0-7.1-vc14-x86\_64.dll

zend\_extension = "C:\laragon\bin\php\php-7.1.7-Win32-VC14-x64\ext\php\_xdebug-2.6.0-7.1-vc14-x86\_64.dll"

（这是我本机地址，这里建议直接拷贝图1-5中网页给出的值）

## 2phpstorm中使用xdebug

2.1修改php配置文件

1 php服务器在本地的情况。

在php.ini文件的末尾处追加如下代码：

;xdebug库文件

zend\_extension = "C:\laragon\bin\php\php-7.1.7-Win32-VC14-x64\ext\php\_xdebug-2.6.0-7.1-vc14-x86\_64.dll"

;开启远程调试

xdebug.remote\_enable = On

;客户机ip

xdebug.remote\_host="127.0.0.1"

;客户机xdebug监听端口和调试协议

xdebug.remote\_port=9001

xdebug.remote\_handler=dbgp

;idekey 区分大小写

xdebug.idekey="PHPSTORM"

xdebug.profiler\_enable = off

xdebug.profiler\_enable\_trigger = off

xdebug.profiler\_output\_name = cachegrind.out.%t.%p

;idekey 区分大小写

xdebug.profiler\_output\_dir = "d:\tmp"

图2-1图中标红的地方因不同主机而不同，请注意

2 PHP服务器不是本机

这种情况我还没有试过，以后再补充，大家有做过的可以给我留言。

## 2.2 phpstorm中的配置

（1）在文件->设置->语言与框架中->PHP->debug 设置端口，端口默认为9000，由于经常会被其他程序占用，这里建议设置为9001

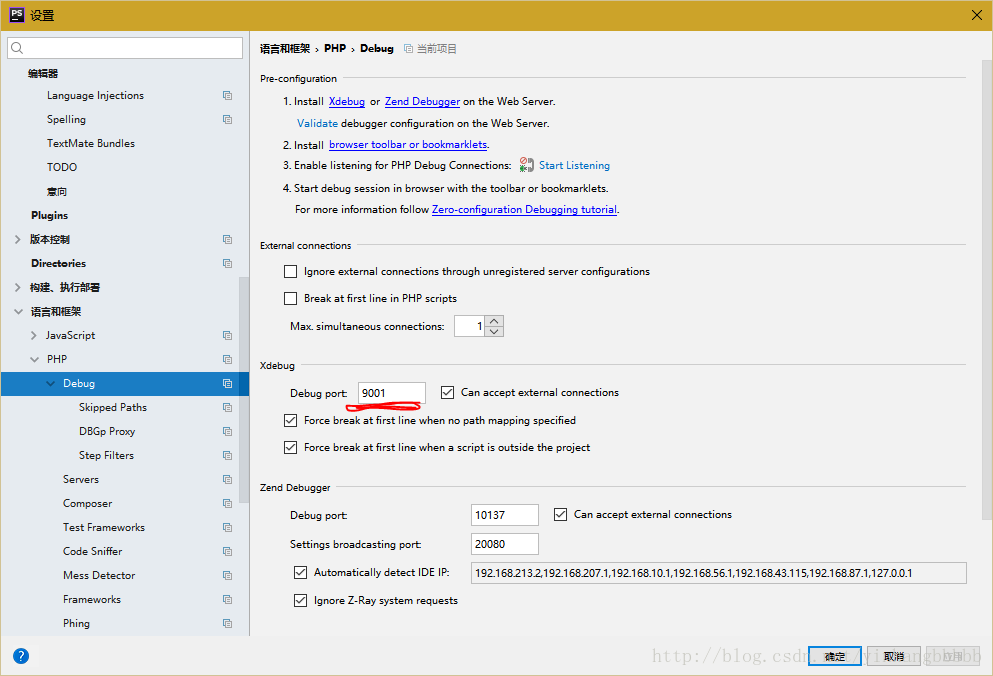


图2-2 修改调试端口

（2）在DBGp Proxy中配置你的idekey，idekey就是你在配置文件中最后一项，host是你的服务器ip或者是已经可以解析的域名，port可以自选，一般默认选80就好了

