

TVOS项目概要设计说明书

1 项目简介

1.1 目的

1.2 实现

2 概要设计

2.1 数据获取

2.1.1 各版块记录获取

2.1.2 记录中详细信息的获取

2.1.3 代码实现

2.2 数据分析

2.2.1 modify-time

2.2.2 name:modify-time

2.2.3 number-branch

2.2.4 number-project

2.2.5 file_path-commens_size

2.3 数据存储

2.3.1 record

2.3.2 info

2.4 Django后台搭建

2.4.1 Model

2.4.2 Template

2.4.3 View

2.5 服务器部署

2.6 前端设计

2.6.1 ECharts

2.6.2 Bootstrap

2.6.3 网页功能

3 后记

3.1 错误与解决

3.1.1 Django中静态文件路径

3.2 后期安排

4 参考

5 附录

5.1 数据获取部分代码

5.1.1 获取三个版块的json文件源码

5.1.2 获取记录文件中对应的详细json文件源码

5.2 数据分析部分代码

5.2.1 获取修改量-时间的list

5.2.2 图2-1源码

5.2.3 图2-2源码

5.3 数据存储部分代码

5.4 Django后台搭建部分代码

5.4.1 view.py的部分代码

5.6 前端设计部分代码

5.6.1 branch.html

项目负责人：

项目开发人员：

项目持续周期：

TVOS项目概要设计说明书

1 项目简介

1.1 目的

分析<http://120.25.200.39:8081>上的数据，并将信息可视化展现在搭建的网站上

1.2 实现

Django+MySQL+ECharts+Bootstrap

2 概要设计

2.1 数据获取

2.1.1 各版块记录获取

首先访问网址，使用F12，动态抓包（Network->Headers），获取到如下信息

对于open, Merged, Abandoned, 显示界面的json文件如下：

- Open <http://120.25.200.39:8081/changes/?n=25&O=81>
- Merged <http://120.25.200.39:8081/changes/?q=status:merged&n=25&O=81>
- Abandoned <http://120.25.200.39:8081/changes/?q=status:abandoned&n=25&O=81>

2.1.2 记录中详细信息的获取

点击网页上面版块任一条目，进入详细信息，其json文件网址为

<http://120.25.200.39:8081/changes/1637/revisions/205f3f9694e931de9779cdbaa82c5bb881751899/files>

分析上面网址，有两个地方是变化的：

- 一个是1637，一个是205f3f9694e931de9779cdbaa82c5bb881751899
1637代表的是该条目的编号，在对应版块的json文件中可获取(_number)
- 第二个为commit的编号，通过查看Network中显示的文件，发现在下面的json文件中可获取
<http://120.25.200.39:8081/changes/1637/detail?O=404>

