

1. 安装git

ssh-keygen -t rsa -C "haiyan.xu.vip@gmail.com"

2. 安装开发环境

1. 安装Anaconda3

1. 在Environments里面安装python3.9环境

2. clone项目

3. 选择虚拟环境

4. export AR=/usr/bin/ar

执行下这个后，再安装

5. 安装sudo pip install -r requirements.txt

3. 文档地址

1. 禅道（任务分配）：<https://zentao.easycodesource.com/my/>

2. confluence:

3. 规范手册：<http://doc.easycodesource.com/pages/viewpage.action?pageId=65815>

4. docker中启动

1. docker run -d --name mysql -p 3307:3306 -e MYSQL_ROOT_PASSWORD=123456 7b6f3978ca29

5. 启动core:

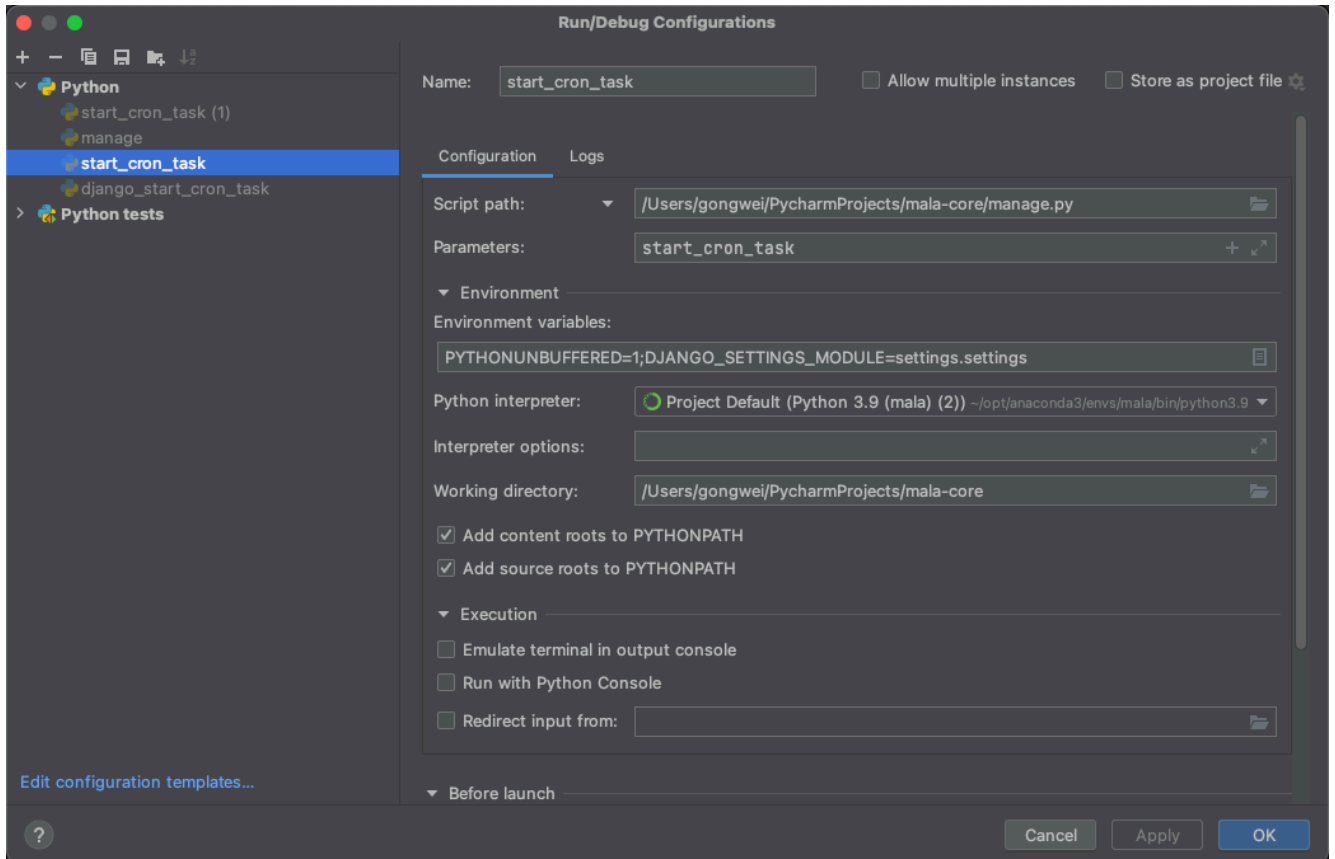
1. python manage.py start_open_server

6. 启动client:

1. python manage.py runserver

7. 游戏链接：<https://game.idn-zenam.com/Cocos-H5-Idn/203/>

8. 运行脚本:



9. Apscheduler:

实例化调度器

```
SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://root:123456@127.0.0.1:3307/test_mala'

jobstores = {
    'mysql': SQLAlchemyJobStore(url=SQLALCHEMY_DATABASE_URI),
    'default': MemoryJobStore()
}

executors = {
    'default': ThreadPoolExecutor(20),
    'processpool': ProcessPoolExecutor(5)
}
scheduler = BackgroundScheduler(jobstores=jobstores, executors=executors)

scheduler.add_job(test_4, trigger="interval", seconds=3, jobstore="mysql")
```

10. Django-apscheduler

```
# 实例化调度器
SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://root:123456@127.0.0.1:3307/test_mala'

executors = {
    'default': ThreadPoolExecutor(20),
    'processpool': ProcessPoolExecutor(5)
}

scheduler = BackgroundScheduler(executors=executors)
scheduler.add_jobstore(DjangoJobStore(), "default")
```

11. 开发环境配置

```
redis:
# url: "redis://127.0.0.1:6379/"
url: "redis://middleware-2.easycodedev.com:6379/"

mongo:
# url: "mongodb://127.0.0.1:27017/"
url: "mongodb://middleware-2.easycodedev.com:27017/"
```