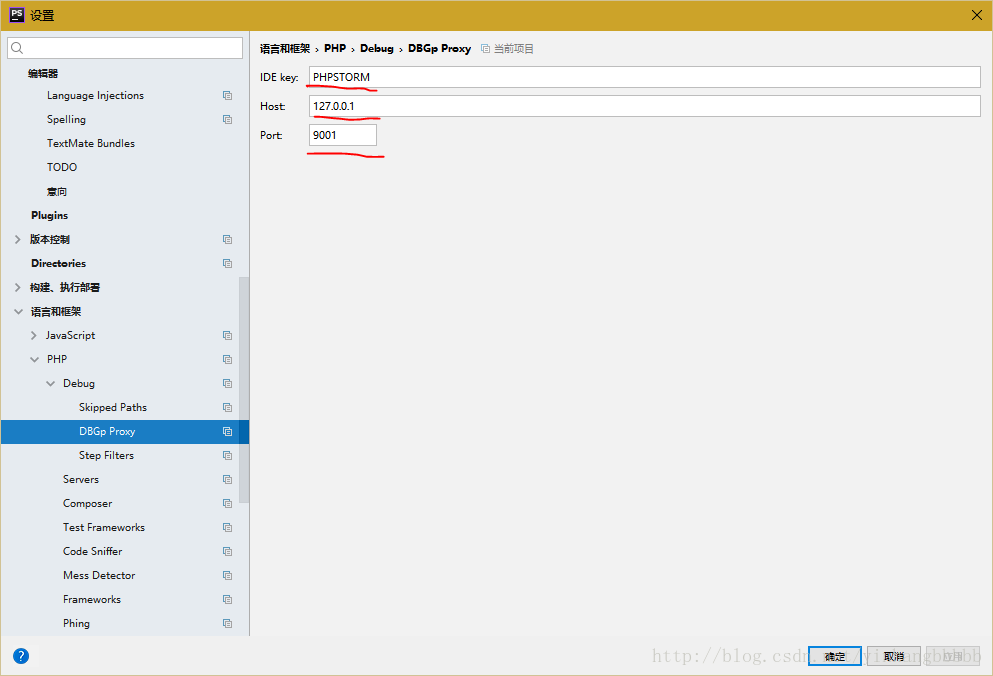


图2-3修改调试协议DBGp的配置

（3）自己手动添加一个，Host填服务器ip，然后port是默认80端口，如果你修改过，你可以更改成你设置的web端口，debug选xdebug

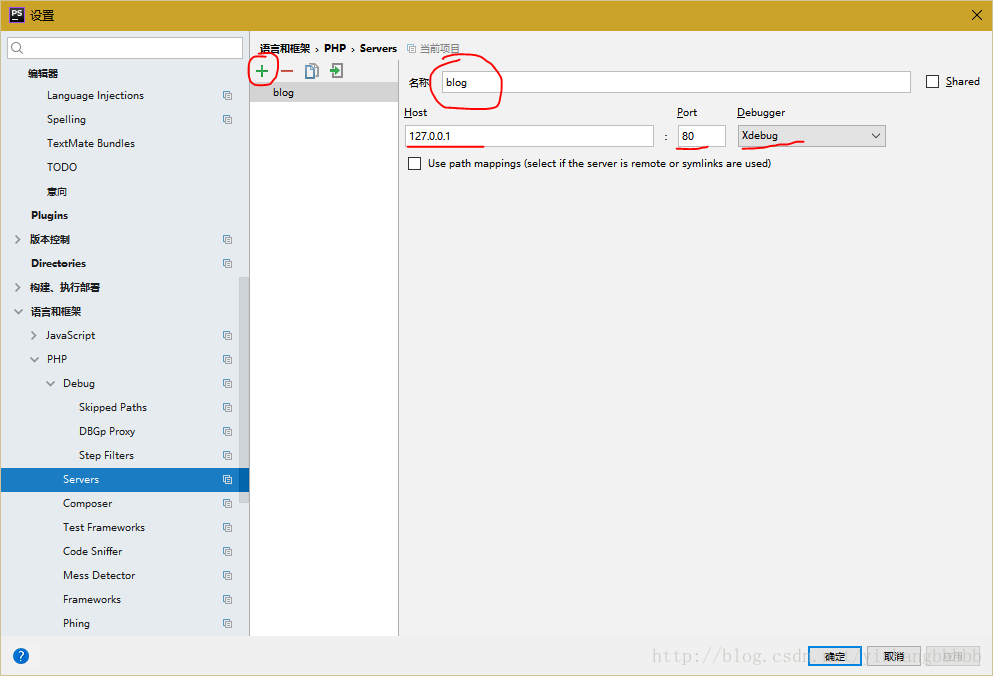


图2-4 创建一个本地调试服务器

（4）测试一下配置是否成功

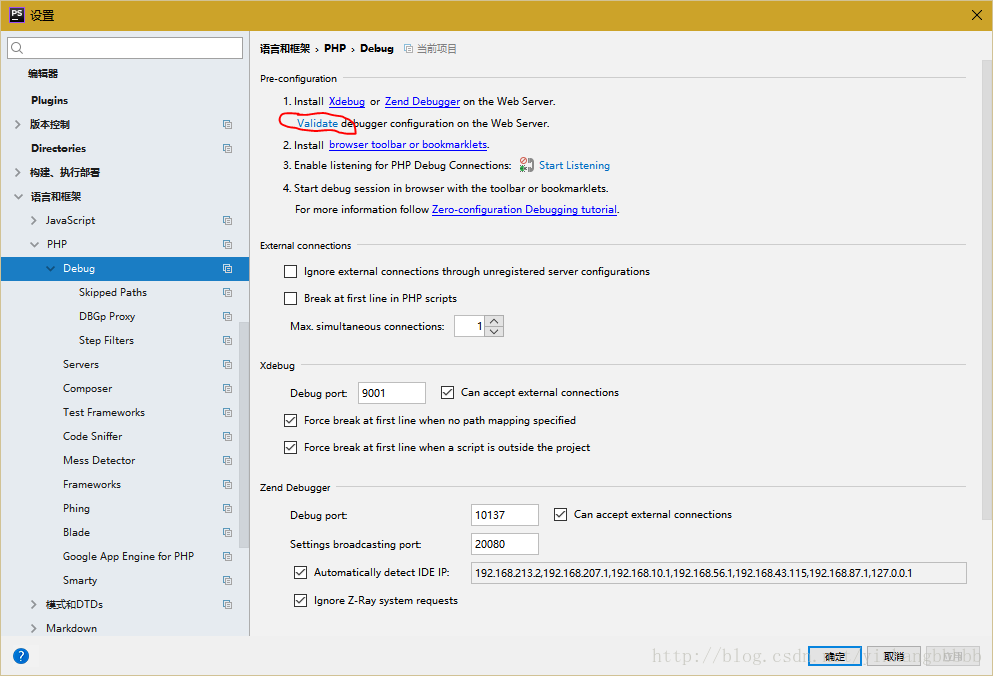


图2-4 创建一个调试服务器，用于与真正服务器同步

（5）测试成功

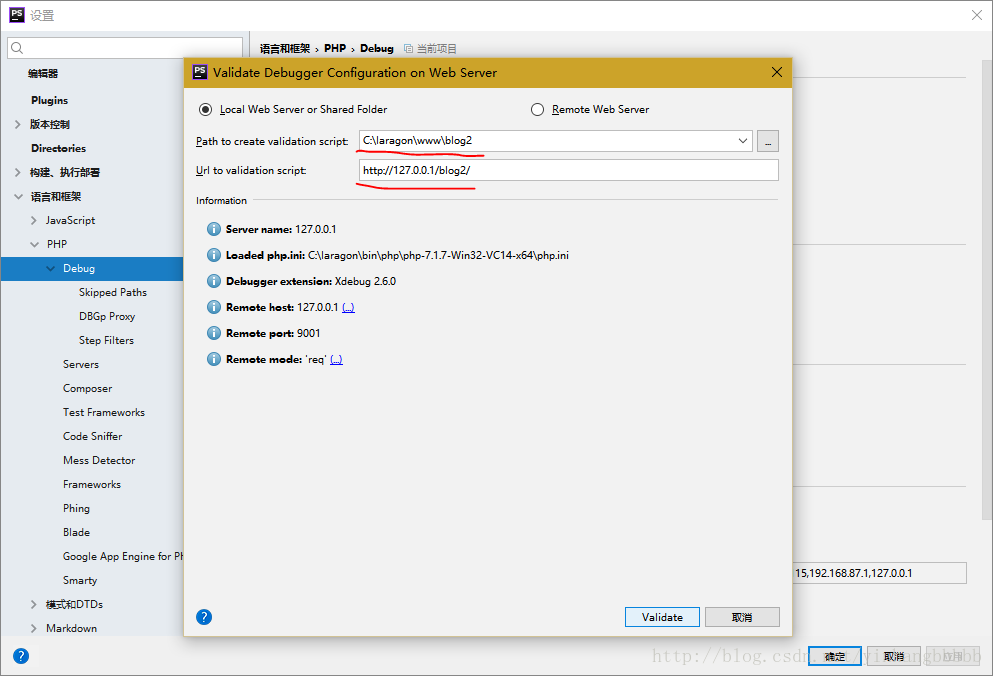