2.1.3 代码实现

见附录5.1数据获取

总共下载了1500多个json文件。

2.2 数据分析

因为网页到底要展现什么是模糊的，因此数据分析只能通过数据的基本特征来分析，现自定如下：

2.2.1 modify-time

获取每天的修改量，制作(modify,time)的曲线图

modify=insertions+deletions

time=created

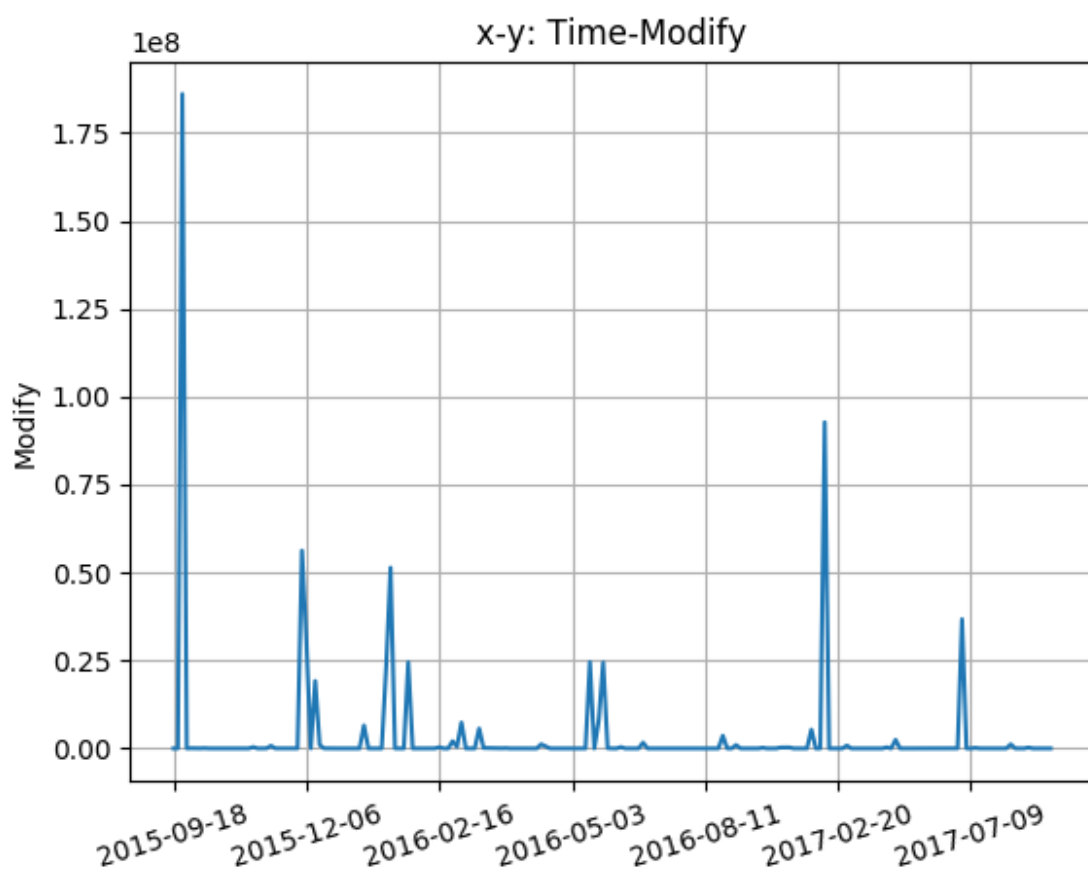


图2-1

——这里同时也可以展现(insertions,time)和(deletions,time)的曲线图

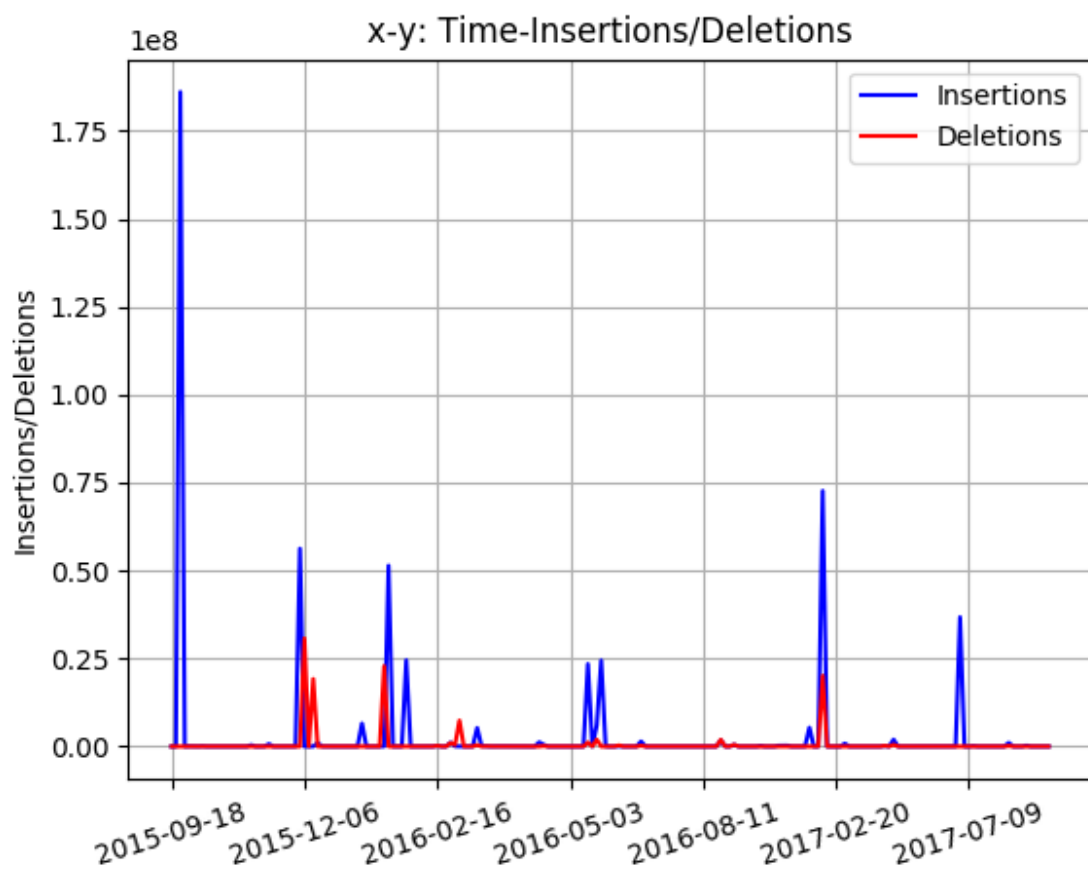


图2-2

——可以对三个版块分别实现1中所述, 和不同的分支(Branch)实现1中所述

2.2.2 name:modify-time

对每个修改者, 即记录中的name, 统计(modify,time)

