注意：所有json文件应该是后端程序文件，需修改，即例如：index.cs、hot.cs，返回数据类型是json字符串，格式和样例保持一致

目录

[**1、** **index.json** 2](#_Toc61015488)

[**2、** **hot.json** 3](#_Toc61015489)

[**3、** **login.json** 4](#_Toc61015490)

[**4、** **return.json** 4](#_Toc61015491)

[**5、** **giveqs.json** 6](#_Toc61015492)

[**6、** **question\_detail.json** 6](#_Toc61015493)

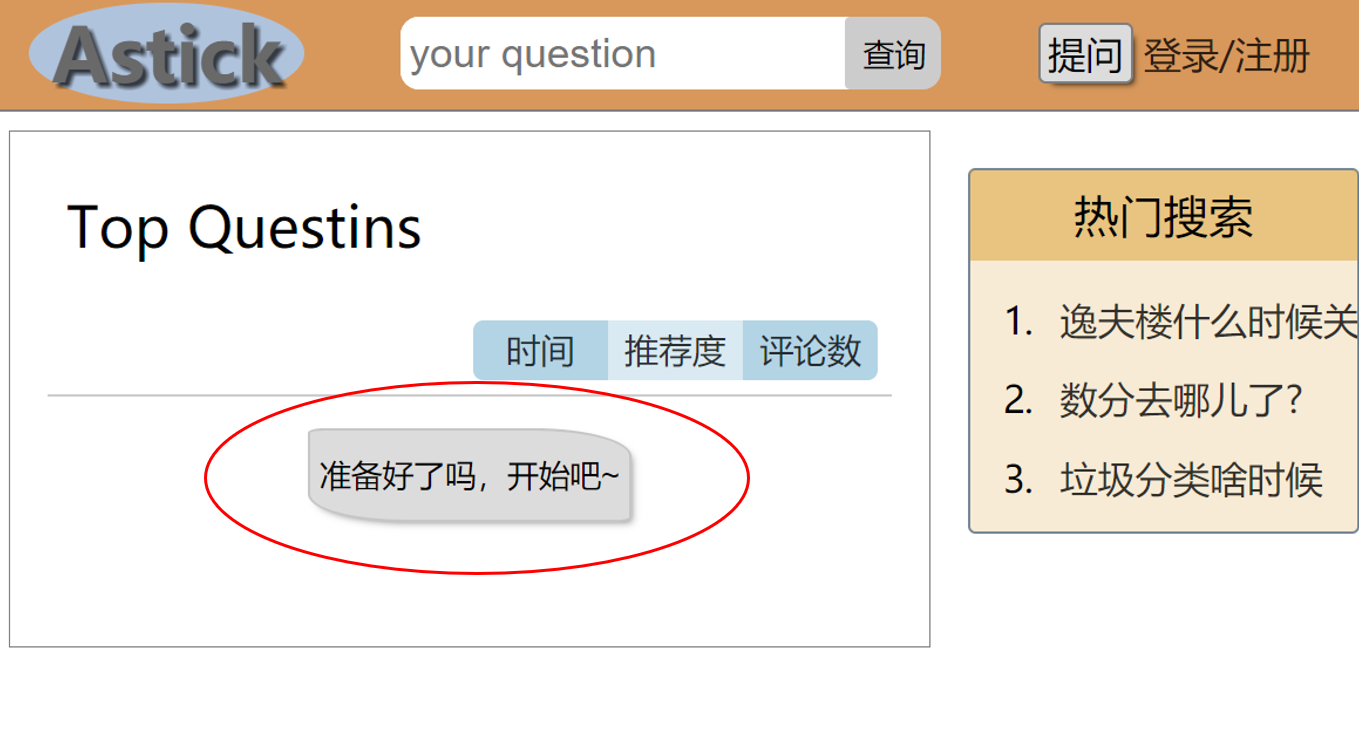
[**7、** **question\_addlike.json** 9](#_Toc61015494)

[**8、** **question\_answerQuestion.json** 9](#_Toc61015495)

[**9、** **comment.json** 10](#_Toc61015496)

