2. 复用计划

后台框架以 tornado 为主，复用 tornado 的 RequestHandler、websocketHandler、Application等类。爬虫复用自定义基类 baseSpider、Table以及 selenium 库。前端复用 bootstrap 框架。

3. 设计概述

3.2. 系统接口设计

3.2.2.内部接口：

**1. 聊天部分**

**1.1 返回用户ID**

url

119.29.161.184:8000?rand=xxx&sex=xxx&nameIndex=xxx

sex : 0为male 1为female

response

{

"status":1,

"id":"xsijfa-r21ja2j-asoj2aa-adh2saj"

}

**1.2 聊天**

url

119.29.161.184:8000/chat

params

id : string(用接口1.1返回的ID)

request

{

"type":"xxxx",

"text":"xxxxxxx",

"id":"xxxxxx-xxxx-xxxxx"

}

response

{

"res":{

"sender":"xxx",

"type":"xxx",

"content":"",

"time":"2016-10-16 13:45:20"

},

"status":1

}

该接口使用websocket协议

**1.3 返回昵称**

url

119.29.161.184:8000/name?sex=xxx

response

{

res: {

7: "郭芙",

11: "李莫愁",

16: "李沅芷",

24: "方怡",

28: "何红药",

31: "安小惠",

37: "杨不悔",

41: "袁紫衣"

},

status: 1

}

sex : 0为male 1为female

**2. 热点新闻部分**

**1.1 返回热点新闻**

url

119.29.161.184:8000?page=xxx

(page表示第几页，从 1 开始计数，每页返回十条热点新闻,按照 pv 进行降序排列)

response

success:

{

"status": 1,

"data": [{"id":xxx,"url":xxx,"title":xxx,"visit\_cnt":xxx},...]

fail:

{"status": 0, "data": "page exceed limits"}

**1.2 热点新闻访问量更新**

**url**

119.29.161.184:8000/news (post)

数据格式：

{

"id":XXX

}

**response** success:

{

"status":1

}

fail:

{

"status":0,

"data":"XXX" (给出失败原因)

}

**1.3 返回笑话**

**url**

119.29.161.184:8000/joke?num=XXX

(返回 num 个笑话)

**response** success:

{

"status":1,

"data":[

{"id":"XXX",

"content":"XXX"

},

{

...

}

...

]

}

fail:

{

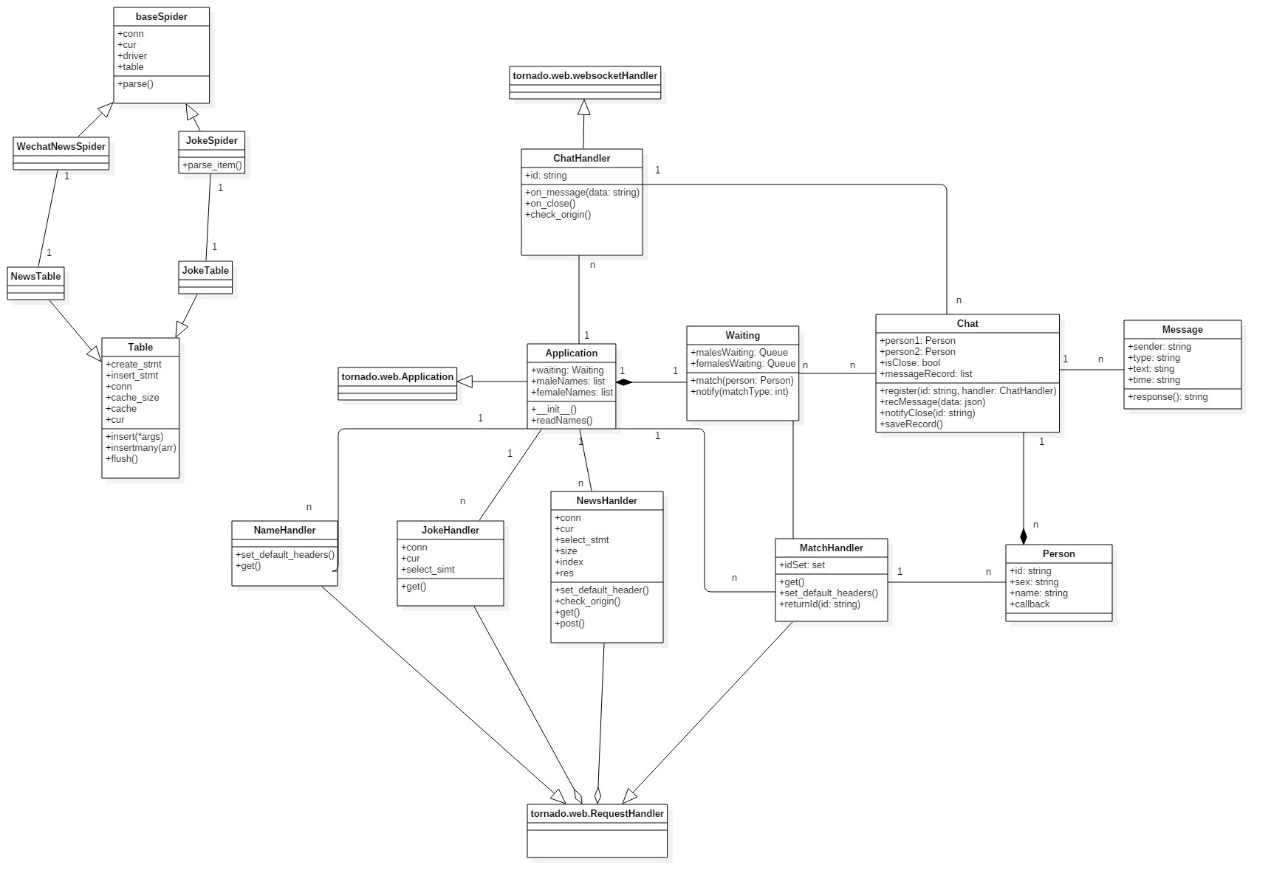
"status":0,

"data":"XXX"

}

3.3 对象模型设计

对象模型：



对象描述：

1. Application

属性：

1. waiting : Waiting
2. maleNames：list

存储男的名字的一个list

1. femaleNames：list

存储女的名字的一个list

方法：

1. \_\_init\_\_()

重载构造函数

1. readNames()

读取名字存到maleNames和femaleNames

1. NameHandler

方法：

1. set\_default\_headers()

设置返回的header，解决跨域问题

1. get()

响应对其相应的url的get请求

1. MatchHandler

属性：

1. idSet : set

存储已分配出去的id，避免重复

方法：

1. set\_default\_headers()

设置返回的header，解决跨域问题

1. get()

响应对其相应的url的get请求

1. returnId(id:string)

返回ID给前端

1. ChatHandler

属性：

1. id : string

用户在MatchHandler的returnId中返回的id

方法：

1. on\_message(data: string)

实现websocket的on\_message方法，data为在连接过程中前端向服务器发送的消息

1. on\_close()

实现websocket的on\_close()方法，这是当连接断开时执行的函数。

1. check\_origin()

解决跨域问题

1. Waiting

属性:

1. malesWaiting: Queue

存储未匹配的女生的队列

1. femalesWaiting: Queue

存储未匹配的男生的队列

方法：

1. match(person:Person)

当一个新的用户进入时，若已有正在等待的用户，则进行匹配，若没有则将用户加入队列进行等待

1. notify(matchType: int)