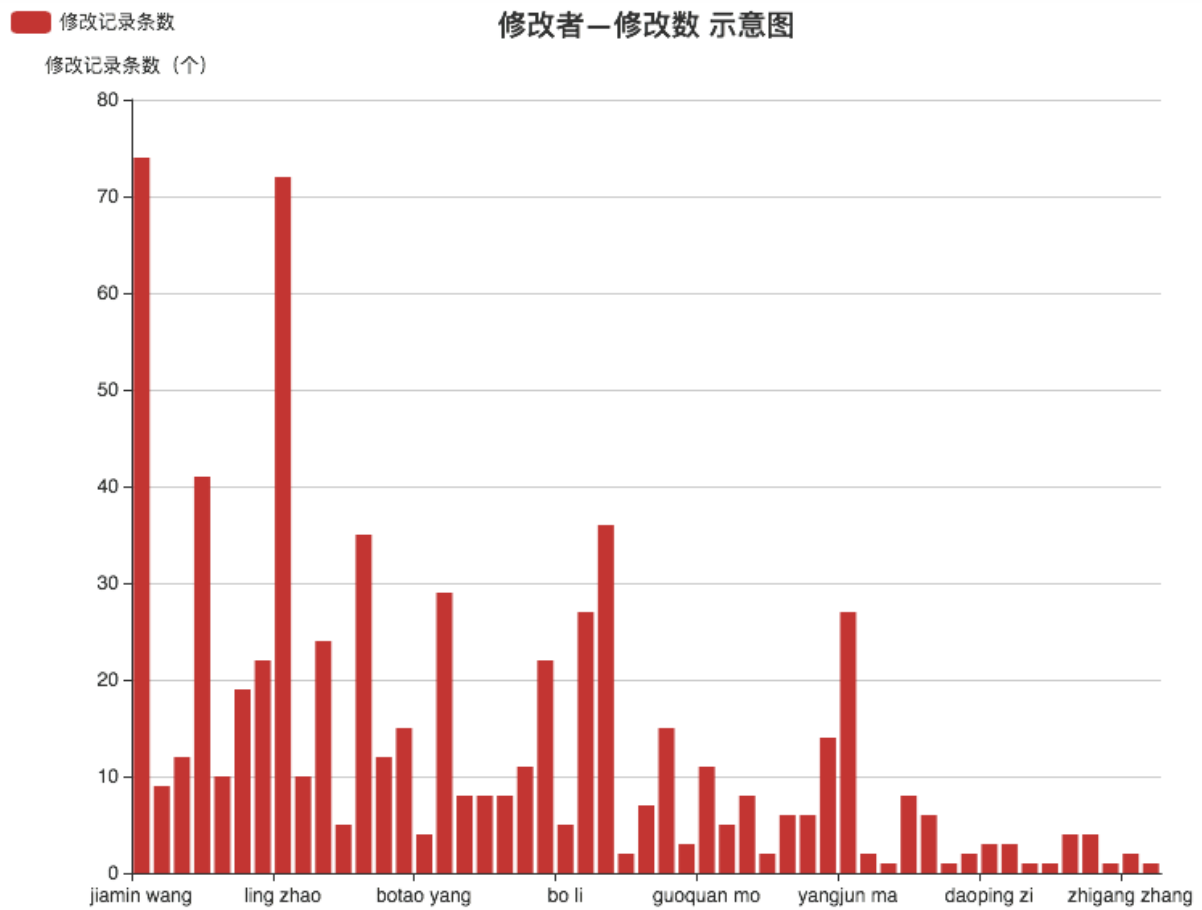


图2-3

2.2.3 number-branch

统计各分支的占比

Branch 分布 示意图

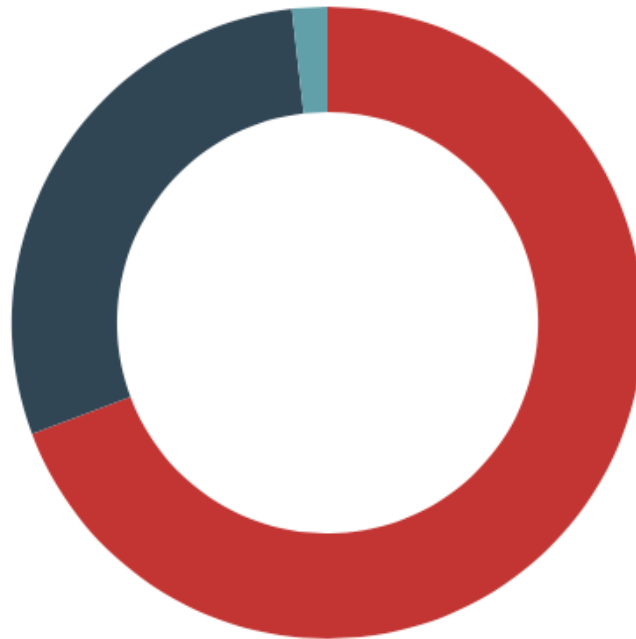


图2-4

2.2.4 number-project

统计项目占比

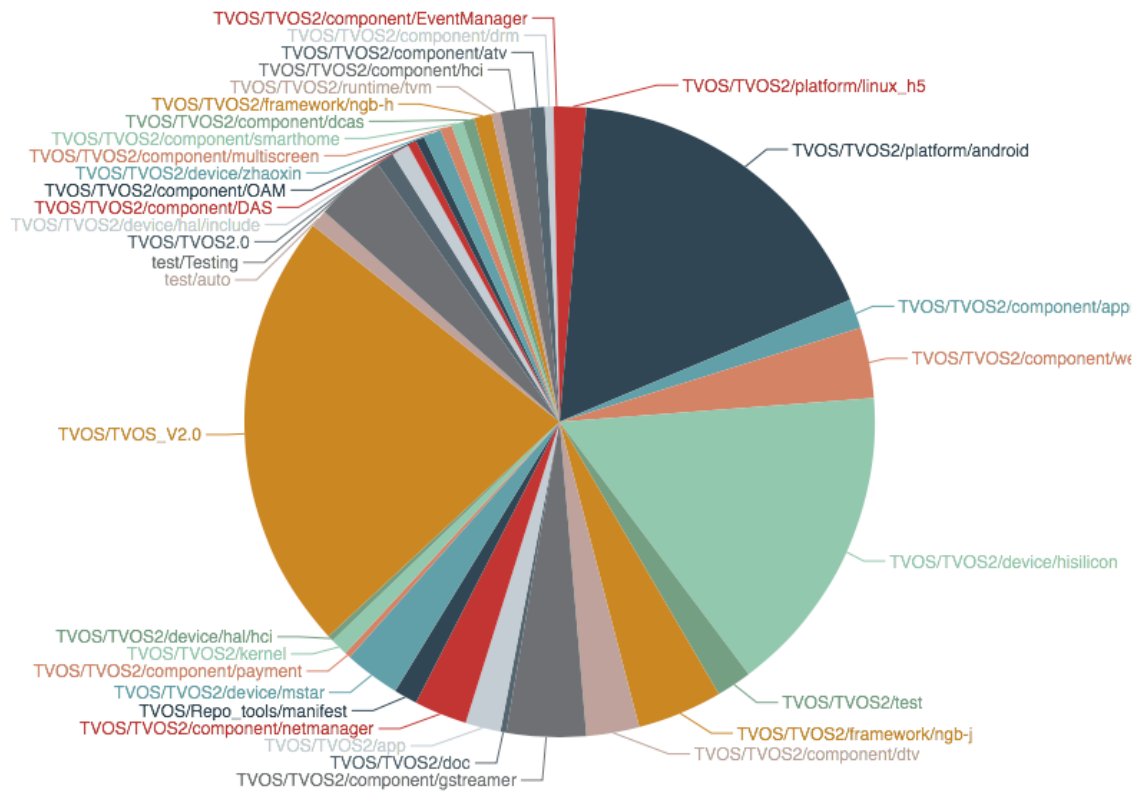


图2-5

2.2.5 file_path-commens_size

进入每个记录，获取其中的File Path和Commens Size，制作(File_Path,Commens_Size)的柱状图

此处的File_Path可以选取一部分，以便更加明显【可作为各文件的活跃度指标】

暂时分析上述，具体后续，此部分需要明确清楚

2.3 数据存储

暂定MySQL，根据数据分析部分，具体建表如下（待定）

2.3.1 record

record_table

字段名称	字段描述	字段类型	备注
number	标号	UNSIGNEDINT	主键
insertions	增量	UNSIGNEDINT	
deletions	删量	UNSIGNEDINT	
updated	更新日期	DATE	
owner	拥有者	CHAR()	
branch	分支	CHAR()	
project	项目名	CHAR()	