图2-5 配置成功

第一个红线是你的工程在服务器上的地址

第二条红线是你的工程的网络地址（最后的斜杠有木有都可以）

# **3调试使用**

（1）新建一个运行调试配置

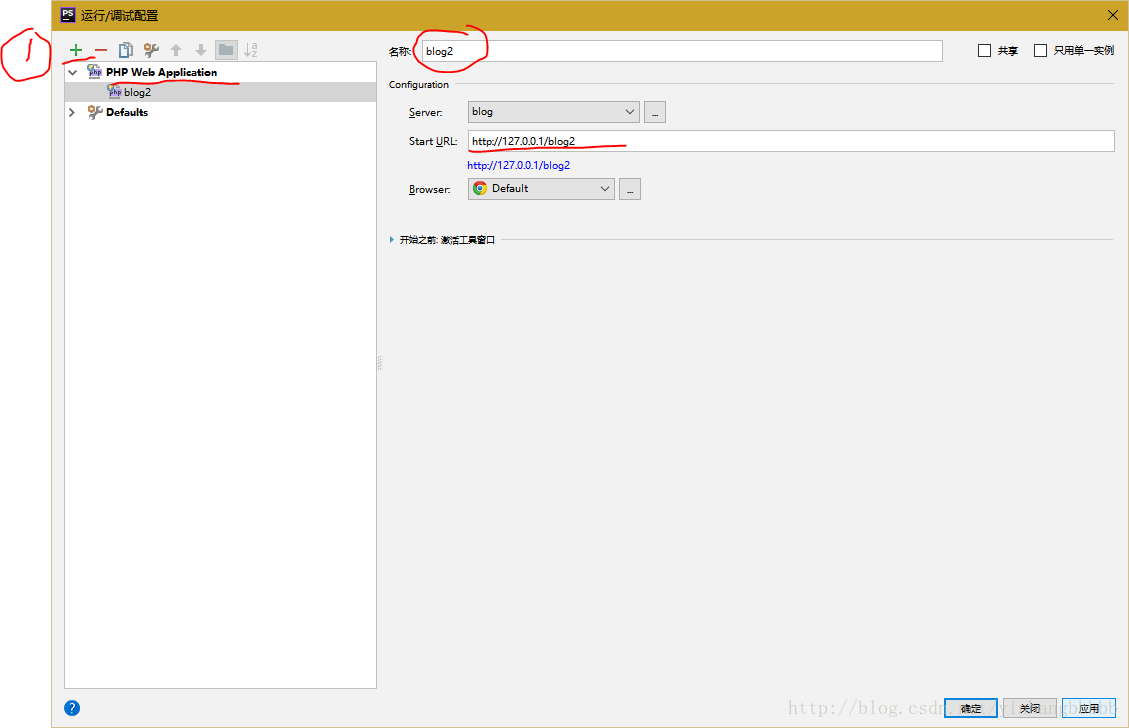


图3-1创建一个调试配置

点击“+”，创建一个 PHP Web Application的配置，右侧输入一个用于记忆的名称，选择服务器

（2）打开IDE的debug监听

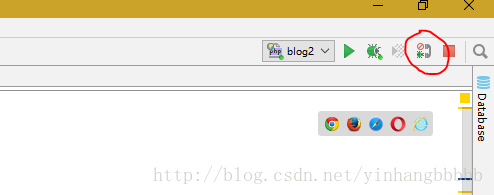


图3-2 图中为关闭状态，单击后为打开状态

（3）开始调试

