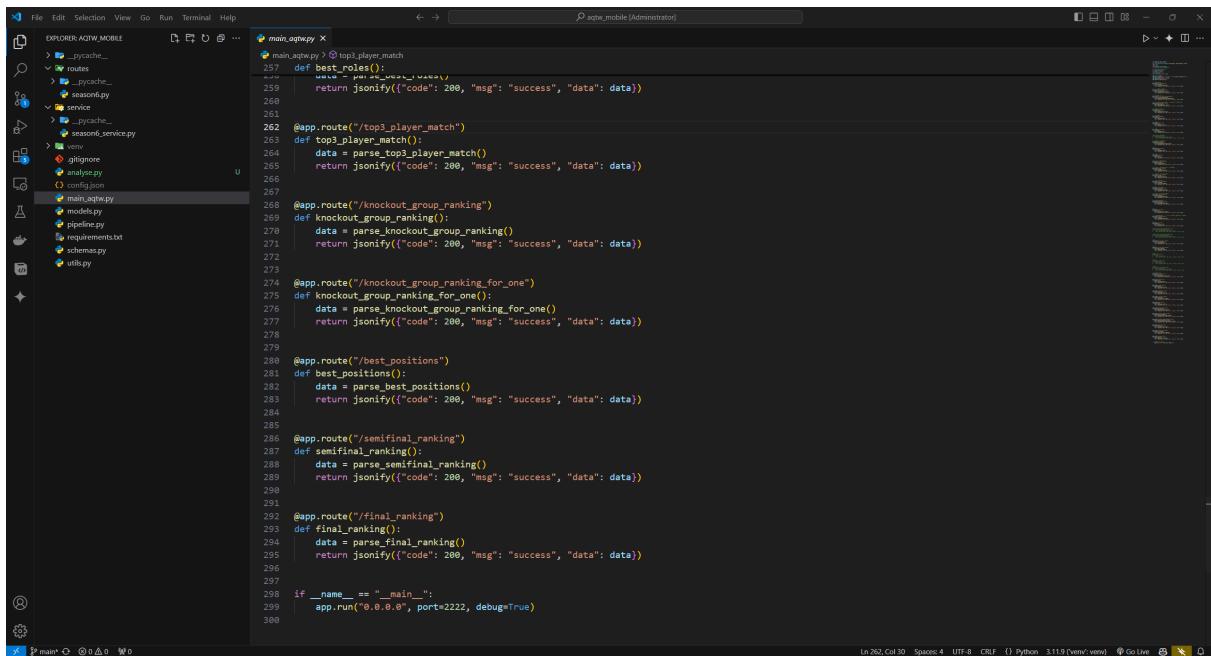


暗区突圍数据执行手册

1. 程序介绍

1.1 aqtw_mobile



```
File Edit Selection View Go Run Terminal Help
EXPLORER: AQTW_MOBILE
main_aqtw.py x
main_aqtw.py > @app.route('/')
def best_player():
    www = parse_best_player()
    return jsonify({'code': 200, "msg": "success", "data": data})
259
260
261
262 @app.route('/top3_player_match')
263     def top3_player_match():
264         www = parse_top3_player_match()
265         return jsonify({'code': 200, "msg": "success", "data": data})
266
267
268 @app.route('/knockout_group_ranking')
269     def knockout_group_ranking():
270         www = parse_knockout_group_ranking()
271         return jsonify({'code': 200, "msg": "success", "data": data})
272
273
274 @app.route('/knockout_group_ranking_for_one')
275     def knockout_group_ranking_for_one():
276         www = parse_knockout_group_ranking_for_one()
277         return jsonify({'code': 200, "msg": "success", "data": data})
278
279
280 @app.route('/best_positions')
281     def best_positions():
282         www = parse_best_positions()
283         return jsonify({'code': 200, "msg": "success", "data": data})
284
285
286 @app.route('/semifinal_ranking')
287     def semifinal_ranking():
288         www = parse_semifinal_ranking()
289         return jsonify({'code': 200, "msg": "success", "data": data})
290
291
292 @app.route('/final_ranking')
293     def final_ranking():
294         www = parse_final_ranking()
295         return jsonify({'code': 200, "msg": "success", "data": data})
296
297
298 if __name__ == "__main__":
299     app.run("0.0.0.0", port=2222, debug=True)
300
```

程序功能：在执行时，用于获取带入物资和配装数据

主程序文件: main_aqtw.py

1.2 aqtw_fetch

```
aqtw2025.py
1 import json
2 from datetime import datetime
3 import requests
4 from sqlalchemy.orm.exc import NoResultFound
5 from sqlalchemy.exc import IntegrityError
6 from models import *
7 from daijn import *
8
9 Session = sessionmaker(bind=ENGINELocal, autocommit=False)
10 session = Session()
11
12 # 读取config.txt中roomId, round, course
13 roomId = ""
14 battle_round = ""
15 battle_course = ""
16 battle_remark = ""
17
18 gamenum = ""
19
20 with open("config.txt", mode="r", encoding="utf-8") as f:
21     line = f.readline()
22     while line:
23         key, value = line.split(":")
24         if key == "roomId":
25             roomId = value.strip()
26         if key == "BATTLE_ROUND":
27             battle_round = value.strip()
28         if key == "BATTLE_COURSE":
29             battle_course = value.strip()
30         if key == "BATTLE_REMARK":
31             battle_remark = value.strip()
32         if key == "gameNum":
33             gamenum = value.strip()
34     line = f.readline()
35 print(roomId, battle_course, battle_round)
36
37 # 全局变量
38 # 已经判断的event数量
39 incre_event_num = (
40     session.query(DynaKillInfo).filter_by(BATTLE_ID=roomId).count()
41     + session.query(DynaItemPickUp).filter_by(BATTLE_ID=roomId).count()
42 )
43 boss_token_first_flag = 0 # 首次拾取boss怪物的人
44 global_player = []
45
46 # 读取选手信息
47 player_data = session.query(BasePlayer.ID, BasePlayer.NAME).filter().all()
```

