* 点击左边栏发布报名那三个
  + 发**GET**请求（script.js的 loadContentOfItem() 中）
    - 参数列表：
      * act\_type: string类型，表示问卷的类型：enroll、recuit、vote
      * time: int类型，目前是2016，表示当前时间
      * user\_id：string类型，目前是1111111111，表示所属用户id
    - 通过request.POST['键值']获取相应value
  + 连接到api的views.py中create\_new\_act(request)函数
    - 需要返回JSON：
      * status: string类型，表示状态，需要是“ok”
      * id: int类型，表示问卷id

* 点击题目
  + 发**GET**请求（design.ajax.html中 createNewQst(qst\_type) 函数）
    - 参数列表：
      * act\_id: int类型，表示所属问卷的id
      * qst\_type: string类型，表示问题类型：single、multi、fillin、vote、mark、sort
      * **qst\_rank: int类型，表示问题序号，从1开始计数**
  + 连接到api的create\_new\_qst(request) 函数
    - 需要返回JSON：
      * status: string类型，表示状态，需要是“ok”
      * id: int类型，表示问题id

* 点击问卷题目和说明那里的”编辑“
  + 显示modal（design\_list.html中）
    - url参数：
      * qst\_type：’qst\_name‘
      * act\_id：问卷中存的act\_id
    - modal的网页id：
      * activity-modal
  + 发POST请求（qst\_name\_modal.html中）
    - 使用handleFormPost，form字典格式：
      * status：尚未定义
      * act\_id：问卷id
      * csrfmiddlewaretoken：不懂
      * description：问卷描述
      * title：问卷题目
  + 连接到api的modify\_name（request）函数
    - 其中的dict表示上述字典
    - 需要返回JSON：
      * status：string类型，表示状态，需要是“ok”