1. **index.json**

（1）请求页面：index.html



（2）请求函数：index.js——getblock()

（3）介绍：

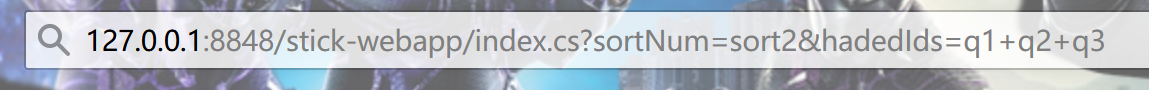
功能：获取问题链接和问题相关信息

触发：

该事件由index.html页面“加载更多”按钮触发，触发时会根据“创建时间”、“推荐度（点赞数）”、“评论数”进行数据请求。

请求方式：

数据请求方式时ajax的GET请求，所有请求信息都在URL信息中，如：



（index.cs是程序处理文件,sortNum表示根据创建时间、点赞数等进行数据请求，hadedIds表示当前页面已经存有的信息，”sort1”表示创建时间，“sort2”表示点赞数，“sort3”表示评论数，hadedIds表示当前页面已返回的数据id，避免重复申请，第一次申请hadedIds=null，之后hadesIds=q1+q2+q3，表示问题识别id）

（4）返回数据格式如下：

{

"q1":{ ——问题id

"title":"逸夫楼什么时候关门？", ——问题标题

"blockbr":{

"likeNum":1, ——点赞数

"commentNum":2, ——评论数

"username":"John", ——问题发表人名称

"givetime":"2020/5/6" ——发表时间

}

},

"q2":{

"title":"逸夫楼什么时候关门？",

"blockbr":{

"likeNum":1,

"commentNum":2,

"username":"John",

"givetime":"2020/5/6"

}

}

}

1. **hot.json**
   1. 请求页面：index.html



（2）请求函数：index.js——getHot()

（3）介绍：

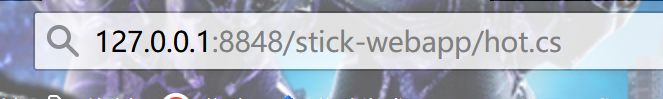
功能：获取后端推荐的关键词，用于检索

触发：

该事件会在index.html页面刷新时触发，触发时会向后端脚本请求数据。

请求方式：

Ajax的Get请求，如：



（4）返回数据格式如下：

{

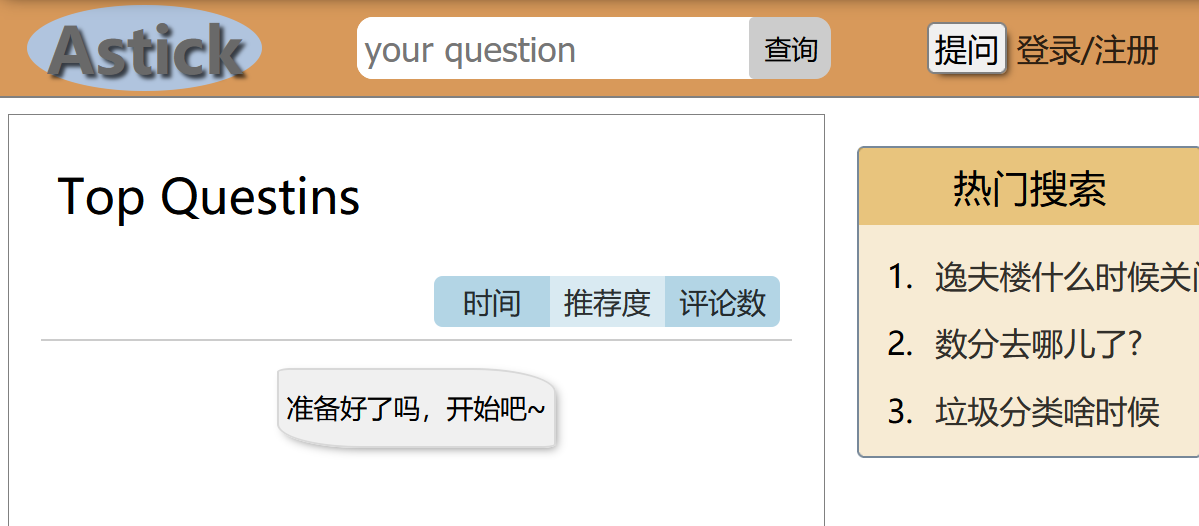
"a1":{"keyword":"逸夫楼什么时候关门"}, ——keyword：查询关键字