12. 芯片错误

```
http://events.jianshu.io/p/7986fbd82884
```

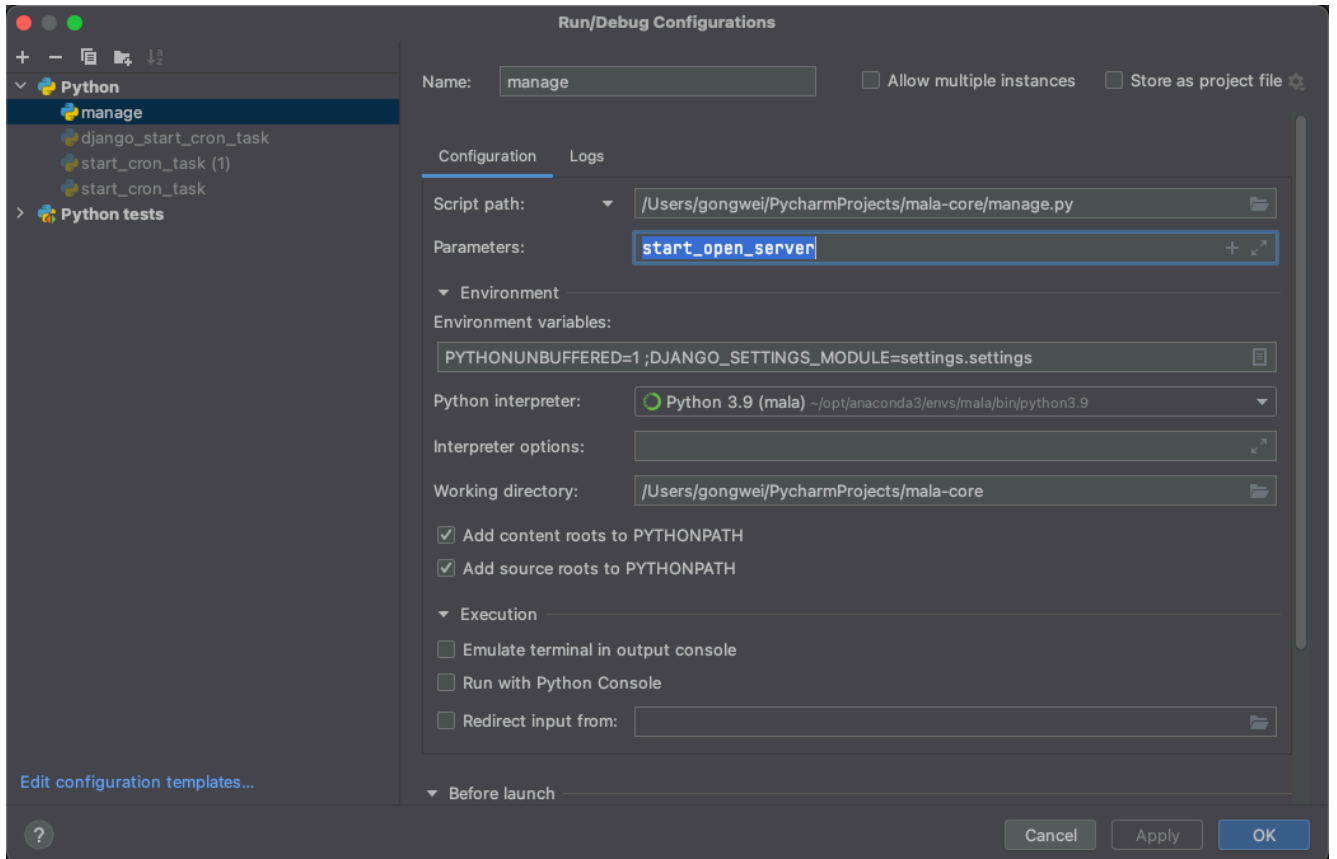
13. Enter_limit_range

client: https://git.easycodesource.com/game-portal/mala-client/-/merge_requests/194/diffs

Idl: https://git.easycodesource.com/game-portal/mala-idl/-/merge_requests/111/diffs

core: https://git.easycodesource.com/game-portal/mala-core/-/merge_requests/492/diffs

14. mala-core运行



15. Mala-client生成protoc文件

```
pip install protobuf
```

需要切换到

proto文件目录，terminal打开

运行该语句，会生成proto文件

```
protoc --python_out=. api.proto
```

保存到hello文件夹下

```
protoc --python_out=./hello --proto_path=api.proto
```

```
protoc fruit.proto --python_out=.
```

```
protoc -I=$SRC_DIR --python_out=$DST_DIR  
$SRC_DIR/addressbook.proto
```

命令解释：

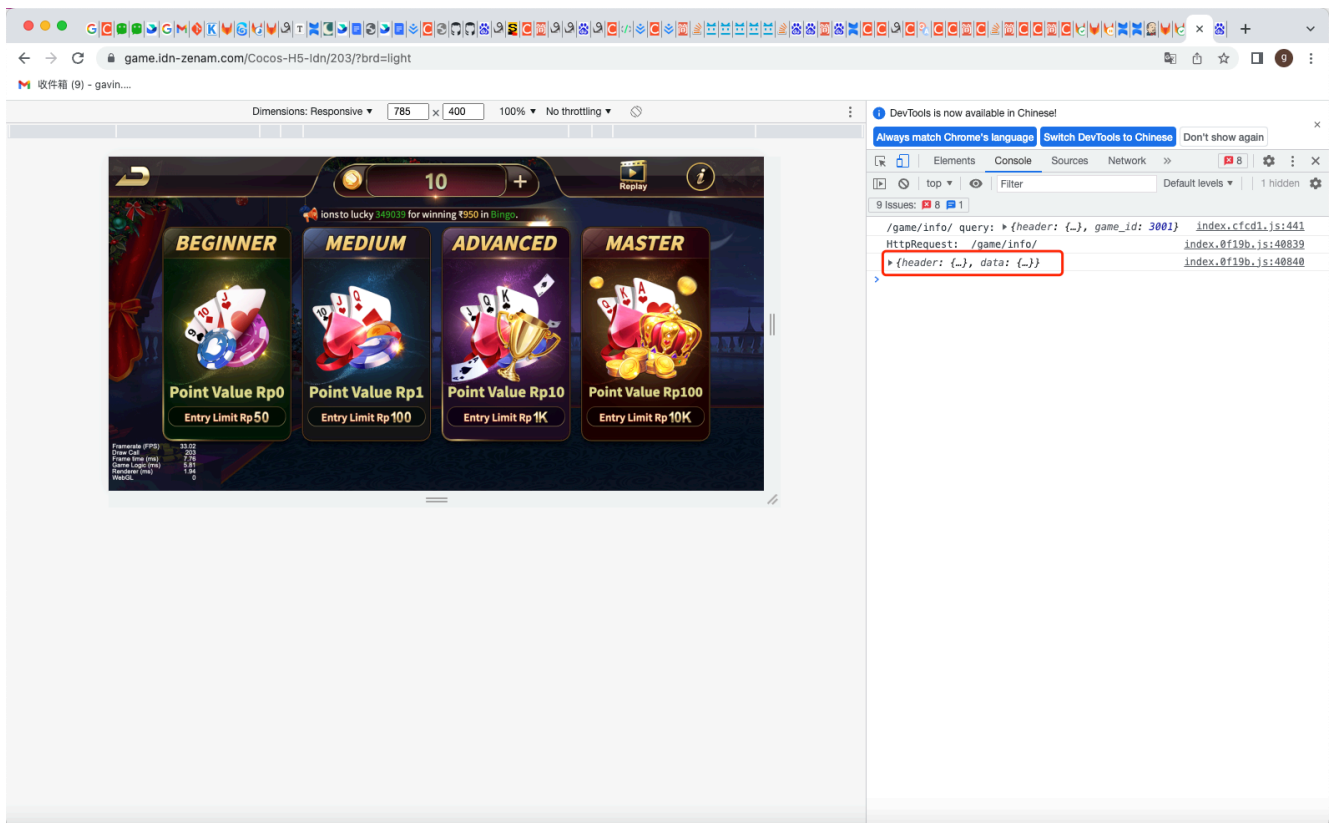
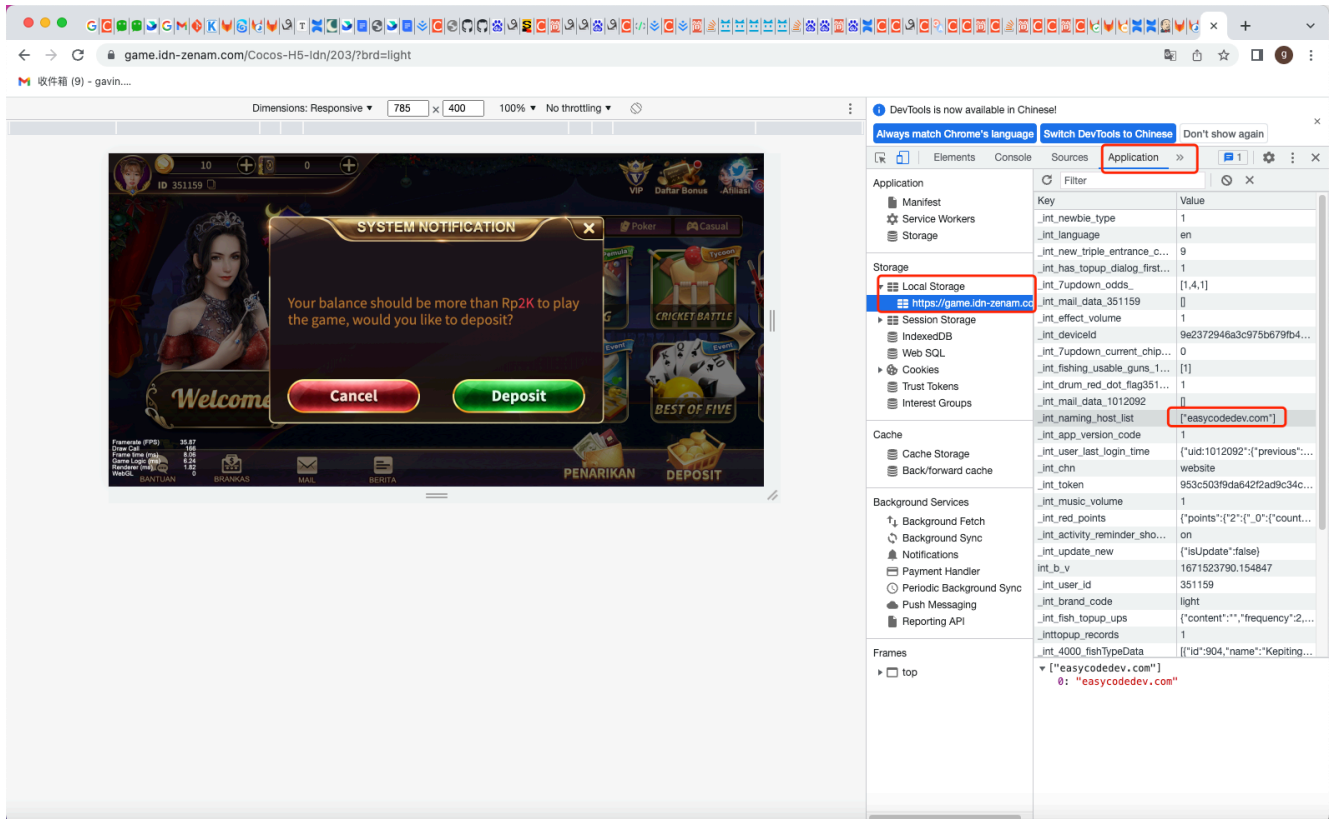
--python_out用于指定生成Python文件要存放的路径，.指当前路径，随后紧接demo.proto文件路径（注意有空格做间隔）

