

## 41 | Linux/Mac实验环境搭建与URI查询参数

2019-10-09 Chrono

《透视HTTP协议》

课程介绍 >



讲述：Chrono

时长 06:20 大小 5.09M



你好，我是 Chrono。

先要说一声“抱歉”。由于工作比较紧张、项目实施频繁出差，导致原本预定的“答疑篇”迟迟没有进展，这次趁着“十一”长假，总算赶出了两期，集中回答几个同学们问得比较多的问题：Linux/Mac 实验环境搭建（[🔗第 7 讲](#)），URI 查询参数（[🔗第 11 讲](#)），还有 DHE/ECDHE 算法的原理（[🔗第 26 讲](#)），后续有时间可能还会再陆续补充完善。

很高兴在时隔一个多月后与你再次见面，废话不多说了，让我们开始吧。

### Linux 上搭建实验环境


我们先来看一下如何在 Linux 上搭建课程的实验环境。

领资料




首先，需要安装 OpenResty，但它在 Linux 上提供的不是 zip 压缩包，而是各种 Linux 发行版的预编译包，支持常见的 Ubuntu、Debian、CentOS 等等，而且 [官网](#) 上有非常详细安装步骤。

以 Ubuntu 为例，只要“按部就班”地执行下面的几条命令就可以了，非常轻松：

 复制代码

```
1 # 安装导入GPG公钥所需的依赖包：
2 sudo apt-get -y install --no-install-recommends wget gnupg ca-certificates
3
4
5 # 导入GPG密钥：
6 wget -O - https://openresty.org/package/pubkey.gpg | sudo apt-key add -
7
8
9 # 安装add-apt-repository命令
10 sudo apt-get -y install --no-install-recommends software-properties-common
11
12
13 # 添加官方仓库：
14 sudo add-apt-repository -y "deb http://openresty.org/package/ubuntu $(lsb_release
15
16
17 # 更新APT索引：
18 sudo apt-get update
19
20
21 # 安装 OpenResty
22 sudo apt-get -y install openresty
```

全部完成后，OpenResty 会安装到“/usr/local/openresty”目录里，可以用它自带的命令行工具“resty”来验证是否安装成功：

 复制代码

```
1 $resty -v
2 resty 0.23
3 nginx version: openresty/1.15.8.2
4 built with OpenSSL 1.1.0k 28 May 2019
```

领资料



有了 OpenResty，就可以从 GitHub 上获取 http\_study 项目的源码了，用“git clone”是最简单快捷的方法：

```
1 git clone https://github.com/chronolaw/http_study
```

在 Git 仓库的“www”目录，我为 Linux 环境补充了一个 Shell 脚本“run.sh”，作用和 Windows 下的 start.bat、stop.bat 差不多，可以简单地启停实验环境，后面可以接命令行参数 start/stop/reload/list：

复制代码

```
1 cd http_study/www/      #脚本必须在www目录下运行，才能找到nginx.conf
2 ./run.sh start          #启动实验环境
3 ./run.sh list           #列出实验环境的Nginx进程
4 ./run.sh reload         #重启实验环境
5 ./run.sh stop           #停止实验环境
```

启动 OpenResty 之后，就可以用浏览器或者 curl 来验证课程里的各个测试 URI，但之前不要忘记修改“/etc/hosts”添加域名解析，例如：

复制代码

```
1 curl -v "http://127.0.0.1/"
2 curl -v "http://www.chrono.com/09-1"
3 curl -k "https://www.chrono.com/24-1?key=1234"
4 curl -v "http://www.chrono.com/41-1"
```

## Mac 上搭建实验环境

看完了 Linux，我们再来看一下 Mac。

这里我用的是两个环境：Mac mini 和 MacBook Air，不过都是好几年前的“老古董”了，系统是 10.13 High Sierra 和 10.14 Mojave（更早的版本没有测试，但应该也都可以）。

首先要保证 Mac 里有第三方包管理工具 homebrew，可以用下面的命令安装：

复制代码

```
1 #先安装Mac的homebrew
2 /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install
```

领资料



然后，要用 homebrew 安装 OpenResty，但它在 Mac 上的安装过程和 Linux 不同，不是预编译包，而是要下载许多相关的源码（如 OpenSSL），然后再用 clang 本地编译，大概要花上五六分钟的时间，整体上比较慢，要有点耐心。

