

前后端交互文档

前后端交互文档

修改记录

登录注册

登录

登出

注册

工人

个人信息界面

获取头像

编辑个人信息

排行榜

已完成

正在进行

访问任务

加入一级任务

查看ALL下属任务

查看具体下属任务

承接下属子任务

提交下属子任务

个人在一级任务中的参与

推荐任务列表

搜索任务

标注模块

登出

客服人员

登录

发布者

个人信息

任务创建

任务列表

任务详情

管理员

登录

创建客服人员

用户列表

数据分析

Staff

未接受的所有审核任务

已接受的所有审核任务

点开具体一级check任务

接受一级check任务

查看一级check任务下属小任务

查看参与的check小任务
接受某个check小任务
访问某个check小任务的第一张图片
访问某个check小任务的下一张图片
获取图片的标注信息
给标注结果评分

修改记录

修改人	修改时间	修改内容
廖均达、訾源	初稿	初稿
廖均达	2018-5-19	添加登录注册部分
訾源	2018-5-21	更新worker个人信息、任务列表等有关交互信息
訾源	2018-5-21	更新个人在任务中的参与部分有关交互信息
廖均达	2018-5-23	将发布者的返回信息中的所有 "fail" 换成了 "failure"
廖均达	2018-5-24	修改发布者任务列表以及任务详细信息的返回字段信息
廖均达	2018-5-28	添加管理员登录、注册部分
廖均达	2018-6-1	修改管理员数据分析部分
訾源	2018-6-3	增加staff部分
廖均达	2018-6-9	发布者创建任务，任务列表，任务详情三个部分添加返回 taskRequirement，发布者添加修改任务优先级接口（见发布者任务详情处）
廖均达	2018-6-16	增加数据图表
廖均达	2018-6-17	管理员用户列表模块点击用户返回该用户统计信息，完善管理员数据分析模块

登录注册

登录

```
url: "login.html"
method: GET
param:
  "email" : "excited@mail.com",
  "password" : "asdfgsdfjk",
  "userType" : "worker|requester"
return
{
  "message" : "balabala|null", // 登录失败为null
  "status" : "success|failure",
  "token" : "AdfaDAdfjakAdjf|null"
}
```

登出

```
url: "logout.html"
method: GET
param:
  "token" : "ASdfaSDFAFasdafas"
no return message
```

注册

```
url: "checkEmail"
method: GET
param:
  "email" : "fasdfds@mail.com"
return
{
  "status" : "valid|invalid|duplicate" // invalid主要是为了验证邮箱是否真实有效
}

url: "signup.html"
method: GET
param:
  "email" : "excited@1926.com"
  "password" : "sajdfkjjae"
  "name" : "frog"
  "userType" : "worker|requester"
return
{
  "message" : "balabala|null", // 注册成功则为null, 失败则返回error原因
  "status" : "success|failure",
  "token" : "AdfSdaCasf|null"
}
```

工人

个人信息界面

前端访问url: /worker/userInfo/main.html

Get方法, 需要参数token。

后端返回:

```
"avatar": "xxxxx", // url
"userName": "xxx",
"email": "xxxx",
"lastVisit": "xxxx",
"rank": 8048000
"joint": "xxxx",
"dollars": x.xx,
"division": "xxx" (可能的取值: Novice, Contributor, Expert, Master, Grandmaster)
"score": xx.x
```

前端访问url: /worker/userInfo/recent.html

Get方法, 需要参数token。

```
"dollarChanges": [
  {"date": "17-03-01", "value": 13.3},
  ...
]
"scoreChanges": [
  {"date": "17-03-01", "value": 18.3},
  ...
]
"Activity": [
  {"date": "17-03-01", "value": 18.3},
  ...
]
] // 365个数字, 表示活跃度, 瑞年要不要考虑回头再议, 反正2018年不是瑞年
```

获取头像

一般说来, 在其它界面, 也要有获取头像的功能。

前端访问url: /worker/userInfo/avatar.html

GET方法, 需要参数:
token

返回一个url, 表明了头像的存储位置。

编辑个人信息

编辑用户名:

前端访问url: /worker/userInfo/editBasicInfo.html

GET方法, 需要参数:

```
token,
userName
返回
state: "xxx" // 可选: success, fail
// 如果success了, 就重新访问一下个人信息界面

编辑头像:
前端访问url: /worker/userInfo/editBasicAvatar.html
Post方法, 需要参数
token,
avatar
返回
state: "xxxx" // 可选: success, fail(不知道为什么失败了)

编辑密码:
前端访问url: /worker/userInfo/editPassword.html
GET方法, 需要参数
token,
oldPassword
newPassword
返回:
state: "xxxx" // 可选: success, failure(不知道为什么失败了), invalid password(原来的密码是错的)
```

排行榜

前端访问url: /worker/userInfo/rank.html

Get方法, 需要参数:

token

返回一个

```
[
  {
    "userName": "xxx",
    "score": xxx.x,
    "division": "xxx",
    "avatar": "xxxxx/xxx.jpg"
  },
  ...
]
```

已完成

前端访问url: /worker/userInfo/finished.html

Get方法, 需要参数:

token

返回一个列表, 列表中的每一项都标志着一个一级任务

```
[
  {
    "name": "xxx" // 任务名
    "taskId": 2333 // 任务ID
    "taskDescription": "xxxx" // 任务描述
    "taskCover": "xxxx/xxx/xx.jpg" // 一个url，表示任务的封面，以固定的尺寸显示！
    "earnedDollars": xx.xx // 2位小数，表示挣来的钱数
    "changeOfScore": xx.xx // 做的这个任务引起的积分变化，可正可负
    "taskTypes": [401, 402] // 这个一级任务中包含的任务的类型
    "endDate": "2017-10-1" // 任务截止的时间
    "taskTags": ["nature", "history"] // 任务的一些标签信息，就是任务的主题
    "totalDollars": xxxx // 显示不显示无所谓
  },
  ...
]
```

正在进行

前端访问url: /worker/userInfo/doing.html

Get方法，需要参数：

token

返回一个列表，列表中的每一项都标志着一个一级任务

```
[
  {
    "name": "xxx" // 任务名
    "taskId": 2333 // 任务ID
    "taskDescription": "xxxx" // 任务描述
    "taskCover": "xxxx/xxx/xx.jpg" // 一个url，表示任务的封面，以固定的尺寸显示！
    "taskTypes": [401, 402] // 这个一级任务中包含的任务的类型
    "endDate": "2017-10-1" // 任务截止的时间
    "taskTags": ["nature", "history"] // 任务的一些标签信息，就是任务的主题
    "totalDollars": xxxx
  },
  ...
]
```

访问任务

点击某个任务是可以访问的！（访问某个具体的一级任务）

前端访问url: /worker/task/index.html

Get方法，需要参数：

token

taskId

后端返回：

```
{
  "name": "xxxx"
```

```

"taskId": xxx
"taskCover": "xxxxx"
"taskDescription": "xxxx"
"endDate": "2017-1-10"
"state": "xxx" // 可能的值 Finished, Going
"participated": true // 是否已经参与了这个任务
"requiredDivision": "xxx" // 表示段位的一个枚举值
"taskTags": ["军事", "历史"]
"taskTypes": [401, 402]
"taskBackground": "一段文字，描述了下任务的背景，这个是迭代二没有的"
"totalDollars": xxxx // 价格
"canAccept": true // 用户没有接过这个任务 && 用户满足接受条件 && 正在进行中，就可以接受
"earnedDollars": xxxx.xx // 如果用户参与了这个任务而且这个任务已经完成，这个字段才有效，否则就是0
"scoreChange": xxxx.x //如果用户参与了这个任务而且这个任务已经完成，这个字段才有效，否则就是0
这两个字段如果无效，就不显示
}

