系统基础管理平台需求

# 模块关系

Web界面

GET/POST JSON

main.py

module.py

module.py

module.py

# 流程说明

# 部署

## python部署

基本要求，系统要有python环境

安装python开发包

#64位

yum install -y python-devel.x86\_64

#32位

yum install -y python-devel.i686

安装web.py框架库，请参考

<http://webpy.org/install>

## uwsgi部署

下载最新版的uwsgi

<http://projects.unbit.it/uwsgi/>

解压后编译即可，不用安装，然后在opt目录建立一个uwsgi文件夹

cd /opt

mkdir uwsgi

cd uwsgi

#建立一个软链接，指向之前编译好的uwsgi执行文件

ln -s /xxx/uwsgi-x.x.x/uwsgi uwsgi

然后把模块文件，uwsgi的启动脚本等代码脚本都copy到/opt/uwgis目录，启动uwsgi进程

run\_uwsgi.sh启动wusgi脚本

|  |
| --- |
| #!/bin/sh  #全部uwsgi的请求都发到main.py文件处理  ./uwsgi -s 127.0.0.1:3031 -M -p 4 --pythonpath /opt/uwsgi --chdir /opt/uwsgi --wsgi-file main.py --daemonize /var/log/uwsgi.log |

## nginx部署

下载最新版的nginx

<http://nginx.org/en/download.html>

解压后编译安装到opt目录

nginx.con配置文件

|  |
| --- |
| user root;  worker\_processes 1;  #error\_log /dev/null;  events { worker\_connections 1024; }  http {  include mime.types;  default\_type application/octet-stream;  sendfile on;  keepalive\_timeout 65;  server {  listen 80;  location / {  root html;  index index.html index.htm;  }  #配置访问uwsgi目录全部都pass到python后端  location ~ /uwsgi/(.\*) {  include uwsgi\_params;  uwsgi\_pass 127.0.0.1:3031;  }  }  } |

## webadmin界面部署

cd /opt

mkdir webadmin

#把webadmin界面代码都copy到/opt/webadmin目录下面，然后在nginx的html文件夹建立一个文件夹的软链接指向/opt/webadmin目录

cd /opt/nginx/html

ln -s /opt/webadmin webadmin

# 模块程序及调用规范

模块目录 /opt/uwsgi/modules

模块名称 所属板块\_功能.py

比如系统时间，它属于硬件板块，因此系统时间的模块名称为hardware\_date.py，如此类推

## URL请求规范

http://hostname/uwsgi/main?module=module.py&opt=opt&....

module为实现功能的模块py脚本

opt为操作指令

后面为其它需要传递的参数

## 返回结果（JSON格式）

ret=返回0代表执行成功，其它为错误号

error=如果执行不成功，则保存错误的提示信息

result=如果执行成功，则保存返回结果

descmap=字段与中文描述的映射表

# 功能列表与接口描述

|  |  |
| --- | --- |
| 功能 | 描述 |
| 系统信息 | 显示一些系统的基本信息如  主机名  操作系统发行版本  内核版本  管理平台版本  CPU信息  系统开机时间  进程数  CPU负荷  内存信息  磁盘信息  模块：system\_info.py  请求GET opt=showinfo  返回  //CPU基本信息  "cpu\_info": {  "cpu\_used\_per": 0,  "vendor\_id": "GenuineIntel",  "model\_name": "Intel Xeon E312xx (Sandy Bridge)",  "processor": 1  }  cpu\_used\_per代表cpu当前的使用率  vendor\_id  model\_name  processor代表处理器个数  //磁盘基本信息  "disk\_info": {  "/dev/mapper/VolGroup-lv\_home": {  "path": "/home",  "type": "ext4",  "fs\_info": {  "total": 50782,  "used": 179,  "free": 48022  }  },  ……  ……  }  依次是每个分区对应该分区的信息  path是分区所挂载的目录  type是分区类型  fs\_info里面包含total是空间大小MB，used是使用了多少MB，free是剩余空间MB  //内存信息  "mem\_info": {  "swap\_total": 1015800,  "mem\_total": 502464,  "swap\_used\_per": 0,  "mem\_used\_per": 33  }  swap\_total交换分区大小KB  mem\_total物力内存大小KB  swap\_used\_per交换分区使用率  mem\_used\_per物力内存使用率  //操作系统信息  "os\_info": "Linux-2.6.32-358.el6.x86\_64-x86\_64-with-centos-6.4-Final" |
| 平台 |  |
| 账号管理 | 增删改查 |
| 日志管理 | 管理平台的操作日志记录 |
| 系统 |  |
| 进程管理 | 进程列表  关闭指定进程  模块：system\_ps.py  查看进程列表：  请求 GET opt=showps  返回进程列表 |
| 账号管理 | 增删改查系统账号 |
| 磁盘管理 | 查看磁盘分区及使用情况 |
| 日志管理 | 查看系统日志文件  查看指定日志文件指定位置内容 |
| 软件管理 | 查看，升级，删除，安装指定软件 |
| 服务 |  |
| SSH服务 | 配置SSH端口，服务停止，重启 |
| 网络 |  |
| 防火墙 | 允许，禁止IP，端口策略 |
| 网络配置 | 查看网络接口信息  配置网络接口IP等 |
| 硬件 |  |
| 系统时间 | 功能描述：  查看，修改系统时间  模块：hardware\_date.py  查看时间：  请求GET opt=showdate  返回系统当前时间字符串 |
| 重启关机 | 重启，关机 |
| 其它 |  |
| 命令行 | 执行命令  自定义命令  执行脚本 |
| 上传下载文件 | 上传，下载文件 |
|  |  |