"a2":{"keyword":"数分去哪儿了?"},

"a3":{"keyword":"垃圾分类啥时候"}

}

1. **login.json**
2. 请求页面：所有页面都可以进行登录



1. 请求函数：loginform.js——loginResultDeal()
2. 介绍：

触发：

用户点击进行注册、登录时（注意：邮箱、名称等都已进行过格式检查处理，返回数据只需包括登录邮箱与密码是否与数据库保持一致）

请求方式：

Ajax的post请求，请求信息打包在数据包中，后端根据语言自身的对post请求的处理方式进行数据处理。

发送给后端的数据信息包括：

deal：1表示登录，2表示注册

username：注册名称（只有在注册时包含）

addr：登录或者注册邮箱（标准邮箱格式）

password：登录或者注册密码

1. 返回数据格式：

{

"status":1, ——操作状态：0：登录失败；1：登陆成功；2：注册失败;3:注册成功

"username":"John", ——用户名称

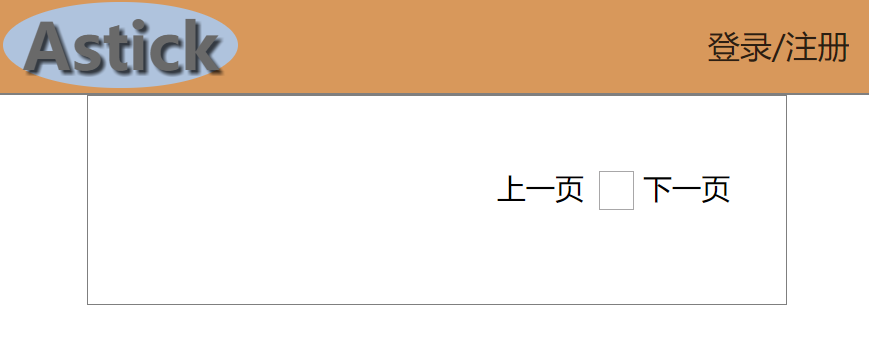
"addr":"965450327@qq.com", ——用户邮箱

"comment":"comment" ——操作备注信息，失败时会弹出备注信息，例如：“注册失败，注册邮箱已经被使用！”

}

1. **return.json**

（1）请求页面：return.html



（2）请求函数：return.js——reGetBlock()

（3）介绍：

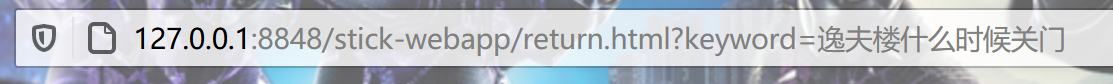
触发：

页面刷新时触发

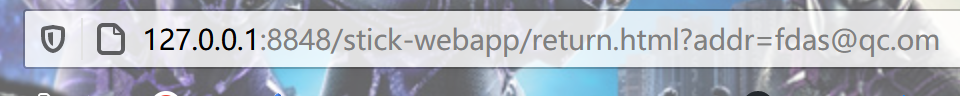
请求方式：

Ajax的Get请求，发送给后端的数据信息都保存在URL中，如：

（1）请求关键词是keyword的问题



1. 请求用户个人邮箱的问题：



· （4）返回数据格式（与index.json返回数据格式相同）：

{

"q1":{

"title":"逸夫楼什么时候关门？",

"blockbr":{

"likeNum":1,

"commentNum":2,

"username":"John",

"givetime":"2020/5/6"