```

加入一级任务

用户可以加入某个一级任务！
 这个在迭代二中没有
 前端访问url: /worker/join.html

需要参数：

token
 taskId

返回

```

{
"result": true
"reason": "xxxxx" // 如果是true的话，那就是空的
}

```

查看ALL下属任务

总的来说，就是和overview并列的bar上面的第二个里面，有可能有很多标注方式(taskType)，这个是根据taskType访问子任务的方法

前端访问url: /worker/task/subTaskSet.html

需要参数：

token
 taskId
 taskType

返回

```

[
{
"subTaskId": xxxxxx
}
]

```



```
    "taskId": xxxx
    "cover": "xxxxx" // 是一个URL
    "picCount": 66 // 图片数量
  },
  ...
]
```

查看具体下属任务

下属任务也是可以具体查看的具体内容。

前端访问url: /worker/task/subTask/details.html

需要参数:

token

taskId

subTaskId

taskType

1. 如果是没有接过的任务

返回:

```
{
  "taskId": "xxx"
  "subTaskId": "xxxx"
  "taskState": "unAccepted",
  "taskName": "xxx",
  "taskType": 401,
  "taskDescription": "xxxxxxxxxx",
  "picList": [
    "/image/xxx/xx.jpg",
    "/image/xxx/xx.jpg"
  ]
}
```

2. 如果是已经接过的任务

返回:

```
{
  "taskId": "xxx"
  "subTaskId": "xxxx"
  "taskState": "Accepted"
  "taskName": "xxx",
  "taskType": 401,
  "taskDescription": "xxxxxxxxxx",
  "finishedPicList": [
    "/image/xxx/xx.jpg",
    "/image/xxx/xx.jpg"
  ],
  "unFinishedPicList": [
    "/image/xxx/xx.jpg",
    "/image/xxx/xx.jpg"
  ]
}
```

```
"endDate": "xxxx/xx/xx xx:xx:xx" /  
}
```

提交或者结算了就不给他看了👀👀

承接下属子任务

worker可以承接已经join的任务的下属子任务

前端访问url: /worker/task/subTask/accept.html

需要参数:

token

taskId

taskType // int来表示

subTaskId

后端返回:

result: true

description: "xxxx" // 如果返回true的话, 就可以无视
// 这个description前端可以不处理, 显示为未知错误就行了

提交下属子任务

worker可以提交已经join的任务的下属子任务

前端访问url: /worker/task/subTask/commit.html

需要参数:

token

taskId

subTaskId

后端返回:

result: true

description: "xxxx" // 如果返回true的话, 就可以无视
// 这个description前端可以不处理, 显示为未知错误就行了

个人在一级任务中的参与

worker可以查看自己在以及任务中的参与状况情况

访问url: /worker/task/myparticipation.html

参数

token

taskId

taskState // 这儿要用户去选择

--> 这个参与情况类似于迭代二的任务状态, 有等待结算, 过期, 进行中。 // waiting, expired, doing

--> 当这个一级任务结束后，所有参与的小任务只有已完成、过期两个状态。 // expired, finished

--> 已完成/过期/等待结算不能点进去看

```
[
  {
    "cover": "xxxx/xxxx" // 封面
    "expiredDate": "xxxxxx"
    "commiteDate": "xxxxx" // 可能是空的，如果没提交就是空的
    "subTaskId": xxxx
    "taskType": xxxx // 任务类型
    "process": 0-100之间的整数
    "picAmount": xxx // 图片数量
  }
]
```

推荐任务列表

推荐任务本身可以加筛选条件的。

前端访问url: /worker/task/recommendation.html

Get方法，需要参数：

token

后端返回 // 这儿的字段和用户正在做的任务的列表是一致的，推荐的任务只能是用户没有接的。

```
[
  {
    "name": "xxx" // 任务名
    "taskId": 2333 // 任务ID
    "taskDescription": "xxxx" // 任务描述
    "taskCover": "xxxx/xxx/xx.jpg" // 一个url，表示任务的封面，以固定的尺寸显示！
    "taskType": [401, 402] // 这个一级任务中包含的任务的类型
    "endDate": "2017-10-1" // 任务截止的时间
    "taskTag": ["nature", "history"] // 任务的一些标签信息，就是任务的主题
    "totalDollars": xxxx // 奖池总金额
  },
  ...
]
```