2.3.2 info

info_table

字段名称	字段描述	字段类型	备注
filepath	修改的文件路径	CHAR()	联合主键
number	标号	UNSIGNEDINT	联合主键
insertions	增量	UNSIGNEDINT	
deletions	删量	UNSIGNEDINT	

【这一块因为Django默认的是sqlite（数据量比较小适合），就先用sqlite实现（这里初步的想法是先将数据进行统计，删除不必要的信息，对筛选的数据再构建数据表进行存储），后期如果不行或者有时间再改】

2.4 Django后台搭建

Django框架中一个项目可以有多个app，在app中完成前后端的交互，分为三个部分类似于MVC模式的MTV模式

2.4.1 Model

负责业务对象和数据库的关系映射

这里对应的是models.py文件，用于定义数据表及其类型【暂时还没有写好，后期补上】

2.4.2 Template

负责如何把页面展示给用户

这里对应的时templates目录下的html文件，前端html文件全放于此，对于前端中使用到的css,js等文件，需要再app（后者其他目录）下新建static目录，后台回去查找static目录，见**3.1.1**

详细见**前端部分**

2.4.3 View

负责业务逻辑，并在适当时候调用Model和Template

这里对应的是views.py文件，在其中定义函数，获取数据，并传入前端界面

函数名和前端访问网址对应，在项目中的urls.py中使用正则匹配获取网址，并对应到view.py中的函数

可见附录5.4.1部分代码（未处理的）

【此处还需要写不同的响应函数，待后期完善】

2.5 服务器部署

[void]

2.6 前端设计

2.6.1 ECharts

按照ECharts官网，下载echarts库，并在html文件的head下导入

```
<script src="/static/echarts.min.js"></script>
```

接下来只需要实现如下脚本在 `<script>` 中

对需要显示的div初始化如下：

```
var ownerChart=echarts.init(document.getElementById('owner'))
```

设置option,在其中接收传入的数据，以json传入，这里需要使用safe，如下：

```
data:{ { name|safe } }
```

最后设置生效：

```
ownerChart.setOption(option);
```

显示部分见2.2

2.6.2 Bootstrap

使用Bootstrap框架来实现前端界面，这里使用官网中的样例

将Bootstrap和样例中的css，js文件放置于static目录下

整个界面分为三部分，由左侧的项目框和各板块的导航条，以及中间的显示部分组成，网页截取一部分如下所示：



这里左侧项目条和上侧导航条部分，在点击过程中，网页不会跳转，只会让中间部分根据不同的数据重新绘制，具体实现如下：

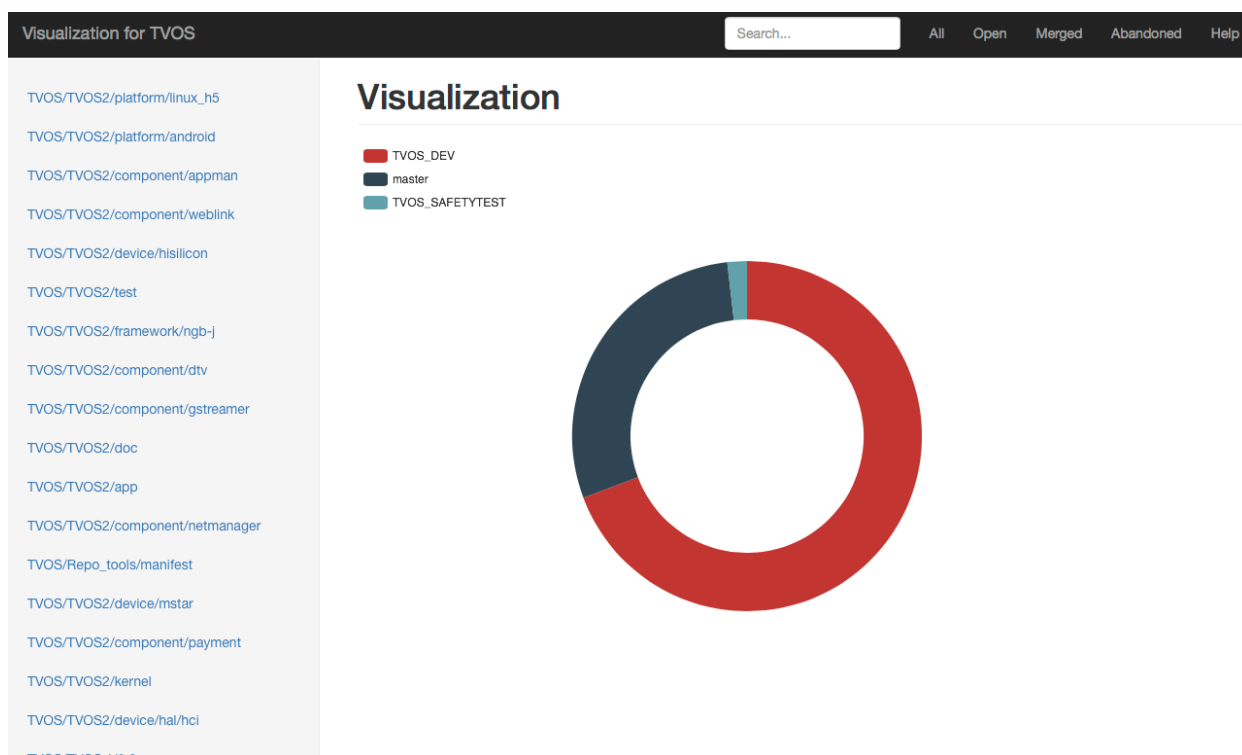
前端部分需要用到jquery，导入jquery文件，以及点击只更新指定div

```
<script type="text/javascript" src="/static/js/jquery-3.2.1.min.js">
</script>
<script type="text/javascript">
    $(document).ready(function () {
        $("#all").click(function () {
            $('#owner').load('../branch');
        })
    })
</script>
```