编译多个proto文件

```
$ mkdir out
```

```
$ protoc --cpp_out=out helloworld.proto route_guide.proto
```

16.



17. 修改hosts

```
sudo vim /etc/hosts
```

18. 巴西运营:

<https://operation.brazil-zenam.com/#/user/information>

dev运营:

<https://operation.easycodedev.com/#/user/information>

19. redis-cli -h middleware-2.easycodedev.com -p 6379

20. redis-cli -h middleware-india.zenam.net -p 6379

```
hset mala:game:config 2004 "[{"id":0,"enterLimit":0,"chips":
[10000,50000,100000,500000,1000000,5000000],"betLimit":100000,"redLimit":
[10000,10000000000],"tableId":1,"enterLimitRange":[1000,2000]}]"
```

```
"[{"id":0,"enterLimit":0,"chips":
[10000,50000,100000,500000,1000000,5000000],"betLimit":100000,"redLimit":
[10000,10000000000],"tableId":1,"enterLimitRange":[1000,2000]}]"
```

21. 发布线上

<https://muse-devops.atlassian.net/browse/INT-363>

magicaper9528@gmail.com

_8Zun+w?rUiWUKA

22. 简单代码改动 (可以学习)

https://git.easycodesource.com/game-portal/mala-client/-/merge_requests/199

23. 昨日工作:

1. apscheduler添加分布式锁, 实现分布式部署 (100%)
2. 解决一个线上问题, 用户查询vip等级显示错误问题, (100%)

今日工作:

1. 测试enter_limit_range需求

24. core代码学习(ALin)

```
https://git.easycodesource.com/game-portal/mala-core/-/merge_requests/510
```

```
https://git.easycodesource.com/game-portal/mala-client/-/merge_requests/200
```

```
https://git.easycodesource.com/game-portal/mala-client/-/merge_requests/201
```

25. core 修复award为空

```
https://git.easycodesource.com/game-portal/mala-core/-/merge_requests/507/diffs
```

26. pre-commit install --allow-missing-config

gavin.g@binarywalk.com

27. 在 git 存储库中仍然安装了 pre-commit 钩子。只需删除存储库中的 .git/hooks/pre-commit 即可删除它们，之后提交时将不再调用 pre-commit。

https://api-op.easycodedev.com/activity/online_bonus_conf/?page_size=20&page_no=1

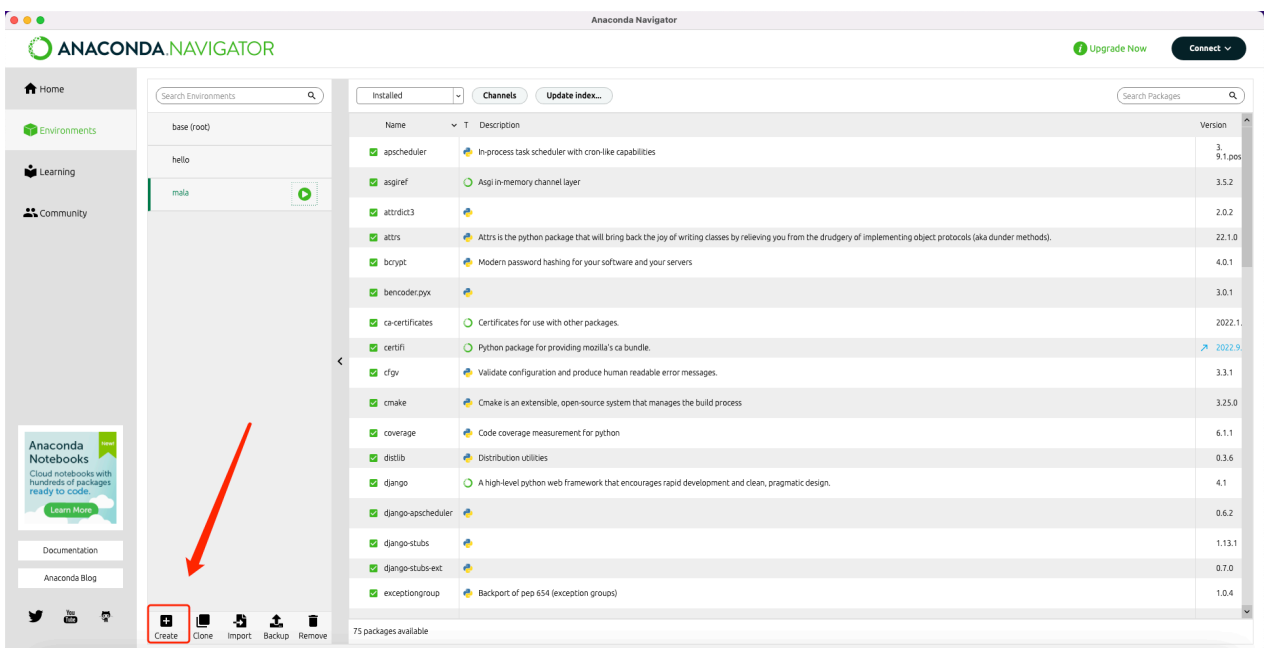
https://api-op.easycodedev.com/activity/online_raffle/conf/?status=0&order_by=-id&page_size=20&page_no=1