搜索任务

搜索任务本身也是可以加筛选条件的。

前端访问url: /worker/task/search.html

Get方法，需要参数：

token

taskType // 指的是401, 402之类的，如果用户没选，那就是传过来0

taskTag // 任务的题材，可能是军事啊，自然啊之类的，之后会有一个可能取值的表格

--> military,
nature,
sports,

```

    humanity,
    science,
    politics,
    others,
    all; // all 表示全部都返回，这个要传回来一个字符串
rankType // 排序方式，提供默认排序，按照金钱数量降序，按照金钱升序
--> DEFAULT,
    MONEY_DESCEND,
    MONEY_ASCEND; // 传回来一个字符串
begin // 起始位置，从0开始
step // 请求的推荐任务的数量，迭代三应该实现下拉刷新，所以就不设置页了
key // 字符串，用户为搜索提供的关键字
canAccept // 筛选条件，指的是用户是否可以接受，如果是false就返回所有的，如果是true就只返回可以接受的

后端返回 // 这儿的字段和用户正在做的任务的列表是一致的，搜索的任务只能是用户没有接的。
[
    {
        "name": "xxx" // 任务名
        "taskId": 2333 // 任务ID
        "taskDescription": "xxxx" // 任务描述
        "taskCover": "xxxx/xxx/xx.jpg" // 一个url，表示任务的封面，以固定的尺寸显示！
        "taskType": [401, 402] // 这个一级任务中包含的任务的类型
        "endDate": "2017-10-1" // 任务截止的时间
        "taskTag": ["nature", "history"] // 任务的一些标签信息，就是任务的主题
        "totalDollars": xxxx // 奖池总金额
    },
    ...
]

```

标注模块

直接复用之前代码，不过user的验证信息全部变成token，外加subTaskId

登出

前端访问url: `/logout.html`

Get参数:

token

无返回信息。

客服人员

登录

和管理员共用一个url

发布者

个人信息

```
url: "requester/userInfo/main.html" // 主要信息
method: GET
param:
token
return
{
  "name" : "xxx",
  "email" : "xxx",
  "signMessage" : "Joined x ago last seen in x day",
  "dollars" : "9,999.20",
  "avatar" : "jsaldfjasdl.png",
}

url: "requester/userInfo/accountInfo.html" // 账户变动
method: GET
param:
"token" : 2341234
"page" : 1
"pageSize" : 20
返回一个矩阵，每个行向量中的值从左到右为：变动金额，变动原因，余额，变动时间， 降序排列
return
[
  ["+1,000.2", "change type", "2,000.2", "2018-5-15 12:00"]
  ["-2,000,000.2", "change type", "0", "2018-4-30 11:00"]
]

url: "requester/userInfo/recharge.html"
method: POST
param:
{
  "token" : "198372",
  "dollars" : 2832000.23,
}

// 如果 status 为 success, 那么 message 为 null
return {
  "message": "balabla",
  "status" : "success|failure"
}

url: "requester/userInfo/checkOldPassword.html" // 检查用户旧密码的正确性

url: "requester/userInfo/changeAvatar.html" // 修改用户信息
method: POST
```

```

param:
{
    "avatar" : "string",
}
url: "requester/userInfo/changeName.html"
param:
{
    "name" : "sdfasfd",
}
url: "requester/userInfo/changePassword.html"
{
    "oldPassword" : "123424",
    "newPassword" : "asfjiesd"
}
// 上面三个方法返回形式均如下
return
{
    "message" : "null|balabala",
    "status" : "success|failure"
}

```

任务创建

```

url: "requester/new/info.html" // 返回系统预设标签, 预设任务类型
method: GET
return
{
    "tags" : ["a", "b", "c"],
    "divisions" : ["Novice", "Contributor", "Expert", "Master", "Grandmaster"]
}

url: "requester/new/taskOrder.html" // 新建任务
method: POST
param:
{
    "token" : "asdfkjdk13df",
    "cover" : "string", // 可选, 和头像上传一样
    "backgroundImage" : "string", // 可选, 和头像上传一样
    "title" : "balabala",
    "tags" : ["a", "b", "c"],
    "description" : "balabala",
    "tasks" : [
        {
            "type" : 100,
            "description" : "bala",
            "labels" : ["a", "b", "c"]
        },
        {
            "type": 101,
            "description" : "bala",
            "labels" : ["a", "b", "c"]
        }
    ]
}