}

},

"q2":{

"title":"逸夫楼什么时候关门？",

"blockbr":{

"likeNum":1,

"commentNum":2,

"username":"John",

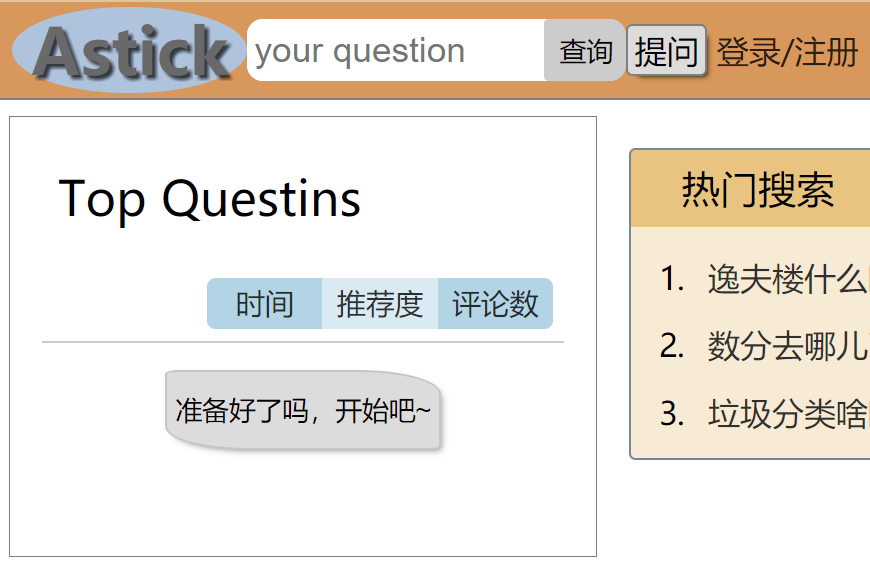
"givetime":"2020/5/6"

}

}

}

1. **giveqs.json**
2. 请求页面：Index.html



1. 请求函数：

index.js——prepareGiveQuesitonForm（）

1. 介绍：

功能：

用户进行提问

触发：

进行登录操作之后才可以进行消息发送，点击“提问”按钮触发

请求方式：

Ajax的post请求

发送给后端的信息如下：

addr：用户邮箱

title：问题标题

content： 问题正文

1. 返回数据格式：

{

"status":"1", ——提问操作状态

"comment":"发表成功，可在个人中心查看具体信息" ——操作备注信息，用于提示操作

}

1. **question\_detail.json**
2. 请求页面：Question.html



1. 请求函数： question.js——getQuestion()
2. 介绍：

功能：

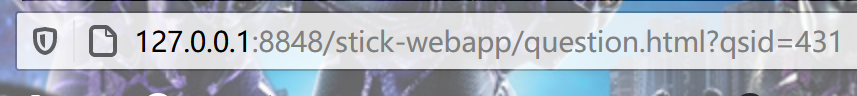
根据问题标识号（qsid）向后端获取具体的页面信息

触发：

页面刷新时触发，将URL中“qsid“关键字发送给后端，获取页面数据

请求方式：

Ajax的get请求，如：



1. 返回数据格式：

{

"status":"1", ——操作状态，1表示成功

"detail":{

"qsid":123414, ——问题id

"useraddr":"96545032@qew.com", ——发布者邮箱

"title":"标题", ——问题标题

"content":"问题详细内容", ——问题内容

"questionbr":{

"likeNum":2, ——问题点赞数量

"contentNum":3, ——评论数量

"username":"John", ——发布者名称

"givetime":"2020/05/06" ——创建时间

},

"answer":{ ——评论模块

"a1":{ ——第一层评论id

"addr":"fdas@qc.om", ——评论者邮箱

"username":"John", ——评论者姓名

"targetname":"\_host\_", ——接收评论者姓名

"comment":"我觉得不错", ——评论信息

"answertime":"2020/5/6", ——评论时间

"answer":{ —评论回复模块

"a1-a1":{ —第二层评论id

"addr":"fdas@da.c",

"username":"Clark",

"comment":"那我也觉得不错",

"targetname":"John",

"targetaddr":"fdfa@da.com",

"answertime":"2020/5/6"

