QQ聊天机器人v0.2开发文档

**目录**

[QQ聊天机器人v0.2开发文档 1](#_Toc47871365)

[一、版本信息： 2](#_Toc47871366)

[二、开发环境： 3](#_Toc47871367)

[三、CQHTTP API插件： 3](#_Toc47871368)

[1、插件使用方法： 3](#_Toc47871369)

[2、插件配置： 4](#_Toc47871370)

[3、插件配置代码： 7](#_Toc47871371)

[4、插件API调用： 8](#_Toc47871372)

[（1）API请求方式： 8](#_Toc47871373)

[（2）API响应： 8](#_Toc47871374)

[（3）异步调用： 9](#_Toc47871375)

[（4）限速调用： 9](#_Toc47871376)

[（5）API列表： 9](#_Toc47871377)

[/send\_private\_msg 发送私聊消息 9](#_Toc47871378)

[/send\_group\_msg 发送群消息 10](#_Toc47871379)

[/send\_discuss\_msg 发送讨论组消息 10](#_Toc47871380)

[/send\_msg 发送消息 11](#_Toc47871381)

[/delete\_msg 撤回消息 11](#_Toc47871382)

[/set\_group\_ban 群组单人禁言 12](#_Toc47871383)

[/set\_group\_whole\_ban 群组全员禁言 12](#_Toc47871384)

[/set\_friend\_add\_request 处理加好友请求 12](#_Toc47871385)

[/set\_group\_add\_request 处理加群请求／邀请 13](#_Toc47871386)

[/get\_login\_info 获取登录号信息 13](#_Toc47871387)

[/get\_friend\_list 获取好友列表 13](#_Toc47871388)

[/get\_group\_list 获取群列表 14](#_Toc47871389)

[/get\_cookies 获取 Cookies 14](#_Toc47871390)

[/get\_csrf\_token 获取 CSRF Token 14](#_Toc47871391)

[/get\_credentials 获取 QQ 相关接口凭证 15](#_Toc47871392)

[/get\_status 获取插件运行状态 15](#_Toc47871393)

[/get\_version\_info 获取酷Q及CQHTTP插件的版本信息 16](#_Toc47871394)

[/set\_restart\_plugin 重启 CQHTTP 插件 16](#_Toc47871395)

[/clean\_data\_dir 清理数据目录 17](#_Toc47871396)

[/clean\_plugin\_log 清理插件日志 17](#_Toc47871397)

[四、Warframe API调用： 17](#_Toc47871398)

[1、API调用地址： 17](#_Toc47871399)

[2、内建接口： 18](#_Toc47871400)

[warframe\_info.py 18](#_Toc47871401)

[（1）get\_alerts() 18](#_Toc47871402)

[（2）get\_arbitration() 18](#_Toc47871403)

[（3）get\_all\_cycle() 19](#_Toc47871404)

[（4）get\_earth\_cycle() 19](#_Toc47871405)

[（5）get\_cetus\_cycle() 19](#_Toc47871406)

[（6）get\_orb\_cycle() 20](#_Toc47871407)

[（7）get\_cetus\_mission() 20](#_Toc47871408)

[（8）get\_conclave\_challenge() （不再开发） 22](#_Toc47871409)

[（9）get\_darvo\_daily\_deal() 22](#_Toc47871410)

[（10）get\_orb\_mission() 23](#_Toc47871411)

[（11）get\_events() 23](#_Toc47871412)

[（12）get\_fissures() 24](#_Toc47871413)

[（13）get\_flash\_sales() （不再开发） 25](#_Toc47871414)

[（14）get\_global\_upgrades() （未开发） 25](#_Toc47871415)

[（15）get\_invasions() 26](#_Toc47871416)

[（16）get\_construction\_progress() 27](#_Toc47871417)

[（17）get\_kuva() （不稳定） 27](#_Toc47871418)

[（18）get\_news() 28](#_Toc47871419)

[（19）get\_nightwave() 29](#_Toc47871420)

[（20）get\_persistent\_enemy() （未开发） 30](#_Toc47871421)

[（21）get\_riven\_statistic() （不再开发） 30](#_Toc47871422)

[（22）get\_sentient\_outpost() 31](#_Toc47871423)

[（23）get\_sanctuary() （不再开发） 32](#_Toc47871424)

[（24）get\_sortie() 32](#_Toc47871425)

[（25）get\_syndicate\_mission() 33](#_Toc47871426)

[（26）get\_void\_trader() 33](#_Toc47871427)

[（27）get\_arcane\_enhancement() （不再开发） 34](#_Toc47871428)

[（28）get\_rest\_time(expiry) 35](#_Toc47871429)

[五、Warframe Market API调用： 35](#_Toc47871430)

[1、API调用地址： 35](#_Toc47871431)

[2、内建接口： 35](#_Toc47871432)

[warframe\_market.py 35](#_Toc47871433)

[（1）get\_market(message\_from\_user) 35](#_Toc47871434)

# 一、版本信息：

**该程序由开发者（QQ：465152177）开发提供，未经允许禁止传播本文档。**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **发布日期** | **改动信息** | **额外信息** |
| V0.1 | 2020.7.26 | / | 初版机器人，具备部分warframe游戏数据的查询功能 |
| V0.2 | 2020.8.9 | 改动部分warframe游戏数据查询输出信息 | 完成warframe游戏数据的查询功能；完成warframe market物品交易查询功能；完成warframe游戏内容翻译功能 |