程序功能：在比赛进行中，持续获取游戏数据，并存入数据库，为API提供实时数据

主程序文件: aqtw2025.py

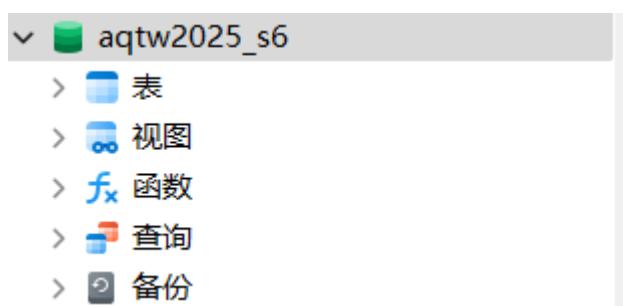
1.3 aqtw_ex

```
handle_after_game_919.py > ...
1  from pipeline import parsepostonematch_data_919
2  from models import *
3  import json
4
5  SessionLocal = sessionmaker(bind=ENGINELocal, autocommit=False)
6  session = SessionLocal()
7
8  MATCH_DAY = ["14"]| TAB to jump here
9
10 battle_today: list = [
11     item[0]
12     for item in session.query(BaseBattle.ID)
13     .filter(BaseBattle.BATTLE_REMARK.in_(MATCH_DAY))
14     .all()
15 ]
16 for battle in battle_today:
17     res = parsepostonematch_data_919(battle)
18
19     # 如果要保存json用以下逻辑
20     # with open(f"game_{battle}.json", "w", encoding="utf-8") as f:
21     #     json.dump(res["req_payload"], f, ensure_ascii=False, indent=4)
22     #     print(f"Game{battle}信息写入完毕!")
23
24     if res["status"] == "success":
25         print(f"数据传输成功, 收到的服务器返回是: {res['data']}"))
26     else:
27         print(f"数据传输失败, 收到的服务器返回是: {res['data']}"))
28
```

程序功能：赛后为游戏平台提供赛后数据

主程序文件: handle_after_game_919.py

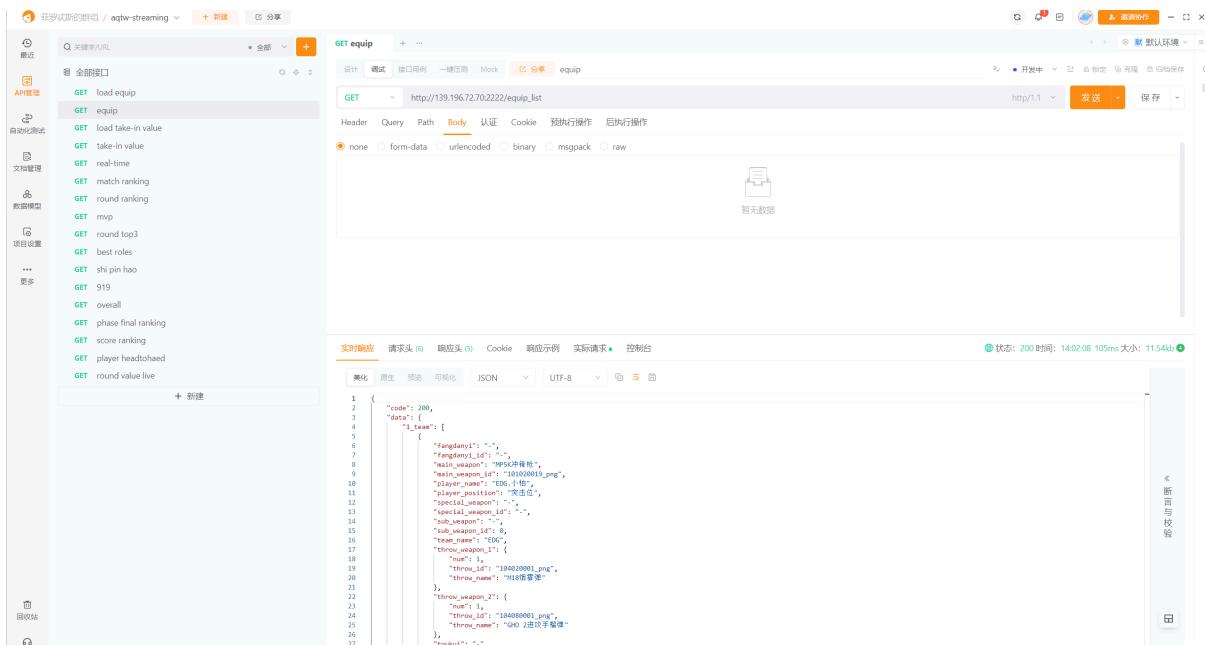
1.4 aqt2025_s6数据库



程序功能：配置赛前选手信息

程序入口：Navicat数据库管理软件(会和其他程序一同打包提供)

1.5 APIPOST API查询程序



程序功能：赛中加载配装、带入物资（需在开启aqtw_mobile后使用）

程序入口：aqtw_streaming目录(会和其他程序一同打包提供)

2. 现场操作顺序

2.1 配置赛前选手信息

2.1.1 涉及程序

aqtw2025_s6数据库

2.1.2 操作方法

step 1

双击打开base_player表

数据库结构树显示了以下内容：

- 数据库：aqtw2025_s6
- 表：
 - base_battle
 - base_player
 - data_battle_player
 - data_battle_team
 - data_course_player
 - data_course_team
 - data_round_player
 - data_round_team
 - data_team_score
 - dyna_battle_player
 - dyna_item_pick_up
 - dyna_kill_info
 - item_list
 - para_course
 - para_f.mvp
 - para_mvp
 - para_player
 - para_round
 - para_team
 - player_camp
 - player_pov_selected
 - player_takein_value
- 视图
- 函数
- 查询
- 备份

