1. 项目创建

在 pycharm中可以创建django项目，所以有别于在ubuntu 中

ubuntu中：django-admin startproject acapp

1. 项目运行

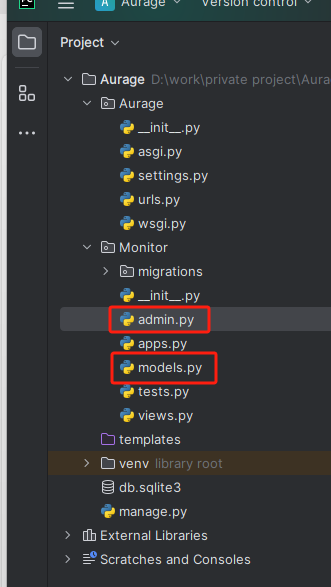
python3 manage.py runserver 0.0.0.0:8000

1. 创建app

python3 manage.py startapp game

1. 数据库

创建 app 后在标红位置设置数据库



修改数据库后需要python3 manage.py migrate，同步修改

1. 创建超级用户

python3 manage.py createsuperuser

1. 修改settings.py

ALLOWED\_HOSTS = ['0.0.0.0'](添加可访问ip)

INSTALLED\_APPS 增加 "Monitor.apps.MonitorConfig"

TIME\_ZONE = "Asia/Shanghai"

STATIC\_ROOT = os.path.join(BASE\_DIR, 'Monitor/static')  
 STATIC\_URL = "/static/"  
 MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')  
 MEDIA\_URL = "/media/"

1. 数据库设计

用户，设备，模式

用户：(id、手机号)

方案：(id、name、手机号、描述)

用户-方案表：（用户id,方案id）

设备：（id、name）

模式：(id、name、描述、url)