**已提供和未来将提供的功能有：**

（1）Warframe国际服：

1、查询游戏实时信息；

2、查询Warframe Market实时交易信息；

3、游戏内容翻译；

4、查询Warframe Wiki内容（暂未开发）；

（2）崩坏三国服（暂未开发）：

（3）机器人管理员（暂未开发）：

（4）腾讯AI聊天（暂未开发）：

# 二、开发环境：

该机器人使用酷Q软件的插件CQHTTP为信息传输平台，通过CQHTTP API插件完成数据的发送及接收，使用Python脚本完成数据处理。

CQHTTP插件的Github链接：<https://github.com/richardchien/coolq-http-api/releases>

CQHTTP插件文档链接：<https://richardchien.gitee.io/coolq-http-api/docs/4.15/#/>

Python开发版本：3.8.1

Python使用的第三方库：flask、requests、json、pytz、re、time、datetime

**注意：**酷Q已经停止提供服务，无法提供未来之更新等支持。

# 三、CQHTTP API插件：

## 1、插件使用方法：

启用后插件将开启一个HTTP服务器来接收请求，监听服务器某端口，首次启用会生成一个默认配置文件，在酷Q的data\app\io.github.richardchien.coolqhttpapi\config文件夹中，文件名为 <user\_id>.json（<user\_id>为登录的QQ号），使用JSON格式填写。此时通过http://<host>:<port>/即可调用酷Q的函数。

## 2、插件配置：

|  |  |  |
| --- | --- | --- |
| **配置项名称** | **默认值** | **说明** |
| host | [::] | HTTP 服务器监听的 IP |
| port | 5700 | HTTP 服务器监听的端口 |
| use\_http | true | 是否开启HTTP接口，即通过HTTP调用API |
| ws\_host | [::] | WebSocket服务器监听的 IP |
| ws\_port | 6700 | WebSocket服务器监听的端口 |
| use\_ws | false | 是否开启WebSocket服务器，可用于调用 API 和推送事件 |
| ws\_reverse\_url | 空 | 反向WebSocket Event和事件上报的共用地址 |
| ws\_reverse\_api\_url | 空 | 反向 WebSocket API 地址，如果为空，则使用ws\_reverse\_url指定的值 |
| ws\_reverse\_event\_url | 空 | 反向 WebSocket 事件上报地址，如果为空，则使用 ws\_reverse\_url 指定的值 |
| ws\_reverse\_reconnect\_interval | 3000 | 反向 WebSocket 客户端断线重连间隔，单位毫秒 |
| ws\_reverse\_reconnect\_on\_code\_1000 | true | 是否在关闭状态码为1000的时候重连 |
| ws\_reverse\_use\_universal\_client | false | 是否使用Universal客户端 |
| use\_ws\_reverse | false | 是否使用反向WebSocket服务，即插件作为WebSocket客户端主动连接指定的API和事件上报地址 |
| post\_url | 空 | 消息和事件的上报地址，通过POST方式请求，数据以JSON格式发送 |
| post\_timeout | 0 | HTTP上报（即访问post\_url）的超时时间，单位秒，0表示不设置超时 |
| access\_token | 空 | API访问token，如果不为空，则会在接收到请求时验证Authorization请求头是否为Bearer xxxxxxxx，xxxxxxxx为access token |
| secret | 空 | 上报数据签名密钥，如果不为空，则会在 HTTP 上报时对 HTTP 正文进行 HMAC SHA1 哈希，使用 secret 的值作为密钥，计算出的哈希值放在上报的 X-Signature 请求头，例如 X-Signature: sha1=f9ddd4863ace61e64f462d41ca311e3d2c1176e2 |
| post\_message\_format | string | 上报消息格式，string 为字符串格式，array 为数组格式 |
| serve\_data\_files | false | 是否提供请求 data 目录的文件的功能 |
| update\_source | v4.12.3 之前 github v4.12.3 及之后 china | 更新源，可选 global/github 和 china/coding |
| update\_channel | stable | 更新通道，目前有 stable、beta、alpha 三个 |
| auto\_check\_update | false | 是否自动检查更新（每次启用插件时检查），不启用的情况下，仍然可以在 酷Q 应用菜单中手动检查更新 |
| auto\_perform\_update | false | 是否自动执行更新，仅在 auto\_check\_update 启用时有效，若启用，则插件将在自动检查到更新后，自动下载新版本（需要手动重启 酷Q 以生效） |
| thread\_pool\_size | 4 | 工作线程池大小，用于异步发送消息和一些其它小的异步任务，应根据计算机性能和实际需求适当调节，若设为 0，则使用 CPU 核心数 \* 2 + 1 |
| server\_thread\_pool\_size | 4 | API 服务器线程池大小，用于异步处理请求，应根据计算机性能和实际需求适当调节，若设为 0，则使用 CPU 核心数 \* 2 + 1 |
| convert\_unicode\_emoji | true | 是否在 CQ:emoji 和实际的 Unicode 之间进行转换，转换可能耗更多时间，但日常情况下影响不大，如果你的机器人需要处理非常大段的消息（上千字），且对性能有要求，可以考虑关闭转换 |
| event\_filter | 空 | 指定事件过滤规则文件，留空将不开启事件过滤器 |
| enable\_backward\_compatibility | false | 是否启用旧版兼容性，启用时事件上报的数据将和 3.x 版本保持兼容 |
| show\_log\_console | true | 是否显示日志输出控制台 |
| max\_log\_file\_size | 6291456 | 最大单日志文件大小，单位字节，默认 6 MB |
| max\_log\_files | 1 | 最大日志文件备份数量（采用日志轮替机制） |
| log\_level | info | 日志文件和日志控制台的日志等级，可选 debug、info、warning、error、fatal |
| use\_extension | false | 是否启用扩展机制 |
| disable\_coolq\_log | true | 是否禁用酷Q原生日志，由于使用 酷Q 原生日志可能会导致快速重启时插件卡死，所以默认禁用，如果你不在乎重启时卡死，并且需要在酷Q原生日志窗口查看插件的日志，可以将此项设为false |
| online\_status\_detection\_method | get\_stranger\_info | QQ在线状态检测方式，默认（get\_stranger\_info）通过陌生人查询接口判断，设为log\_db可切换成从 酷Q的日志数据库判断 |
| enable\_heartbeat | false | 是否启用心跳机制，启用时会产生类型为heartbeat的元事件 |
| heartbeat\_interval | 15000 | 产生心跳元事件的时间间隔，单位毫秒 |
| enable\_rate\_limited\_actions | false | 是否启用限速API调用的支持 |
| rate\_limit\_interval | 500 | 限速API调用的排队间隔时间，单位毫秒 |