step 2

一键修改所有选手信息

1.修改所有选手的is_playing=0

ID	NAME	IS_PLAYING	TEAM_ID	TEAM_NAME	PLAYER_POSITION	PLAYER_PINGTAI	ESCAPE_RATIO	BATTLE_COUNT	AVERAGE_DAMAGE	KILL_DEATH_RATIO	MAX_KILL_OF_SINGLE	MAX_DAMAGE_OF_SI	MAX_DIST
		# int(1)	# int(1)	# varchar(100)	# varchar(100)	# varchar(255)	# decimal(10,4)	# int(11)	# int(11)	# decimal(10,4)	# int(11)	# int(11)	# int(11)
10006220913	KM.李阳	0	0	KM	指挥位	抖音游戏	17.1000	41	41643	2.4706	10	279881	
10073522063	TEC.兵兵	0	0	TEC	物资位	抖音游戏	25.6000	293	77851	3.0367	14	416658	
10342387112	XxX.瑞瑞	0	0	XxX	(Null)	算理物理	0.0000	0	0	0.0000	0	0	
10383675287	XxX.阿斯	0	0	XxX	物资位	抖音游戏	33.3000	15	50956	4.7000	8	185657	
10545904835	宜春TJB.少子	1	0	宜春TJB	物资位	抖音游戏	11.6000	310	57001	2.2664	11	288457	
10602224718	TianBa.阿雷	0	0	TianBa	物资位	抖音游戏	23.2000	155	68455	3.2269	11	366641	
10614618908	XHG.脚脚	1	0	XHG	狙击位	抖音游戏	16.9000	308	58303	2.5352	16	518090	
106321927066	XxX.阿斯	1	0	XxX	物资位	算理物理	29.2000	253	59428	3.6480	16	494141	
106233981713	T7.灵	0	0	T7	物资位	抖音游戏	26.4000	87	42578	2.7656	11	195590	
10676748787	XHG.吉他乐	0	0	XHG	物资位	抖音游戏	21.1000	128	34386	2.3663	12	153755	
10695353921	TianBa.白黑	0	0	TianBa	指挥位	抖音游戏	18.0000	133	61818	2.9817	13	378153	
10710567291	宜春TJB.放一	1	0	宜春TJB	狙击位	抖音游戏	11.4000	306	64444	3.2399	18	286457	
10749692479	MU.暖明	1	0	MU	物资位	抖音游戏	29.9000	97	47078	2.4559	8	281638	
10804086800	Ly.五十	0	0	Ly	狙击位	抖音游戏	14.4000	160	47588	2.4088	8	182816	
10808986243	情久.酒渺	0	0	情久	狙击位	抖音游戏	22.9000	48	52390	2.4865	9	421782	
10828712011	MU.摩拉维	1	0	MU	狙击位	抖音游戏	28.6000	28	45364	3.5500	8	161318	
10997458796	T7.白止	0	0	T7	物资位	抖音游戏	13.6000	22	32161	1.6842	5	125150	
11101848834	FIN.森林	0	0	FIN	突击位	抖音游戏	26.8000	220	91223	4.7702	14	576424	
11279347949	FIN.明星	0	0	FIN	指挥位	抖音游戏	27.8000	212	66548	3.9085	12	307181	
11366541913	EDG.小莫	1	0	EDG	指挥位	抖音游戏	22.5000	178	49836	2.8768	12	296364	
11402693461	NOVA.初见	0	0	NOVA	突击位	抖音游戏	19.2000	198	67708	3.6500	15	390783	
11404147552	TianBa.战队	0	0	TianBa	狙击位	抖音游戏	10.0000	10	53289	3.2222	9	150045	
11411081375	LY.TY	0	0	Ly	(Null)	抖音游戏	11.8000	161	39640	2.2676	10	245857	
11520052702	XHG.胡世群	0	0	XHG	(Null)	抖音游戏	0.0000	0	0	0.0000	0	0	
11620645494	XHG.胡震	1	0	XHG	狙击位	抖音游戏	27.8000	216	60870	3.2628	13	271994	
11699171825	宜春TJB.茶cc	1	0	宜春TJB	指挥位	抖音游戏	10.0000	310	52801	2.4337	11	254227	
11704664090	TianBa.十六	0	0	TianBa	突击位	抖音游戏	17.4000	167	42807	2.2391	13	217923	
11727437172	Ly.独狼	0	0	Ly	突击位	抖音游戏	18.4000	179	65057	4.2808	16	541870	
11806766430	狂月.阿神	0	0	狂月	突击位	抖音游戏	25.0000	24	37333	2.4444	9	167461	
11843732500	KM.名分	0	0	KM	指挥位	抖音游戏	12.3000	114	51364	2.7500	13	441567	
11869956430	STE.萧白	1	0	STE	狙击位	抖音游戏	18.8000	186	62951	2.9536	15	403409	
11874116483	XHG.小李	1	0	XHG	突击位	抖音游戏	17.3000	300	63301	3.1694	19	427926	
11995915840	情久.南笙	0	0	情久	物资位	抖音游戏	19.2000	261	78694	4.3649	33	982913	
12004183107	狂月.迪奥	0	0	狂月	指挥位	抖音游戏	17.8000	45	79312	3.1351	14	261841	
12059593454	Ly.美丽	0	0	Ly	指挥位	抖音游戏	18.3000	213	49596	3.1724	15	345225	
1219921294	FIN.莫雷	0	0	FIN	狙击位	抖音游戏	38.0000	200	90721	4.5645	19	755583	
12341476629	STE.白进桃	0	0	STE	突击位	抖音游戏	19.8000	192	57252	2.8312	12	363793	

2.点击勾选(有修改记录会显示黑色可点击样式)

已选择 1 列

第 1 条记录 (共 80 条) 于第 1 页 上次刷新时间: 1m

step 3

按照当日的首发名单，一个个队伍的筛选，并为首发选手更新信息

1.点击筛选

3.选择队伍

4.点击应用筛选

5.为每个首发选手设置is_playing=1

点击勾选确定更改

ID	NAME	IS_PLAYING	TEAM_ID	TEAM_NAME	PLAYER_POSITION	PLAYER_PINGTAI	BATTLE_COUNT	AVERAGE_DAMAGE	KILL_DEATH_RATIO	MAX_KILL_OF_SINGLE	MAX_DAMAGE_OF_SI	MAX_DIST
		# int(1)	# int(1)	# varchar(100)	# varchar(100)	# varchar(255)	# int(11)	# int(11)	# decimal(10,4)	# int(11)	# int(11)	# int(11)
10623988173	T7.灵	0	0	T7	物资位	抖音游戏	26.4000	87	42578	2.7656	11	195590
10997458796	T7.白止	0	0	T7	物资位	抖音游戏	13.6000	22	32161	1.6842	5	125150
8870868646	T7.小白	0	0	T7	指挥位	抖音游戏	31.4000	86	58420	4.1017	11	208643
8913077294	T7.迎上南	0	0	T7	狙击位	抖音游戏	23.3000	86	65135	4.4242	17	366909
9009595111	T7.坡	0	0	T7	突击位	抖音游戏	28.0000	82	51773	3.3390	9	233081

```
SELECT * FROM `aqtw2025_s6`.`base_player` WHERE `TEAM_NAME` LIKE '%T7%' LIMIT 0,1000
```

第 3 条记录 (共 5 条) 于第 1 页 上次刷新时间: 4m

step 4

检查配置是否正确



结果类似下图

team_name	name
EDG	EDG.小黑
EDG	EDG.黑总
EDG	EDG.单东
EDG	EDG.小白
MU	MU.西瓜
MU	MU.黎明
MU	MU.烈焰
MU	MU.烈火
STE	STE.幻灯
STE	STE.幻灯
STE	STE.小新
STE	STE.黑白
STE	STE.白恰恰
XHG	XHG.脚脚
XHG	XHG.小雨
XHG	XHG.小李
XHG	XHG.安度
XXX	XXX.路
XXX	XXX.飘风
XXX	XXX.手大刀
XXX	XXX.阿离断
直营TJB	直营TJB.爱丽丝离
直营TJB	直营TJB.冰冰
直营TJB	直营TJB.欣一
直营TJB	直营TJB.沙子

检查无误后，赛前配置即为成功。

附：

查询代码

```
select team_name, `name` from base_player where IS_PLAYING=1  
ORDER BY TEAM_NAME;
```