 复制代码

```
1 #使用homebrew安装OpenResty
2 brew install openresty/brew/openresty
```

安装完 OpenResty，后续的操作就和 Linux 一样了，“git clone”项目源码：

 复制代码

```
1 git clone https://github.com/chronolaw/http_study
```

然后，进“http\_study/www”目录，用脚本“run.sh”启停实验环境，用 Safari 或者 curl 测试。

## Linux/Mac 下的抓包

Linux 和 Mac 里都有图形界面版本的 Wireshark，抓包的用法与 Windows 完全一样，简单易用。

所以，今天我主要介绍命令行形式的抓包。

命令行抓包最基本的方式就是著名的 tcpdump，不过我用得不是很多，所以就尽可能地“藏拙”了。

简单的抓包使用“-i lo”指定抓取本地环回地址，“port”指定端口号，“-w”指定抓包的存放位置，抓包结束时用“Ctrl+C”中断：

 复制代码

```
1 sudo tcpdump -i lo -w a.pcap
2 sudo tcpdump -i lo port 443 -w a.pcap
```

抓出的包也可以用 tcpdump 直接查看，用“-r”指定包的名字：

 领资料



```
1 tcpdump -r a.pcap
2 tcpdump -r 08-1.pcapng -A
```

不过在命令行界面下可以用一个更好的工具——tshark，它是 Wireshark 的命令行版本，用法和 tcpdump 差不多，但更易读，功能也更丰富一些。

```
1 tshark -r 08-1.pcapng
2 tshark -r 08-1.pcapng -V
3 tshark -r 08-1.pcapng -O tcp|less
4 tshark -r 08-1.pcapng -O http|less
```

tshark 也支持使用 keylogfile 解密查看 HTTPS 的抓包，需要用“-o”参数指定 log 文件，例如：

```
1 tshark -r 26-1.pcapng -O http -o ssl.keylog_file:26-1.log|less
```

tcpdump、tshark 和 Linux 里的许多工具一样，参数繁多、功能强大，你可以课后再找些资料仔细研究，这里就不做过多地介绍了。

## URI 的查询参数和头字段

在 [第 11 讲](#) 里我留了一个课下作业：

“URI 的查询参数和头字段很相似，都是 key-value 形式，都可以任意自定义，那么它们在使用时该如何区别呢？”

从课程后的留言反馈来看，有的同学没理解这个问题的本意，误以为问题问的是这两者在表现上应该如何区分，比如查询参数是跟在“？”后面，头字段是请求头里的 KV 对。

这主要是怪我没有说清楚。这个问题实际上想问的是：查询参数和头字段两者的形式很相近，query 是 key-value，头字段也是 key-value，它们有什么区别，在发送请求时应该如何正确地使用它们。



换个说法就是：应该在什么场景下恰当地自定义查询参数或者头字段来附加额外信息。

当然了，因为 HTTP 协议非常灵活，这个问题也不会有唯一的、标准的答案，我只能说说我自己的理解。

因为查询参数是与 URI 关联在一起的，所以它针对的就是资源（URI），是长期、稳定的。而头字段是与一次 HTTP 请求关联的，针对的是本次请求报文，所以是短期、临时的。简单来说，就是两者的作用域和时效性是不同的。

从这一点出发，我们就可以知道在哪些场合下使用查询参数和头字段更加合适。

比如，要获取一个 JS 文件，而它会有多个版本，这个“版本”就是资源的一种属性，应该用查询参数来描述。而如果要压缩传输、或者控制缓存的时间，这些操作并不是资源本身固有的特性，所以用头字段来描述更好。

除了查询参数和头字段，还可以用其他方式来向 URI 发送附加信息，最常用的一种方式就是 POST 一个 JSON 结构，里面能够存放比 key-value 复杂得多的数据，也许你早就在实际工作中这么做了。

