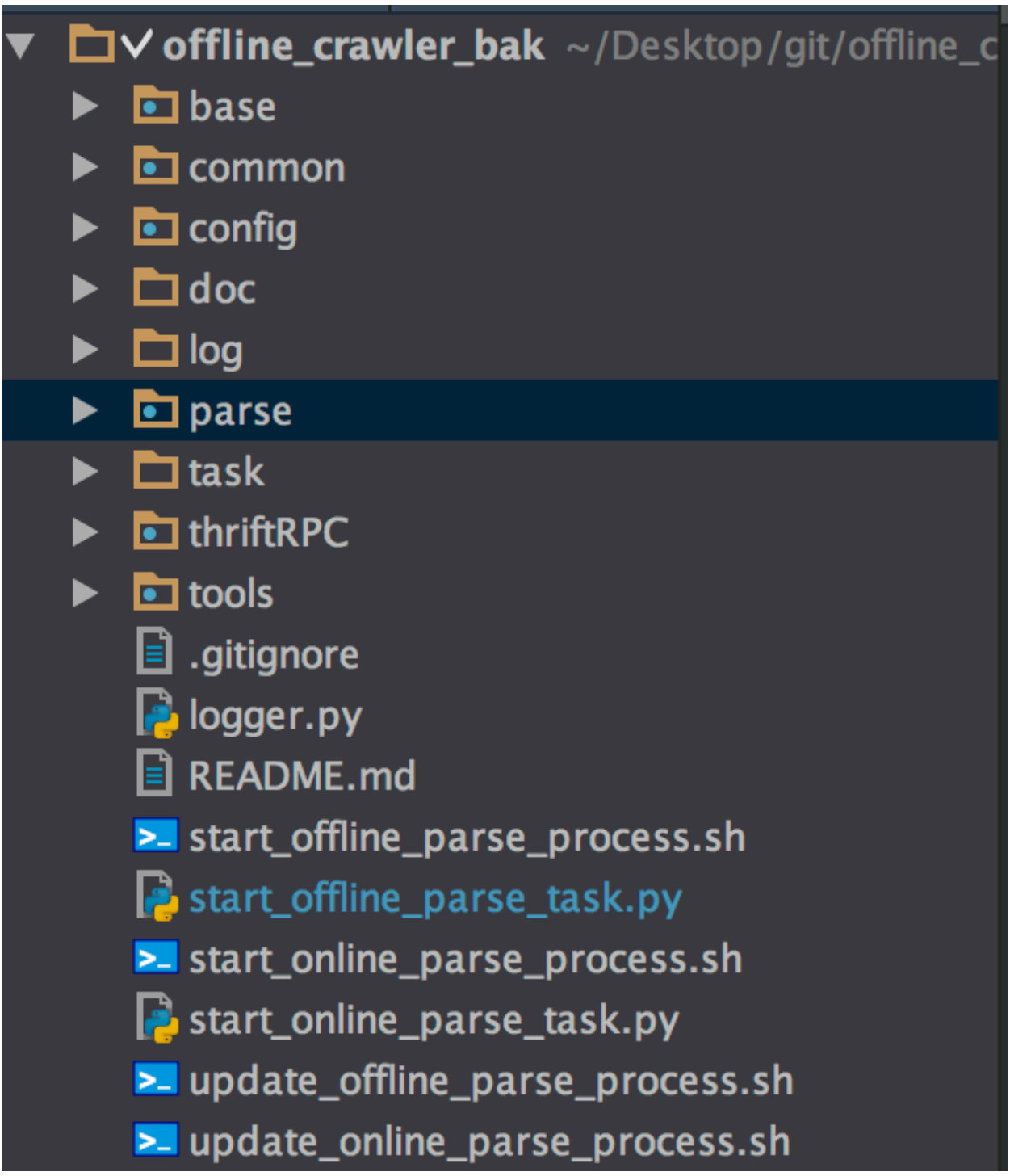
工商解析系统

1.功能说明

对工商捉取网页库进行解析
网页库位于mongdb crawl_data库 表格命名方式为: online_crawl*省份*new

> 案例:江苏省 --- online_crawl_jiangsu_new

2.程序文件目录说明



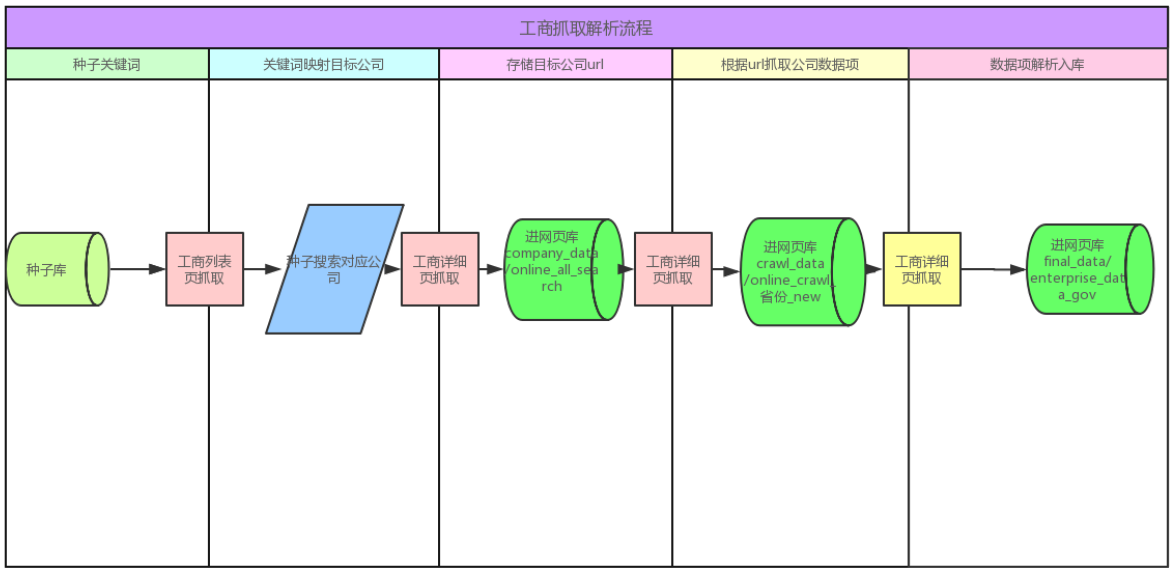
- /base —— 存放基类