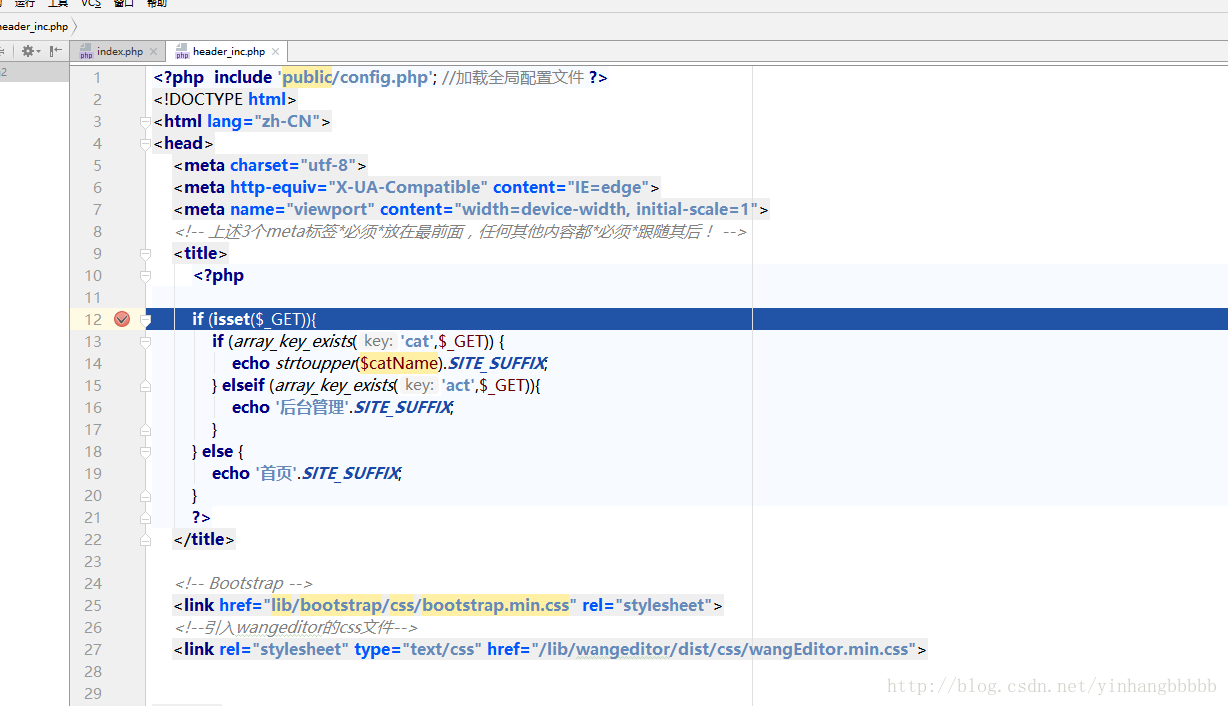


图3-3 断点生效

红色圆中有个对号，是说明改断点生效了

几点注意

（1）Apache服务器默认的端口是80，按照上面设置就可以了，Nginx默认是8080有URL的地方需要调整一下。URL的组成结构中包括端口号，只是默认值是80，很多时候省略了。

（2）遇到问题仔细分析一下，都是可以解决的，如果实在有困难，可以在评论区留言。

4调试的意义

调试对于PHP开发的真正意义，后续补充。

5参考资料

1 phpStorm+XDebug进行断点调试的配置

http://www.chenxuanyi.cn/xampp-phpstorm-xdebug.html

2利用下面网页中的Start debug按钮在COOKIE中设置Xdebug所需的变量。

http://www.jetbrains.com/phpstorm/marklets/

3 Cannot accept external Xdebug connection:Cannot evaluate expression 'isset($\_SERVER['PHP\_IDE\_CONFIG'])

http://devnet.jetbrains.com/message/5463083?tstart=0

4Xdebug调试原理

http://my.oschina.net/atanl/blog/371424