```

```

    }
  ],
  "dataset" : "multipart-file",
  "readme" : "balabala",
  "deadline" : "2018-5-20",
  "lowestDivision" : "novice",
  "taskRequirement" : "common, speed, quality"
}
return
{
  "status" : "success|faliure",
  "message" : "null|balabala",
  // 如果返回的 status 是成功则有下面两个字段, 否则没有
  "orderId" : "dsfdsf", // 订单编号
  "pictureNum" : 1234,
  "payLowerBound" : 21412342.11, // 最少付多少
}

url: "requester/new/pay.html"
method: GET
param:
"token" : "1234AS234234",
"status" : "accept|reject", // 如果拒绝了, 后面的就不用传了或是传一个null, 方便为好
"orderId" : "dfadsfa",
"dollars" : 123432.11,
"advertisementDollars" : 2341234

return {
  "status" : "success|failure",
  "message" : "null|balabala"
}

```

任务列表

```

url: "requester/task/ongoing.html" // 正在进行中的任务列表
method: GET
param:
"token" : 21423
return
[
  {
    "taskId" : 12312,
    "title" : "balabala",
    "description" : "balabala",
    "cover" : "url",
    "tags" : ["a", "b", "c", "d"],
    "dollars" : 200.1,
    "participantsNum" : 1234,
    "pictureNum" : 1234,
    "timeInfo" : "upload 2 months ago, 30 days to go",
    "workerRequirement" : "novice",
  }
]

```

```
        "taskRequirement" : "common, speed, quality"
    }
]

url: "requester/task/done.html" // 已经完成的任务列表
method: GET
return 和上面一模一样
```

任务详情

```
url: "requester/task/sketch.html" // 任务基本描述，即上面那一块显示的内容
method: GET
param:
    "taskId" : 1234
return
{
    "title" : "balabala",
    "taskId" : 12324,
    "cover" : "url",
    "description" : "balabala",
    "tags" : ["a", "b", "c", "d"],
    "dollars" : 12324.11,
    "backgroudImage" : "url",
    "participantsNum" : 1234,
    "pictureNum" : 1234,
    "types" : [100, 101, 200],
    "timeInfo" : "upload 2 months ago, 30 days to go",
    "deadline" : "2018-5-12 23:59",
    "process": 0.86,
    "state" : "on_going|finished",
    "taskRequirement" : "common, speed, quality"
}

url: "requester/task/participants.html" // 参与者排行榜，前20将被显示
method: GET
param:
    "taskId" : 1234
return
[
    {
        "name" : "frog",
        "avatar" : "frog.jpg",
        "level" : "novice",
        "pictureNum" : 123,
    }
]

url: "requester/task/readme.html" // 任务的详细描述，类似于github的readme
method: GET
param:
    "taskId" : 1234
```



```
return
{
    "readme" : "balabala"
}
```

url: "requester/task/subtask.html" // 子任务信息

method: GET

param:

"taskId" : 1234,

return

```
[
    {
        "type" : "type1",
        "process" : 0.87,
        "participantsNum" : 100,
        "description" : "asfadsfss",
        "participants" : [
            {
                "name" : "frog",
                "avatar" : "frog.jpg",
                "level" : "novice",
                "pictureNum" : 123,
            }
        ]
    }
]
```

url: "requester/task/download.html" // 下载任务结果，需要先判断总任务进度是否为1

method: GET

param:

"taskId" : 23413

return

"filepath"

url: "requester/task/editReadme.html" // 修改readme的内容

method: POST

param:

```
{
    "taskId" : 1234,
    "readme" : "content"
}
```

return

```
{
    "message" : "null|content",
    "status" : "success|fail"
}
```

url: "requester/task/changeTaskRequirement.html"

method: GET

param:

"taskId" : 232423

"taskRequirement" : "common, speed, quality"

```
return {
  "message" : "null|something",
  "status" : "success|failure"
}
```

管理员

登录

```
url: "inside/login"
method: GET

param:
{
  "username" : "admin", // 客服人员实际上是email, 但是此处名称仍为username
  "password" : "admin" // 初始值均为 admin
}
return
{
  "status" : "success|failure",
  "message" : "null|balabala",
  "token" : "aslkfjasdvx313jlk4j32"
}

url: "inside/new/admin" // 创建新的管理员
method: GET

param:
{
  "token" : "asfjasldfjei", // 主要用于验证当前管理员权限
  "newAdminName" : "admin2",
  "password" : "password",
}

return {
  "status" : "success|failure",
  "message" : "null|balabala"
}
```

创建客服人员

```
url: "admin/newStaff"
method: GET

param:
{
  "token" : "bas34jk31ld1334nddf",
```

```

    "email" : "frog@naive.com", // staff name
    "password" : "sometimesnaive"
  }

  return {
    "status" : "success|failure",
    "message" : "null|balabala"
  }
}

```

用户列表

说明：

- GET 方法传递的参数中的 `page` , `pageSize` 主要是用来实现下拉显示更多的内容, `page`表示第几页, `pageSize` 用来表示一次刷新获得多少 user 的信息

```

url: "admin/workers.html" // 默认返回排行的工人
method: GET
param:
  "page" : 1 // start from 1
  "pageSize" : 20
return
[
  {
    "name" : "frog",
    "email" : "12341234@mail.com",
    "avatar" : "frog.jpg",
    "signupTime" : "2018-5-15",
    "dollars" : 2000,
    "division" : "novice",
    "score" : 1342
  }
]

url: "admin/workerdetails.html"
"email" : "123@mail.com",
"token" : "sdi12331sdf"
return {
  "taskType" : [12, 23, 43, 12, 21, 12, 12, 12], // 雷达图, 各个标注类型(8)的能力值
  "taskTag" : [1, 1, 1, 1, 1, 1], // 6 中标签类型
  "workerSpeed" : 13, // pictures per day
  "workerTaste" : [123, 213, 213, 22, ..., 132] // 对八种标注类型的喜爱程度
}

url: "admin/requester.html" // 默认返回发布任务数量最多的发布者
method: GET
param:
  "page" : 1
  "pageSize" : 20 // 如果是0的话就会返回所有的
return
[

```

```

    {
        "name" : "frog",
        "avatar" : "frog.jpg",
        "signupTime" : "2018-5-15",
        "dollars" : 2000,
        "tasksNum" : 1342,
        "email" : "2134213@mail.com"
    }
]

url: "admin/requesterTasks.html" // 返回发布者各个任务的一些统计数据
param:
"email" : "requester@mail.com",
"token" : "idsafji3j4i2j3ij42"
return [
    {
        "tagError" : 0.99, // 到目前为止的评估标注错误率, 如果还没有人进行标注就为0
        "taskId" : 1232,
        "adDollars" : 1232.12, // 该任务所缴纳广告费
        "wokerGetDollars" : 21323.11, // 工人总共拿了多少
        "prizeDollars" : 1234213, // 奖金总共多少
        "backDollars" : 123423, // 返回给了发布者多少
        "costTime" : 12, // 天为单位
        "participatedPeopleNum" : 1231
    }
]

url: "admin/requesterDetails.html"
param:
"email" : "requester@mail.com",
"token" : "diasldfj32jj32ji4j23"
return {
    "exptected"
}

url: "admin/searchUser.html" // 搜索用户
method: GET
param:
"key" : "balabala"
"userType" : "worker|requester"
"page" : 1
"pageSize" : 20
return: 和上面的worker, requester的信息一模一样

url: "admin/modify/password.html" // 修改用户数据
method: POST
param:
{
    "token" : "asldfj3j42314j234", // admin 的账号
    "email" : "email@email.com", // user 的账号
    "newPassword" : "1234123",
}