https://api-op.easycodedev.com/activity/online_raffle/conf/?status=2&order_by=-id&page_size=20&page_no=1

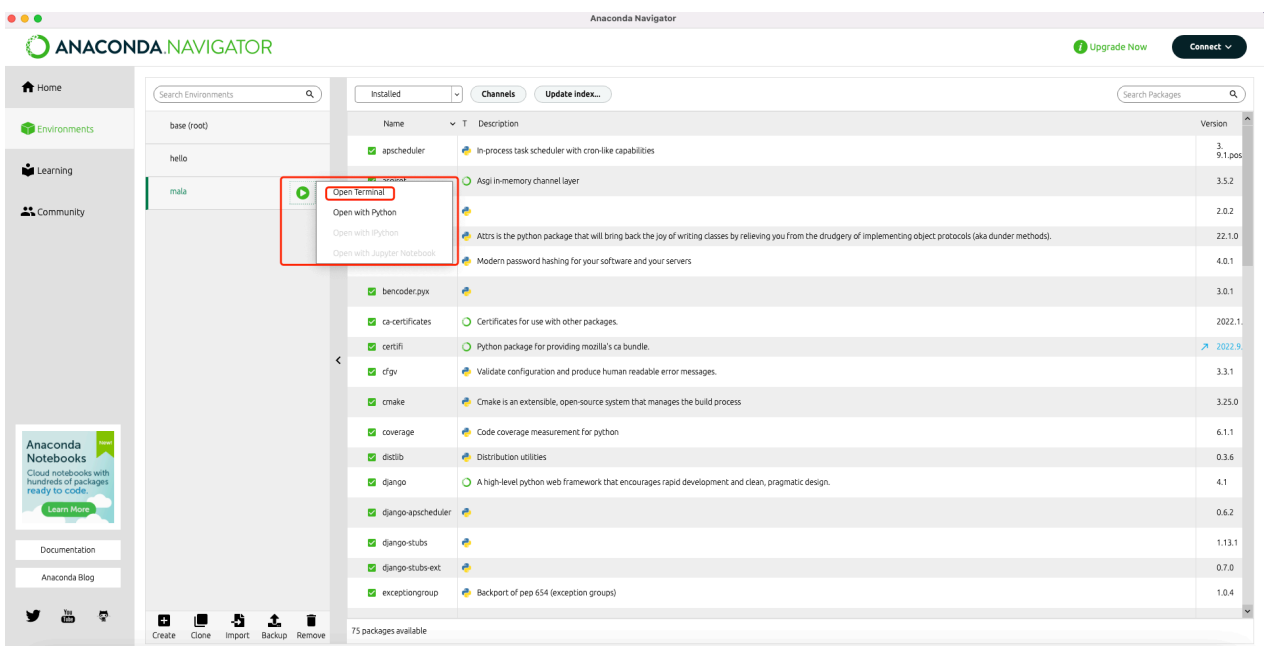
https://api-op.easycodedev.com/activity/online_raffle/record/?status=2&order_by=-id&page_size=20&page_no=1&export=true

28. 创建虚拟环境

1.



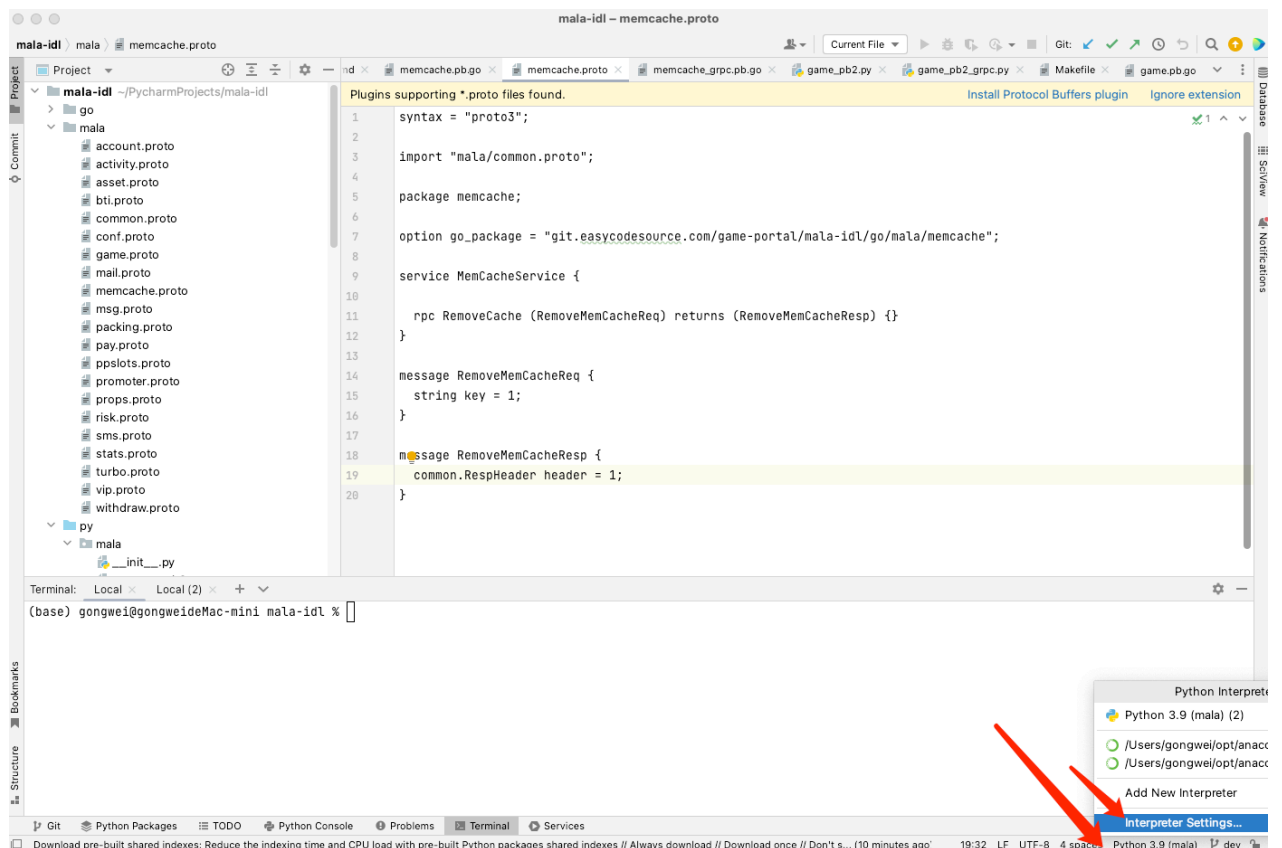
2.



