1. 用户注册

|  |  |
| --- | --- |
| 路径 | /users/user |
| 请求方式 | POST |
| 参数类型 | 无特殊说明均为普通请求方式 |
| 参数 | {  email:xxx,  username:xxx,  password1:xxx,  password2:xxx } |
| 参数说明 | email:用户邮箱  username:用户昵称  password1:密码1  password2:密码2 |
| 返回类型 | json |
| 返回 | 成功：{“flag”:”success”}  失败：{“flag”:”fail”,content:”失败原因”} |

1. 用户登录

|  |  |
| --- | --- |
| 路径 | /users/loginResult |
| 请求方式 | POST |
| 参数类型 | application/json |
| 参数 | {email:xxx,password:xxx} |
| 参数说明 |  |
| 返回类型 | json |
| 返回 | 成功：{“flag”:”success”}  失败：{“flag”:”fail”} |

1. 查看id为{id}的用户个人信息

|  |  |
| --- | --- |
| 路径 | /users/user/{id} |
| 请求方式 | GET |
| 参数类型 | 整型 |
| 参数 | {id} |
| 参数说明 | 用户id（在请求路径上） |
| 返回类型 | json |
| 返回 | 成功：  {"result":"success","user":{"id":xxx,"email":"xxx","userName":"xxx","sex":xxx,"integral":xxx}}  失败：{“result”:”fail”} |

1. 更新id为{id}的用户性别

|  |  |
| --- | --- |
| 路径 | /users/u/{id}/sex |
| 请求方式 | POST |
| 参数类型 |  |
| 参数 | {sex:0（女）或1（男）} |
| 参数说明 | 用户id（在请求路径上） |
| 返回类型 | json |
| 返回 | 成功：{"result":"success"}  失败：{“result”:”fail”} |

发布问题

|  |  |
| --- | --- |
| 路径 | /questions/question |
| 请求方式 | POST |
| 参数类型 | application/json |
| 参数 | {  authorId:authorId,  authorName:authorName,  title:title,  content:content  } |
| 参数说明 | authorId:发表的用户id  authorName:发表的用户名  title:问题题目  content:问题内容 |
| 返回类型 | json |
| 返回 | 成功e.g:  {"flag":"success","content":{"id":5,"authorId":1,"authorName":"dragon","title":"test","content":"1111","numOfAnswers":null,"time":1483280177495}}  失败：{“flag”:”fail”} |

查看问题完整内容

|  |  |
| --- | --- |
| 路径 | /questions/q/{id}/detail |
| 请求方式 | GET |
| 参数类型 | 整型 |
| 参数 | {id} |
| 参数说明 | 问题id（在请求路径上） |
| 返回类型 | json |
| 返回 | 成功：  {  "result": "success",  "content": "现在大四\n\n大二升大三那年暑假，由于家里帮忙找了兼职，就没有会老家。正好舍友也有像简直就住在学校旁边，老房子像2000年左右的了，老鼠有成年汉子手掌那么大，反正很破。由于三个人住也没什么，这里说a和b(另外俩舍友)。没住几天a回老家当兵去了，和b在住后来b也走了去杭州玩了，mdzz我明显是被坑的那个。后来就我一个人住那个破房子单间，一楼，靠路边，旁边没人家。很恐怖。\n\n以上背景。\n\n我之后就一个人，后来住的地方旁边有条流浪狗，我觉得可怜，一般我吃晚饭快餐点两份，我吃一份(大荤)，帮狗带一份(最便宜的六块蛋炒饭)。然后一两天之后狗狗就在我门口不走了，我开门它还要进来，然后就趴在门口边，我怕野狗半夜发疯咬我喉咙怎么办，一直不许进来。它也乖，就趴在我门口，就这样半个月后。有天晚上差不多两三点左右，狗狗开始叫了(以前很安静一直没叫过)，就在我门口的地方，很凶的那种，很凶的那种，很凶的那种。我被吵醒，觉得很怪，但是害怕就没下床开门去看。那只狗就在门口叫了40分钟左右都没有停过，一直叫。我也听不见有人过路，一点声音都没有，只有狗叫，就算过路也不用过这么久，我也没在意，继续睡了。\n\n第二天早上上班出门，看见离我不远处的有户人家办丧，半夜死人了。我看了毛骨悚然。因为老家在农村，知道狗可以看见鬼。我就估计昨晚狗应该看见了脏东西，然后就为了护着我一直在门口叫，但是一想到昨晚有个脏东西就在我门外，屎差点吓出来。然后晚上请狗吃了顿好的，毕竟过命的交情。现在想想也蛮恐怖的。\n\n之后的日子狗也没有叫过，更加肯定了我的想法。"  }  失败：  {  "result": "fail"  } |