2.2 运行aqtw_fetch程序

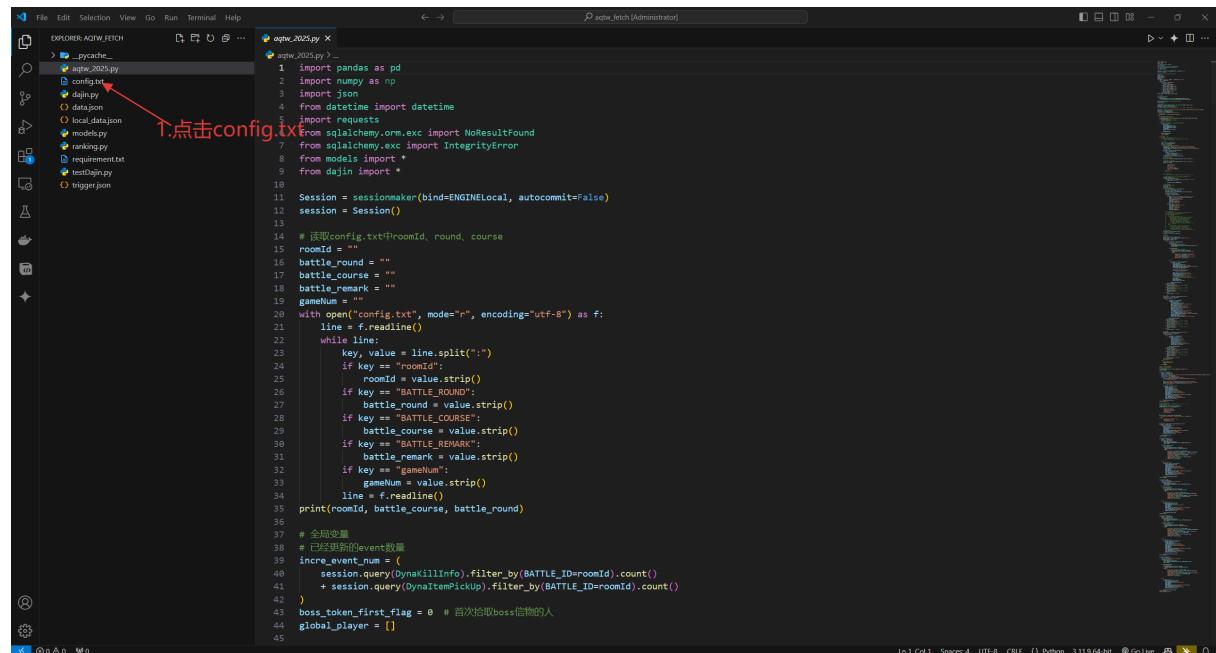
2.2.1 涉及程序

aqtw_fetch

2.2.2 操作方法

step 1

打开config.txt文本文件，配置本场比赛信息



```
aqtw_2025.py > ...  
aqtw_2025.py > ...  
1 import pandas as pd  
2 import numpy as np  
3 import json  
4 from datetime import datetime  
5 import requests  
6 from sqlalchemy.orm.exc import NoResultFound  
7 from sqlalchemy.exc import IntegrityError  
8 from models import *  
9 from daijn import *  
10  
11 Session = sessionmaker(bind=ENGINElocal, autocommit=False)  
12 session = Session()  
13  
14 # 读取config.txt中roomId, round, course  
15 roomId = ""  
16 battle_round = ""  
17 battle_course = ""  
18 battle_remark = ""  
19 gameNum = ""  
20 with open("config.txt", mode="r", encoding="utf-8") as f:  
21     line = f.readline()  
22     while line:  
23         key, value = line.split(":")  
24         if key == "roomId":  
25             roomId = value.strip()  
26         if key == "BATTLE_ROUND":  
27             battle_round = value.strip()  
28         if key == "BATTLE_COOURSE":  
29             battle_course = value.strip()  
30         if key == "BATTLE_REMARK":  
31             battle_remark = value.strip()  
32         if key == "gameNum":  
33             gameNum = value.strip()  
34     line = f.readline()  
35 print(roomId, battle_course, battle_round)  
36  
37 # 全局变量  
38 # 已经刷新的event数量  
39 incre_event_num = (  
40     session.query(DynaKillInfo).filter_by(BATTLE_ID=roomId).count()  
41     + session.query(DynaItemPickUp).filter_by(BATTLE_ID=roomId).count()  
42 )  
43 boss_token_first_flag = 0 # 首次拾取boss信物的人  
44 global_player = []
```

```
config.txt
1 roomId:6580055364550715
2 BATTLE_ROUND:26
3 BATTLE.Course:8
4 BATTLE.REMARK:14
5 gameNum:87
```

2. 在 ":" 后直接输入对应的信息

step 2

运行aqtw2025.py程序

```
aqtw_2025.py
# ...
1 import pandas as pd
2 import numpy as np
3 import json
4 from datetime import datetime
5 import requests
6 from sqlalchemy.orm.exc import NoResultFound
7 from sqlalchemy.exc import IntegrityError
8 from models import *
9 from dajin import *
10
11 Session = sessionmaker(bind=ENGINELocal, autocommit=False)
12 session = Session()
13
14 # 读取config.txt中roomId、round、course
15 roomId = ""
16 battle_round = ""
17 battle_course = ""
18 battle_remark = ""
19 gameNum = ""
20 with open("config.txt", mode="r", encoding="utf-8") as f:
21     line = f.readline()
22     while line:
23         key, value = line.split(":")
24         if key == "roomId":
25             roomId = value.strip()
26         if key == "BATTLE_ROUND":
27             battle_round = value.strip()
28         if key == "BATTLE.Course":
29             battle_course = value.strip()
30         if key == "BATTLE.REMARK":
31             battle_remark = value.strip()
32         if key == "gameNum":
33             gameNum = value.strip()
34     line = f.readline()
35 print(roomId, battle_course, battle_round)
36
37 # 全局变量
38 # 已经更新的event数量
39 incre_event_num = (
40     session.query(DynakillInfo).filter_by(BATTLE_ID=roomId).count()
41     + session.query(DynaiItemPickUp).filter_by(BATTLE_ID=roomId).count()
42 )
43 boss_token_first_flag = 0 # 首次拾取boss信物的人
44 global_player = []
45
```

附：

半决赛config.txt信息

roomId:根据当时比赛实际情况填写
BATTLE_ROUND:27
BATTLE_COURSE:9

BATTLE_REMARK:15
gameNum:88 (打完一场比赛就+1)

总决赛config.txt信息

roomId:根据当时比赛实际情况填写
BATTLE_ROUND:28
BATTLE_COURSE:10
BATTLE_REMARK:16
gameNum:93 (打完一场比赛就+1)

如果线下赛是round5的话
就是从93开始，请根据实际情况再核对一下

2.3运行aqtw_mobile程序

2.3.1 涉及程序

aqtw_mobile

2.3.2 操作方法

step 1

选择main_aqtw.py文件

```
main_aqtw.py
...
# 获取选手装备信息
@app.route("/get_equip_list")
def get_equip_list():
    data = parseequip_list()
    return jsonify({"code": 200, "msg": "success", "data": data})

# @app.route("/longest_elim")
def longest_elim():
    data = parselongest_elim()
    return jsonify({"code": 200, "msg": "success", "data": data})

# @app.route("/get_dajin_state")
def get_dajin_state():
    data = parse_dajin_state()
    return jsonify({"code": 200, "msg": "success", "data": data})

# @app.route("/bracket_ranking")
def bracket_ranking():
    data = parse_bracket_ranking()
    return jsonify({"code": 200, "msg": "success", "data": data})

# @app.route("/best_roles")
def best_roles():
    ...