用来通知被匹配的用户，matchType表示匹配的类型。

1. Chat

属性：

1. person1: Person

用户1

1. person2 : Person

用户2

1. isClose : bool

表示是否有人退出聊天

1. messageRecord : list

存储聊天记录的一个list

1. Person

属性：

1. Id : string

用户的id

1. Sex: string

用户的性别

1. Name : string

用户的名字

1. Callback

该用户建立连接对应的MatchHandler实例中的returnId这个回调函数

1. Message

属性：

1. Sender : string

消息发送者的名字

1. Type : string

发送的消息的类型

1. Text : string

发送的消息的内容

1. Time : string

发送消息的时间

方法：

1. response(): string

返回给另一个用户的消息

1. Table

属性：

1. conn: pymysql.cnnections.Connection

与数据库的连接

1. cur: pymysql.cursors.Cursor

游标

1. cache\_size: int

缓存大小

1. cache: list

缓存

1. spider\_name: string

爬虫名字

1. create\_stmt: string

创建数据库表语句

1. insert\_stmt: string

插入数据库表语句

方法：

(1). Insert(\*args): void

向缓存或者数据库插入记录

(2). Insertmany(arr): void

向数据库插入多条记录

(3). Flush(): void

将缓存写入数据库

1. baseSpider

属性：

1. conn: pymysql.cnnections.Connection

与数据库的连接

(2). cur: pymysql.cursors.Cursor

游标

(3). driver: selenium.webdriver

浏览器内核

(4). table: Table

table 实例

方法：

(1). Parse(OA\_name): void

给定公众号名字，爬取该公众号最近十条文章

1. NewsHandler

属性：

1. conn: pymysql.cnnections.Connection

与数据库的连接

1. cur: pymysql.cursors.Cursor

游标

1. select\_stmt: string

选择语句

1. size: int

数据库记录大小

1. index: tuple(int,int)

一个tuple,用来读取长度为10的文章记录

1. res: list[list[string]]

数据库文章

方法：

1. set\_default\_header()

设置允许跨域

1. check\_origin

检查请求来源

1. get()

响应客户端发起的 get 请求

1. post()