}

}

},

"a2":{

"addr":"fdas@qc.om",

"username":"Marry",

"targetname":"host",

"comment":"我觉得不错",

"answertime":"2020/5/6",

"answer":{

"a2-a1":{

"addr":"fdas@qc.om",

"username":"Clark",

"comment":"那我也觉得不错",

"targetname":"Marry",

"targetaddr":"fdas@fdafda.cd",

"answertime":"2020/5/6"

}

}

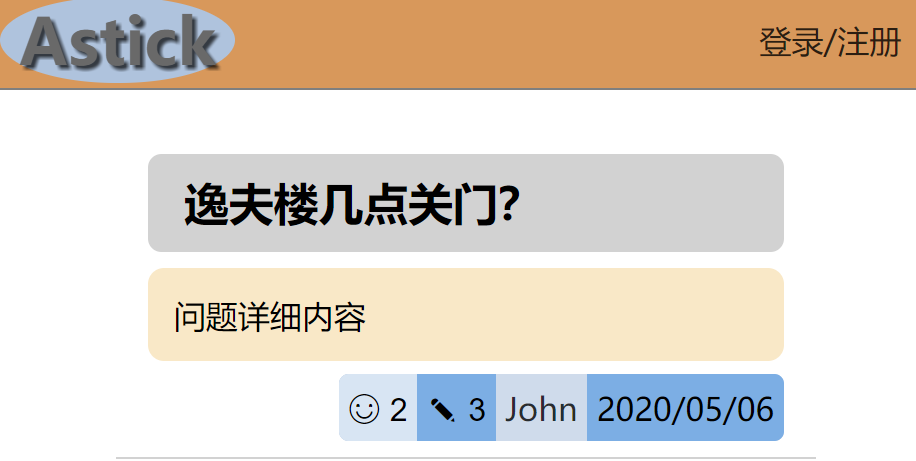
}

}

}

}

1. **question\_addlike.json**
   1. 请求页面：question.html



* 1. 请求函数： question.js——addLikeNum()
  2. 介绍：

功能：

实现用户对问题点赞功能

触发：

用户登录后点击“点赞”按钮实现触发

请求方式：

Ajax的POST请求

发送给后端的数据信息如下：

qsid：问题id

useraddr：点赞用户的id

* 1. 返回数据格式：

{

"status":1, ——操作状态，1表示点赞成功

"qsid":428717, ——问题id

"likeNum":8, ——问题的点赞数

"comment":"添加成功" ——操作备注信息

}

1. **question\_answerQuestion.json**
   1. 请求页面：question.html



* 1. 请求函数：question.js——answerQuestion（）

（3） 介绍：

功能：

实现评论页面的功能

触发：

用户登录以后，并在文本框中输入相应内容后，点击“回复”按钮进行回复

请求方式：

Ajax的post请求

发送给后端的数据信息如下：

qsid：问题id

addr：评论用户id

answer：评论内容

1. 返回数据格式：

{

"status":1, ——操作状态信息

"detail":{

"id":"a4", ——注意：需要给这条评论新建id

"username":"Marry", ——评论用户姓名

"addr":"fdasf@eq.com", ——评论用户邮箱

"comment":"我觉得不错", ——评论内容

"givetime":"2020/5/6" ——评论时间

}

}

1. **comment.json**
   1. 请求页面：question.html



* 1. 请求函数：question.js——answerComment（）
  2. 介绍：

功能：

实现给评论信息评论的功能

触发：

用户登录后，点击页面评论后“回复”按钮，输入回复信息后，点击“确定”按钮实现回复

请求方式：

Ajax的post请求，

发送给后端的请求数据信息如下：

addr：回复用户邮箱

qsid：问题id

answered：评论的id

comment：回复的具体内容

* 1. 返回数据格式：

{

"status":1, ——操作状态

"detail":{

"qsid":"q1", ——问题id

"where":1, ——接收回复的评论在第一层还是第二层（页面渲染时需要使用）

"answerid":"a1", ——接收回复的评论的id

"id":"a1", ——给新评论新建的id

"addr":"4739@eq.com", ——回复者的id

"username":"James", ——回复者的名称

"targetname":"Kobe", ——接收回复者的姓名

"targetaddr":"41@eq.cim", ——接受回复者的邮箱

"comment":"真好看" ——回复内容

}

}