这里载入的 `../branch`，是网址对应的上级目录下的branch目录，后台会先在urls.py文件中匹配该网址，找到对应的name参数，再在views.py中找到以name参数命名的函数名，如下

```
def branch(request):
    branch_name = ['TVOS_DEV', 'master', 'TVOS_SAFETYTEST']
    branch_num = [460, 192, 12]
    return render(request, 'branch.html',
        {'branchName': json.dumps(branch_name), 'branchNum': json.dumps(branch_num)})
```

在该函数中获取数据，加载到branch.html上，branch.html的功能是根据传入的数据绘制Branch图，branch.html就会替换原有的div显示部分，仅该div刷新，如下图：



[这里只是举个例子，将该div显示部分换成Branch图]

2.6.3 网页功能

- 点击导航条的All标签，中间会显示三幅图：修改量-时间，项目分布图，Branch分布图；同理点击Merged和Abandoned标签会显示对应版块下的这三幅图
- 点击左边的项目，会显示该项目修改量-时间的统计，以及该项目各文件夹的活跃程度
- 待增加

3 后记

3.1 错误与解决

3.1.1 Django中静态文件路径

比如把 `echarts.min.js` 文件放置在templates下和index.html同级目录，按理直接使用

```
<script src="echarts.min.js"></script>
```

便可，但是访问时却显示：Not Found charts.min.js

解决：在app目录下（这个随意，会自动查找static文件夹）新建static文件夹，然后将js文件存放在其中，再在settings.py文件中声明static目录：

```
STATIC_PATH=os.path.join(os.path.dirname(__file__),'static').replace('\\','/')
')
```

在index.html中使用

```
<script src="/static/echarts.min.js"></script>
```

即可

3.2 后期安排

1. 首先是实现MySQL数据库查找相应数据，分析，然后实现model中的数据表
2. 以及对每条记录中的信息分析，实现各项目各文件活跃度的分析
3. views.py实现前端对应的函数
4. 前端界面优化，或添加新功能
5. 数据更新的问题，自动化实现
6. 搭建服务器，这个可早可晚
7. 完结！

4 参考

- 【1】 [Django官方文档](#)
- 【2】 [自强学堂 Django基础教程](#)
- 【3】 [ECharts](#)
- 【4】 [Bootstrap官方文档](#)

5 附录

5.1 数据获取部分代码

5.1.1 获取三个板块的json文件源码

```

import json
import requests

#获取三个板块的json文件
def get_record_data():
    #动态抓包获取各板块的json文件网址,存储在字典中
    section_dict={}
    section_dict['open']='http://120.25.200.39:8081/changes/?n=25&O=81'
    section_dict['merged']='http://120.25.200.39:8081/changes/?
q=status:merged&n=25&O=81'
    section_dict['abandoned']='http://120.25.200.39:8081/changes/?
q=status:abandoned&n=25&O=81'

    filename_list=[]#获取记录的文件名

    #遍历访问网址获取数据,以json存储在data中
    for key,value in section_dict.items():
        flag = 1
        num = 0
        while flag:
            json_open = requests.get(value + "&S=" + str(num)).text#获取文件
            的文本形式
            json_open = json_open.strip().split(']}\\'\n')[1]#去除文本中的第一行
            #这里小于5表示空,即next到头了,结束next;否则将数据写入文件
            if len(json_open) < 5:
                flag = 0
            else:
                with open('data/record/'+key+'_' + str(num) + '.json', 'w')
as f:
                    f.write(json_open)
                    filename_list.append(key+'_'+str(num))
                    print(key+str(num)+"has wirtten !")
                num += 25
    return filename_list

```

5.1.2 获取记录文件中对应的详细json文件源码

#获取记录详细信息

```
def get_record_info_data(filename_list):
    for filename in filename_list:
        with open('data/record/'+filename+'.json','r') as f:
            records_json=json.load(f)
        for record in records_json:
            number=record['_number']
            print(number)

url_str2='http://120.25.200.39:8081/changes/'+str(number)+'/' + 'detail?O=404'
detail=requests.get(url_str2).text
detail = detail.strip().split(']}\\'\\n') # 去除文本中的第一行
if len(detail)<2:continue
detail=detail[1]
detail='['+detail+']'
with open ('data/info/'+filename.split('_')
[0]+'/' + 'detail/detail'+str(number)+'.' + 'json','w') as f:
    f.write(detail)
    with open ('data/info/'+filename.split('_')
[0]+'/' + 'detail/detail'+str(number)+'.' + 'json','r') as f:
        detail=json.load(f)
        print(filename+"detail"+str(number)+"has written!")
        revision=detail[0]['current_revision']

url_str1 =
'http://120.25.200.39:8081/changes/'+str(number)+'/' + 'revisions/'+str(revision)
+'/' + 'files'

info=requests.get(url_str1).text
info = info.strip().split(']}\\'\\n') # 去除文本中的第一行
if len(info)<2:continue
info=info[1]
with open('data/info/' + filename.split('_')[0] + "/" +
str(number) + '.' + 'json','w') as f:
    f.write(info)
    print(filename+str(number)+"has written!")
```