## 3、插件配置代码：

酷Q路径下：data\app\io.github.richardchien.coolqhttpapi\config\<user\_id>.json



其中，host使用本地回传地址127.0.0.1，端口号8080，数据post URL地址<http://127.0.0.1:5701/api/message>，数据post格式为array数组，设置响应延时500ms，日志记录等级为warning。

## 4、插件API调用：

### （1）API请求方式：

所有API都同时支持GET和POST两种请求方式（除获取data目录中的文件），参数可通过URL参数、表单或JSON传入，后两者分别对应Content-Type application/x-www-form-urlencoded和application/json。如果使用JSON传入，参数要放在根级别的JSON对象中，且参数的数据类型必须符合API列表中给出的每个参数所要求的数据类型。为避免可能的问题，建议使用JSON方式传入数据。

如果配置文件中填写了access\_token，则需要在每次请求时在请求头中加入验证头。如：

GET /send\_private\_msg?access\_token=<access\_token>&user\_id=123456&message=hello HTTP/1.1

**数据包示例：**

data={"user\_id": 123456, "message": "hello"}

以JSON格式，加入API列表中的参数及对应数据。

### （2）API响应：

对于所有请求：

1. 如果access token未提供，状态码为401；
2. 如果access token不符合，状态码为403；
3. 如果POST请求的Content-Type不支持，状态码为406；
4. 如果POST请求的正文格式不正确，状态码为400；
5. 如果API不存在，状态码为404；
6. 其余情况下，状态码均为200；

响应内容为JSON格式，结构如下：

{

"status": "ok",

"retcode": 0,

"data": {

"id": 123456,

"nickname": "滑稽"

}

}

响应返回码retcode说明如下表：

|  |  |
| --- | --- |
| **retcode** | **说明** |
| 0 | 同时status为ok，表示操作成功 |
| 1 | 同时status为async，表示操作已进入异步执行，具体结果未知 |
| 100 | 参数缺失或参数无效，通常是因为没有传入必要参数，某些接口中也可能因为参数明显无效（比如传入的QQ号小于等于0，此时无需调用酷Q函数即可确定失败），此项和以下的status均为failed |
| 102 | 酷Q函数返回的数据无效，一般是因为传入参数有效但没有权限，比如试图获取没有加入的群组的成员列表 |
| 103 | 操作失败，一般是因为用户权限不足，或文件系统异常、不符合预期 |
| 104 | 由于酷Q提供的凭证（Cookie和CSRF Token）失效导致请求QQ相关接口失败，可尝试清除酷Q缓存来解决 |
| 201 | 工作线程池未正确初始化（无法执行异步任务） |

### （3）异步调用：

所有API都可以通过附加后缀 \_async 进行异步调用。当进行异步调用时，以 get\_ 开头的接口无法获得调用结果，不存在调用意义；某些接口本身即为异步调用（status=async）时，使用后缀不会出现区别。

### （4）限速调用：

设置配置项 enable\_rate\_limited\_actions 为true可开启限速调用（默认为false）。

API接口可通过附加后缀 \_rate\_limited 来进行限速调用，主要用于发送接口，以避免消息发送频率过快。

所有限速调用将以指定速度排队执行，速度由配置项rate\_limit\_interval控制，单位：毫秒。

限速调用时，status=async。

### （5）API列表：

#### /send\_private\_msg 发送私聊消息

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| user\_id | number | - | 对方QQ号 |
| message | message | - | 要发送的内容 |
| auto\_escape | boolean | false | 消息内容是否作为纯文本发送（即不解析CQ码），只在message字段是字符串时有效 |

**响应数据：**

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| message\_id | number (int32) | 消息ID |

#### /send\_group\_msg 发送群消息

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| group\_id | number | - | 群号 |
| message | message | - | 要发送的内容 |
| auto\_escape | boolean | false | 消息内容是否作为纯文本发送（即不解析CQ码），只在message字段是字符串时有效 |

**响应数据：**

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| message\_id | number (int32) | 消息ID |

#### /send\_discuss\_msg 发送讨论组消息

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| discuss\_id | number | - | 讨论组ID（正常情况下看不到，需要从讨论组消息上报的数据中获得） |
| message | message | - | 要发送的内容 |
| auto\_escape | boolean | false | 消息内容是否作为纯文本发送（即不解析CQ码），只在message字段是字符串时有效 |

**响应数据：**

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| message\_id | number (int32) | 消息ID |

#### /send\_msg 发送消息

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| message\_type | string | - | 消息类型，支持private、group、discuss，分别对应私聊、群组、讨论组，如不传入，则根据传入的\*\_id参数判断 |
| user\_id | number | - | 对方QQ号（消息类型为private时需要） |
| group\_id | number | - | 群号（消息类型为group时需要） |
| discuss\_id | number | - | 讨论组ID（消息类型为discuss时需要） |
| message | message | - | 要发送的内容 |
| auto\_escape | boolean | false | 消息内容是否作为纯文本发送（即不解析CQ码），只在message字段是字符串时有效 |