响应客户端发起的 post 请求

1. JokeHandler

属性：

1. conn: pymysql.cnnections.Connection

与数据库的连接

1. cur: pymysql.cursors.Cursor

游标

1. select\_stmt: string

选择语句

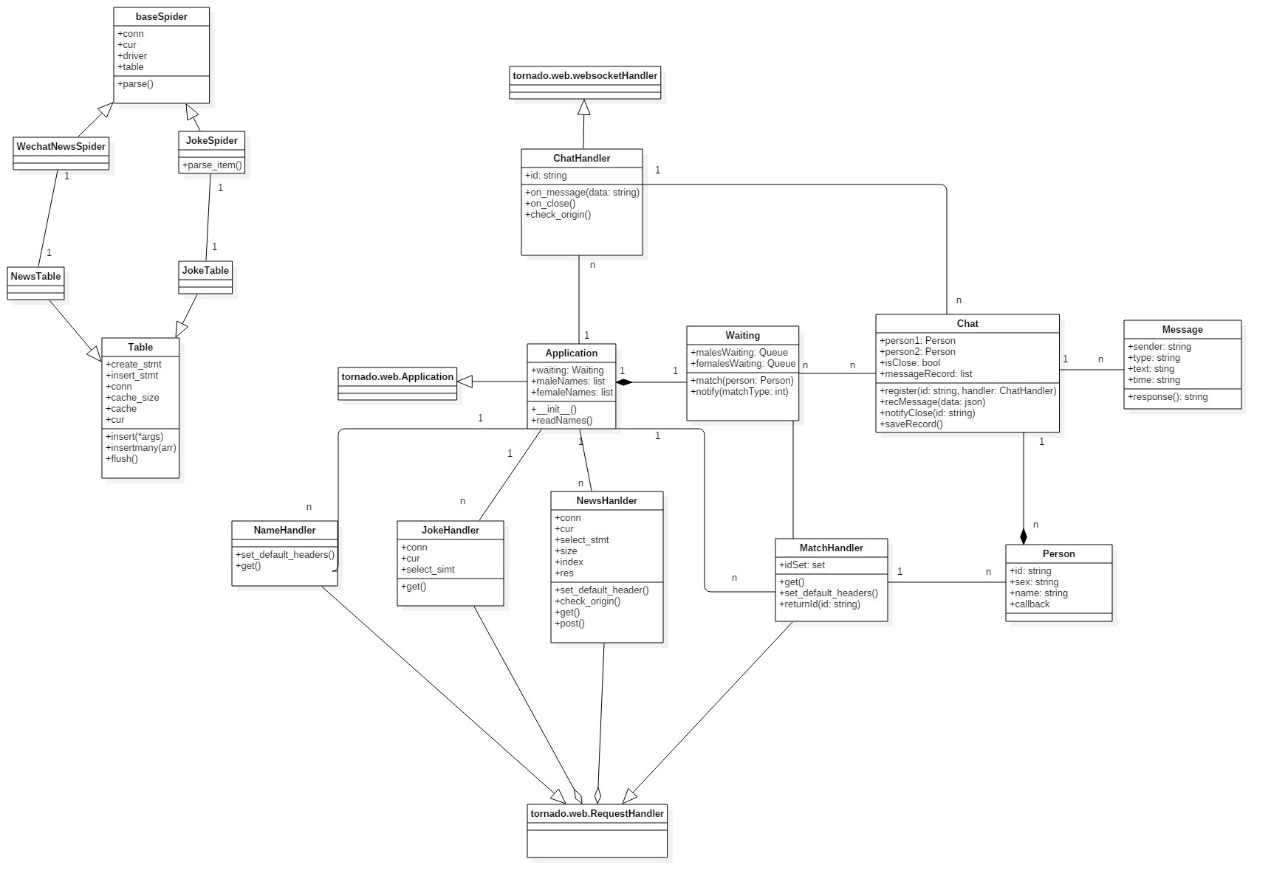
方法：

1. get()

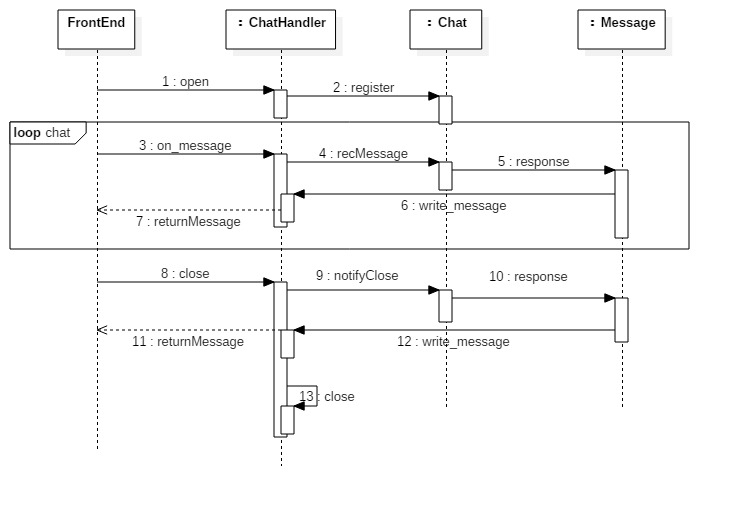
响应客户端的 get 请求

3.4 动态

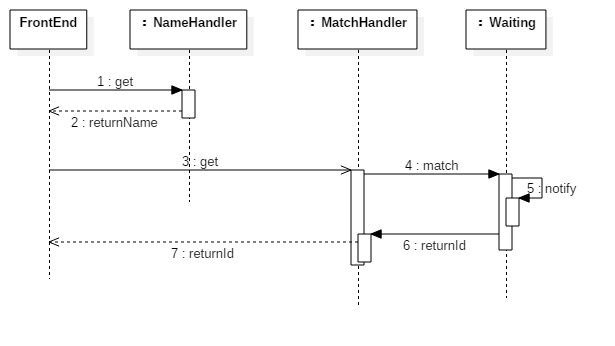
类图：



聊天时序图：



匹配时序图：



热点时序图：

