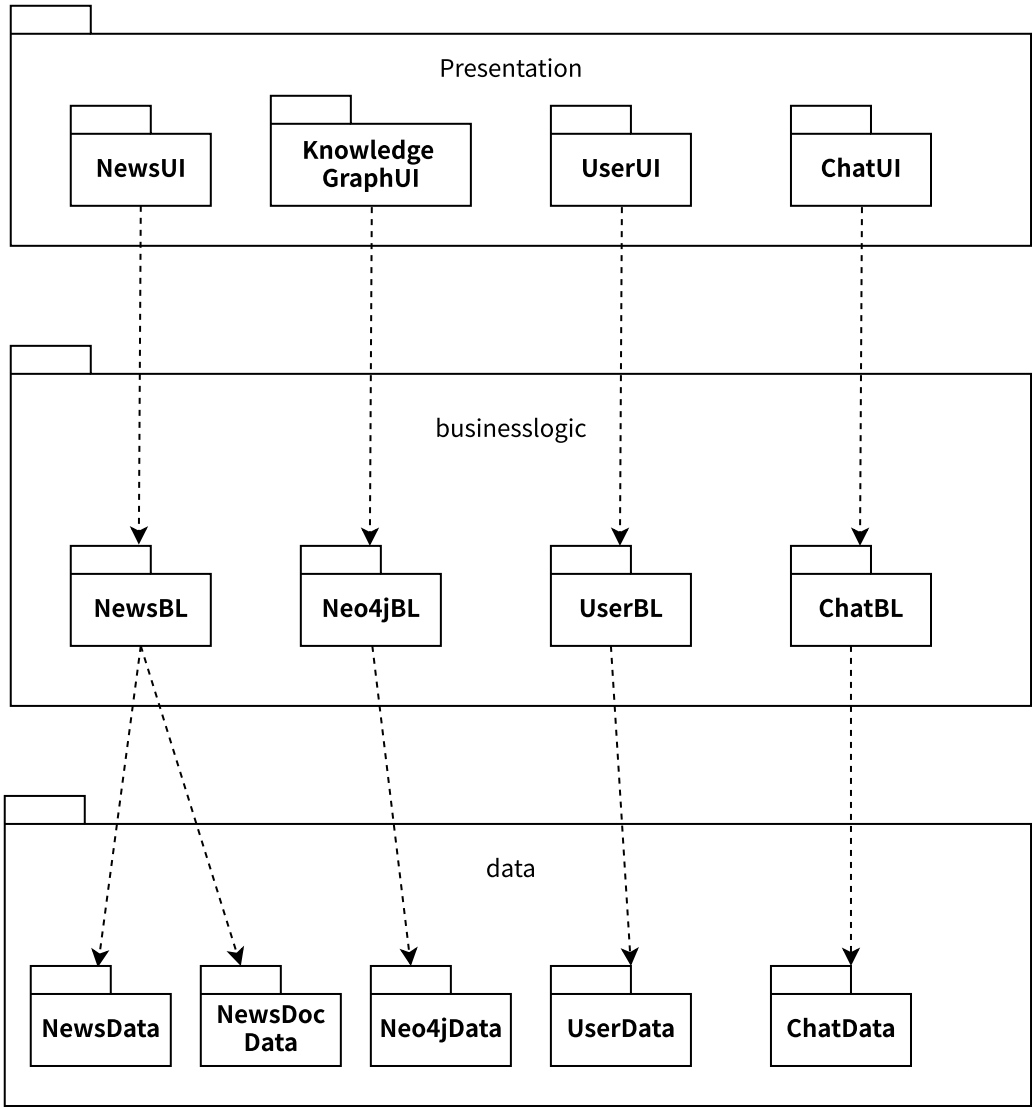


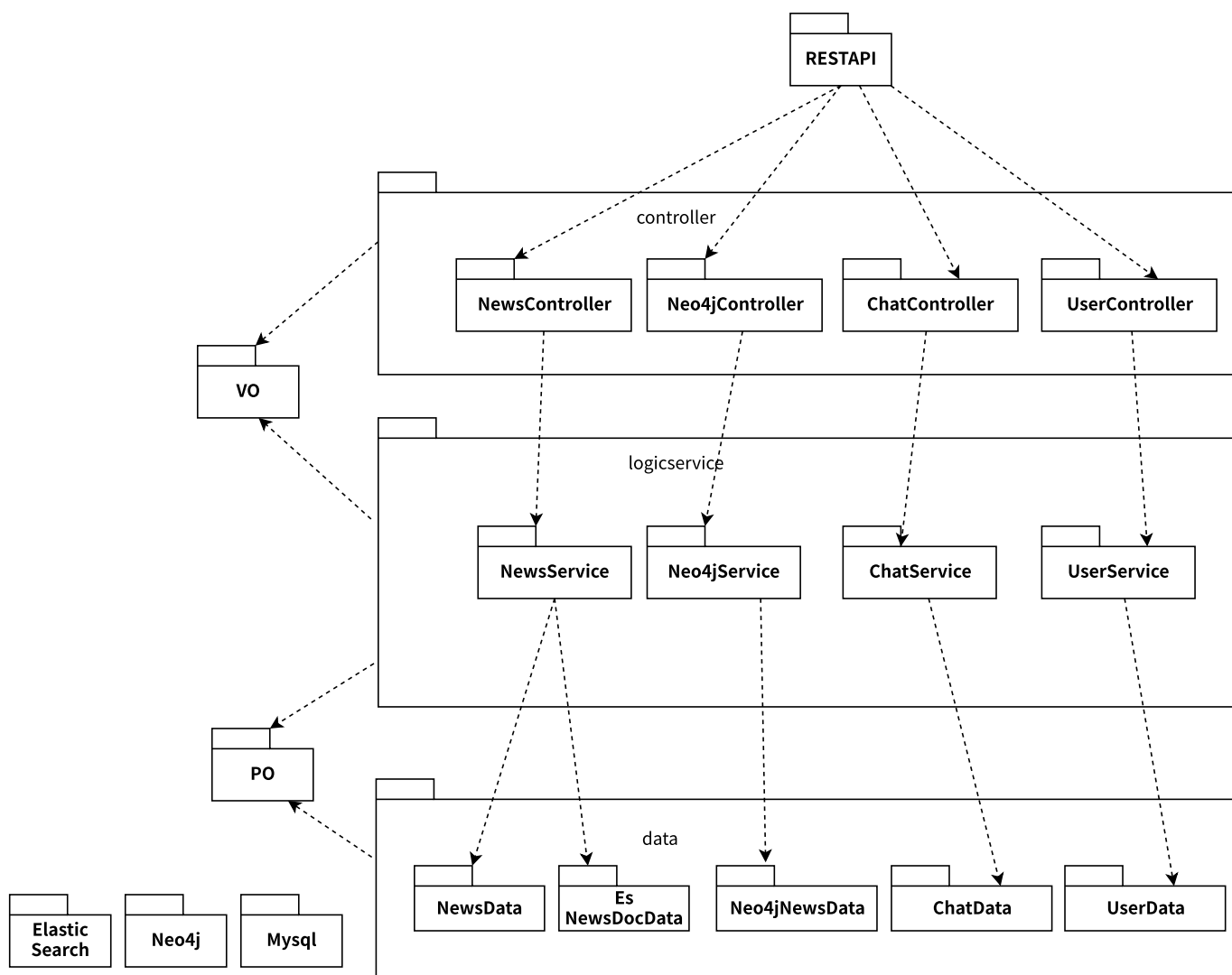
