# 智能车项目——服务器接口搭建

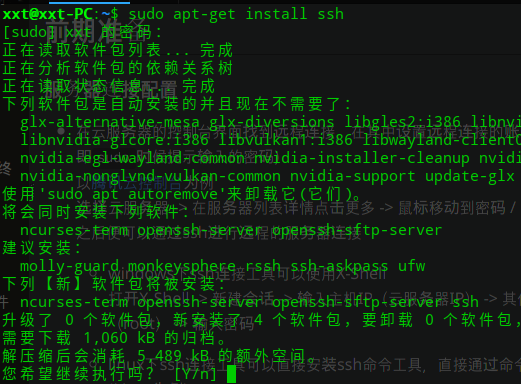
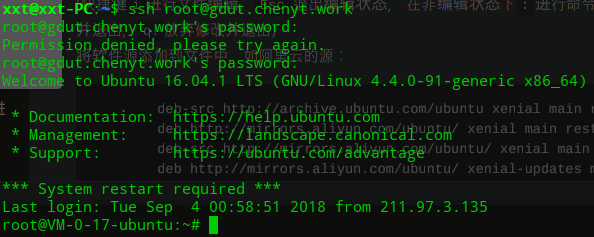
在云端购买服务器之后，需要依次进行系统选择(在本项目中选用ubuntu系统)、环境搭建(AMP)、接口项目配置(ThinkphpRestApi)

# 目录

* [前期准备](#前期准备)
  + [服务器连接配置](#服务器连接配置)
  + [软件源配置](#软件源配置)
  + [环境搭建](#ubuntu下环境搭建)
  + [导入接口项目](#导入接口项目)
  + [域名解析配置](#域名解析配置)
  + [SSL协议配置](#SSL协议配置)
* [正式使用](#正式使用)
  + [wifi模块](#wifi模块)
  + [python算法文件](#python算法文件)

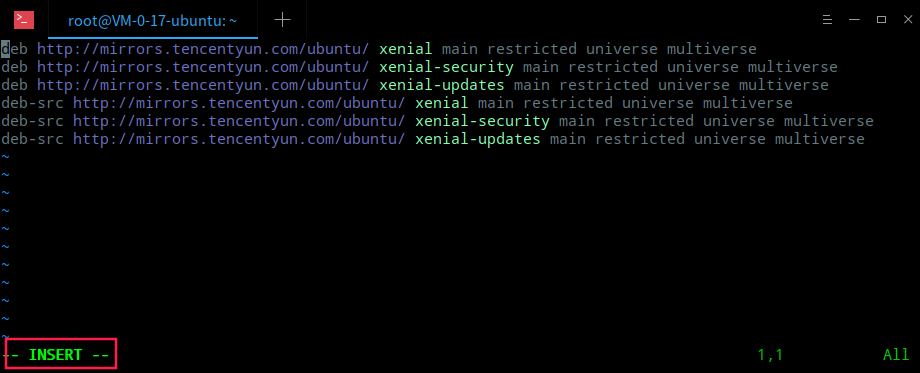
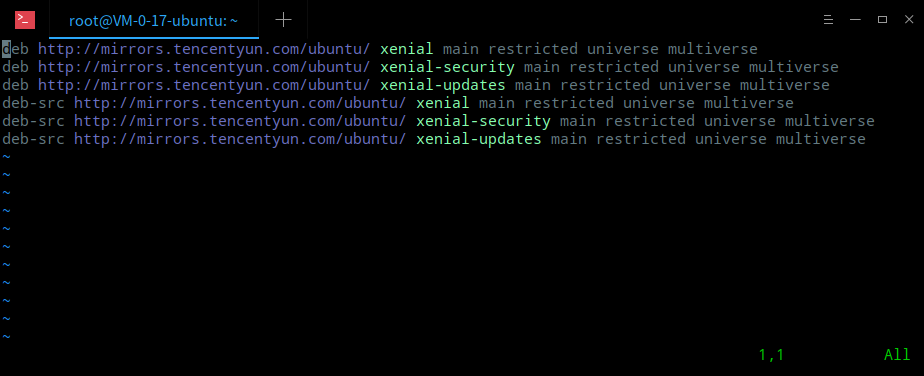
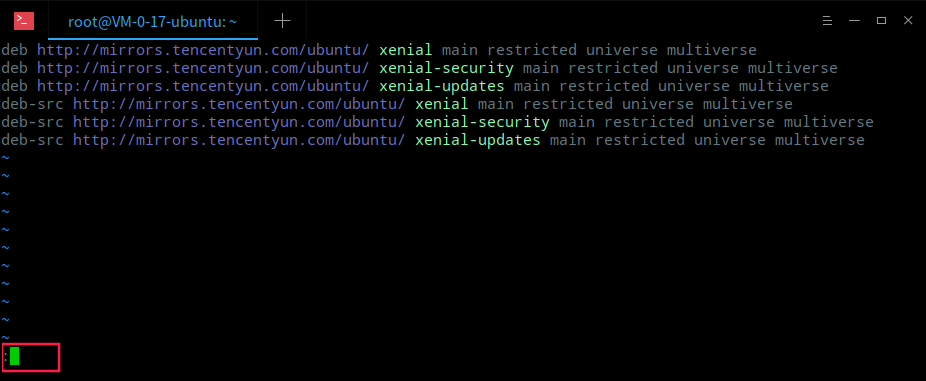
## 前期准备