**响应数据：**

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| message\_id | number (int32) | 消息ID |

#### /delete\_msg 撤回消息

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| message\_id | number (int32) | - | 消息 ID |

**响应数据：**

无

#### /set\_group\_ban 群组单人禁言

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| group\_id | number | - | 群号 |
| user\_id | number | - | 要禁言的QQ号 |
| duration | number | 30\*60 | 禁言时长，单位秒，0表示取消禁言 |

**响应数据：**

无

#### /set\_group\_whole\_ban 群组全员禁言

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| group\_id | number | - | 群号 |
| enable | boolean | true | 是否禁言 |

**响应数据：**

无

#### /set\_friend\_add\_request 处理加好友请求

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| flag | string | - | 加好友请求的flag（需从上报的数据中获得） |
| approve | boolean | true | 是否同意请求 |
| remark | string | 空 | 添加后的好友备注（仅在同意时有效） |

**响应数据：**

无

#### /set\_group\_add\_request 处理加群请求／邀请

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| flag | string | - | 加群请求的flag（需从上报的数据中获得） |
| sub\_type或type | string | - | add或invite，请求类型（需要和上报消息中的sub\_type字段相符） |
| approve | boolean | true | 是否同意请求／邀请 |
| reason | string | 空 | 拒绝理由（仅在拒绝时有效） |

**响应数据：**

无

#### /get\_login\_info 获取登录号信息

**参数：**

无

**响应数据：**

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| user\_id | number (int64) | QQ号 |
| nickname | string | QQ昵称 |

#### /get\_friend\_list 获取好友列表

**参数：**

无

**响应数据：**

响应内容为JSON数组，每个元素如下：

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| user\_id | number (int64) | QQ号 |
| nickname | string | 昵称 |
| remark | string | 备注名 |

#### /get\_group\_list 获取群列表

**参数：**

无

**响应数据：**

响应内容为JSON数组，每个元素如下：

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| group\_id | number (int64) | 群号 |
| group\_name | string | 群名称 |

#### /get\_cookies 获取 Cookies

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| domain | string | 空 | 需要获取cookies的域名 |

**响应数据：**

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| cookies | string | Cookies |

#### /get\_csrf\_token 获取 CSRF Token

**参数：**

无

**响应数据：**

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| token | number (int32) | CSRF Token |

#### /get\_credentials 获取 QQ 相关接口凭证

即上面两个接口的合并。

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| domain | string | 空 | 需要获取cookies的域名 |

**响应数据：**

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| cookies | string | Cookies |
| csrf\_token | number (int32) | CSRF Token |

#### /get\_status 获取插件运行状态

**参数：**

无

**响应数据：**

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| app\_initialized | boolean | CQHTTP插件已初始化 |
| app\_enabled | boolean | CQHTTP插件已启用 |
| plugins\_good | object | CQHTTP的各内部插件是否正常运行 |
| app\_good | boolean | CQHTTP插件正常运行（已初始化、已启用、各内部插件正常运行） |
| online | boolean | 当前QQ在线，null表示无法查询到在线状态 |
| good | boolean | CQHTTP插件状态符合预期，意味着插件已初始化，内部插件都在正常运行，且QQ在线 |

通常情况下建议只使用online和good这两个字段来判断运行状态，因为随着插件的更新，其它字段有可能频繁变化。

其中，online字段的在线状态检测有两种方式，可通过online\_status\_detection\_method配置项切换，默认通过读取酷Q日志数据库实现，可切换为get\_stranger\_info以通过测试陌生人查询接口的可用性来检测。具体区别如下：

|  |  |  |
| --- | --- | --- |
| **在线检测方式** | **优点** | **缺点** |
| get\_stranger\_info（默认） | 正常情况下比log\_db准确，但请求频率过高时可能变得不准确（在线被认为不在线）；需要发送网络请求 | （几乎不可能）会因为酷Q更新而失效 |
| log\_db | 查询速度较快；无需网络请求（不会触发腾讯风控）；不会因为请求频率过高而不准确 | 可能因为酷Q修改数据库表名、文件名而失效；月尾掉线，月初无法检测到 |

#### /get\_version\_info 获取酷Q及CQHTTP插件的版本信息

**参数：**

无

**响应数据：**

|  |  |  |
| --- | --- | --- |
| **字段名** | **数据类型** | **说明** |
| coolq\_directory | string | 酷Q根目录路径 |
| coolq\_edition | string | 酷Q版本，air或pro |
| plugin\_version | string | CQHTTP插件版本，例如2.1.3 |
| plugin\_build\_number | number | CQHTTP 插件build号 |
| plugin\_build\_configuration | string | CQHTTP插件编译配置，debug或release |

#### /set\_restart\_plugin 重启 CQHTTP 插件

由于重启插件同时需要重启API服务，这意味着当前的API请求会被中断，因此需在异步地重启插件，接口返回的status是async。

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **默认值** | **说明** |
| delay | number | 0 | 要延迟的毫秒数，如果默认情况下无法重启，可以尝试设置延迟为2000左右 |

**响应数据：**

无

#### /clean\_data\_dir 清理数据目录

用于清理积攒了太多旧文件的数据目录，如image。

**参数：**

| **字段名** | **数据类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| data\_dir | string | - | 收到清理的目录名，支持image、record、show、bface |

**响应数据：**

无

#### /clean\_plugin\_log 清理插件日志

用于清空插件的日志文件。

**参数：**

无

**响应数据：**

无

# 四、Warframe API调用：

## 1、API调用地址：

Warframe API链接：<https://api.warframe.us/>

Warframe API Doc链接：<https://docs.warframestat.us/>

API链接末尾可附加平台，包括pc、swi、ps4、xb1。

Warframe相关开发Github项目查看：

Warframe Community Developers：<https://github.com/WFCD>

请求头参数设置仅支持接受语言。请求头格式示例：



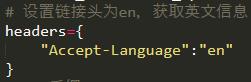
响应数据均为JSON格式。通过from json import loads，使用data=loads(data)将响应数据转换为dict类型。

