

Extract_Excel_Fields

版本信息:

- **版本 v1.0** time: 2020年3月17日16:24:13
 - 半自动化整理字段, 需要手动整理csv格式的文件 /semi_automatic
- **版本 v2.0** time: 2020年3月17日20:43:57
 - 自动从Excel中提取字段, 需要手动进行翻译, 更新v1.0出现的遗留问题, 字段后有多余逗号
- **版本 v2.1** time: 2020年3月18日01:18:32
 - 加入爬虫进行全自动解析字段并生成对应的DDL语句, 字段类型不可选, 默认STRING类型
 - 爬虫对应的有道API接口, 整理字段不规则符号, 修复文件不存在时的报错
- **版本 v2.2** time: 2020年3月19日00:15:19(Deep day)
 - 有道API访问过快会封IP,为了能长时间使用,加了延迟, 相对来说, 也比自己手动扣来的快。
 - 添加可选项, 生成的DDL 可以选择 hive 和 mysql 字段类型较单一 STRING 和 VARCHAR。
- **版本 v2.3** time: 2020年3月19日11:23:41
 - 优化了DDL选项操作, 默认生成 hive 和 mysql ddl,不用进行手动选择
 - 优化了显示格式, 在文本文件中也可以看清楚

使用方法:

1.环境要求:

系统必须有 **python 3环境**, 且安装 **openpyxl 模块**与 **requests 模块**

如果有Py3环境执行以下操作,如果有已安装, 请忽略

安装模块:

```
1 # 操作 Excel 模块
2 pip install openpyxl
3 # 网络请求模块
4 pip install requests
```

2.在满足环境条件下:

- 将要解析的 Excel文件放到 当前【Extract_Excel_Fields】文件夹下

3.在当前目录下执行

- 切记, 运行时, 目标Excel文件是关闭状态的, 换句好理解的话, 运行时Excel不能是打开的状态

```
1 | python get_fields.py
```

4.在PyCharm中直接导入项目运行

功能说明:

- 获取Excel**每列列名**，并生成支持Hive/Mysql的建表语句，默认分隔符为逗号
- 格式:

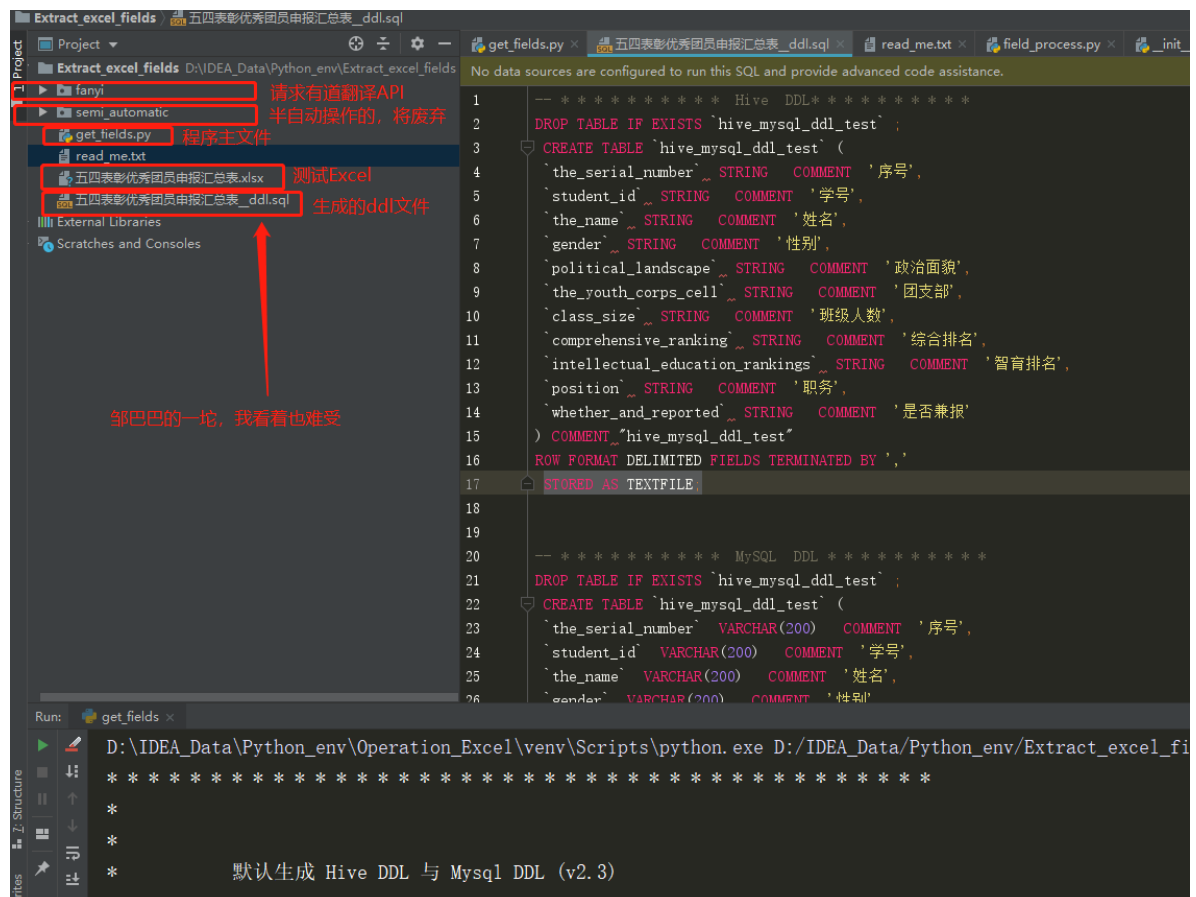
```
1 DROP TABLE IF EXISTS `hive_mysql_ddl_test`;  
2 CREATE TABLE `hive_mysql_ddl_test`  
3 (  
4 `data` STRING COMMENT '数据',  
5 `first` STRING COMMENT '第一',  
6 `code` STRING COMMENT '编码'  
7 ) COMMENT "hive_mysql_ddl_test"  
8 ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
9 STORED AS TEXTFILE;
```

- 生成的文件:
 - 生成的文件格式与源文件名字相同，不同的是后缀有 “__ddl.sql”
 - 例如:

```
1 abc.xlsx    源文件  
2 abc__ddl.sql 生成文件
```

Example:

项目文件说明:



Excel 文件说明:

序号	学号	姓名	性别	政治面貌	团支部	班级人数	综合排名	团费排名	职务	是否兼报
1							平均排名(精确到小数点后一位)	平均排名(精确到小数点后一位)	副班长	否
2									学习委员	否
3									无	否
4									团支书	是
5									宣传委员	否
6									心理委员	否
7									副班长、体育委员	否
8									团支部干事	否

fanyi文件夹

fanyi文件夹是请求有道API的 (request_youdao.py)

具体的逻辑, 在源码里面写的

```
request_youdao.py x
16 import requests
17
18 class get_request_youdao():
19
20     # Start function
21     def start(self):...
22
23
24     # Python constructor function
25     def __init__(self, keywords):...
26
27
28     # Constructor salt param
29     def get_salt(self, ts):...
30
31
32     # Constructor sign param
33     def get_sign(self, key, salt):...
34
35
36     # Constructor timestamp param
37     def get_ts(self):...
38
39
40     # Request Youdao
41     def get_fanyi_url(self):...
```