#### 服务器连接配置

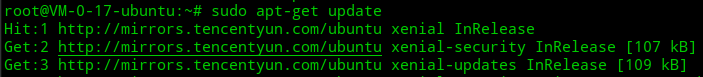
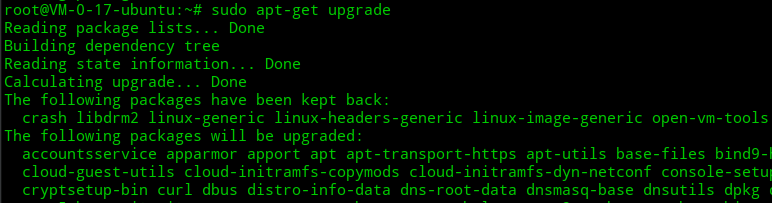
* 在云服务器的控制台界面找到远程连接，在其中设置远程连接的账号密码(也是在服务器中的账号密码，即sudo时候提示输入的密码)  
  以[腾讯云控制台](https://console.cloud.tencent.com/)为例  
  选择云服务器 -> 在服务器列表详情点击更多 -> 鼠标移动到密码 / 密钥选项 -> 点击重置密码  
  之后便可以通过ssh进行远程的服务器连接
  + windows下ssh连接工具可以使用X-Shell  
    打开X-Shell -> 新建会话 -> 输入主机IP（云服务器IP） -> 其他选项保持默认保存 -> 提示输入用户名（root） -> 输入密码
  + linux下ssh连接工具可以直接安装ssh命令工具，直接通过命令行终端进行连接  
    (以apt为例)  
    sudo apt-get install ssh  
      
    ssh account@youripOrdomain 之后输入密码即可连接进入服务器 

#### 软件源配置

在云服务器上的Ubuntu系统初次使用一般是没有配置软件源的，需要进行软件源的配置

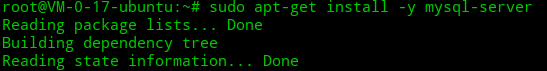
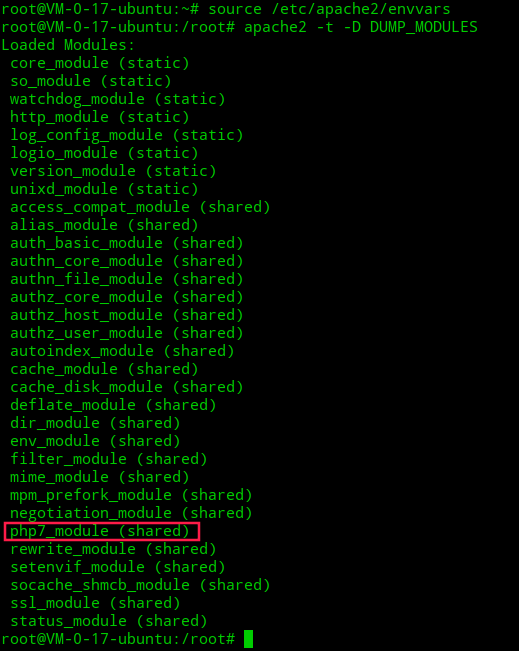
1. 对source.list文件进行备份 sudo cp /etc/apt/sources.list /etc/apt/sources.list\_backup
2. 编辑source.list文件，sudo vi /etc/apt/sources.list
   * 快捷键i进行文件编辑 
   * Esc 退出编辑状态 
   * 在非编辑状态下:进行命令输入 
   * 命令wq 保存当前修改并退出  (如果不需要保存则通过q!放弃修改并退出)  
     在编辑模式下将软件源添加到文件中，如阿里云的源：