## 2、内建接口：

相关代码文件均放于warframe文件夹下。

### warframe\_info.py

该文件代码用于通过Warframe API查询游戏内容，其中，设定链接请求头为：



获取英文内容，以避免翻译问题导致的错误。

下列函数返回的数据均为已经过处理的数据，返回的内容为要回复的消息。格式均为string。

#### （1）get\_alerts()

获取警报内容。请求链接为：

<https://api.warframstat.us/pc/alerts>

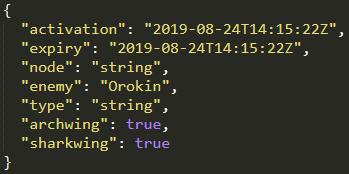
由于游戏不再提供任何警报任务，因此该请求链接回复为空数组，即[]。

#### （2）get\_arbitration()

获取仲裁内容。请求链接为：

<https://api.warframstat.us/pc/arbitration>

**数据示例：**



**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| activation | string | 开始时间 |
| expiry | string | 结束时间 |
| node | string | 节点 |
| enemy | string | 敌人派系 |
| type | string | 任务类型 |
| archwing | bollean | 是否需要archwing |
| sharkwing | bollean | 是否需要sharkwing |

#### （3）get\_all\_cycle()

获取所有的循环状态，该模块需要获得其它3个模块返回的数据，包括：

get\_earth\_cycle() ：获取地球循环状态

get\_cetus\_cycle() ：获取希图斯循环状态

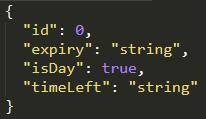
get\_orb\_cycle() ：获取奥布山谷循环状态

#### （4）get\_earth\_cycle()

获取地球循环状态。API链接为：

<https://api.warframstat.us/pc/earthCycle>

**数据示例：**



**数据参数：**

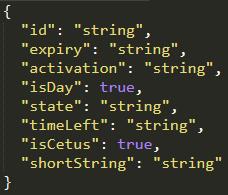
|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| id | number | 事件ID |
| expiry | string | 结束时间 |
| isDay | boolean | 是否为白天 |
| timeLeft | string | 剩余时间 |

#### （5）get\_cetus\_cycle()

获取希图斯循环状态，API链接为：

<https://api.warframestat.us/pc/cetusCycle>

**数据示例：**



**数据参数：**

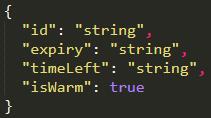
|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| id | string | 事件ID |
| expiry | string | 结束时间 |
| isDay | boolean | 是否为白天 |
| timeLeft | string | 剩余时间 |
| isCetus | boolean | 是否为希图斯相关数据 |
| shortString | string | 剩余循环时间 |

#### （6）get\_orb\_cycle()

获取福尔图娜循环状态，API链接为：

<https://api.warframestat.us/pc/vallisCycle>

**数据示例：**



**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| id | string | 事件ID |
| expiry | string | 结束时间 |
| timeLeft | string | 剩余时间 |
| isWarm | boolean | 是否为温暖状态 |

#### （7）get\_cetus\_mission()

获取希图斯赏金内容，API链接为：

<https://api.warframestat.us/pc/syndicateMissions>

希图斯赏金内容来源于集团任务API，从中提取出含有”syndicate”:“Ostrons”字段的即为希图斯赏金。

**数据示例：**



**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| nodes | array | 节点 |
| eta | string | 预计结束时间 |
| jobs | array | 任务列表 |
| syndicate | string | 集团名称 |
| id | string | 事件ID |
| expiry | string | 事件结束时间 |
| activation | string | 事件开始时间 |

**“jobs”字段内参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| activation | string | 任务开始时间 |
| expiry | string | 任务结束时间 |
| rewardPool | array | 奖励池 |
| type | string | 任务类型 |
| enemyLevels | array | 敌人等级 |
| standingStages | array | （未知） |
| minMR | number | （未知） |

#### （8）get\_conclave\_challenge() （不再开发）

获取PVP任务内容，API链接为：

<https://api.warframestat.us/pc/conclaveChallenges>

**数据示例：**

**数据参数：**

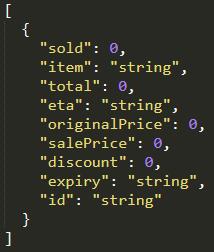
|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
|  |  |  |
|  |  |  |
|  |  |  |

#### （9）get\_darvo\_daily\_deal()

获取Darvo的每日特惠内容，API链接为：

<https://api.warframestat.us/pc/dailyDeals>

**数据示例：**



**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| sold | number | 已出售数量 |
| item | string | 出售的物品名称 |
| total | number | 出售总数 |
| eta | string | 预计结束时间 |
| originalPrice | number | 原价 |
| salePrice | number | 售价 |
| discount | number | 折扣百分比 |
| expiry | string | 售卖结束时间 |
| id | string | 事件ID |

#### （10）get\_orb\_mission()

获取奥布山谷赏金内容，API链接为：

<https://api.warframestat.us/pc/syndicateMissions>

奥布山谷赏金内容来源于集团任务API，从中提取出含有”syndicate”:“Solaris United”字段的即为奥布山谷赏金。

数据示例及数据参数参考：[（7）get\_cetus\_mission()](#_（7）get_cetus_mission())