- /common —— 程序工具共同类
- /config —— 配置文件
- /doc —— 说明文档
- /log —— 日志文档
- /parse —— 解析业务代码类
- /tools —— 工具脚本
- logger.py —— 日志类
- start_offline_parse_task.py —— 程序入口

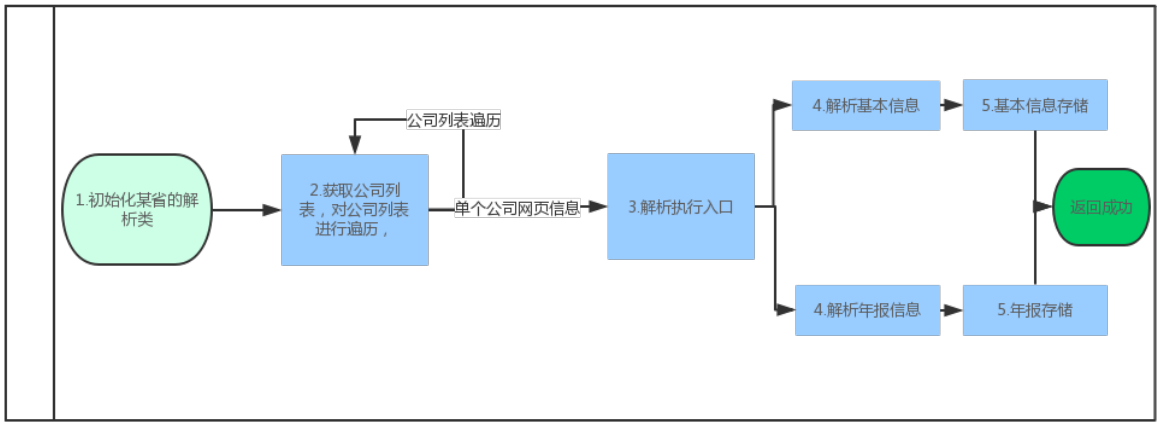
> 案例1:江苏省具体业务代码:/parse/parse_jiangsu_worker.py