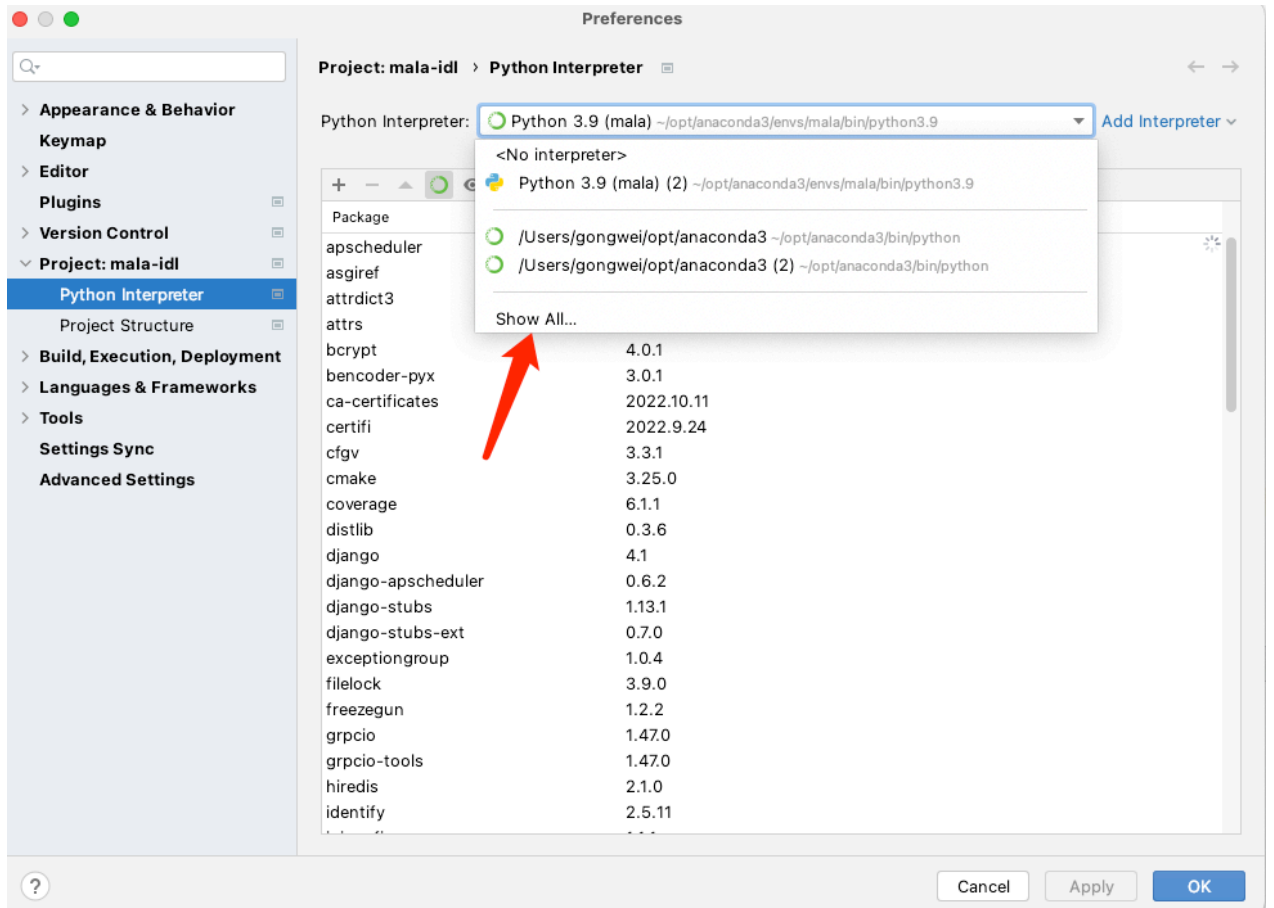
3.

```
gongwei — -zsh — 80x24
Last login: Thu Dec 29 15:27:19 on ttys013
. /Users/gongwei/opt/anaconda3/bin/activate && conda activate /Users/gongwei/opt/anaconda3/envs/mala;
(base) gongwei@gongweideMac-mini ~ % . /Users/gongwei/opt/anaconda3/bin/activate
&& conda activate /Users/gongwei/opt/anaconda3/envs/mala;
(mala) gongwei@gongweideMac-mini ~ %
```

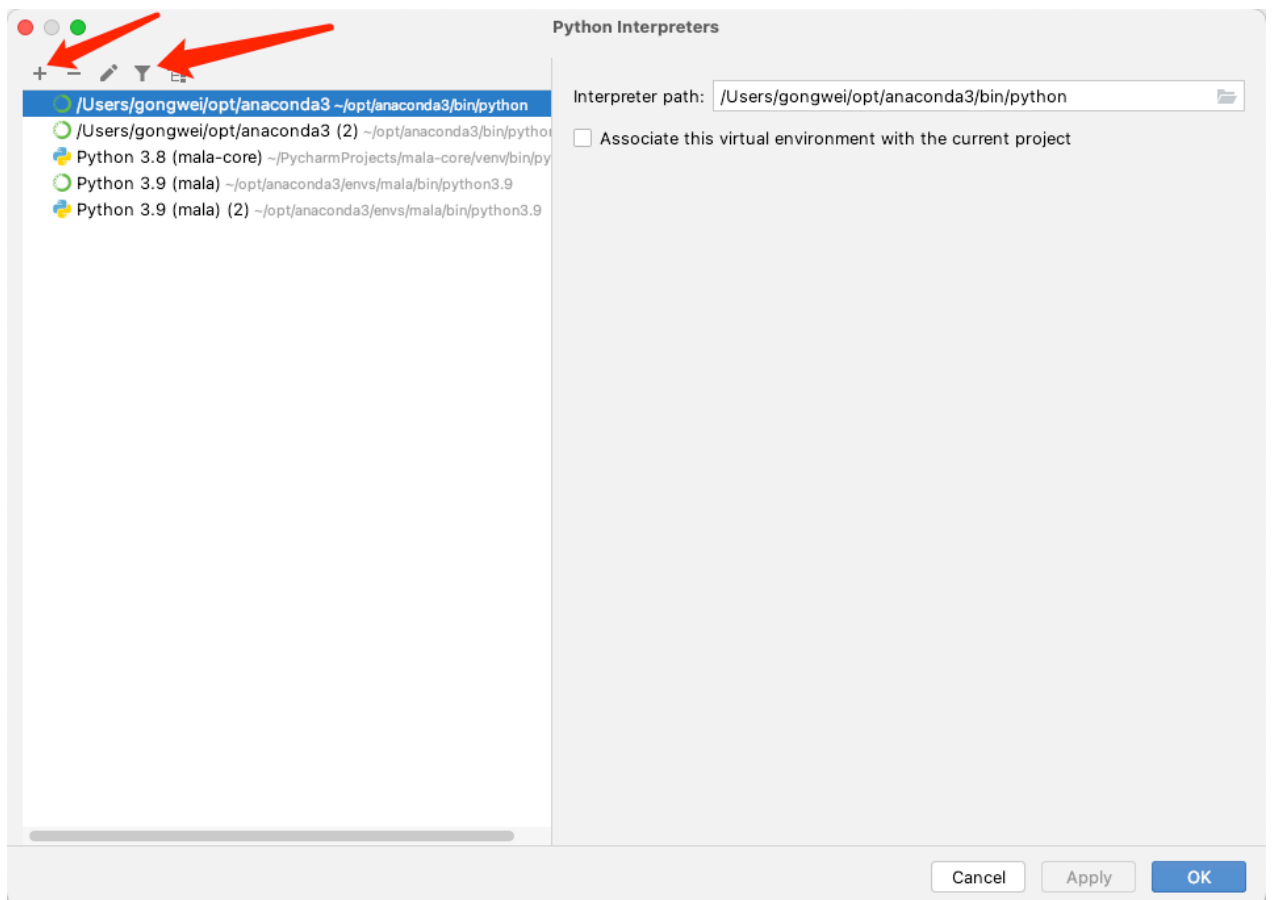
4.