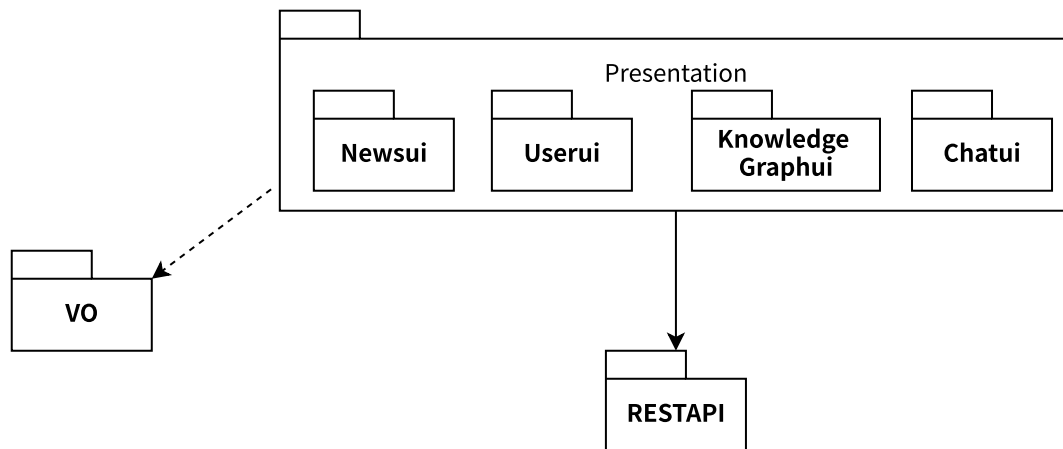
SpY项目设计文档