在这种情况下，就可以完全不使用查询参数和头字段，服务器从 JSON 里获取所有必需的数据，让 URI 和请求头保持干净、整洁（^\_^）。

今天的答疑就先到这里，我们下期再见，到时候再讲 ECDHE 算法。

领资料







## == 课外小贴士 ==

- 01 落笔之时我使用的 OpenResty 是 1.15.8.2，这个版本修复了一个 HTTP/2 的安全漏洞，内核仍然是 Nginx 1.15.8，不过 agentzh 承诺很快就会推出基于 Nginx 1.17 的更新版本。
- 02 在 Linux 里使用预编译包安装 OpenResty 会在开机时自动运行，与实验环境的 80 端口冲突，可以用命令“`sudo service openresty stop`”手动停止，或者用“`sudo update-rc.d openresty disable`”彻底禁止自动运行。
- 03 较早期的 OpenResty 在 Mac 上安装使用的是“`brew install homebrew/nginx/openresty`”，如果你这样做了，需要用“`brew untap homebrew/nginx`”先移除，然后用新的官方地址“`brew install openresty/brew/openresty`”。

分享给需要的人，Ta订阅超级会员，你将得 50 元

Ta单独购买本课程，你将得 20 元

生成海报并分享

赞 6 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 40 | HTTP性能优化面面观（下）

下一篇 42 | DHE/ECDHE算法的原理

## 学习推荐

# JVM + NIO + Spring

各大厂面试题及知识点详解

限时免费



## 精选留言 (19)



GitHubGanKai

2019-10-09

真好，又见到你了，而且我最近换个了mac，😁正愁这个。

作者回复: we meet again.

写留言

领资料





**dongge**

2019-10-18

老师好,

按文章指导搭建了MAC的环境:

openresty -v

nginx version: openresty/1.11.2.2

在~/git/http\_study/www目录下执行

./run.sh start

Password:

nginx: [emerg] "/Users/xiaodong/git/http\_study/www/conf/ssl/ticket.key" must be 48 bytes in /Users/xiaodong/git/http\_study/www/conf/nginx.conf:34

报了这个错误, 在网上google没找到解决方法。

尝试在nginx.conf中注销相关代码, 也会报其他错误。

老师能指点一下吗?

作者回复: 这是提示Nginx要求48字节的密钥文件, 按理说附带的80字节也是可以的, 你可以用命令“openssl rand 48 > ticket.key”重新生成www/conf/ssl/ticket.key。

详细可参见[http://nginx.org/en/docs/http/nginx\\_http\\_ssl\\_module.html#ssl\\_session\\_tickets](http://nginx.org/en/docs/http/nginx_http_ssl_module.html#ssl_session_tickets)

另外, 你用的openresty版本太老了, 用最新的1.15.8.2可能就不会出现这样的问题。

共 2 条评论 &gt;

2

**Change**

2020-03-28

老师请教个问题: Mac 环境下安装以后, 按照命令./run.sh start 启动后访问 localhost 显示 403 Forbidden: 终端返回的错误信息是下面的错误信息, 这是所有端口都被占用了? 我查了一下好像也没有被占用啊, 不知道这是啥原因

nginx: [emerg] bind() to 0.0.0.0:80 failed (48: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:8080 failed (48: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:443 failed (48: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:8443 failed (48: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:440 failed (48: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:441 failed (48: Address already in use)

nginx: [emerg] bind() to 0.0.0.0:442 failed (48: Address already in use)

领资料



作者回复: 可能需要sudo, 是否权限的问题。

可以在网上搜一下错误信息, Nginx的问题一般都有现成的解决办法。

共 7 条评论 >



1



**SmNiuhe**

2019-11-07

这个大家有遇到嘛, 是不是资源的问题

brew install openresty/brew/openresty : DownloadError: Failed to download resource  
"openresty-openssl--patch"

Download failed: [https://raw.githubusercontent.com/openresty/openresty/master/patches/openssl-1.1.0d-sess\\_set\\_get\\_cb\\_yield.patch](https://raw.githubusercontent.com/openresty/openresty/master/patches/openssl-1.1.0d-sess_set_get_cb_yield.patch)

作者回复: 如果mac安装有问题, 可以用virtualbox装个Linux虚拟机, 暂解燃眉之急。

共 6 条评论 >



1



**超轶主**

2021-12-13

mac环境运行 run run.sh 返回 nginx version: openresty/1.19.9.1

format : run.sh [start|stop|reload|list]是什么情况呢

作者回复: 后面要加参数, start|stop|reload|list。

脚本比较简单, 可以用vi看看。



**浪里淘沙的小法师**

2021-11-10

讲一下用M1芯片 mac 搭建搭建环境的遇到的问题 and 解决方法。

1. 运行 ./run.sh start 报错 /usr/local/bin/openresty: command not found

这是因为 M1 芯片mac 的 homebrew 安装软件的位置与以往不同, 先通过 which openresty 查询 openresty 的位置 /opt/homebrew/bin/openresty, 然后打开 run.sh 脚本替换一下老师写的位置

```
if [ $os != "Linux" ]; then
```

```
    openresty="/usr/local/bin/openresty"
```

```
fi
```