#### （11）get\_events()

获取事件活动内容，API链接为：

<https://api.warframestat.us/pc/events>

**数据示例：**

**数据参数：**

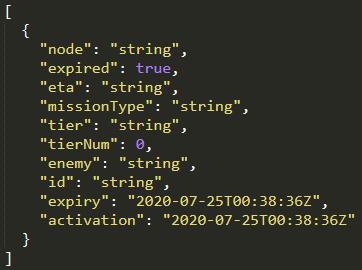
|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| activation | string | 事件开始时间 |
| expiry | string | 事件结束时间 |
| maximumScore | number | 分数上限 |
| currentScore | number | 目前分数 |
| smallInterval | number | （未知） |
| largeInterval | number | （未知） |
| faction | string | 敌人派系 |
| description | string | 事件描述 |
| tolltip | string | 事件副描述 |
| node | string | 节点 |
| concurrentNodes | array | 其它节点列表 |
| victimNode | string | 牺牲节点 |
| scoreLocTag | string | 分数标签 |
| rewards | array | 主要奖励表 |
| expired | boolean | 事件是否到期 |
| health | number | 剩余生命值 |
| affiliatedWith | string | 事件陪同集团 |
| jobs | array | 事件任务 |
| interimSteps | array | 阶段目标内容 |
| progressSteps | dict | 阶段进度 |
| progressTotal | number | 总进度 |
| showTotalAtEndOfMission | boolean | 任务结束时是否显示总数 |
| isPersonal | boolean | 是否为个人任务 |
| isCommunity | boolean | 是否为团队任务 |
| regionDrops | array | 地区掉落列表 |
| archwingDrops | array | 空战掉落列表 |
| asString | string | 掉落总表 |
| metadata | dict | （未知） |
| completionBonuses | array | 事件完成额外奖励 |
| scoreVar | string | 分数 |
| altExpiry | string | 事件切换结束时间 |
| altActivation | string | 事件切换开始时间 |
| nextAlt | dict | 下一事件切换时间表 |

#### （12）get\_fissures()

获取虚空裂缝任务内容，API链接为：

<https://api.warframestat.us/pc/fissures>

**数据示例：**



**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| node | string | 节点 |
| expired | boolean | 是否到期 |
| eta | string | 预计结束时间 |
| missionType | string | 任务类型 |
| tier | string | 关联遗物级别名称 |
| tierNum | number | 关联遗物级别 |
| enemy | string | 敌人类型 |
| id | string | 事件ID |
| expiry | string | 结束时间 |
| activation | string | 开始时间 |

#### （13）get\_flash\_sales() （不再开发）

获取商店热卖内容，API链接为：

<https://api.warframestat.us/pc/flashSales>

**数据示例：**

**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
|  |  |  |
|  |  |  |
|  |  |  |

#### （14）get\_global\_upgrades() （未开发）

获取当前全服加成内容，API链接为：

<https://api.warframestat.us/pc/globalUpgrades>

**数据示例：**

**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
|  |  |  |
|  |  |  |
|  |  |  |

#### （15）get\_invasions()

获取入侵任务内容，API链接为：

<https://api.warframestat.us/pc/invasions>

**数据示例：**

**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| defenderReward | dict | 防守方奖励 |
| attakingFaction | string | 攻击方派系 |
| completion | number | 完成百分比 |
| attakerReward | dict | 攻击方奖励 |
| count | number | 攻击次数计数 |
| completed | boolean | 是否已完成 |
| requiredRuns | number | 要求参与次数 |
| vsInfestation | boolean | 是否为对抗I系怪物 |
| node | string | 节点 |
| eta | string | 预计结束时间 |
| defendingFaction | string | 防守方派系 |
| id | string | 事件ID |
| activation | string | 开始时间 |
| rewardTypes | array | 奖励类型 |
| desc | string | 任务描述 |

其中，字段 count 、 requiredRuns 、 completion有如下关系：

(requiredRuns+count)/2/requiredRuns=completion

可得出：

attaker\_rest\_time=requiredRuns-count

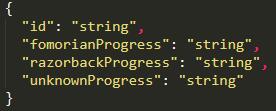
defender\_rest\_time=requiredRuns+count

#### （16）get\_construction\_progress()

获取入侵舰队建造进度，API链接为：

<https://api.warframestat.us/pc/constructionProgress>

**数据示例：**



**数据参数：**

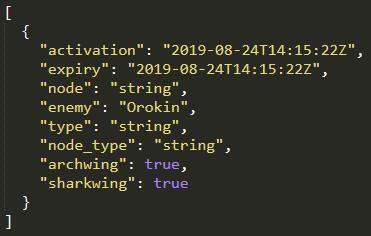
|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| id | string | 事件ID |
| fomorianProgress | string | 利刃豺狼建造进度百分比 |
| razorbackProgress | string | 巨人舰队建造百分比 |
| unknownProgress | string | 未知内容 |

#### （17）get\_kuva() （不稳定）

获取赤毒虹吸器任务内容，API链接为：

<https://api.warframestat.us/pc/kuva>

**数据示例：**



**数据参数：**

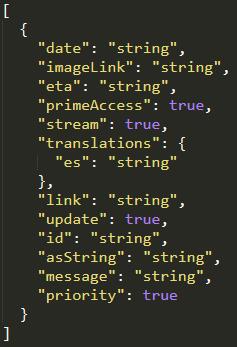
|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| activation | string | 开始时间 |
| expiry | string | 结束时间 |
| node | string | 节点 |
| enemy | string | 敌人派系 |
| type | string | 任务类型 |
| node\_type | string | 节点类型 |
| archwing | boolean | 是否需要archwing |
| sharkwing | boolean | 是否需要sharkwing |