1. 体系结构设计

1.1 逻辑视角



1.2 组合视角



2. 数据持久化对象

MySQL

```
1 public class news
2 {
3     @Id
4     private Long id;
5     private String title;//标题
6     private Date date;//日期
7     private String content;//内容
8     private String source;//来源
9     private List<String> categories;//类别
10 }
```

```
1 public class NewsEvent {
2     @Id
3     private Long id;
4     private Long newsId;
5     private String event;
6     private String time;
7 }
```

```
1 public class ProvinceAndCity {
2     @Id
3     private Long id;
4     private String province;
5
6     @ElementCollection(fetch = FetchType.EAGER)
7     @CollectionTable(name = "province_city_mapping", joinColumns =
8         @JoinColumn(name = "province_id"))
9     @Column(name = "city")
10    private List<String> city;
11 }
```

```
1 public enum Category {
2     //财经、彩票、房产、股票、家居、教育、科技、社会、时尚、时政、体育、星座、游戏、娱乐。
3     // 成员
4     FINANCE("财经"),
5     LOTTERY("彩票"),
6     HOUSE("房产"),
7     STOCK("股票"),
8     HOME("家居"),
9     EDUCATION("教育"),
10    TECHNOLOGY("科技"),
```

```
11     SOCIETY("社会"),
12     FASHION("时尚"),
13     POLITICS("时政"),
14     SPORTS("体育"),
15     CONSTELLATION("星座"),
16     GAME("游戏"),
17     ENTERTAINMENT("娱乐");
18 }
```

Neo4j

```
1 public class NewsNeo4j {
2     @Id
3     @GeneratedValue
4     private Long id;
5     private String value;
6     private String type;
7     private Long newsId;
8
9     @Relationship(type = "Relation",direction = OUTGOING)
10    private List<RelationNeo4j> relationList;
11
12 }
```

```
1 public class RelationNeo4j {
2     @Id
3     @GeneratedValue
4     Long id;
5
6     @TargetNode
7     private NewsNeo4j newsEntity;
8
9     private String relationName;
10 }
```

ElasticSearch

```
1 public class NewsDoc {
2     @Id ()
3     private Long id;
4 }
```

```

5     @Field(type = FieldType.Text, searchAnalyzer = "ik_smart", analyzer =
      "ik_max_word")
6     private String title;          //标题
7
8     @Field(type = FieldType.Date)
9     private Date date;             //日期
10
11    @Field(type = FieldType.Text, searchAnalyzer = "ik_smart", analyzer =
      "ik_max_word")
12    private String content;         //内容
13
14    @Field(type = FieldType.Text)
15    private String source;          //来源
16
17    @Field(type = FieldType.Keyword)
18    private Category category;      //类别
19
20    @Field(type = FieldType.Double)
21    private Double standPoint;      //立场 保存的是小数
22
23    @Field(type = FieldType.Text)
24    private List<String> keywords;  //关键词
25 }

```

3. 接口

GET数据库记录总数

url:localhost:9010/spy/news/count

参数：

无需传递参数。

返回值：

```

1 {
2     "code":str,
3     "message":str,
4     "result":long
5 }

```

GET单个新闻 3.25新增

News getNews(Long id)

url:localhost:9010/spy/news/{id}

参数：

无需传递参数。请注意：news的id请在url的路径中传递。

返回值：

```
1 {
2   "code":str,
3   "message":str,
4   "result":{
5     "id":Long,
6     "title":str,
7     "date":str,
8     "content":str,
9     "source":str,
10    "category":str,
11    "standPoint":Integer, //立场分析, 3.25新增,范围-1~1; 中立 (-0.3~0.3) , 正面
    (0.3~1) , 负面 (-1~-0.3)
12    "keyWords":[str] //关键词数组, 3.25新增
13  }
14 }
```