5.2 数据分析部分代码

5.2.1 获取修改量-时间的list

```

def get_modify_time():
    mod_ti_dict={}
    filelist=os.listdir('data/record')#获取data/record目录下的全部文件名
    for filename in filelist:
        with open('data/record/'+str(filename),'r') as f:
            record_data=json.load(f)
            if len(record_data)<1:continue
            for record in record_data:
                updated_time=record['updated'][:10]
                if updated_time not in mod_ti_dict.keys():
                    mod_ti_dict[updated_time]='0#0'

                insert,delete=mod_ti_dict[updated_time].split('#')
                insert=int(insert)+record['insertions']
                delete=int(delete)+record['deletions']
                mod_ti_dict[updated_time]=str(insert)+'#+str(delete)
    #print(mod_ti_dict)

    ti_list=[key for key in mod_ti_dict.keys()]
    ti_list=sorted(ti_list)#对时间排序
    mod_list=[]
    mod_insert_list=[]
    mod_delete_list=[]
    #按照排好序的时间list来获取value值
    for ti in ti_list:
        mod_insert,mod_delete=mod_ti_dict[ti].split('#')
        mod_insert_list.append(int(mod_insert))
        mod_delete_list.append(int(mod_delete))
        mod_list.append(int(mod_insert)+int(mod_delete))

    return ti_list,mod_list,mod_insert_list,mod_delete_list

```

5.2.2 图2-1源码

#绘制时间-修改量的图

```
def draw_modify_time(ti_list,mod_list,tag_str):
    x=[i+1 for i in range(len(ti_list))]
    #plt.bar(x,mod_list,width=5,color='y')#条形图
    plt.grid(True)
    plt.plot(x,mod_list)
    # 设置x轴的标签, 第一个参数对应x的位置, 第二个参数对应x的位置上的text (这里是日期), 第三个参数是旋转度
    plt.xticks(x[::30],ti_list[::30],rotation=17)

    plt.xlabel('Time')
    plt.ylabel(tag_str)
    plt.title(r'x-y: Time-'+tag_str)

    plt.show()
```

5.2.3 图2-2源码

#绘制时间-增加量, 时间-删除量的图

```
def draw_ins_de_time(ti_list,ins_list,de_list):
    x=[i+1 for i in range(len(ti_list))]
    #plt.bar(x,mod_list,width=5,color='y')#条形图
    plt.grid(True)
    plt.plot(x,ins_list,color='b',label='Insertions')
    plt.plot(x,de_list,color='r',label='Deletions')
    # 设置x轴的标签, 第一个参数对应x的位置, 第二个参数对应x的位置上的text (这里是日期), 第三个参数是旋转度
    plt.xticks(x[::30],ti_list[::30],rotation=17)

    plt.xlabel('Time')#x轴
    plt.ylabel('Insertions/Deletions')#y轴
    plt.title(r'x-y: Time-Insertions/Deletions')#标题
    plt.legend()#这行表示显示标签 (是plot中的标签)
    plt.show()
```

5.3 数据存储部分代码

[void]