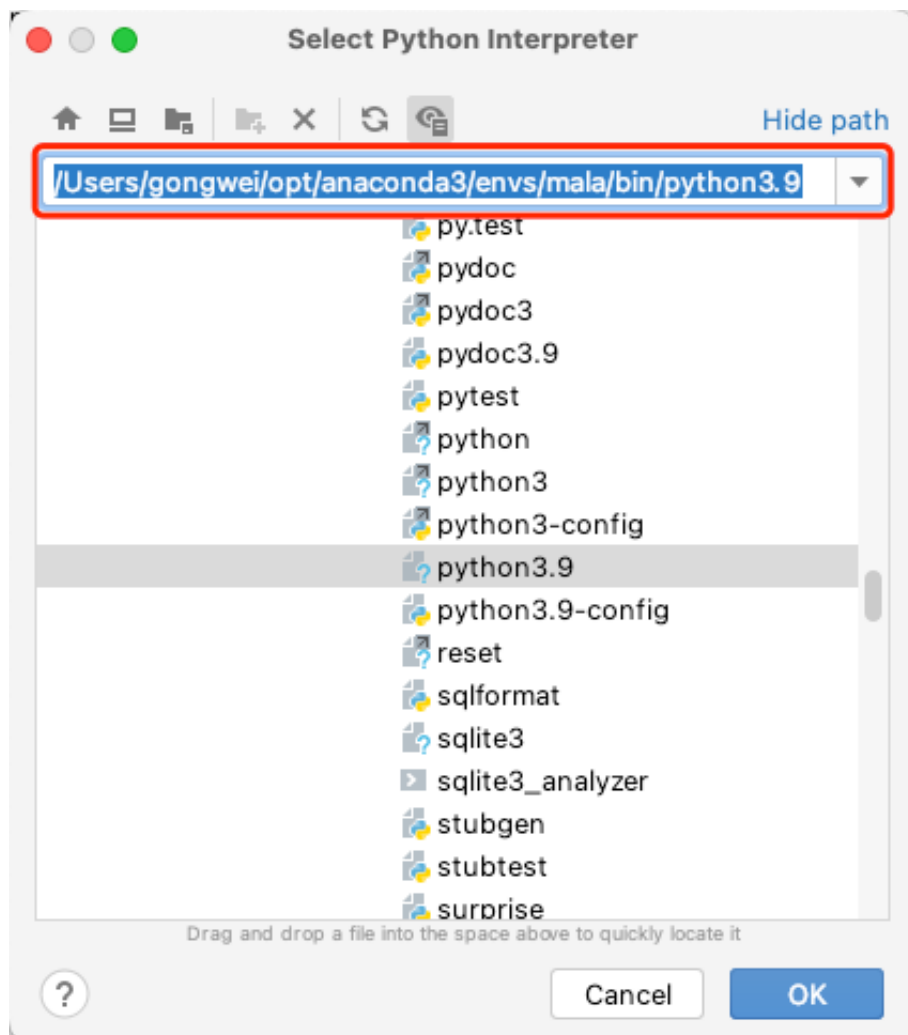
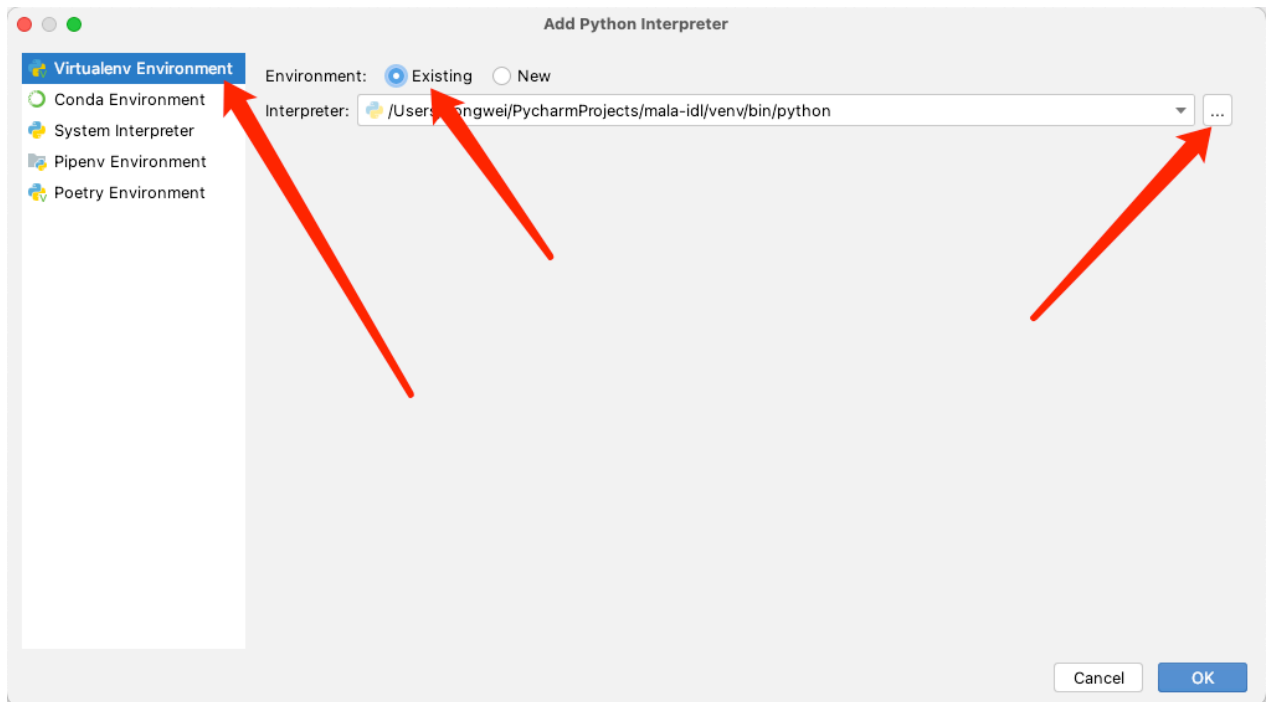
5.