```

step 2

运行

```
main_aqtw.py
...
# 获取选手装备信息
@app.route("/get_equip_list")
def get_equip_list():
    data = parseequip_list()
    return jsonify({"code": 200, "msg": "success", "data": data})

# @app.route("/longest_elim")
def longest_elim():
    data = parselongest_elim()
    return jsonify({"code": 200, "msg": "success", "data": data})

# @app.route("/get_dajin_state")
def get_dajin_state():
    data = parse_dajin_state()
    return jsonify({"code": 200, "msg": "success", "data": data})

# @app.route("/bracket_ranking")
def bracket_ranking():
    data = parse_bracket_ranking()
    return jsonify({"code": 200, "msg": "success", "data": data})

# @app.route("/best_roles")
def best_roles():
    ...

```

2.4 运行APIPOST API查询

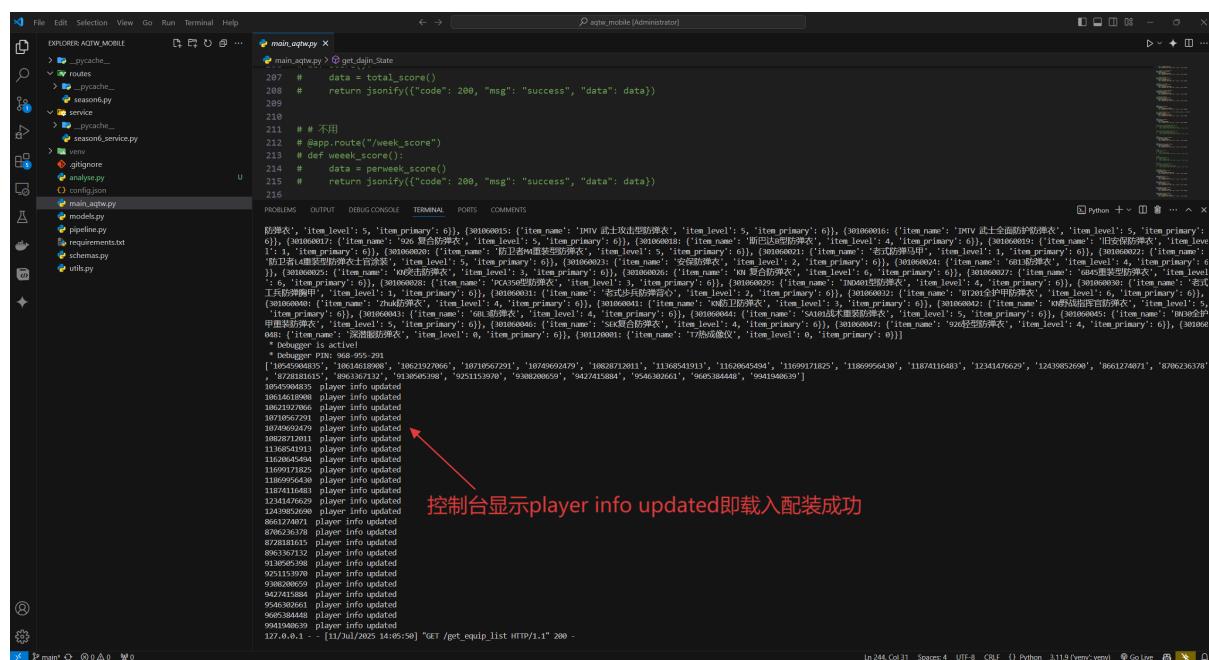
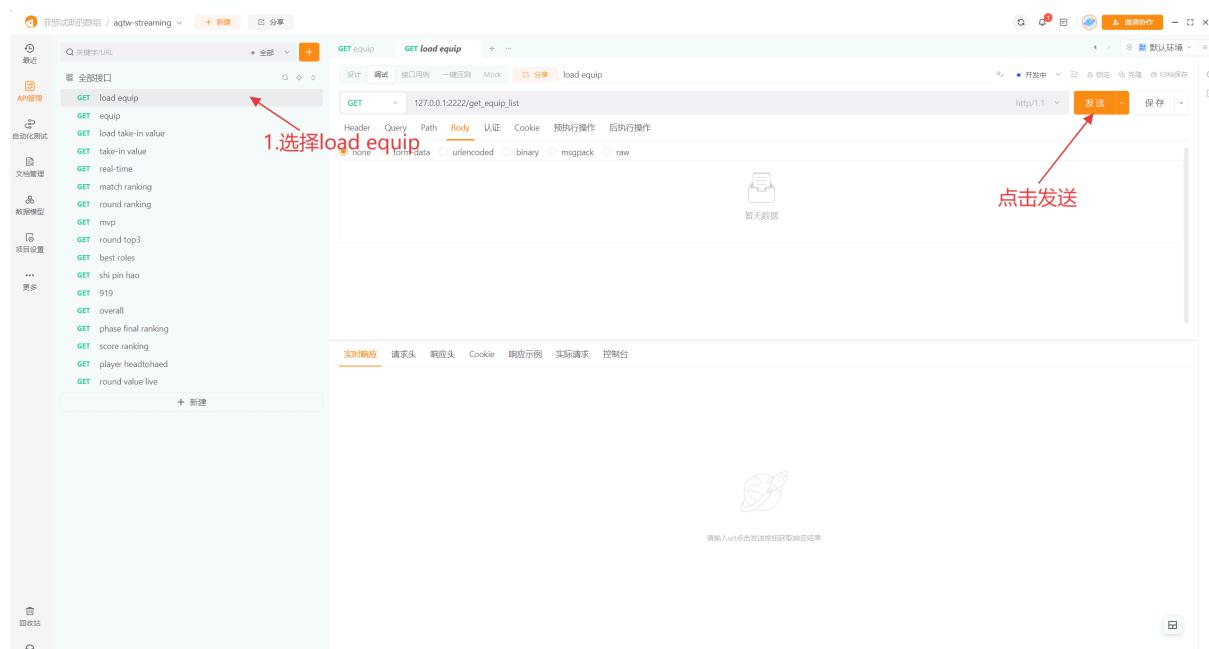
2.4.1 涉及程序