GET搜索及根据时间、来源进行筛选

url:localhost:9010/spy/news/criteria

参数：（注意时间的格式@DateTimeFormat(pattern = "yyyy-MM-dd")）

```
1 {
2   "currentPage":long, //必要参数
3   "pageSize":long, //必要参数
4   "keyword":str, //必要参数, 无筛选项时传递空字符串
5   "startDate":str, //不是必要参数, 无筛选项时可传递空字符串, 传递参数的格式为 "yyyy-MM-dd"
6   "endDate":str, //不是必要参数, 无筛选项时可传递空字符串, 传递参数的格式为 "yyyy-MM-dd"
7   "source":str //必要参数, 无筛选项时传递空字符串
8 }
```

返回值：

```
1 {
```

```
2     "code":str,
3     "message":str,
4     "result":{
5         "list":[{
6             "id":long,
7             "title":str,
8             "date":str,
9             "content":str,
10            "source":str,
11            "categories":[{"category":str}]
12        }],
13        "totalCount":long
14    }
15 }
```

GET获取所有的来源

url:localhost:9010/spy/sources

返回值:

```
1 {
2     "code":str,
3     "message":str,
4     "result":[{"
5         "source":str
6     }]
7 }
```

有关于时间的格式

这个类对应的 JSON 形式可能如下所示：

```
json Copy code

{
  "id": "some_id_value",
  "title": "some_title_value",
  "date": "2024-02-29T12:00:00Z",
  "content": "some_content_value",
  "source": "some_source_value",
  "categories": ["category1", "category2", "category3"]
}
```

这是一个典型的 JSON 格式，其中包含了类中定义的字段。需要注意的是，日期字段在 JSON 中表示为字符串，并且遵循 ISO 8601 格式。

POST增加新闻

Void addNews(News news);

url:localhost:9010/spy/news

参数：

```
1 {
2   "news":{
3     "title":str,
4     "date":str,
5     "content":str,
6     "source":str,
7     "categories":[{"category":str}]
8   }
9 }
```

返回值：

```
1 {
2   "code":str,
3   "message":str,
4   "result":null
5 }
```


PUT修改新闻

Void updateNews (News news);

curl:localhost:9010/spy/news/{id}

参数：

请注意：news的id请在url的路径中传递。

```
1 {
2     "news":{
3         "title":str,
4         "date":str,
5         "content":str,
6         "source":str,
7         "categories":[{"category":str}]
8     }
9 }
```

返回值：

```
1 {
2     "code":str,
3     "message":str,
4     "result":null
5 }
```

DELETE删除新闻

Void deleteNews(Long id);

url:localhost:9010/spy/news/{id}

参数：

无需传递参数。请注意：news的id请在url的路径中传递。

返回值：

```
1 {
2     "code":str,
3     "message":str,
```

```
4     "result":null
5 }
```

数据分析模块

GET 获取近14日新闻数量 3.25新增

Url:<localhost:9010/spy/news/14num>

说明：返回最近十四天每日新闻数量，每周的可以算出来

参数：

```
1 {
2     "date":str, //需要获取的日期，以这个日期为当日，也就是需要根据实际需要的日期返回
3 }
```

返回值：

```
1 {
2     "code":str,
3     "message":str,
4     "result": [{
5         "date":str, //2024年3月12日类似日期
6         "num":long, //当日新闻数量
7     }]
8 }
```

GET 获取近两日每种新闻数量 3.25新增

Url:<localhost:9010/spy/news/2days/category>

说明：返回最近两天每种类别新闻数量，每种类别相比昨日变化量可以算出来

参数：

```
1 {
2     "date":str, //需要获取的日期，以这个日期为当日，也就是需要根据实际需要的日期返回
3 }
```

返回值：

```

1 {
2     "code":str,
3     "message":str,
4     "result": [{
5         "category":str,//类别
6         "todayNum":long,//当日该类别新闻数量
7         "lastdayNum":long,//昨日该类别新闻数量
8     }]
9 }
10 }