领资料



替换成

```
if [ $os != "Linux" ]; then  
    openresty="/opt/homebrew/bin/openresty"  
fi
```

2. 再运行 ./run.sh start 报错 nginx: [emerg] could not build server\_names\_hash, you should increase server\_names\_hash\_bucket\_size: 32

网上查寻了一下，放大 bucket\_size 即可，打开 www/conf/nginx.conf 文件添加这一句server\_names\_hash\_bucket\_size 64; 即可

```
# http conf  
http {  
    #include http/common.conf;  
    #include http/cache.conf;  
    #include http/resty.conf;  
    #include http/mime.types;  
    server_names_hash_bucket_size 64;  
  
    include http/*.conf;  
  
    include http/servers/*.conf;  
}
```

作者回复: 太高端了，都用上M1的Mac。

也可以参考GitHub里的Dockerfile，构建出基于arm的镜像。



👍 1



**silence**

2021-08-23

请问安装好环境后在www目录执行./run.sh start 老是command not found怎么解决

作者回复: 是不是没有安装好openresty，看看是哪个命令没找到，再按照课程正文是否遗漏了哪个步骤。

领资料

共 3 条评论 >



**Miroticwillbeforever**

2021-06-22

老师我有个问题。实验环境搭建好了。前两讲的实验也做成功了。

但是当我用浏览器 访问 [www.chrono.com](http://www.chrono.com) 时，它跳转到的 地址为 <https://dp.diandongzhi.com/?acct=660&site=chrono.com> 然后wireshark抓包并没有任何反应。我想问一下是我操作不当的原因还是怎么回事。课程大部分听完了。但是后面实验没做成挺难受的，没有去验证。等老师给个答复准备二刷！

作者回复: 域名解析这个问题确实困扰了不少同学，因为我们实验环境的域名是假的，只能在本地图用，所以有可能会与某些真实的域名冲突，导致输入域名跑到了外网而不是实际环境。

解决方法是改hosts文件，但因为域名会有缓存，所以有时候改了hosts也不会生效。如果遇到这种情况，最好是换个实验环境，改成虚拟机或者docker。

共 2 条评论 >



Austin

2021-03-20

老师你好、我安装好了openresty后、启动服务说 nginx: invalid option: http，请问是怎么回事呀

作者回复: 这个提示信息太模糊了，不好帮你。

可以网上搜一下错误信息，一般都能找到解决方案。或者把OpenResty、GitHub项目重新安装看看。

还是不行就改用docker试试。



小童

2021-03-02

不行啊，老师，我的那个openresty界面出来了，就是抓不到包！用的wireshark .搞了好久。那个telnet也安装了。是不是那步出错了，我就直接运行openresty，然后用抓包工具过滤信息，然后浏览器输入localhost，浏览器洁界面也出来了。

作者回复: 可以先试试Wireshark能不能抓其他网络的包，然后再试试抓本地（loopback），再加上过滤器，逐步分析来缩小范围。

如果还不行，可以试着换个环境，用虚拟机+Linux或者docker，这样的环境比较单纯隔离。

共 2 条评论 >



旗木卡卡

2020-08-19

领资料



Mac电脑，耗费本人2个晚上的环境，终于搭好了，碰到了2个坑，第一个是dns查找不到，brew install openresty时，需要在本机的hosts文件，加上解析不到的url的ip地址，第二个是启动一直bind不上，nginx就自动启动了，但是很明显不是openresty，然后用root权限启动成功，也可以正常访问，发现是nginx.conf的用户权限问题，修改成本机的用户user kaka(你的用户名) staff;即可。

作者回复: 虽然都是unix，但mac的环境和Linux还是不太一样，辛苦了。

共 3 条评论 >



**Luka!3055**

2020-05-07

记录下问题：

brew install openresty/brew/openresty 后，报错：

curl: (7) Failed to connect to raw.githubusercontent.com port 443: Connection refused  
Error: An exception occurred within a child process:  
DownloadError: Failed to download resource "openresty-openssl--patch"  
Download failed: https://raw.githubusercontent.com/openresty/openresty/master/patches/openssl-1.1.0d-sess\_set\_get\_cb\_yield.patch