APIPOST

2.4.2 操作方法

step 1

加载选手配装



step 2

加载队伍带入物资

同step 1, 点击load take in value, 再点击发送(此处省略)

step 3

发送equip和take-in value API请求(如上述发送请求方式，此处省略)检查配装和带入物资是否有误，检查点如下：

1. equip返回数据中装备是否有正常刷出
 2. take-in value返回数据中，队伍是否为本场队伍，带入物资有无为0的（根据情况和游戏内再次核对）

2.5 赛后检查选手击杀是否有误（可选）

2.5.1 涉及程序

APIPOST, Navicat数据库管理软件, aqtw_mobile

2.5.2 操作方法

step 1

查看接口返回的eliminate_player_count是否和游戏客户端的结果一致

```

1 {
2     "code": 200,
3     "data": [
4         {
5             "eliminate_value": 150000,
6             "eliminate_value_withcommas": "150,000",
7             "eliminated_player_count": 3, ← 查看接口返回的队伍击杀
8             "player1_damage": "175",
9             "player1_eliminate_count": 1,
10            "player1_live_time": "18m57s",
11            "player1_name": "EDG.小莫",
12            "player1_pingtao": "抖音游戏",
13            "player1_position": "指挥位",
14            "player1_total_out_value": "246,003",
15            "player2_damage": "876",
16            "player2_eliminate_count": 1,
17            "player2_live_time": "19m17s",
18            "player2_name": "EDG.小柏",
19            "player2_pingtao": "抖音游戏",
20            "player2_position": "突击位",
21            "player2_total_out_value": "396,507",
22            "player3_damage": "0",
23            "player3_eliminate_count": 0,
24            "player3_live_time": "19m58s",
25            "player3_name": "EDG.声乐",
26            "player3_pingtao": "爱奇艺直播",
27            "player3_position": "指挥位",
28            "player3_total_out_value": "0",
29            "player4_damage": "2,267",
30            "player4_eliminate_count": 1,
31            "player4_live_time": "18m58s",
32            "player4_name": "EDG.掉豆",
33            "player4_pingtao": "抖音游戏",
34            "player4_position": "物资位",
35            "player4_total_out_value": "500",
36            "rank": 1,
37            "team_name": "EDG",
38            "total_damage": 3319,
39            "total_damage_withcommas": "3,319",
40            "total_out_value": 643010,
41            "total_out_value_withcommas": "643,010",
42            "total_value": 793010,
43            "total_value_withcommas": "793,010"
44        },
45        {
46            "eliminate_value": 50000,
47            "eliminate_value_withcommas": "50,000",
48            "eliminated_player_count": 1
49        }
50    ]
51 }

```

step 2

情况1：若一致，则无需改动

情况2：不一致，则进入step 3

step 3

根据核对后，数据出错的队伍，一个个按照队伍和battle_id(fetch里的config.txt的roomId)的去筛选并查看出错的选手

2.点击筛选，并选择对应的队伍和游戏房间号 (同fetch config.txt)

查看eliminate_player_count数据列，查看哪位选手的击杀数据和游戏客户端不同，请注意不要在这里改数据！

BATTLE_ID	PLAYER_ID	BATTLE_ROUND	BATTLE_COURSE	PLAYER_NAME	TEAM_ID	TEAM_NAME	BOSS_TOKEN	ELIMINATE_PLAYER_C	LIVE_TIME	TOTAL_DAMAGE	TOTAL_OUT_VALUE	CREATE_TIME	UPDATE_TIME
# varchar(20)	# varchar(20)	# varchar(20)	# varchar(20)	# varchar(100)	# int(11)	# varchar(100)	# int(11)	# int(11)	# int(11)	# int(11)	# int(11)	# datetime	# datetime
6579688141647185	1062398173	2	1	17.夷	4	T7	0	1	1184	60596	77733	2025-05-30 19:50:12	2025-05-30 19:
6579688141647185	887086646	2	1	17.白	4	T7	0	0	1183	43410	7805	2025-05-30 19:50:12	2025-05-30 19:
6579688141647185	8913078294	2	1	17.迎上海	4	T7	0	0	1181	29517	488790	2025-05-30 19:50:12	2025-05-30 19:
6579688141647185	900959111	2	1	17.破	4	T7	0	1	1183	53885	264421	2025-05-30 19:50:12	2025-05-30 19:

step 4

把所有出错的队伍填写到aqtw_mobile的update文件

点击update文件

输入所有出错的选手的名字和队名，以及增加的击杀数

点击运行

```

update_player_kill.py
update_player_kill.py > ...
update_player_kill.py > ...
1  from models import *
2  from sqlalchemy import and_
3  from datetime import datetime
4
5  battle_id = "6580955364550715"
6  battle_course = 8
7  battle_round = 26
8  update_player_list = [
9      {"player_name": "STE.黄白", "team_name": "STE", "add_num": 2},
10 ]
11 SessionLocal = sessionmaker(bind=ENGINElocal, autocommit=False)
12 session = SessionLocal()
13 try:
14     log_entries = []
15     for player in update_player_list:
16         player_name = player['player_name']
17         team_name = player['team_name']
18         add_num = player['add_num']
19         # 算出当前的击杀数
20         player_battle_kill = (
21             session.query(DataBattlePlayer.ELIMINATE_PLAYER_COUNT)
22             .filter(
23                 and_(
24                     DataBattlePlayer.BATTLE_ID == battle_id,
25                     DataBattlePlayer.PLAYER_NAME == player_name,
26                 )
27             )
28             .scalar()
29         )
30         team_battle_kill = (
31             session.query(DataBattleTeam.ELIMINATE_PLAYER_COUNT)
32             .filter(
33                 and_(
34                     DataBattleTeam.BATTLE_ID == battle_id,
35                     DataBattleTeam.TEAM_NAME == team_name,
36                 )
37             )
38             .scalar()
39         )
40         player_course_kill = (
41             session.query(DataCoursePlayer.ELIMINATE_PLAYER_COUNT)
42             .filter(
43                 and_(
44                     DataCoursePlayer.BATTLE_COURSE == battle_course,
45                     DataCoursePlayer.PLAYER_NAME == player_name,
46                 )
47             )
48             .scalar()
49         )
50         log_entries.append({
51             "player_name": player_name,
52             "team_name": team_name,
53             "battle_id": battle_id,
54             "battle_course": battle_course,
55             "battle_round": battle_round,
56             "add_num": add_num,
57             "player_battle_kill": player_battle_kill,
58             "team_battle_kill": team_battle_kill,
59             "player_course_kill": player_course_kill
60         })
61     session.commit()
62 except Exception as e:
63     session.rollback()
64     print(f"Error: {e}")
65 finally:
66     session.close()
67
68 if __name__ == "__main__":
69     update_player_kill()