方案-模式表：（方案id,模式id）

模式-设备表：（模式id,设备id）

指令记录：

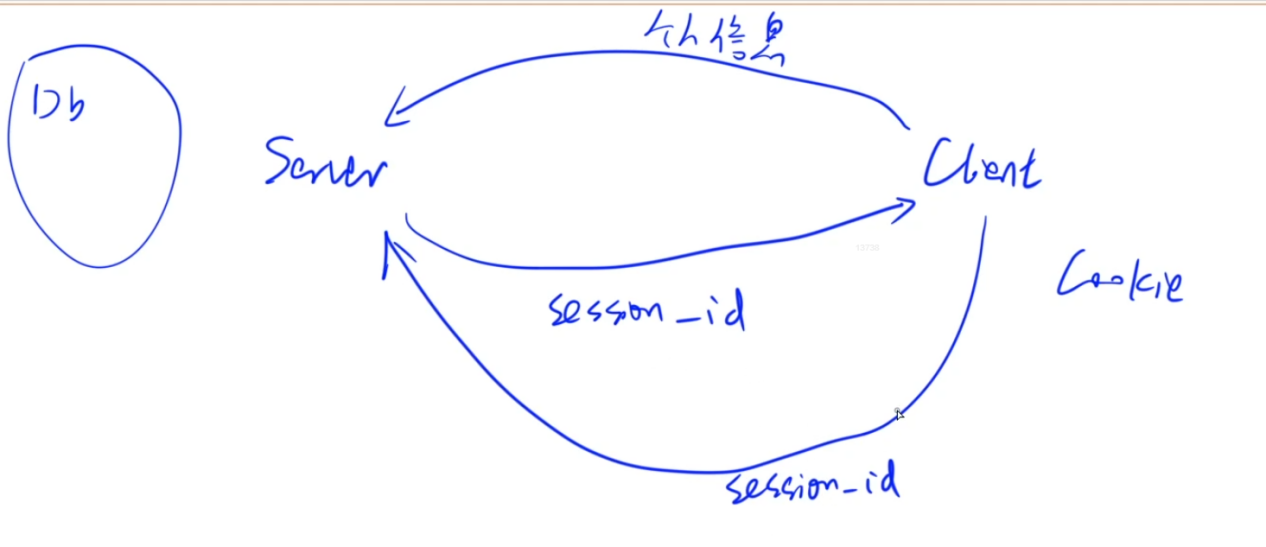
创建页表：

python manage.py makemigrations Monitor

迁移创建数据库：

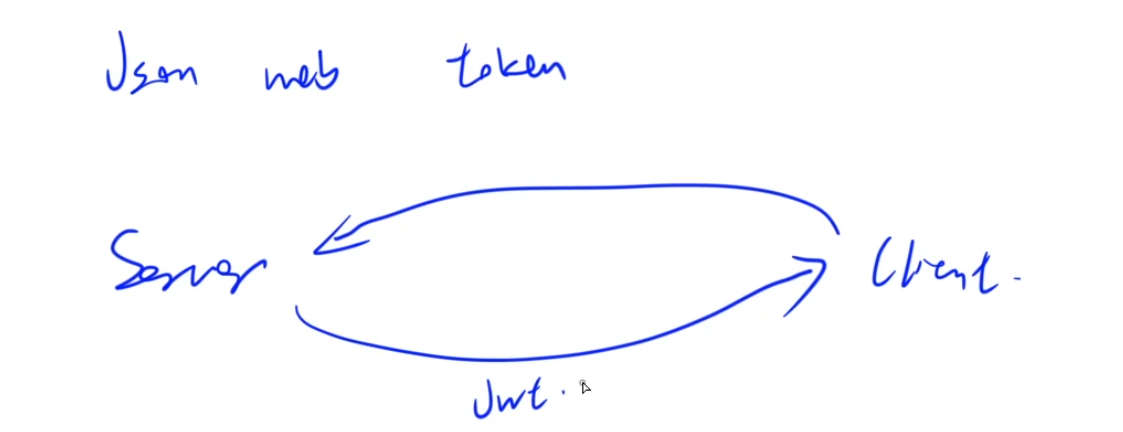
python manage.py migrate

Django用户验证



用户使用个人信息创建，Django会在后台创建usr - session\_id 对应表，其中标明了session\_id的创建时间，django后台向用户发送session-id(存入每次请求时使用的cookie内,cookie存入http-only内) 这种验证方式在浏览器中可完成web中完成。(在同一个域名中可以完成 websocket 以及 http的验证), 弊端js或者ts无法读取cookie中的session-id.

在跨域的情况下,cookie不容易把session\_id带上。换个地方存储session-id即可完成跨域验证。



Docker 部分：

查看镜像：

docker images

查看容器

docker ps(ps -a 查看全部，包括停止运行的)

激活容器

docker start docker\_ name

镜像-》容器

docker run -p 20000:22 -p 8000:8000 -p 80:80 -p 443:443 --name CONTAINER\_NAME -itd 镜像name

容器-》镜像

docker commit 镜像名称

删除容器

docker rm 容器名称

配置redis(docker)

进入配置好的redis

sudo docker exec -it my-redis redis-cli -a your\_password

查看docker 网络

sudo docker network inspect django-redis-net

查看log

sudo docker logs my-redis

使用docker网络创建容器

sudo docker run -d --name my-redis --network django-redis-net -p 6379:6379 swr.cn-east-3.myhuaweicloud.com/library/redis:latest redis-server --requirepass "your\_password" --appendonly yes

创建网络

docker network create django-redis-net

下载华为云的redis docker镜像

sudo docker pull swr.cn-east-3.myhuaweicloud.com/library/redis:latest

重启docker

sudo systemctl restart docker

关闭 docker

sudo systemctl daemon-reload

Daphne 部分 实现websocket 通信

daphne -b 0.0.0.0 -p 5015 Aurage.asgi:application

Redis 部分

哨兵模式

Nginx：

开启服务

sudo /etc/init.d/nginx start

ssl文件位置：

将**acapp.key**中的内容写入服务器**/etc/nginx/cert/acapp.key**文件中。  
将**acapp.pem**中的内容写入服务器**/etc/nginx/cert/acapp.pem**文件中。

uwsgi部分

[uwsgi]

socket = 127.0.0.1:8000

chdir = /home/sdz/project/hyld

wsgi-file = hyld/wsgi.py

master = true

processes = 2

threads = 5

vacuum = true

uwsgi --ini scripts/uwsgi.ini

树莓派Debian

nano 编辑

Ctrl + o保存

Ctrl + x 退出