#### （18）get\_news()

获取游戏新闻内容，API链接为：

<https://api.warframestat.us/pc/news>

**数据示例：**



**数据参数：**

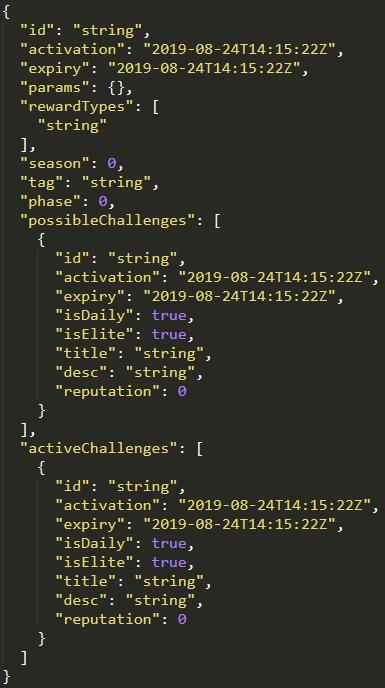
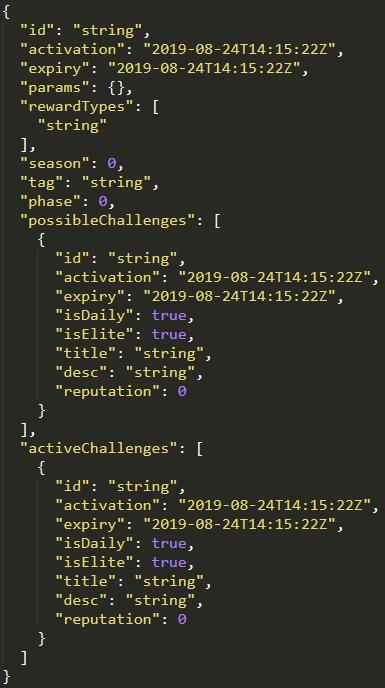
|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| date | string | 新闻日期 |
| imageLink | string | 图片链接 |
| eta | string | 预计结束时间 |
| primeAccess | boolean | 是否为PA新闻 |
| stream | boolean | 是否为平台新闻 |
| translations | dict | 新闻信息各语言翻译 |
| link | string | 新闻链接 |
| update | boolean | 是否更新 |
| id | string | 新闻ID |
| asString | string | 新闻字段 |
| message | string | 新闻信息 |
| priority | boolean | 是否优先显示 |

#### （19）get\_nightwave()

获取午夜电波任务内容，API链接为：

<https://api.warframestat.us/pc/nightwave>

**数据示例：**

****

**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| id | string | 事件ID |
| activation | string | 开始时间 |
| expiry | string | 结束时间 |
| params | list | 参数表 |
| rewardTypes | array | 奖励类型 |
| season | number | 电波季度 |
| tag | string | 电波活动标签 |
| phase | number | 电波阶段 |
| possibleChallenges | list | 可能的挑战内容 |
| activeChallenges | list | 正在进行的挑战内容 |

**“activeChallenges”字段内参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| id | string | 事件ID |
| activation | string | 开始时间 |
| expiry | string | 结束时间 |
| isDaily | boolean | 是否为每日任务，若不是，则该项不存在 |
| isElite | boolean | 是否为精英任务 |
| title | string | 任务名称 |
| desc | string | 任务描述 |
| reputation | number | 任务奖励声望 |

#### （20）get\_persistent\_enemy() （未开发）

获取追随者信息，API链接为：

<https://api.warframestat.us/pc/persistentEnemies>

**数据示例：**

**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
|  |  |  |
|  |  |  |
|  |  |  |

#### （21）get\_riven\_statistic() （不再开发）

获取指定紫卡信息，API链接为：

[https://api.warframestat.us/pc/rivens/search/{query}](https://api.warframestat.us/pc/rivens/search/%7bquery%7d)

链接后需要加上武器名称才可查询到具体的紫卡信息。

**数据示例：**

**数据参数：**

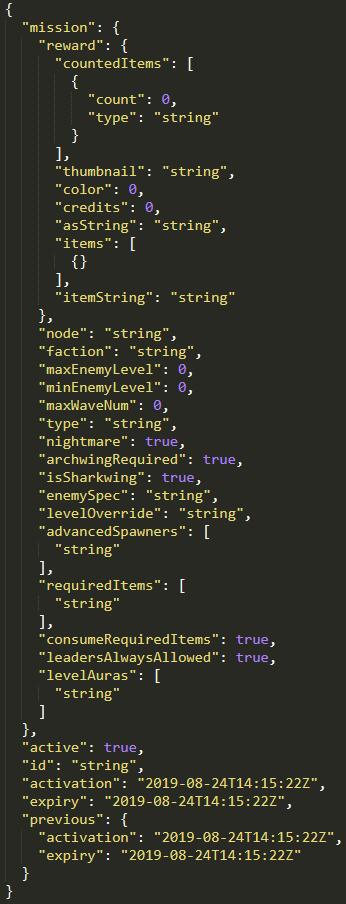
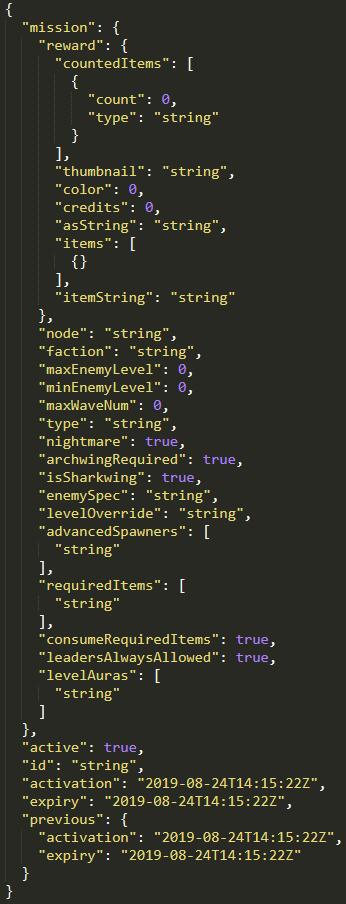
|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
|  |  |  |
|  |  |  |
|  |  |  |