```

step 5(可选)

若出现队伍犯规，需要清空本round的所有击杀，则启用set_player_kill_zero.py文件

```

update_player_kill.py
1 from models import *
2 from sqlalchemy import and_
3 from datetime import datetime
4
5 battle_id = "658005364550715"
6 battle_round = 8
7 update_player_list = [
8     {"player_name": "STE.萧白", "team_name": "STE"}, 2.填写需要归零的选手信息
9 ]
10 SessionLocal = sessionmaker(bind=ENGINELocal, autocommit=False)
11 session = SessionLocal()
12
13 try:
14     log_entries = []
15     for player in update_player_list:
16         player_name = player["player_name"]
17         team_name = player["team_name"]
18         add_num = player["add_num"]
19         # 查询原先的数据
20         player_battle_kill = (
21             session.query(DataBattlePlayer.ELIMINATE_PLAYER_COUNT)
22             .filter(
23                 and_(
24                     DataBattlePlayer.BATTLE_ID == battle_id,
25                     DataBattlePlayer.PLAYER_NAME == player_name,
26                 )
27             )
28             .scalar()
29         )
30         team_battle_kill = (
31             session.query(DataBattleTeam.ELIMINATE_PLAYER_COUNT)
32             .filter(
33                 and_(
34                     DataBattleTeam.BATTLE_ID == battle_id,
35                     DataBattleTeam.TEAM_NAME == team_name,
36                 )
37             )
38             .scalar()
39         )
40         player_course_kill = (
41             session.query(DataCoursePlayer.ELIMINATE_PLAYER_COUNT)
42             .filter(
43                 and_(
44                     DataCoursePlayer.BATTLE_COURSE == battle_course,
45                     DataCoursePlayer.PLAYER_NAME == player_name,
46                 )
47             )
48             .scalar()
49         )
50         log_entries.append({
51             "player_name": player_name,
52             "team_name": team_name,
53             "battle_id": battle_id,
54             "battle_round": battle_round,
55             "player_battle_kill": player_battle_kill,
56             "team_battle_kill": team_battle_kill,
57             "player_course_kill": player_course_kill,
58             "add_num": add_num
59         })
60     session.commit()
61
62     # 根据实际情况，以此类推的填写...
63
64 ]
65
66 # 若是给选手击杀归零，则不需要add_num,比如:
67 update_player_list = [
68     # 一个花括号内的数据行即代表一名选手
69     {"player_name": "STE.萧白", "team_name": "STE"}, 1.同理, 填写信息
70     {"player_name": "Team1.player1", "team_name": "Team1"}, 2.填写需要归零的选手信息
71     {"player_name": "Team2.player2", "team_name": "Team2"}, 2.填写需要归零的选手信息
72     {"player_name": "Team3.player3", "team_name": "Team3"}, 2.填写需要归零的选手信息
73     # 根据实际情况，以此类推的填写...
74
75 ]
76
77 # 提示：通常来说，你不需要复制这些内容，因为文件里已经写好了至少一行了

```

附：

更新/归零击杀所用的信息填写模板：

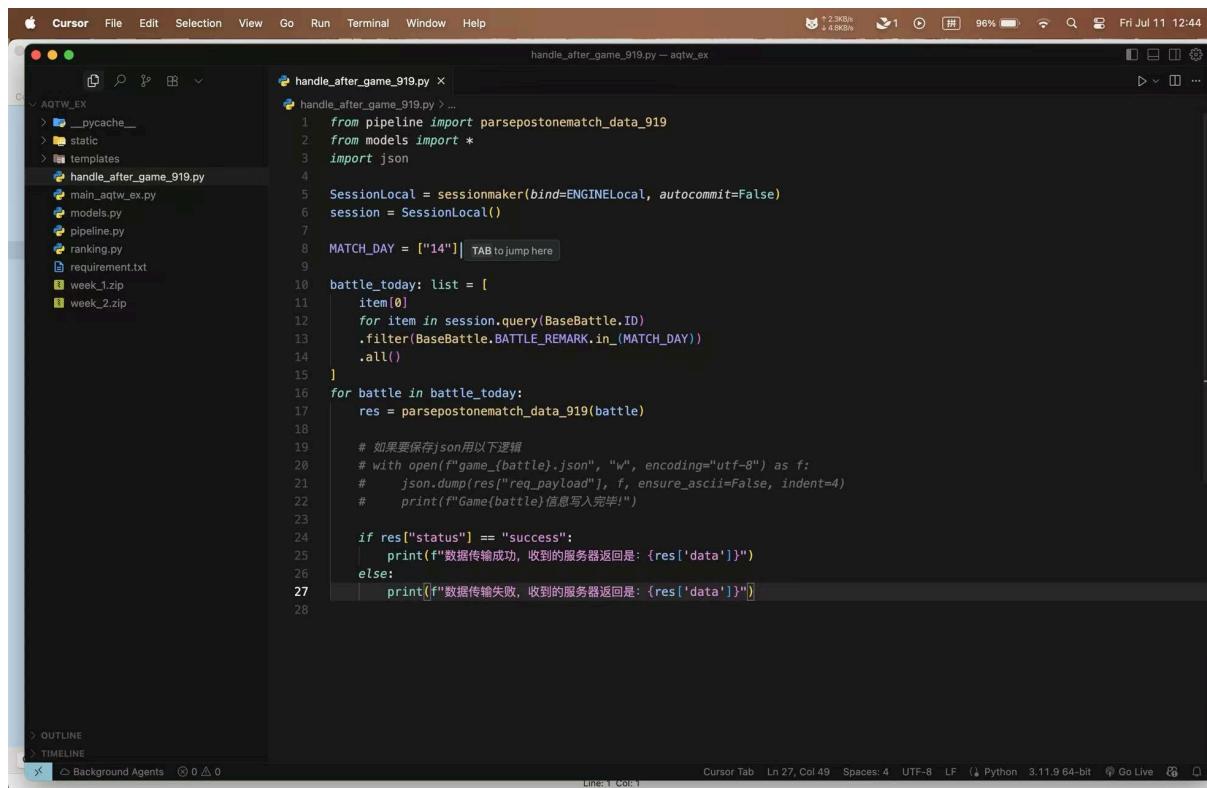
```

update_player_list = [
    # 一个花括号内的数据行即代表一名选手
    {"player_name": "STE.萧白", "team_name": "STE", "add_num": 1},
    {"player_name": "Team1.player1", "team_name": "Team1", "add_num": 2},
    {"player_name": "Team2.player2", "team_name": "Team2", "add_num": 3},
    {"player_name": "Team3.player3", "team_name": "Team3", "add_num": 4},
    # 根据实际情况，以此类推的填写...
]
# 若是给选手击杀归零，则不需要add_num,比如:
update_player_list = [
    # 一个花括号内的数据行即代表一名选手
    {"player_name": "STE.萧白", "team_name": "STE"}, 1.同理, 填写信息
    {"player_name": "Team1.player1", "team_name": "Team1"}, 2.填写需要归零的选手信息
    {"player_name": "Team2.player2", "team_name": "Team2"}, 2.填写需要归零的选手信息
    {"player_name": "Team3.player3", "team_name": "Team3"}, 2.填写需要归零的选手信息
    # 根据实际情况，以此类推的填写...
]
# 提示：通常来说，你不需要复制这些内容，因为文件里已经写好了至少一行了