* 点击题目下拉菜单中的按钮（除”编辑“外）
  + 发GET请求（question文件夹中index.html的operation\_qst（operation）函数）
    - 参数列表：
      * act\_id：int类型，表示问卷id
      * qst\_id：int类型，表示问题id
      * operation：string类型，表示进行的操作，具体分为以下三个：
        + “UP” 表示上移
        + “DOWN” 表示下移
        + “REMOVE” 表示删除
  + 连接到api的operation\_qst（request）函数
    - 需要返回JSON：
      * status：string类型，需要为“ok”
* 点击每个问卷操作中的“删除”
  + 发GET请求（script.js中的remove\_act（act\_id, url）函数）
    - 参数列表：
      * act\_id：int类型，表示要删除的问卷id
  + 连接到api的remove\_act（request）函数
    - 需要返回JSON：
      * status：string类型，需要为“ok”
* 点击设计问卷中的“保存”和“发布”
  + 发GET请求（design.ajax中的script）
    - 参数列表：
      * act\_id：int类型，表示要操作的问卷id
  + 分别连接到api的save\_act(request)和publish\_act(request)函数
    - 需要返回JSON：
      * status：string类型，需要为“ok”
* **点击单选题问题中的“编辑”**
  + **发POST请求（single\_modal中的script）：**
    - **使用handleFormPost，form字典格式：**
      * **status：目前未定义**
      * **qst\_type：string类型，当前问题的类型（即single）**
      * **questions\_id：int类型，当前问题id**
      * **option\_num：int类型，选项个数**
      * **csrfmiddlewaretoken：没用**
      * **qst\_title：string类型，问题的题面描述**
      * **option[x]\_field：string类型，表示第x个选项的文本内容，其中[x]为可变数字，取值从1到option\_num，比如“option1\_field“，表示第一个选项内容**
  + **连接到api的modify\_qst（request）函数**
    - **其中的dict表示上述字典**
    - **需要返回JSON：**
      * **status：string类型，表示状态，需要是“ok”**
* **获得所有<某状态>的问卷**
  + **传入一个string，表示问卷的状态（pending、already）**
  + **返回一个数组，数组中每个元素是一个字典，代表一个问卷，每个字典中的信息如下：**
    - **name：可支持中文编码的string，表示问卷标题**
    - **subscribe\_time：暂时只有日期，比如“2016.10.21”**
    - **status：string类型，表示问卷状态（pending、already）分别表示待发布和已发布**
    - **type：string类型，表示问卷类型（enroll、recruit、vote）**
    - **id：int类型，表示问卷id**
    - **description：string类型，表示问卷说明**
    - **fillin：int类型，表示填写人数，只有发布的问卷有该域**
* **通过<id>获得问卷**
  + **传入一个int，表示问卷id**
  + **返回一个字典，表示问卷具体题目的信息，具体定义如下：**
    - **act\_type：string类型，表示问卷类型（enroll、recruit、vote**
    - **act\_status：string类型，表示问卷状态（pending、already**
    - **act\_title：string类型，表示问卷标题**
    - **act\_description：string类型，表示问卷说明**
    - **qst\_num：int类型，表示问卷中问题的数目**
    - **questions：一个数组，其中每个元素是一个字典，表示每个问题，字典中的具体信息如下：**
      * **qst\_type：string类型，表示问题类型（single等**
      * **qst\_title：string类型，表示问题标题**
      * **qst\_id：int类型，表示问题id**
      * **option\_num：int类型，表示选项个数，只有选择题有**
      * **rows：int类型，表示填空题行数，只有填空题有**
      * **hint：string类型，表示提示文字，只有填空题有**
      * **check：string类型，表示输入检查类型，只有填空题**
      * **option：一个数组，其中每个元素是一个string，表示每个选项的文本，只有选择题有，具体要求如下**
        + **每个string需要以###开头和结尾，不然由于一些原因前端无法分辨，例：“###选项1###”**
* **点击多选题问题中的“编辑”**
  + **目前和单选题完全相同，在multi\_modal.html的script中发请求**
* **点击填空题问题中的“编辑”**
  + **发POST请求（fillin\_modal中的script）：**
    - **使用handleFormPost，form字典格式：**
      * **status：目前未定义**
      * **qst\_type：string类型，当前问题的类型（即fillin）**
      * **questions\_id：int类型，当前问题id**
      * **fillin\_row：int类型，表示输入框行数**
      * **csrfmiddlewaretoken：没用**
      * **qst\_title：string类型，问题的题面描述**
      * **fillin\_check：string类型，表示验证类型**
      * **fillin\_hint：string类型，表示输入提示**
  + **连接到api的modify\_qst（request）函数**
    - **其中的dict表示上述字典**
    - **需要返回JSON：**
      * **status：string类型，表示状态，需要是“ok”**
* **点击左边栏中的“统计信息”获得问卷的所有填写**
  + **连接到api的get\_participants(act\_id)函数**
    - **参数为问卷id**
  + **需要返回所有该问卷所有填写信息的数组，每一个元素是一个字典，包含以下内容：**
    - **id：int类型，填写id**
    - **fillin\_time：string类型，填写时间**
    - **ip：string类型，填写的ip**
    - **city：string类型，填写的城市（有中文）**
* **获得某个填写者对于某个问卷某个问题的回答**
  + **interface的views.py中的** **questionnaire\_publish\_question()函数在‘fillin\_id’参数不为空时调用**
    - **连接到api的get\_result\_of\_question(act\_id, qst\_id, fillin\_id)函数，参数意义如下：**
      * **act\_id：问卷id**
      * **qst\_id：问题id**
      * **fillin\_id：填写id**
    - **需要返回该用户对于该问题的回答**
      * **如果是单选题，返回一个int类型，表示第几个选项**
      * **如果是多选题，返回一个数组，每个是一个int类型，表示选了哪几个选项**
      * **如果是填空题，返回一个string，表示填写的内容**
* **获得某个问题的详细统计信息**
  + **interface的views.py在statistics\_question\_list函数中调用，用来获取某问题的统计列表中的content**
    - **连接到api的get\_statistics\_of\_question(qst\_id)函数**
      * **参数qst\_id为问题id**
    - **需要返回一个数组，数组的每个元素是一个字典，表示该题的一个选项的统计信息，具体包含以下：**
      * **id：int类型，选项的序号（顺序）**
      * **content：string类型，选项的文本内容（选择题），如果是填空题这里放填写的内容**
      * **count：int类型，选项被选择的次数（选择题需要）**
      * **total：int类型，表示该题的有效填写人数**
      * **percentage：int类型，表示选项被选择的次数占总填写人数的百分比，不加百分号（取值0-100）**
* 点击每个问卷操作中的“停止”
  + 发GET请求（script.js中的stop\_act（act\_id, url, item）函数）
    - 参数列表：
      * act\_id：int类型，表示要删除的问卷id
  + 连接到api的stop\_act（request）函数
    - 需要返回JSON：
      * status：string类型，需要为“ok”