```

```
url: "admin/modify/dollars"
method: POST
param:
{
  "token" : "a234j34jl3m34i23n",
  "email" : "email@email.com",
  "dollars" : "134555"
}
return
{
  "message" : "null|content",
  "status" : "success|fail"
}
```

数据分析

```
url: "admin/statistic/activeUser.html" \\ 活跃用户
method: GET
return:
{
  "time" : ["time", "time", "time", "time"],
  "totalData" : [123, 123, 234, 234],
  "workerData" : [123, 123, 234, 234],
  "requesterData" : [123, 123, 234, 234],
  "exptected" : 123,
  "workerExpected" : 123,
  "requesterExptected" : 123
}

url: "admin/statistic/signUpUser.html"
method: GET
return:
{
  "time" : ["time", "time", "time", "time"],
  "userData" : [123, 123, 234, 234],
  "workerData" : [123, 123, 234, 234],
  "requesterData" : [123, 123, 234, 234],
  "exptected" : 123
}

url: "admin/statistic/tasks.html"
return:
{
  "time" : ["time", "time", "time", "time"],
  "doneNumber" : [123, 123, 234, 234],
  "commitNumber" : [123, 123, 234, 234],
  "releaseNumber" : [123, 321, 133, 312]
}

url: "admin/statistic/taskPieChart.html"
```

```

return:
{
    "t_100" : 0.2,
    "t_101" : 0.1,
    ...,
    "t_401" : 0.05// 八种标注方式
}

url: "admin/statistic/workerdistribution.html" // worker 段位分布
return:
{
    "Novice" : 0.1,
    "Contributor" : 0.1,
    "Expert" : 0.2,
    "Master" : 0.2,
    "GrandMaster" : 0.4
}

url: "admin/statistic/workerevaluation.html" // worker的能力统计(按照段位)
return:
{
    // 数组中从左到右分别是 Novice, Contributor, Expert, Master, GrandMaster 的数值
    "tagError" : [0.23, 0.34, 0.11, 0.23, ..., 0.01], // 均值
    "speed" : [1234, 234, 34, ..., 34], // 单位是 pictures per day(均值)
    "ability" : [234, 324, 34, ..., 423], // 这个能力值统计, 数值代表用户的能力向量的长度
    (均值)
    "acceptedTasks" : [
        [1010, 1200, 4300, ..., 2301],
        [3001, 2000, 4011, ..., 1010],
        [3002, 2011, 2012, ..., 1000],
        ...,
        [312, 32132, 13, ..., 312]
    ], // 各个段位用户各种标注方式接受次数统计(接受一个小任务算一次)
}

url : "admin/statistic/taskevaluation.html" // 各类型标注任务质量和速度指标(平均), 参与者信息统计
return:
{
    // 都用用条形图表示, 横坐标是八种表述方式, 数据按照100,101,200,201...排序返回
    "tagError" : [0.34, 0.23, 0.02, 0.12, ..., 0.12],
    "speed" : [12, 23, 23, 12, 23, 34, 23, 34], // pictures per day
    "participantsNum" : [340, 230, 304, ..., 234],
}

url : "admin/statistic/dollars.html" // 关于金钱的统计图
method: GET
return:
{
    "date" : ["2018-6-1", "2018-6-2", ..., "2018-6-17"],
    "recharge" : [234532.22, 321.2, 321.2] // 发布者累计充值
    "prize" : [1234213.23, 21423.3, 343.12], // 累计发放奖金
}

```

```
"payWorker" : [231324.23, 2132.2, 32412], // 工人获得的累计回报
"advertisement" : [1234.234, 1234.3, 1343.4], // 累计广告费
"expectedPrize" : 3224, // 任务的平均奖金数目
"expectedPayWorker" : 2312, // worker 平均能从每个任务获得的金钱
"exptectedAdRate" : 0.23, // 推广费占奖金的平均比率
"expectedPayBackRate" : 0.11 // 平均返回给发布者的剩余奖金(没用完的)
}
```