* deb-src http://archive.ubuntu.com/ubuntu xenial main restricted #Added by software-properties  
   deb http://mirrors.aliyun.com/ubuntu/ xenial main restricted  
   deb-src http://mirrors.aliyun.com/ubuntu/ xenial main restricted multiverse universe #Added by software-properties  
   deb http://mirrors.aliyun.com/ubuntu/ xenial-updates main restricted  
   deb-src http://mirrors.aliyun.com/ubuntu/ xenial-updates main restricted multiverse universe #Added by software-properties  
   deb http://mirrors.aliyun.com/ubuntu/ xenial universe  
   deb http://mirrors.aliyun.com/ubuntu/ xenial-updates universe  
   deb http://mirrors.aliyun.com/ubuntu/ xenial multiverse  
   deb http://mirrors.aliyun.com/ubuntu/ xenial-updates multiverse  
   deb http://mirrors.aliyun.com/ubuntu/ xenial-backports main restricted universe multiverse  
   deb-src http://mirrors.aliyun.com/ubuntu/ xenial-backports main restricted universe multiverse #Added by software-properties  
   deb http://archive.canonical.com/ubuntu xenial partner  
   deb-src http://archive.canonical.com/ubuntu xenial partner  
   deb http://mirrors.aliyun.com/ubuntu/ xenial-security main restricted  
   deb-src http://mirrors.aliyun.com/ubuntu/ xenial-security main restricted multiverse universe #Added by software-properties  
   deb http://mirrors.aliyun.com/ubuntu/ xenial-security universe  
   deb http://mirrors.aliyun.com/ubuntu/ xenial-security multiverse
* 以上文本可以复制后通过控制台粘贴至编辑状态的sources.list文件中，之后wq保存退出

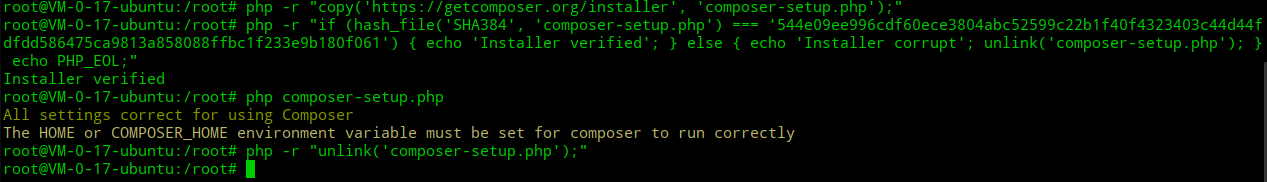
1. 使修改生效  
   sudo apt-get update 更新软件列表  
     