5.4 Django后台搭建部分代码

5.4.1 view.py的部分代码

```
def home(request):
```



```
name=['jiamin wang', 'yanshan jiang', 'haifeng yan', 'dandong guan',
'gerrit', 'hongguo qiu', 'jinxiao liu', 'ling zhao', 'weifan li', 'jianrong
ye', 'zhijian liang', 'mingyong sun', 'jingang dong', 'haili bai', 'botao
yang', 'zhetian zhu', 'ming zhang', 'guofeng sun', 'jianlong ye', 'baogang
zhang', 'yuanbao zan', 'bo li', 'tao zhou', 'yonggang huang', 'shihua
huang', 'zhenlei xie', 'boqin cheng', 'songxing ding', 'guoquan mo', 'zeqin
zhou', 'weilong lv', 'xinwei liu', 'chun tao', 'xueqing zhao', 'bo jiang',
'yangjun ma', 'tongsi xue', 'chao chu', 'zhenming chen', 'jie wang',
'jianfeng lai', 'yujian zhao', 'daoping zi', 'yubin zhang', 'qiang liu',
'chao wu', 'lei tong', 'jiulin guo', 'qiyuan zhong', 'zhigang zhang',
'zhikui wang']

number=[74, 9, 12, 41, 10, 19, 22, 72, 10, 24, 5, 35, 12, 15, 4, 29, 8,
8, 8, 11, 22, 5, 27, 36, 2, 7, 15, 3, 11, 5, 8, 2, 6, 6, 14, 27, 2, 1, 8, 6,
1, 2, 3, 3, 1, 1, 4, 4, 1, 2, 1]

date_list=['2015-09-18', '2015-09-30', '2015-10-09', '2015-10-13',
'2015-10-16', '2015-10-23', '2015-10-26', '2015-11-03',
'2015-11-04', '2015-11-05', '2015-11-06', '2015-11-12', '2015-11-15',
'2015-11-16', '2015-11-17', '2015-11-18',
'2015-11-19', '2015-11-20', '2015-11-23', '2015-11-24', '2015-11-25',
'2015-11-26', '2015-11-27', '2015-11-28',
'2015-11-30', '2015-12-01', '2015-12-02', '2015-12-03', '2015-12-04',
'2015-12-05', '2015-12-06', '2015-12-07',
'2015-12-08', '2015-12-09', '2015-12-10', '2015-12-11', '2015-12-14',
'2015-12-15', '2015-12-16', '2015-12-17',
'2015-12-18', '2015-12-19', '2015-12-20', '2015-12-21', '2015-12-22',
'2015-12-23', '2015-12-24', '2015-12-29',
'2015-12-30', '2015-12-31', '2016-01-11', '2016-01-14', '2016-01-16',
'2016-01-17', '2016-01-18', '2016-01-19',
'2016-01-25', '2016-01-28', '2016-01-29', '2016-02-01', '2016-02-16',
'2016-02-17', '2016-02-19', '2016-02-24',
'2016-02-27', '2016-02-28', '2016-03-01', '2016-03-02', '2016-03-07',
'2016-03-08', '2016-03-09', '2016-03-15',
'2016-03-17', '2016-03-21', '2016-03-23', '2016-03-24', '2016-03-30',
'2016-04-01', '2016-04-05', '2016-04-06',
'2016-04-07', '2016-04-11', '2016-04-12', '2016-04-13', '2016-04-14',
'2016-04-15', '2016-04-18', '2016-04-22',
'2016-04-27', '2016-04-28', '2016-05-03', '2016-05-18', '2016-05-20',
'2016-05-21', '2016-05-24', '2016-05-25',
'2016-05-27', '2016-06-01', '2016-06-03', '2016-06-05', '2016-06-06',
'2016-06-07', '2016-06-08', '2016-06-10',
'2016-06-12', '2016-06-13', '2016-06-27', '2016-06-28', '2016-06-30',
'2016-07-01', '2016-07-04', '2016-07-06',
'2016-07-08', '2016-07-11', '2016-07-26', '2016-07-29', '2016-08-01',
'2016-08-03', '2016-08-06', '2016-08-08',
'2016-08-11', '2016-08-17', '2016-08-24', '2016-08-30', '2016-09-05',
'2016-09-10', '2016-09-21', '2016-10-08',
'2016-10-21', '2016-10-28', '2016-11-07', '2016-11-16', '2016-11-28',
'2016-12-01', '2016-12-08', '2016-12-12',
```

```
'2016-12-13', '2016-12-18', '2016-12-19', '2016-12-20', '2016-12-22',
'2016-12-26', '2016-12-27', '2016-12-30',
'2017-01-05', '2017-01-16', '2017-01-22', '2017-01-24', '2017-01-25',
'2017-02-09', '2017-02-20', '2017-02-23',
'2017-03-07', '2017-03-08', '2017-03-15', '2017-03-17', '2017-03-21',
'2017-03-25', '2017-04-06', '2017-04-25',
'2017-04-28', '2017-05-03', '2017-05-08', '2017-05-12', '2017-05-15',
'2017-05-25', '2017-06-04', '2017-06-08',
'2017-06-09', '2017-06-16', '2017-06-20', '2017-06-21', '2017-06-22',
'2017-06-23', '2017-06-26', '2017-06-30',
'2017-07-05', '2017-07-06', '2017-07-07', '2017-07-08', '2017-07-09',
'2017-07-10', '2017-07-11', '2017-07-12',
'2017-07-14', '2017-07-17', '2017-07-18', '2017-07-19', '2017-07-20',
'2017-07-21', '2017-07-22', '2017-07-23',
'2017-07-24', '2017-07-27', '2017-07-28', '2017-07-29', '2017-07-31',
'2017-08-01', '2017-08-02']
```

```
"""
```

```
[1, 0, 186065206, 0, 60128, 14758, 16074, 75815, 7, 2206, 127, 1961,
14303, 1, 2673, 12641, 2890, 796, 388207, 866,
1516, 25339, 732857, 18957, 15908, 24, 37391, 875, 18665, 56242944,
30715078, 16854, 19190105, 966803, 26572, 16,
431, 2099, 7053, 1976, 208, 125, 133, 6554751, 17, 6, 4, 2919,
22840645, 51403721, 14191, 1397, 103, 24518976, 20,
3, 2332, 2, 66, 12660, 351012, 1477, 1, 2050247, 482069, 7381904, 3380,
55, 56, 5670416, 141090, 106141, 12499, 2,
39881, 84476, 2711, 6243, 1672, 0, 0, 6, 3, 1187693, 624308, 51, 0,
2726, 43, 4, 2, 303, 37109, 81, 24574513, 172,
7960421, 24446076, 68101, 0, 1082, 391088, 11, 3853, 1728, 36, 1662489,
0, 0, 15761, 440, 24, 129, 28, 0, 27, 422,
32, 179, 3631, 1164, 38, 29, 3, 3610378, 212, 2, 960402, 4, 4, 8, 658,
5, 193679, 6, 2812, 1268, 253627, 241815,
282085, 3594, 0, 568, 7520, 5360380, 980, 38, 92739855, 979, 2, 17, 4,
811295, 186, 4, 572, 6, 1193, 74, 60, 1189,
317929, 1, 2452985, 0, 49, 1599, 40, 458, 1, 120, 53893, 2, 974, 4,
2165, 33698, 29906, 36753345, 266, 27, 162940,
8693, 7, 2, 41, 3, 1207, 3499, 1175070, 6057, 203, 153, 253575, 246, 4,
906, 14334, 216]
```

```
"""
```

```
insert_list=[1, 0, 186065206, 0, 60111, 14273, 14890, 75814, 7, 1094,
66, 1958, 13245, 1, 2673, 9327, 2634, 727, 381285, 860,
1404, 25338, 673548, 9014, 12210, 15, 27203, 870, 17611, 56242048,
1070, 16817, 338, 966672, 25353, 7, 255, 1751,
5422, 1494, 115, 108, 33, 6552005, 10, 4, 3, 0, 9612, 51403520, 10491,
1139, 65, 24518794, 18, 2, 2170, 1, 25,
10047, 175633, 1192, 1, 1062117, 54338, 0, 3379, 46, 47, 5275085,
140994, 59860, 11443, 1, 39791, 42402, 1758,
3144, 1352, 0, 0, 3, 3, 1187693, 483437, 44, 0, 2722, 40, 3, 2, 282,
22102, 80, 23430598, 45, 6019982, 24445630,
```

```
68092, 0, 380, 87561, 9, 3771, 850, 23, 1432702, 0, 0, 15745, 348, 24,
128, 23, 0, 18, 417, 23, 115, 2743, 1003,
32, 22, 3, 1705167, 13, 2, 465016, 2, 2, 6, 488, 5, 193668, 3, 2812,
1268, 224812, 215063, 254862, 3594, 0, 564,
7520, 5360380, 980, 38, 72647675, 979, 2, 10, 3, 810979, 0, 2, 387, 3,
698, 23, 53, 0, 138458, 1, 1948704, 0, 49,
1599, 40, 121, 1, 61, 35939, 1, 853, 4, 1969, 21067, 29837, 36753323,
158, 17, 148561, 8690, 5, 1, 41, 3, 1200,
3349, 1014905, 550, 3, 147, 253392, 208, 2, 889, 14327, 176]
```

```
delete_list=[0, 0, 0, 0, 17, 485, 1184, 1, 0, 1112, 61, 3, 1058, 0, 0,
3314, 256, 69, 6922, 6, 112, 1, 59309, 9943, 3698, 9,
10188, 5, 1054, 896, 30714008, 37, 19189767, 131, 1219, 9, 176, 348,
1631, 482, 93, 17, 100, 2746, 7, 2, 1, 2919,
22831033, 201, 3700, 258, 38, 182, 2, 1, 162, 1, 41, 2613, 175379, 285,
0, 988130, 427731, 7381904, 1, 9, 9,
395331, 96, 46281, 1056, 1, 90, 42074, 953, 3099, 320, 0, 0, 3, 0, 0,
140871, 7, 0, 4, 3, 1, 0, 21, 15007, 1,
1143915, 127, 1940439, 446, 9, 0, 702, 303527, 2, 82, 878, 13, 229787,
0, 0, 16, 92, 0, 1, 5, 0, 9, 5, 9, 64, 888,
161, 6, 7, 0, 1905211, 199, 0, 495386, 2, 2, 2, 170, 0, 11, 3, 0, 0,
28815, 26752, 27223, 0, 0, 4, 0, 0, 0, 0,
20092180, 0, 0, 7, 1, 316, 186, 2, 185, 3, 495, 51, 7, 1189, 179471, 0,
504281, 0, 0, 0, 0, 337, 0, 59, 17954, 1,
121, 0, 196, 12631, 69, 22, 108, 10, 14379, 3, 2, 1, 0, 0, 7, 150,
160165, 5507, 200, 6, 183, 38, 2, 17, 7, 40]
```

```
branch_name=['TVOS_DEV', 'master', 'TVOS_SAFETYTEST']
branch_num=[460, 192, 12]
```

```
project_name=['TVOS/TVOS2/platform/linux_h5',
'TVOS/TVOS2/platform/android', 'TVOS/TVOS2/component/appman',
'TVOS/TVOS2/component/weblink', 'TVOS/TVOS2/device/hisilicon',
'TVOS/TVOS2/test', 'TVOS/TVOS2/framework/ngb-j',
'TVOS/TVOS2/component/dtv', 'TVOS/TVOS2/component/gstreamer',
'TVOS/TVOS2/doc', 'TVOS/TVOS2/app',
'TVOS/TVOS2/component/netmanager', 'TVOS/Repo_tools/manifest',
'TVOS/TVOS2/device/mstar',
'TVOS/TVOS2/component/payment', 'TVOS/TVOS2/kernel',
'TVOS/TVOS2/device/hal/hci', 'TVOS/TVOS_V2.0', 'test/auto',
'test/Testing', 'TVOS/TVOS2.0', 'TVOS/TVOS2/device/hal/include',
'TVOS/TVOS2/component/DAS',
'TVOS/TVOS2/component/OAM', 'TVOS/TVOS2/device/zhaoxin',
'TVOS/TVOS2/component/multiscreen',
'TVOS/TVOS2/component/smarthome', 'TVOS/TVOS2/component/dcas',
'TVOS/TVOS2/framework/ngb-h',
'TVOS/TVOS2/runtime/tvm', 'TVOS/TVOS2/component/hci',
'TVOS/TVOS2/component/atv', 'TVOS/TVOS2/component/drm',
'TVOS/TVOS2/component/EventManager']
```

```

        project_num=[9, 115, 10, 24, 106, 12, 29, 18, 27, 2, 12, 18, 8, 19, 2,
6, 2, 150, 6, 24, 6, 6, 3, 3, 6, 4, 4, 4, 6, 3, 10, 5, 3,
2]

        return render(request, 'home.html',
{'name':json.dumps(name), 'number':json.dumps(number), 'dateList':json.dumps(d
ate_list), 'insertList':json.dumps(insert_list), 'deleteList':json.dumps(delet
e_list),

'branchName':json.dumps(branch_name), 'branchNum':json.dumps(branch_num), 'pro
jectName':json.dumps(project_name), 'projectNum':json.dumps(project_num)})#re
nder为渲染