Staff

未接受的所有审核任务

访问URL: /staff/check/index.html

参数: token

返回一个列表

```
[
  {
    "name": "xxx" // 任务名
    "taskId": 2333 // 任务ID
    "taskDescription": "xxxx" // 任务描述
    "taskCover": "xxxx/xxx/xx.jpg" // 一个url, 表示任务的封面, 以固定的尺寸显示!
    "taskTypes": [401, 402] // 这个一级任务中包含的任务的类型
    "endDate": "2017-10-1" // 任务截止的时间
    "taskTags": ["nature", "history"] // 任务的一些标签信息, 就是任务的主题
    "totalDollars": xxxx
  },
  ...
]
```

已接受的所有审核任务

访问URL: /staff/check/myParticipation.html

参数: token

返回一个列表

```
[
  {
    "name": "xxx" // 任务名
    "taskId": 2333 // 任务ID
    "taskDescription": "xxxx" // 任务描述
    "taskCover": "xxxx/xxx/xx.jpg" // 一个url, 表示任务的封面, 以固定的尺寸显示!
    "taskTypes": [401, 402] // 这个一级任务中包含的任务的类型
    "endDate": "2017-10-1" // 任务截止的时间
    "taskTags": ["nature", "history"] // 任务的一些标签信息, 就是任务的主题
    "totalDollars": xxxx
  }
]
```

```
},  
...  
]
```

点开具体一级check任务

访问URL: /staff/check/main.html

参数: token, taskId

返回:

```
{  
  "name": "xxx" // 任务名  
  "taskId": 2333 // 任务ID  
  "taskDescription": "xxxx" // 任务描述  
  "taskCover": "xxxx/xxx/xx.jpg" // 一个url, 表示任务的封面, 以固定的尺寸显示!  
  "taskTypes": [401, 402] // 这个一级任务中包含的任务的类型  
  "endDate": "2017-10-1" // 任务截止的时间  
  "taskTags": ["nature", "history"] // 任务的一些标签信息, 就是任务的主题  
  "totalDollars": xxxx ,  
  "participated": false // 是否已经参与  
  "taskBackground": "xxxxxxx" // 任务的背景信息  
  "backgroundImgae": "xxx/xxx/x.jpg" // 任务的背景图片URL  
}
```

接受一级check任务

访问URL: /staff/check/join.html

参数: token, taskId

返回:

```
{  
  "result": true  
  "reason": "xxxx" // 没什么用, 主要还是看result的结果, 这个可以不解析  
}
```

查看一级check任务下属小任务

访问URL: /staff/check/subTasks.html

参数: token, taskId, taskType // int!

返回:

```
[
  {
    "subPartId": xxx, // 小任务的id
    "picAmount": xxx, // 图片的数量
    "cover": "xxxx/xxx.jpg" // 任务的封面
  },
  ...
]
```

查看参与的check小任务

访问URL: /staff/check/subTask/ongoing.html

参数: token, taskId, taskType

返回:

```
[
  {
    "subPartId": xxx, // 小任务的id
    "picAmount": xxx, // 图片的数量
    "cover": "xxxx/xxx.jpg" // 任务的封面
  },
  ...
]
```

接受某个check小任务

访问URL: /staff/check/subTask/accept.html

参数

token, taskId, subPartId

返回:

```
{
  "result": true
  "description": "xxxx" // 没什么用, 主要还是看result的结果, 这个可以不解析
}
```

访问某个check小任务的第一张图片

访问url: /staff/subCheck/first.html

参数

token, subPartId

返回一个字符串, 表示图片url

访问某个check小任务的下一张图片

访问url: /staff/subCheck/more.html

参数

token, subPartId

返回一个字符串, 表示图片url, 如果返回了一个空字符串, 表示审查已经结束了, 跳转到查看一级check任务下属小任务的界面。

获取图片的标注信息

访问url: /staff/subCheck/label.html

参数

token, subPartId, url

返回在迭代二中定义的标注的结构字符串, 前端可以直接进行解析

给标注结果评分

访问url: /staff/subCheck/mark.html

参数

token, subPartId, url, accept

返回固定值"ok"