```

, 你可以直接复制，
然后，再检查格式是否和模板相同

2.6 给游戏平台发送数据



```
handle_after_game_919.py > ...
1  from pipeline import parsepostonematch_data_919
2  from models import *
3  import json
4
5  SessionLocal = sessionmaker(bind=ENGINELocal, autocommit=False)
6  session = SessionLocal()
7
8  MATCH_DAY = ["14"]| TAB to jump here
9
10 battle_today: list = [
11     item[0]
12     for item in session.query(BaseBattle.ID)
13     .filter(BaseBattle.BATTLE_REMARK.in_(MATCH_DAY))
14     .all()
15 ]
16 for battle in battle_today:
17     res = parsepostonematch_data_919(battle)
18
19     # 如果要保存json用以下逻辑
20     # with open(f"game_{battle}.json", "w", encoding="utf-8") as f:
21     #     json.dump(res["red_payload"], f, ensure_ascii=False, indent=4)
22     #     print(f"Game{battle}信息写入完毕!")
23
24     if res["status"] == "success":
25         print(f"数据传输成功, 收到的服务器返回是: {res['data']}")
26     else:
27         print(f"数据传输失败, 收到的服务器返回是: {res['data']}")
```

2.6.1 涉及程序

aqtw_ex

2.6.2 操作方法

```
handle_after_game_919.py > ...
1  from pipeline import parsepostonematch_data_919
2  from models import *
3  import json
4
5  SessionLocal = sessionmaker(bind=ENGINELocal, autocommit=False)
6  session = SessionLocal()
7
8  MATCH_DAY = ["14"] TAB to jump here
9
10 battle_today: list = [ item[0]
11     for item in session.query(BaseBattle.ID)
12         .filter(BaseBattle.BATTLE_REMARK.in_(MATCH_DAY))
13         .all()
14 ]
15
16 for battle in battle_today:
17     res = parsepostonematch_data_919(battle)
18
19 # 如果要保存json用以下逻辑
20 # with open(f"game_{battle}.json", "w", encoding="utf-8") as f:
21 #     json.dump(res["req_payload"], f, ensure_ascii=False, indent=4)
22 #     print(f"Game{battle}信息写入完毕!")
23
24 if res["status"] == "success":
25     print(f"数据传输成功, 收到的服务器返回是: {res['data']}"))
26 else:
27     print(f"数据传输失败, 收到的服务器返回是: {res['data']}"))
28
```

3. 注意事项和QA

3.1 注意事项

1. 务必严格按照上述顺序进行数据执行
2. 务必**不要**擅自修改/删除代码文件
3. 务必**不要**操纵aqtw2025_s6以外的所有数据库
4. 务必注意所有直接对数据库的删除操作，比如：

60	小组赛第二轮DE组	5	3
61	小组赛第二轮DE组	5	3
62	小组赛第二轮DE组	5	3
		5	4
		5	4
		5	4
		5	5
		5	5
68	小组赛第二轮BD组	5	5
69	小组赛第二轮CE组	5	6
70	小组赛第二轮CE组	5	6
-	-	-	-

48 KB	InnoDB	308
16 KB		
16 KB	确认删除	X
16 KB		
16 KB	你确定要删除 "base_battle" 吗?	
16 KB	外键检查:	默认
16 KB		
16 KB	<input type="checkbox"/> 我了解此操作是永久性的且无法撤销	
16 KB		
224 KB	删除	取消

请优先点击“取消”，再确认刚刚进行的是什么操作，避免误删除

3.2 常见问题

- 运行fetch程序的时候，控制台显示文字“Key Error (gid)”类似字样时候怎么办？

答：腾讯服务器抽风了，不用着急，可以稍等几秒再试

- 运行fetch程序的时候，控制台报错，其中包含ssl错误字样怎么办？

答：检查是不是开VPN了，关掉就行

3. 我该在什么时机和字幕老师沟通

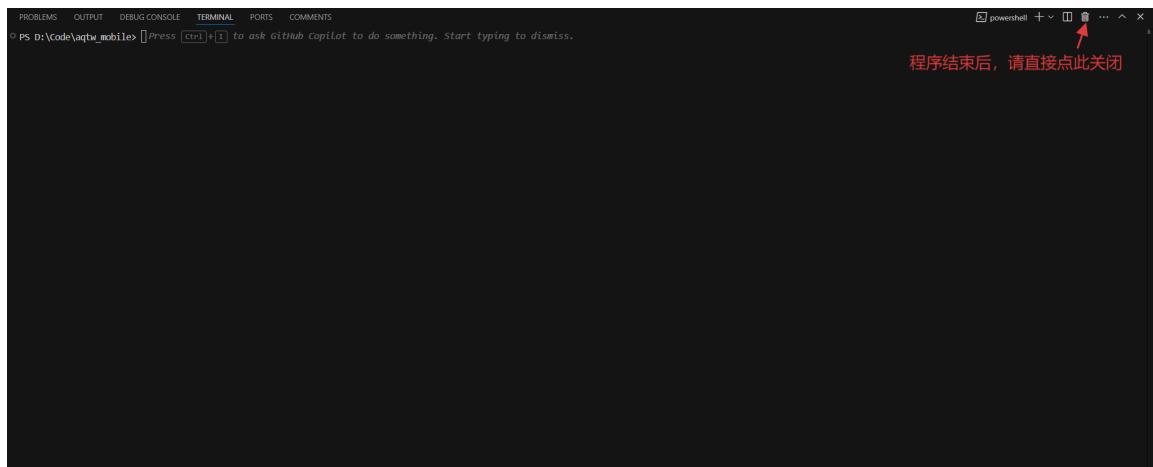
答：fetch程序正常运行后：“实时数据准备好了”

API POST 配装接口和带入物资接口发送完毕并检查无误后：“选手配装和带入物资准备好了”

和人工数据核对数据无误后：“赛后数据准备好了”

4. 控制台唤出和销毁

点**垃圾桶**图标关闭



点此唤出控制台