```

5.6 前端设计部分代码

5.6.1 branch.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <script src="/static/echarts.min.js"></script>
</head>
<body>
<div id="branch" style="width:800px;height: 600px" ></div>
<script type="text/javascript">
    var branchChart=echarts.init(document.getElementById( 'branch' ))

    var branch_num={{ branchNum|safe }}
    var branch_name={{ branchName|safe }}
    var data_branch=new Array()

    for(var i=0;i<branch_name.length;i++){
        var dic={value:branch_num[i],name:branch_name[i] }
        data_branch[i]=dic
    }

    var option3 = {
    tooltip: {
        trigger: 'item',
        formatter: "{a} <br/>{b}: {c} ({d}%)"
    },
    legend: {
        orient: 'vertical',
        x: 'left',

```

```
        data:{{ branchName|safe }}
    },
    series: [
        {
            name:'Branch',
            type:'pie',
            radius: ['40%', '60%'],
            avoidLabelOverlap: false,
            label: {
                normal: {
                    show: false,
                    position: 'center'
                },
                emphasis: {
                    show: true,
                    textStyle: {
                        fontSize: '30',
                        fontWeight: 'bold'
                    }
                }
            },
            labelLine: {
                normal: {
                    show: false
                }
            },
            data:data_branch
        }
    ]
};
branchChart.setOption(option3)
</script>
</body>
</html>
```