   sudo apt-get upgrade 更新软件  
   

#### ubuntu下环境搭建

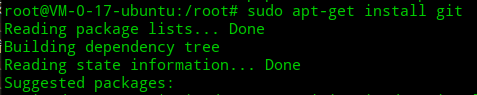
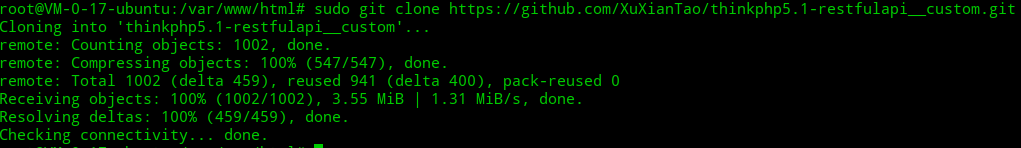
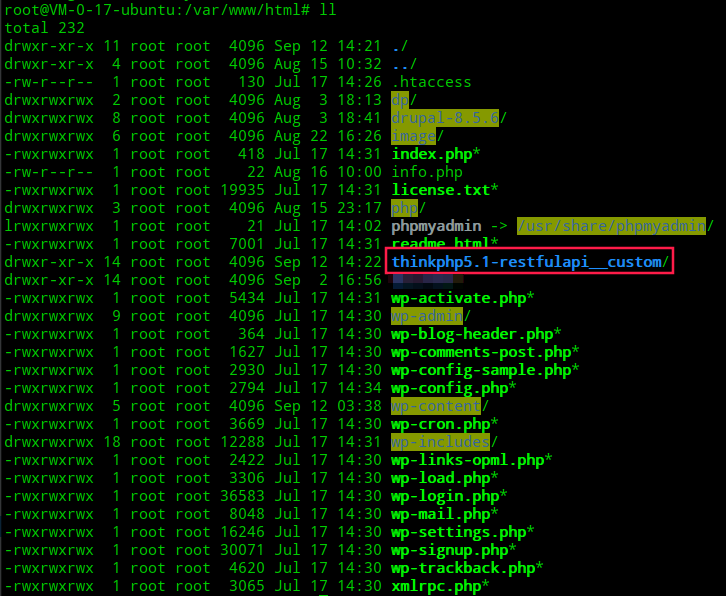
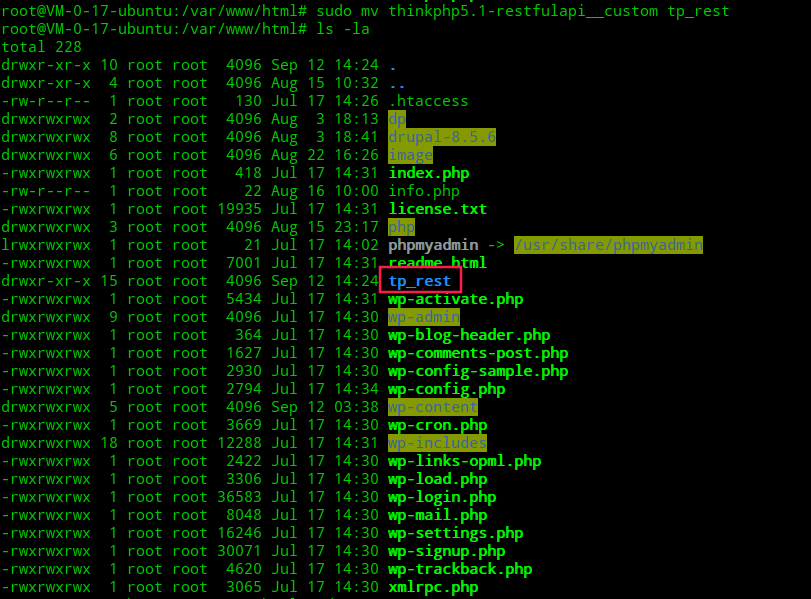
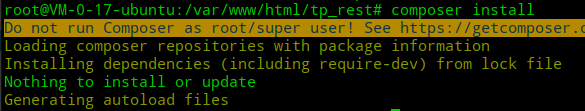
在得到云服务器之后需要进行AMP环境搭建 1. Apache服务器搭建，通过sudo apt-get install apache2 从Ubuntu软件源下载安装apache服务器，按照提示一路y允许安装即可 

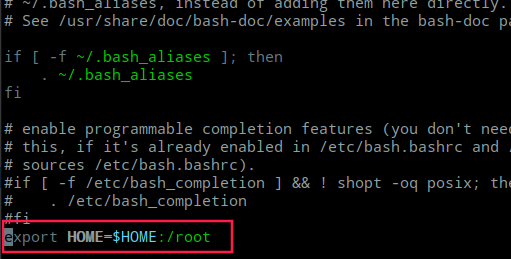
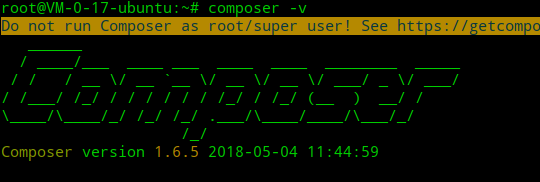
1. php7.1运行环境安装，sudo apt-get install -y php7.1 php7.1-mysql php7.1-mcrypt php7.1-curl php7.1-mbstring php7.1-zip php7.1-xml  (如果发生找不到php7.1库的情况，可以安装php7.0替代)
2. mysql数据库安装，sudo apt-get install -y mysql-server,安装中按照提示配置账号密码即可 
3. 配置完成之后，检查apache2相关配置
   * source /etc/apache2/envvars 导入apache的环境变量
   * apache2 -t -D DUMP\_MODULES 查看apache启用的模块是否有php7的相关模块，保证apache的php运行环境，如果没有php模块，需要运行sudo apt-get install libapache2-mod-php7.1安装对应php7.1版本的apache插件(这里安装了php7.0作为替代)  
     