6.



7.



8.

mala-client: https://git.easycodesource.com/game-portal/mala-client/-/merge_requests/207

Mala-core: https://git.easycodesource.com/game-portal/mala-core/-/merge_requests/520

Mala-op: https://git.easycodesource.com/game-portal/mala-op/-/merge_requests/396

29. 周报地址: <http://doc.easycodesource.com/pages/viewpage.action?pageId=5120058>

30. 修改mala-client项目后, 需要修改api项目里面的protoc文件
api/http/hall/api.proto文件

31. mala-client和mala-core进行通信, 需要配置

1. 添加域名到ip的映射

```
sudo vim /etc/hosts
```

添加如下配置:

```
127.0.0.1      mala-core-grpc
```

2. mala-client里面RpcCaller需要配置MALA_RPC_HOST

```
class RpcCaller:
```

```
    def __init__(self):
        self._channel = insecure_channel(
            settings.MALA_RPC_HOST,
            settings.MALA_RPC_AUTH_TOKEN,
        )
```

mala-client里面RpcCaller有如上配置, 去连接

```
        settings.MALA_RPC_HOST,
        settings.MALA_RPC_AUTH_TOKEN,
```

3. mala-client里面需要配置grpc对应的host和端口

mala-client项目env.yaml文件里面有如下配置:

```
mala_grpc_server:
```

```
    host: "mala-core-grpc:50051"
    token: "36E81134-79EF-478F-844E-0580C3D9D8EE"
```

4. mala-core项目也需要配置grpc服务启动的host和端口

mala-core项目env.yaml文件里面有如下配置:

```
grpc_server:
```

```
    host: "0.0.0.0:50051"
    token: "36E81134-79EF-478F-844E-0580C3D9D8EE"
```

mala-core项目settings文件里面有如下配置:

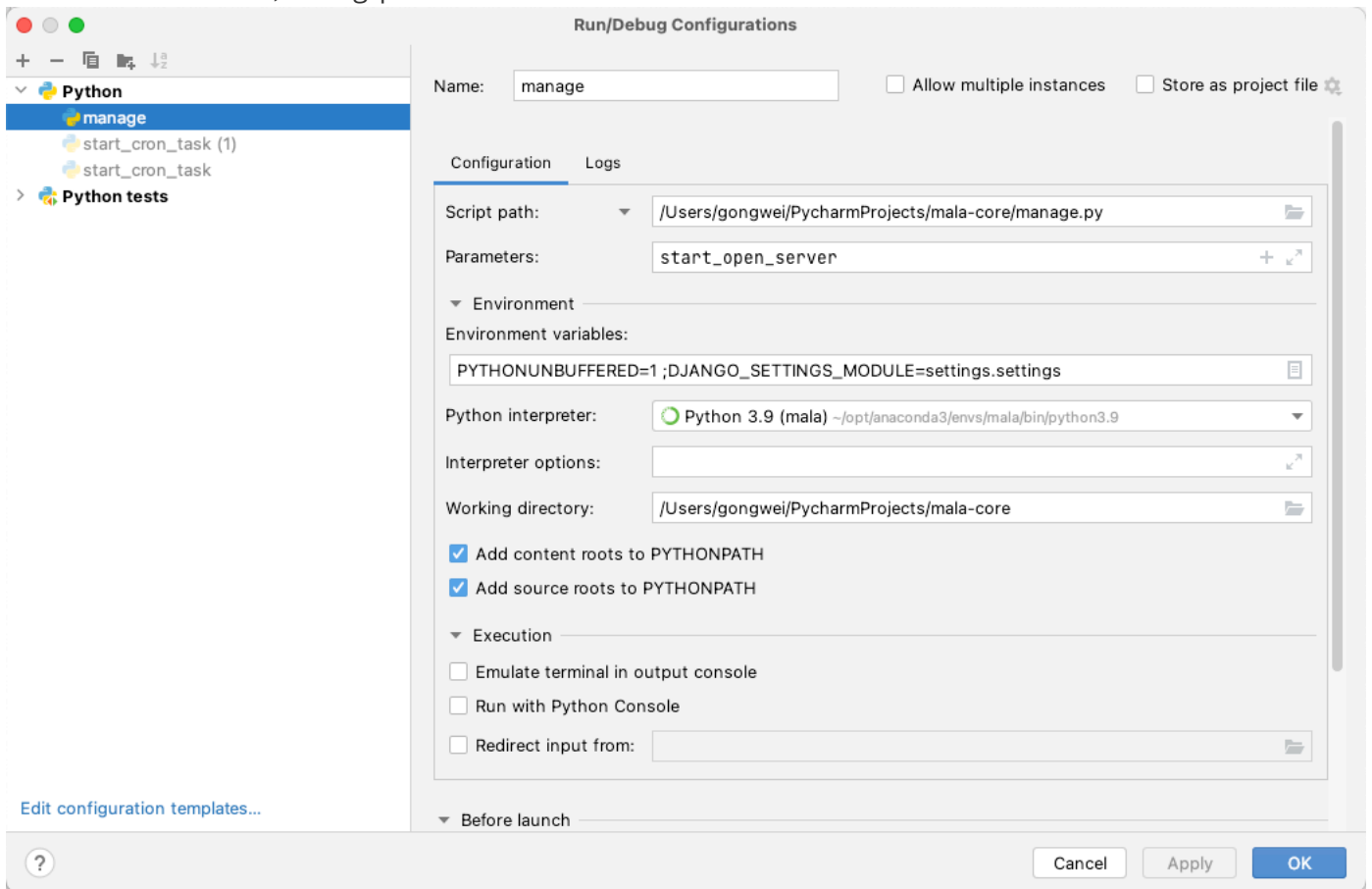
```
GRPC_SERVER_LISTEN_HOST = ENV["grpc_server"]["host"]
GRPC_SERVER_TOKEN = ENV["grpc_server"]["token"]
```

然后mala-core项目下有一个脚本, 名字叫做start_open_server

所以mala-core项目启动时候, 就是启动grpc服务, 使用如下命令

```
python manage.py start_open_server
```

在mala-core项目，启动grpc服务



32. 从工程目录下获取env.yaml文件

```
"""
从工程根目录下读取 env.yaml 文件内容并解析成 ENV 变量。
"""
import os
import yaml
from yaml import Loader

_env_file = open(os.path.join(os.getcwd(), "env.yaml"), "r")
ENV = yaml.load(_env_file, Loader)
_env_file.close()
```

33. protobuf中使用default和enum

```
message SearchRequest {
    required string query = 1;
    optional int32 page_number = 2;
    optional int32 result_per_page = 3 [default = 10];
    enum Corpus {
        UNIVERSAL = 0;
```

```

    WEB = 1;
    IMAGES = 2;
    LOCAL = 3;
    NEWS = 4;
    PRODUCTS = 5;
    VIDEO = 6;
}
optional Corpus corpus = 4 [default = UNIVERSAL];
}