查看问题列表(按时间排序)

|  |  |
| --- | --- |
| 路径 | /questions/time |
| 请求方式 | GET |
| 参数类型 |  |
| 参数 | {  page:xxx,  size:xxx  } |
| 参数说明 | page：要查看的页数，不给则默认为第0页  size:每页要显示的个数，不给默认为2 |
| 返回类型 | json |
| 返回 | 成功e.g:  {  "last": true,  "content": [  {  "id": 11,  "authorId": 1,  "authorName": "dragon",  "title": "Python中list赋值时， L1=L 与 L1=L[:] 有什么区别？",  "summary": "严格的说，python没有赋值，只有名字到对象的绑定。\n\n所以L1=L是把L所指的对象绑定到名字L1上，而L2=L[:]则是把L通过切片运算取得的新列表对象绑定到L2上。前者两个名字指向同一个对象，后者两个名字指向不同对象。\n\n换句话说，L1和L是指的同一个东西，那么修改L1也就修改了L；L2则是不同的东西，修改L2不会改变L。\n\n注意这个引用的概念对于所有的东西都成立，例如容器内部存储的都是引用……\n",  "numOfAnswers": 0,  "time": 1483362920000  },  {  "id": 10,  "authorId": 1,  "authorName": "dragon",  "title": "什么是有效的关心？",  "summary": "现在大四\n\n大二升大三那年暑假，由于家里帮忙找了兼职，就没有会老家。正好舍友也有像简直就住在学校旁边，老房子像2000年左右的了，老鼠有成年汉子手掌那么大，反正很破。由于三个人住也没什么，这里说a和b(另外俩舍友)。没住几天a回老家当兵去了，和b在住后来b也走了去杭州玩了，mdzz我明显是被坑的那个。后来就我一个人住那个破房子单间，一楼，靠路边，旁边没人家。很恐怖。\n\n以上背景。\n\n我之后就一个人，后来住的地方旁边有条流浪狗，我觉得可怜，一般我吃晚饭快餐点两份，我吃一份(大荤)，帮狗带一份(最便",  "numOfAnswers": 1,  "time": 1483360431000  }  ]  }  失败：  {  "last": true,  "content": []  } |

查看问题列表(按热度(回答人数)排序)

|  |  |
| --- | --- |
| 路径 | /questions/hot |
| 请求方式 | GET |
| 参数类型 |  |
| 参数 | {  page:xxx,  size:xxx  } |
| 参数说明 | page：要查看的页数，不给则默认为第0页  size:每页要显示的个数，不给默认为2 |
| 返回类型 | json |
| 返回 | 成功e.g:  {  "last": true,  "content": [  {  "id": 10,  "authorId": 1,  "authorName": "dragon",  "title": "什么是有效的关心？",  "summary": "现在大四\n\n大二升大三那年暑假，由于家里帮忙找了兼职，就没有会老家。正好舍友也有像简直就住在学校旁边，老房子像2000年左右的了，老鼠有成年汉子手掌那么大，反正很破。由于三个人住也没什么，这里说a和b(另外俩舍友)。没住几天a回老家当兵去了，和b在住后来b也走了去杭州玩了，mdzz我明显是被坑的那个。后来就我一个人住那个破房子单间，一楼，靠路边，旁边没人家。很恐怖。\n\n以上背景。\n\n我之后就一个人，后来住的地方旁边有条流浪狗，我觉得可怜，一般我吃晚饭快餐点两份，我吃一份(大荤)，帮狗带一份(最便",  "numOfAnswers": 1,  "time": 1483360431000  },  {  "id": 11,  "authorId": 1,  "authorName": "dragon",  "title": "Python中list赋值时， L1=L 与 L1=L[:] 有什么区别？",  "summary": "严格的说，python没有赋值，只有名字到对象的绑定。\n\n所以L1=L是把L所指的对象绑定到名字L1上，而L2=L[:]则是把L通过切片运算取得的新列表对象绑定到L2上。前者两个名字指向同一个对象，后者两个名字指向不同对象。\n\n换句话说，L1和L是指的同一个东西，那么修改L1也就修改了L；L2则是不同的东西，修改L2不会改变L。\n\n注意这个引用的概念对于所有的东西都成立，例如容器内部存储的都是引用……\n",  "numOfAnswers": 0,  "time": 1483362920000  }  ]  }  失败：  {  "last": true,  "content": []  } |

插入评论：

|  |  |
| --- | --- |
| 路径 | /comments/comment |
| 请求方式 | POST |
| 参数类型 | application/json |
| 参数 | 对问题的回答：  {  questionId:xxx,  authorId:xxx  authorName:”xxx”,  summary:”xxx”  }  对评论的回答：  {  parentId:xxx,  questionId:xxx,  authorId:xxx,  authorName:”xxx”,  replyId:xxx,  replyName:”xxx”,  thread:”xxx”,  summary:”xxx”  } |
| 参数说明 | summary:回复的内容  parentId:回复的评论id  replyId:回复的用户id  replyName:回复的用户名  thread:回复的评论的thread+”/”+parentId |
| 返回类型 | json |
| 返回 | 成功e.g:  {  “flag”:”success”,  “content”:{  id:xxx  parentId:xxx,  questionId:xxx,  authorId:xxx,  authorName:”xxx”,  replyId:xxx,  replyName:”xxx”,  thread:”xxx”,  summary:”xxx”,  time:xxx  }  }  失败：{“flag”:”fail”} |
|  |  |