4. php库管理命令composer安装，直接将以下命令行copy到ubuntu控制台运行即可，视情况在命令行首添加sudo

php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"  
php -r "if (hash\_file('SHA384', 'composer-setup.php') === '544e09ee996cdf60ece3804abc52599c22b1f40f4323403c44d44fdfdd586475ca9813a858088ffbc1f233e9b180f061') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP\_EOL;"  
php composer-setup.php  
php -r "unlink('composer-setup.php');"

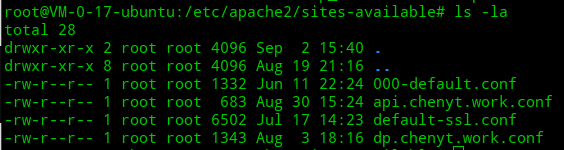
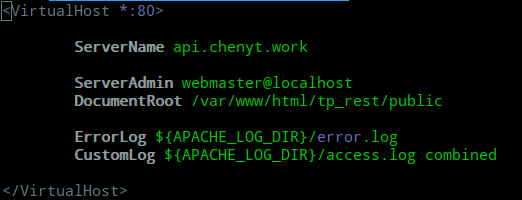


#### 导入接口项目

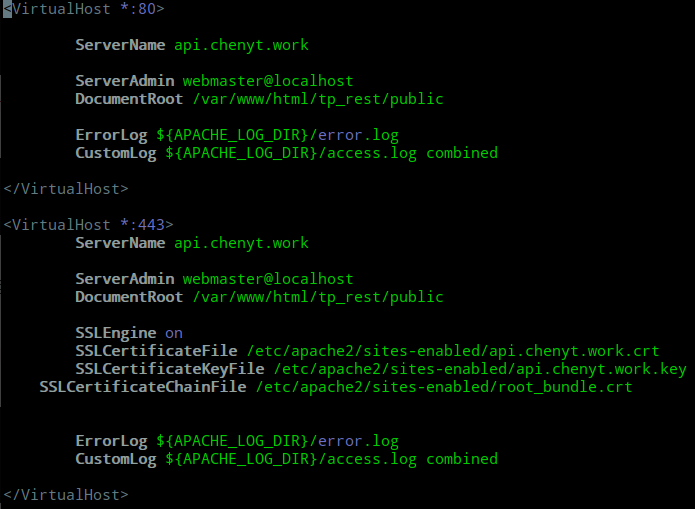
1. sudo apt-get install git安装git 代码版本管理工具，这里仅用于下载项目代码  
   
2. cd /var/www/html定位目录到服务器项目路径下 
3. sudo git clone https://github.com/XuXianTao/thinkphp5.1-restfulapi\_\_custom.git 将项目拷贝下来,会在该项目路径下生成文件夹thinkphp5.1-restfulapi\_\_custom  生成项目文件夹/thinkphp5.1-restfulapi\_custom 
4. 为了方便，将目录重命名为tp\_restsudo mv thinkphp5.1-restfulapi\_\_custom tp\_rest 
5. 命令ll或者ls -la可以查看当前工作路径下的文件夹
6. cd tp\_rest进入项目目录下
7. composer install 安装项目所需的核心库即可 

如果composer 运行提示环境变量问题，则要手动添加HOME环境变量 - 在用户根目录下打开.bashrc文件  
 - 在文件中添加HOME环境变量，如下  - :wq保存退出后运行一遍使刚才添加的环境变量生效  - composer -v 查看是否正常运行 