```

GET 获取14日所有新闻立场分析统计 3.25新增

Url:localhost:9010/spy/news/standpoint

说明：获取14日内所有新闻立场平均值,范围-1~1；以及中立（-0.3~0.3），正面（0.3~1），负面（-1~-0.3）的总新闻个数

参数：

```

1 {
2     "date":str,//需要获取的日期，以这个日期为当日，也就是需要根据实际需要的日期返回
3 }

```

返回值：

```

1 {
2     "code":str,
3     "message":str,
4
5     "result": {
6         "averageStandPoint": double,
7         "standPointNums": [
8             {
9                 "date": string,
10                "positiveNum": long,
11                "negativeNum": long,
12                "neutralNum": long
13            }
14        ]
15    }
16 }

```

GET 获取今日所有新闻关键词统计 3.25新增

Url:localhost:9010/spy/news/keyword

说明：获取今日所有新闻关键词统计，只取了出现最多的前1000个词

参数：

```
1 {  
2     "date":str, //需要获取的日期，以这个日期为当日，也就是需要根据实际需要的日期返回  
3 }
```

返回值：

```
1 {  
2     "code":str,  
3     "message":str,  
4     "result": [{  
5         "name":str, //关键词名称  
6         "value":long, //关键词数量  
7     }]  
8 }
```

GET 获取最近14天新闻的省份计数 4.8新增 缓

Url:<http://localhost:9010/spy/news/provinceCount>

参数：

```
1 {  
2     "date":str, //需要获取的日期，以这个日期为当日，也就是需要根据实际需要的日期返回  
3 }
```

返回值：

```
1 {  
2     "code": "2000",  
3     "message": "成功",  
4     "result": [  
5         {  
6             "name": "山东",  
7             "value": 108  
6         }  
5     ]  
4     }  
3     }  
2     }  
1 }
```

```
8      },
9      {
10         "name": "福建",
11         "value": 71
12     },
13     .....
14 }
```

GET 获取某篇新闻的事件 4.11新增

Url:<http://localhost:9010/spy/news/event/{id}>

参数：新闻的id

返回值：

```
1 {
2     "code": "2000",
3     "message": "成功",
4     "result": [
5         {
6             "id": 200,
7             "newsId": 1,
8             "event": "《中国企业家杂志》发布了小鹏汽车 CEO、董事长何小鹏的专访",
9             "time": "3月8日"
10        },
11        {
12            "id": 201,
13            "newsId": 1,
14            "event": "何小鹏谈到了当前车市竞争、公司发展及小米汽车相关话题",
15            "time": "3月8日"
16        },
17        {
18            "id": 202,
19            "newsId": 1,
20            "event": "何小鹏表示，“这就是一场正常的市场竞争，我觉得最好市场归市场。我觉得市场竞争就是用合适的资源，获得合适的能力，然后在竞争的情况下获取合适的价值。”",
21            "time": "3月8日"
22        },
23        {
24            "id": 203,
25            "newsId": 1,
26            "event": "何小鹏称希望上半年保持每个月同比增长百分之几十的增速，下半年同比增长超过 100%",
27            "time": "3月8日"
28        },
29    ]
30 }
```

```

29     {
30         "id": 204,
31         "newsId": 1,
32         "event": "何小鹏还谈到了自己的“老熟人”雷军和他的小米汽车：“我们有过很多交流，我觉得压力不大也是不可能的。小米有小米的优势，雷总把小米的优势用好，就大有机会。”",
33         "time": "3月8日"
34     }
35 ]
36 }

```

Neo4j模块

GET 根据newsId，返回对应新闻的三类信息(entityList、categories、relationLinks) 4.12新增

url: <http://localhost:9010/neo4j/getNews/{newsId}>

说明：

- 查询的时候先去查数据库如果数据库没有就使用大模型得到结果
- 大模型出错就返回response.error告诉出错了
- 如果数据库有数据就直接返回

返回值：

```

1  {
2      "code": "2000",
3      "message": "成功",
4      "result": {
5          "entityList": [
6              {
7                  "id": 0,
8                  "category": "地点",
9                  "name": "纽约州"
10             },
11             {
12                 "id": 1,
13                 "category": "机构",
14                 "name": "金融服务部 (New York Department of Financial Services, NYDFS) "
15             },
16             {
17                 "id": 2,
18                 "category": "机构",
19                 "name": "加密货币交易所Gemini Trust"