```

34. proto等说明

ProtoBuf是一种序列化数据结构的协议

protoc是一个能把proto数据结构转换为各种语言代码的工具

RPC是一种通信协议

gRPC是一种使用ProtoBuf作为接口描述语言的一个RPC实现方案

35. 路径含义

"./": 代表目前所在的目录。

" ../"代表上一层目录。

" /": 代表根目录。

36. 获取项目的根目录

方案一:

```
root_path = os.getcwd()
```

方案二:

```
cur_path = os.path.abspath(os.path.dirname(__file__))
```

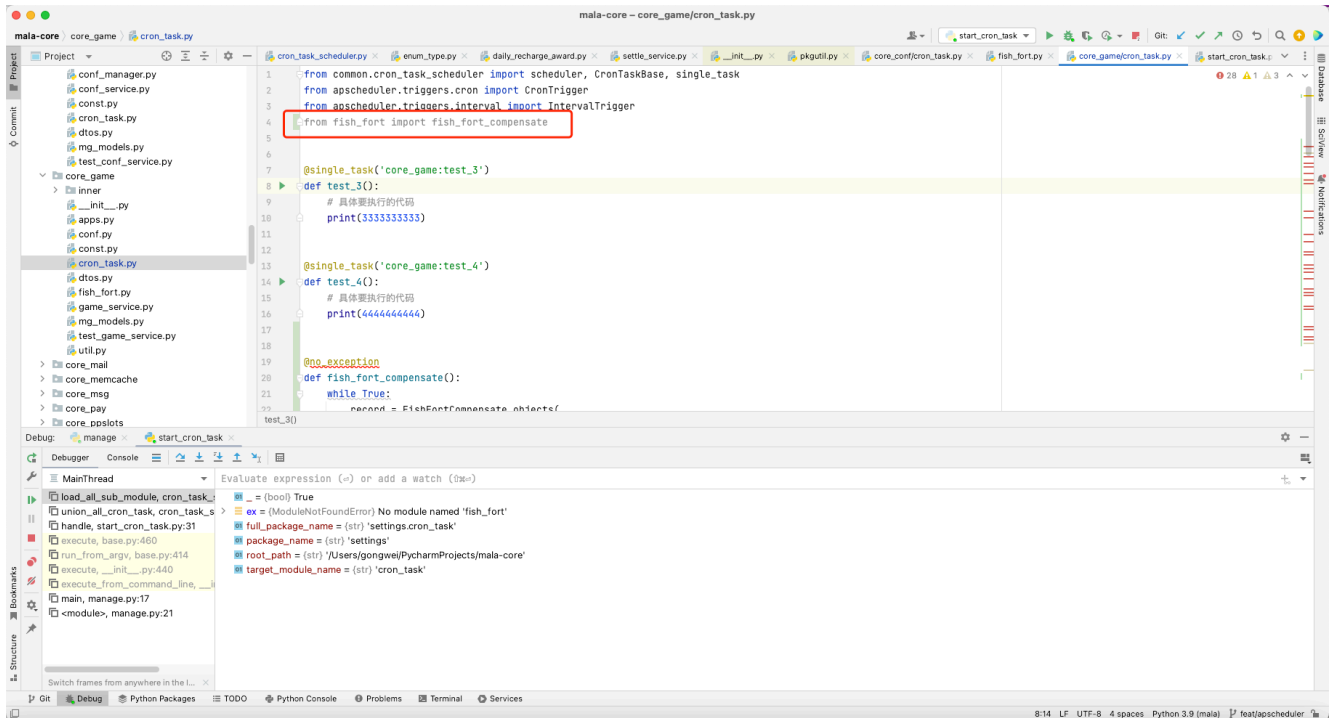
```
root_path = cur_path + "../"
```

手动调整目录

37. 获取当前文件的上级目录

```
path = os.path.dirname(__file__)
```

38. 在core_game/core_task中添加导入，在scheduler里面不能正常导入



解决办法: from .fish_fort import fish_fort_compensate

39. 两层装饰器和三层装饰器区别

两层装饰器: 没有参数

三层装饰器: 传递参数

两层装饰器:

```
from functools import wraps

def add_log(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        res = func(*args, **kwargs)
        return res
    return wrapper
```

三层装饰器:

```
from functools import wraps

def single_task(task):
    def wrapper(func):
        @wraps(func)
        def inner(*args, **kwargs):
            add_result = add_lock(task)
            if add_result:
                _LOGGER.info("当前节点获取任务成功, task_name:{}".format(task))
            try:
                result = func(*args, **kwargs)
            except Exception as e:
                _LOGGER.error("任务执行失败, task_name:{}".format(task), exc_info=e)
            return result
        return inner
    return wrapper
```

```

        _LOGGER.info("task_name:{}执行成功, 执行时间:{}".format(task,
dt.local_now()))
        return result
    except Exception as e:
        raise e
    finally:
        delete_lock(task)
    else:
        _LOGGER.info("当前节点未获取任务, task_name:{}".format(task))
        return
    return inner
return wrapper

```

40. 在Python中, 定义和使用装饰器有先后顺序, 必须先定义后使用

```

def add_log(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        print("哈哈")
        res = func(*args, **kwargs)
        return res
    return wrapper

@add_log
def add_job(*args, **kwargs):
    scheduler.add_job(*args, **kwargs)
    joker_bonus_amount

```

41. Python打印调用栈

```

import traceback

try:
    result = func(*args, **kwargs)
    return result
except Exception as ex:
    traceback.print_exc()

```

42. 日志平台

```

https://kibana-logs.easycodesource.com/app/discover#/?_a=(columns:!(
(message),filters:!(t,hideChart:!(t,index:fbbb3ce0-2cdd-11ed-b9a5-
5dfa4c4d40eb,interval:auto,query:(language:kuery,query:''),sort:!(
('@timestamp',desc)))&_g=(filters:!(t,refreshInterval:(pause:!(t,value:0),time:
(from:now-3d,to:now))

```

43. docker相关问题

1. failed to solve with frontend dockerfile.v0: failed to create LLB definition: no build stage in current context

问题及解决方案：在DockerFile中，第一行必须是FROM xxx，否则就会出现上述报错。

2. Dockerfile文件名一个字都不能错，也没有后缀

44. i18n, 翻译网址

<https://docs.google.com/spreadsheets/d/1A-2M9XfMnqXVnfGzy0Vcq9FSc6LfspW/edit#gid=1017811402>

45. 临时保存url

https://blog.csdn.net/weixin_30230009/article/details/125401456

<https://www.baidu.com/s?ie=UTF-8&wd=%E5%89%8D%E7%AB%AF%E6%80%8E%E4%B9%88%E6%A0%B9%E6%8D%AEproto%E7%BC%96%E8%AF%91%E5%8F%91%E9%80%81%E6%95%B0%E6%8D%AE>

<https://www.cnblogs.com/vickylinj/p/16599236.html>

https://blog.csdn.net/qg_36587420/article/details/105443206

https://blog.csdn.net/qg_22918243/article/details/126827997