#### 域名解析配置

1. 在域名交易系统上购买域名yourdomain(本项目域名为chenyt.work)，并在云服务器的控制台中配置域名解析(二级域名api.chenyt.work)，以及进行域名备案等相关工作
   * [域名解析](https://console.cloud.tencent.com/domain)： 在域名列表中选择域名进行解析 -> 点击添加记录 -> 输入主机记录（api） -> 记录类型（A） -> 记录值填入云服务器IP -> 其他选项保持默认保存即可
2. 定位apache服务器站点配置目录cd /etc/apache2/sites-available 
3. sudo cp 000-default.conf yourdomain.conf拷贝一份新的站点配置文件yourdomain.conf(本项目为api.chenyt.work.conf)(复制后结果如下)，进入编辑sudo vi yourdomain.conf 
4. i进入编辑状态
   * 去掉ServerName 的注释并修改为购买的域名yourdomain(本项目ServerName api.chenyt.work)
   * 修改DocumentRoot为 接口项目文件的路径DocumentRoot /var/www/html/tp\_rest/public
   * 其他不变  
       
     Esc 退出编辑  
     :wq 保存修改并退出
5. sudo a2ensite yourdomain.conf 启用刚刚配置的站点
6. sudo service apache2 reload 重载apache服务器，使站点生效 

#### SSL协议配置

由于小程序的接口有安全性要求，所以需要为服务器配置相关的https服务，即ssl协议的配置 - sudo vi /etc/apache2/sites-available/yourdomain.conf  
在服务器的站点配置文件中添加监听端口443并配置相应的ssl设置 

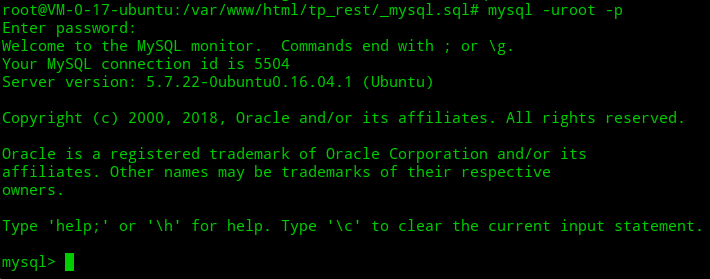
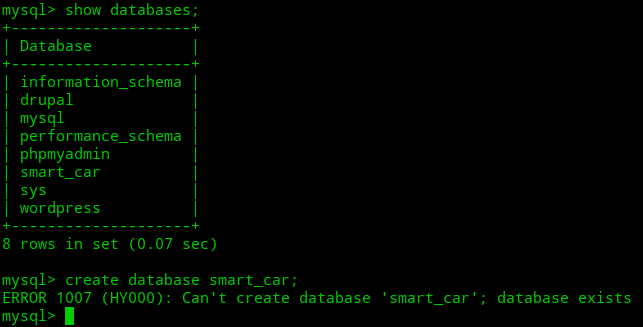
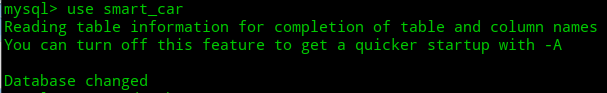
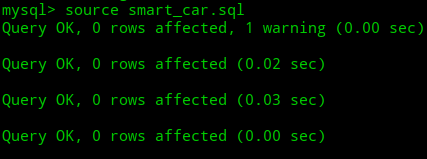
* 证书配置
  + [获取证书](https://buy.cloud.tencent.com/ssl)
    - 选择证书种类
    - 点击申请
    - 按提示操作即可
    - 配置证书：  
      Apache文件夹内获得证书文件 1\_root\_bundle.crt，2\_www.domain.com\_cert.crt 和私钥文件 3\_www.domain.com.key
      * 1\_root\_bundle.crt 文件包括一段证书代码 “—–BEGIN CERTIFICATE—–”和“—–END CERTIFICATE—–”,
      * 2\_www.domain.com\_cert.crt 文件包括一段证书代码 “—–BEGIN CERTIFICATE—–”和“—–END CERTIFICATE—–”,
      * 3\_www.domain.com.key 文件包括一段私钥代码“—–BEGIN RSA PRIVATE KEY—–”和“—–END RSA PRIVATE KEY—–”。
  + 证书安装  
    编辑Apache根目录下 conf/httpd.conf 文件， 找到 #LoadModule ssl\_module modules/mod\_ssl.so 和 #Include conf/extra/httpd-ssl.conf，去掉前面的#号注释； 编辑Apache根目录下 conf/extra/httpd-ssl.conf 文件，修改如下内容：
  + <VirtualHost 0.0.0.0:443>  
     DocumentRoot "/var/www/html"  
     ServerName www.domain.com  
     SSLEngine on  
     SSLCertificateFile /usr/local/apache/conf/2\_www.domain.com\_cert.crt  
     SSLCertificateKeyFile /usr/local/apache/conf/3\_www.domain.com.key  
     SSLCertificateChainFile /usr/local/apache/conf/1\_root\_bundle.crt  
     </VirtualHost>
  + 配置完成后，重新启动 Apache 就可以使用https://www.domain.com来访问了。
  + 注：