```

```

20         },
21         {
22             "id": 3,
23             "category": "人物",
24             "name": "Cameron和Tyler Winklevoss"
25         },
26         .....
27     ],
28     "categories": [
29         {
30             "name": "地点"
31         },
32         {
33             "name": "机构"
34         },
35         {
36             .....
37         },
38     "relationLinks": [
39         {
40             "source": 0,
41             "target": 11,
42             "name": "总部地点"
43         },
44         {
45             "source": 1,
46             "target": 0,
47             "name": "属于"
48         },
49         {
50             "source": 3,
51             "target": 2,
52             "name": "创始人"
53         },
54         .....
55     ]
56 }
57 }

```

```

1 {
2     "code": "-1",
3     "message": "大模型生成知识图谱失败",
4     "result": null
5 }

```

~~GET 根据newsId，返回实体的categories 4.2新增（已删除）~~

Url:localhost:9010/neo4j/category/{newId}

参数：

```
1 @PathVariable("newId") Long newId
```

返回值：

```
1 {
2     "code": "2000",
3     "message": "成功",
4     "result": [
5         {
6             "name": "组织"
7         },
8         {
9             "name": "人物"
10        },
11        {
12            "name": "地点"
13        }
14        .....
15    ]
16 }
```

~~GET 根据newsId，返回实体的linkData 4.2新增（已删除）~~

Url:localhost:9010/neo4j/relation/{newId}

参数：

```
1 @PathVariable("newId") Long newId
```

返回值：

```
1 {
2     "code": "2000",
3     "message": "成功",
4     "result": [
```



```

5      {
6          "source": 0,
7          "target": 16,
8          "name": "属于"
9      },
10     {
11         "source": 2,
12         "target": 1,
13         "name": "创始人"
14     },
15     .....
16 ]
17 }

```

~~GET 根据newsId，返回实体节点Data 4.2新增（已删除）~~

Url:<localhost:9010/neo4j/node/{newId}>

参数：

```
1 @PathVariable("newId") Long newId
```

返回值：

```

1 {
2     "code": "2000",
3     "message": "成功",
4     "result": [
5         {
6             "id": 0,
7             "category": "组织",
8             "name": "金融服务部"
9         },
10        {
11            "id": 1,
12            "category": "组织",
13            "name": "Gemini Trust"
14        },
15        .....
16    ]
17 }

```

GET 新闻可信度

备注:

- 如果新闻可信度已经计算，直接获得
- 如果新闻可信度未计算，计算后获得（有15秒左右延迟）

url:<http://localhost:9010/spy/news/credibility/{id}>

参数:

无需传递参数。请注意：news的id请在url的路径中传递。

返回值:

```
1 {
2     "code": "2000",
3     "message": "成功",
4     "result": {
5         "id": 1839, //不用管
6         "newsId": 7, //对应的新闻ID
7         "sourceScore": 0.0, //来源得分
8         "wordAndLogicScore": 85.0, //语言和逻辑得分
9         "sentimentScore": 81.74327313899994, //情感立场得分
10        "categoryScore": 100.0, //类别对应得分（新闻类别与媒体类别是否对
    应）
11        "totalScore": -0.17451574230242017 //归一化后的得分（-1, 1）
12    }
13 }
```

用户模块

POST注册

url:<http://localhost:9010/usr/register>

参数

```
1 //均不能为空
2 @RequestParam String userId
3 @RequestParam String pwd
```

返回值

```
1 {
2     "code": 200, //成功返回200, 失败返回500
3     "msg": str, //成功会自动登录, 所以返回登录成功; 失败返回注册失败
4     "data": null
5 }
```

GET登录

url:<http://localhost:9010/usr/doLogin>

参数

```
1 //均不能为空
2 @RequestParam String userId
3 @RequestParam String pwd
```

返回值

```
1 {
2     "code": 200, //成功返回200, 失败返回500
3     "msg": str, //成功会自动登录, 所以返回登录成功; 失败返回注册失败
4     "data": null
5 }
```

GET退出登录

url:<http://localhost:9010/usr/logout>

参数:无参数

返回值

```
1 {
2     "code": 200,
3     "msg": "退出登录成功",
4     "data": null
5 }
```

Chat模块

GET交互 缓

url:<http://localhost:9010/chat/question>

参数:

```
1 {  
2   question: str  
3 }
```

返回值

```
1 {  
2   "code":str,  
3   "message":str,  
4   "result": [{  
5     "answer":str  
6   }]  
7 }
```