#### （22）get\_sentient\_outpost()

获取航道星舰S船出没信息，API链接为：

<https://api.warframestat.us/pc/sentientOutposts>

**数据示例：**



**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| mission | list | 任务信息 |
| active | boolean | 是否进行中 |
| id | string | 事件ID |
| activation | string | 开始时间 |
| expiry | string | 结束时间 |
| previous | list | 上一任务信息 |

**“mission”字段内参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| node | string | 任务节点 |
| faction | string | 敌人派系 |
| type | string | 任务类型 |

#### （23）get\_sanctuary() （不再开发）

获取圣殿任务内容，API链接为：

<https://api.warframestat.us/pc/simaris>

**数据示例：**

**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
|  |  |  |
|  |  |  |
|  |  |  |

#### （24）get\_sortie()

获取突击任务内容，API链接为：

<https://api.warframestat.us/pc/sortie>

**数据示例：**



**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| id | string | 事件ID |
| activation | string | 开始时间 |
| expiry | string | 结束时间 |
| rewardPool | string | 奖励池 |
| variants | list | 任务详细内容 |
| boss | string | BOSS名称 |
| faction | string | 敌人派系 |
| expired | boolean | 是否已结束 |
| eta | string | 结束剩余时间 |

**“variants”字段内参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| node | string | 任务节点 |
| boss | string | BOSS名称 |
| missionType | string | 任务类型 |
| planet | string | 星球 |
| modifier | string | 任务状态 |
| modifierDescription | string | 任务状态描述 |

#### （25）get\_syndicate\_mission()

获取集团任务内容，API链接为：

<https://api.warframestat.us/pc/syndicateMissions>

集团任务内容来源于集团任务API，从中提取出含有下列集团特定字段的即为集团任务。

|  |  |  |
| --- | --- | --- |
| 字段名 | **字段值** | **对应集团** |
| syndicate | Steel Meridian | 钢铁防线 |
| Arbiters of Hexis | 均衡仲裁者 |
| Cephalon Suda | 中枢苏达 |
| Perrin Sequence | 佩兰数列 |
| Red Veil | 血色面纱 |
| New Loka | 新世间 |

数据示例及数据参数参考：[（7）get\_cetus\_mission()](#_（7）get_cetus_mission())

#### （26）get\_void\_trader()

获取虚空商人内容，API链接为：

<https://api.warframestat.us/pc/voidTrader>

**数据示例：**



**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| id | string | 事件ID |
| activation | string | 开始时间 |
| expiry | string | 结束时间 |
| character | string | 到访角色名 |
| location | string | 到访位置 |
| inventory | list | 携带物品 |
| psID | string | 事件副ID |
| active | boolean | 是否已到访 |
| startString | string | 开始剩余时间 |
| endString | string | 结束剩余时间 |

**“inventory”字段内参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
| item | string | 物品名称 |
| ducats | number | 需求杜卡德金币数 |
| credits | number | 需求星币数 |

#### （27）get\_arcane\_enhancement() （不再开发）

获取指定附能信息，API链接为：

[https://api.warframestat.us/arcanes/search/{query}](https://api.warframestat.us/arcanes/search/%7bquery%7d)

查询指定附能，链接后需要指定附能名称，且链接不能指定平台。

**数据示例：**

**数据参数：**

|  |  |  |
| --- | --- | --- |
| 字段名 | **数据类型** | **说明** |
|  |  |  |
|  |  |  |
|  |  |  |

#### （28）get\_rest\_time(expiry)

传入GMT时区结束时间，获取当前GMT时区时间并计算获得时间差，返回时间差字符串。

# 五、Warframe Market API调用：

## 1、API调用地址：

Warframe Market API链接：<https://api.warframe.market/v1>

Warframe Market网站链接：<https://warframe.market>

添加 **/items/{item\_name}** 可查询物品信息；

添加 **/items/{item\_name}/orders** 可查询物品交易信息。

其它API链接地址查看：<https://github.com/WFCD/market-api-spec>

Warframe相关开发Github项目查看：

Warframe Community Developers：<https://github.com/WFCD>

## 2、内建接口：

相关代码文件均放于warframe文件夹下。

### warframe\_market.py

#### （1）get\_market(message\_from\_user)

获取物品交易信息，提取数据并整理，返回要发送的消息内容。

**1) 提取API返回的交易订单：**

data.get(“payload”).get(“orders”)

**2) 筛选交易订单：**

data[i].get("user").get("status")=="ingame"

and

data[i].get("order\_type")=="sell"

and

data[i].get("platform")=="pc"

and

data[i].get("region")=="en"

**3) 对筛选后的交易订单进行排序（按价格从低到高）：**

for i in range(online\_length-1):

ex\_flag=False

for j in range(online\_length-i-1):

if (online\_list[j].get("platinum")>online\_list[j+1].get("platinum")): online\_list[j],online\_list[j+1]=online\_list[j+1],online\_list[j]

ex\_flag=True

if not ex\_flag:

break

**4）建立返回的内容：**

返回的内容最多显示10名在线卖家。

**5）从物品信息中获取物品的英文全名：**

根据API查询物品信息，并提取出全名。

data=data.get("payload").get("item").get("items\_in\_set")

items\_length=len(data)

for n in range(items\_length):

if (data[n].get("url\_name")==url\_name):

item\_name="（"+data[n].get("en").get("item\_name")+"）"

break