|  |  |
| --- | --- |
| 配置文件参数 | 说明 |
| SSLEngine on | 启用SSL功能 |
| SSLCertificateFile | 证书文件 |
| SSLCertificateKeyFile | 私钥文件 |
| SSLCertificateChainFile | 证书链文件 |

* 针对具体子站点的配置，将已有的80端口配置内容复制并粘贴，修改监听端口为443，同时添加SSL解析配置，具体文件内容如下
* <VirtualHost \*:80>  
    
   ServerName api.chenyt.work  
    
   ServerAdmin webmaster@localhost  
   DocumentRoot /var/www/html/tp\_rest/public  
    
   ErrorLog ${APACHE\_LOG\_DIR}/error.log  
   CustomLog ${APACHE\_LOG\_DIR}/access.log combined  
    
   </VirtualHost>  
    
   <VirtualHost \*:443>  
   ServerName api.chenyt.work  
    
   ServerAdmin webmaster@localhost  
   DocumentRoot /var/www/html/tp\_rest/public  
    
   SSLEngine on  
   SSLCertificateFile /etc/apache2/sites-enabled/api.chenyt.work.crt  
   SSLCertificateKeyFile /etc/apache2/sites-enabled/api.chenyt.work.key  
   SSLCertificateChainFile /etc/apache2/sites-enabled/root\_bundle.crt  
    
    
   ErrorLog ${APACHE\_LOG\_DIR}/error.log  
   CustomLog ${APACHE\_LOG\_DIR}/access.log combined  
    
   </VirtualHost>
* 保存修改后退出重启apache2服务即可，sudo service apache2 reload

## 正式使用

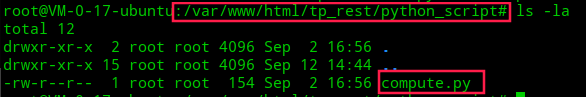
#### 数据库导入

* 定位路径到项目下的数据库文件目录cd /var/www/html/tp\_rest/\_mysql.sql
* 登录mysqlmysql -uroot -p(其中-u后面紧跟的参数是在安装mysql时候设置的账号，-p后面留空)回车后再输入对应账号的密码即可进入mysql 
* 查看当前拥有的数据库，创建新的数据库smart\_car, create database smart\_car;  
  
* 切换到smart\_car数据库 use smart\_car  
  
* 导入数据库文件source smart\_car.sql  
  
* 完成后即可退出mysqlexit

#### wifi模块

* 数据请求的结构部分请参照[README文档](./README.md)进行构建测试
* 以每15ms为周期，将数据打包上传到服务器对应接口，分别上传到以为数据接口与图片接口，且二者请求数据的gid请保持一致，每15ms的请求递增一次，而内部的打包数据序号(list下的数据键值与图片的键值)则每次都从序号1开始

#### python算法文件

* python相关算法请在项目路径下的/python\_script/compute.py中完成  (推荐通过git将github项目拷贝到本地进行开发，修改python文件后提交修改请求即可) (或者本地调试完成后将单独的python文件提交到负责人处更新仓库)
* 运算参数已通过json形式的字符串传入，请用sys.argv[1]获取并解构得到输入数据,输入数据样例如下
* 运算结果请保证用单独一个string输出(通过print输出的第一个字符串即是结果)