此时把 DNS 设置为 114.114.114.114 或者 8.8.8.8 就好了，最好再挂个梯子

作者回复: nice



**Jinlee**

2020-04-21

Welcome to HTTP Study Page! 还好我看得迟，成功在ubuntu下搭建起环境😊😊😊

作者回复: nice



领资料



**人人都是数学家**

2020-01-06

最喜欢实验环境了，之前学习就是苦于没实验环境浪费了几年时间。

作者回复: 搭好后一定要多做实验，实践出真知。





无名

2019-11-15

Updating Homebrew...

==> Auto-updated Homebrew!

Updated 1 tap (homebrew/core).

==> Updated Formulae

handbrake

==> Installing openresty from openresty/brew

==> Downloading <https://openresty.org/download/openresty-1.15.8.2.tar.gz>

Already downloaded: /Users/hejunbin/Library/Caches/Homebrew/downloads/4395089f0fd423261d4f1124b7beb0f69e1121e59d399e89eaa6e25b641333bc--openresty-1.15.8.2.tar.gz

==> ./configure -j8 --prefix=/usr/local/Cellar/openresty/1.15.8.2 --pid-path=/us

Last 15 lines from /Users/hejunbin/Library/Logs/Homebrew/openresty/01.configure:

DYNASM host/buildvm\_arch.h

HOSTCC host/buildvm.o

HOSTLINK host/buildvm

BUILDVM lj\_vm.S

BUILDVM lj\_ffdef.h

BUILDVM lj\_bcdef.h

BUILDVM lj\_folddef.h

BUILDVM lj\_recdef.h

BUILDVM lj\_libdef.h

BUILDVM jit/vmdef.lua

make[1]: \*\*\* [lj\_folddef.h] Segmentation fault: 11

make[1]: \*\*\* Deleting file `lj\_folddef.h'

make[1]: \*\*\* Waiting for unfinished jobs....

make: \*\*\* [default] Error 2

ERROR: failed to run command: gmake -j8 TARGET\_STRIP=@: CCDEBUG=-g XCFLAG S='-msse4.2 -DLUAJIT\_NUMMODE=2 -DLUAJIT\_ENABLE\_LUA52COMPAT' CC=cc PREFIX=/usr/local/Cellar/openresty/1.15.8.2/luajit

If reporting this issue please do so at (not Homebrew/brew or Homebrew/core):

<https://github.com/openresty/homebrew-brew/issues>

These open issues may also help:

Can't install openresty on macOS 10.15 <https://github.com/openresty/homebrew-brew/issues/10>

领资料





Fails to install OpenResty <https://github.com/openresty/homebrew-brew/issues/5>  
The openresty-debug package should use openresty-openssl-debug instead <https://github.com/openresty/homebrew-brew/issues/3>

macOS 10.15.1 安装失败。参考给出的链接也没有解决问题，求老师解惑。

作者回复: 10.15是新出的，我没升级，看这些信息像是openresty在这上面安装有问题（luaJIT编译失败），可以向官方反应一下，只能期待官方更新包了。

共 2 条评论 >



**Dream.**

2019-10-17

Linux-CentOS 7下，修改了/etc/hosts的域名与IP的映射关系后

再使用./run.sh start启动OpenResty之后

curl localhost 或者 curl http://www.chrono.com都是返回403

按之前课程里的url访问<https://www.chrono.com/11-1>什么的，都返回404

第一次接触OpenResty，麻烦老师回复下是哪里没配置好嘛？

作者回复: 我这里没有遇到你这样的现象。

先用run.sh list看看Nginx进程是否正常运行，然后用netstat等工具检查一下监听端口，是否有防火墙什么的其他应用阻碍了服务。

可以问问周围熟悉Linux运维的同事。

共 4 条评论 >



**dongge**

2019-10-16

这个专栏这么好玩，留言的人这么少，真可惜。

作者回复: 答疑篇来的太晚了，没赶上当初的热度，不过总会有需要的同学看到的。



**许童童**

2019-10-13

老师又来了，很高兴再次见到老师。

作者回复: nice to meet you again.



果果

2019-10-09

当初费了好些时间，才在mac上搭建了环境

作者回复: 这篇答疑来晚了，实在是抱歉。



领资料