> 案例2:数据库密码配置:/config/conf.py

3.程序执行流程

工商抓取解析系统流程





使用说明ctrl +f 搜索代码

3-1.初始化解析类:start_offline_parse_task.py

```
self.worker_list[key] = create_crawl_object(value, key)
```

> 案例:解析江苏省 ,此时:
>
> key为:'jiangsu' value{'class':'GsxtParseGuiZhouWorker','host':''gz.gsxt.gov.cn''.....}

3-2.获取公司列表，并对其遍历 start_offline_parse_task.py

```
# 公司列表
#分两个数据库装数据，老库(source_db)已满，所以添加新库(source_db_new)
company_list_old=self.__get_iterator_company_list(self.source_db)
company_list_new=self.__get_iterator_company_list(self.source_db_new)

count = 0
# 网页库遍历
for item in company_list_old:
  count += 1
  self.worker_list[self.province].query_offline_task(item,  CHOOSE_DB_OLD)
  #....

# 新版网页库 遍历
for item in company_list_new:
  count += 1
  self.worker_list[self.province].query_offline_task(item, CHOOSE_DB_NEW)
 #....
```

3-3.解析执行入口:parse_base_worker.py

query_offline_task() 内 会 调用 query_company()

```
def query_company(self, item):
```

3-4.1解析基本信息:parse_base_worker.py

```
gs_flag = self.parse_gs_info(company, data_list, u_time,
                                    _in_time, province, in_time,
                                    base_info_url, item)
```

3-4.2解析年报基本信息:parse_base_worker.py

```
nb_flag = self.parse_nb_info(company, data_list, u_time,
                             _in_time, province, in_time,
                             base_info_url)
```

3-5.1存储基本信息:parse_base_worker.py

```
# 位于 parse_gs_info()中
# 存储解析信息
return self.__store_model(company, base_info_url, in_time, model)
```

3-5.2存储年报信息:parse_base_worker.py

```
#位于 parse_nb_info()中
# 年报需要单独存储
if not self.__store_annual_model(company, year, base_info_url, in_time,
nb_model):
```

4.程序运行案例

各省解析类(parse_jiangsu_worker.py) —— 继承 —— parse_base_worker.py —— 继承—— task_base_worker.py

各省网页结构不同，解析代码不同。各省的解析代码 均在/parse下各个解析器下 如:江苏省的parse_jiangsu_worker.py

> 案例: 省份:江苏省 公司名:伊诗夏兰薇时装（南京）有限公司

4.1.入口参数修改

文件start_offline_parse_task.py

```
def main():
    config = 'config/offline_gsxt_parse.conf'
    province = 'jiangsu'
    max_count = 100000
```

province : 省份参数

config : 配置文件参数

> config= 'config/offline_gsxt_parse.conf' # 以config/offline_gsxt_parse.conf为配置文件

4.2.配置文件修改

```
#14
[jiangsu]
province=jiangsu
logfile=offline_parse_jiangsu
host=www.jsgsj.gov.cn:58888
clazz=GsxtParseJiangSuWorker
source_table=online_crawl_jiangsu_new
target_table=enterprise_data_gov
annual_table=annual_reports
crawl_flag=crawl_online
success_flag=50
source_select_param={'_id':'伊诗夏兰薇时装（南京）有限公司','crawl_online':{'$ne':100}}
is_nb_db_open=False
is_nb_mq_open=True
is_gs_db_open=False
is_gs_mq_open=True
gs_topic=49
gs_nb_topic=136
```

文件config/offline_gsxt_parse.conf

前往[jiangsu]的代码段

source_select_param: 解析参数

_id:公司名称

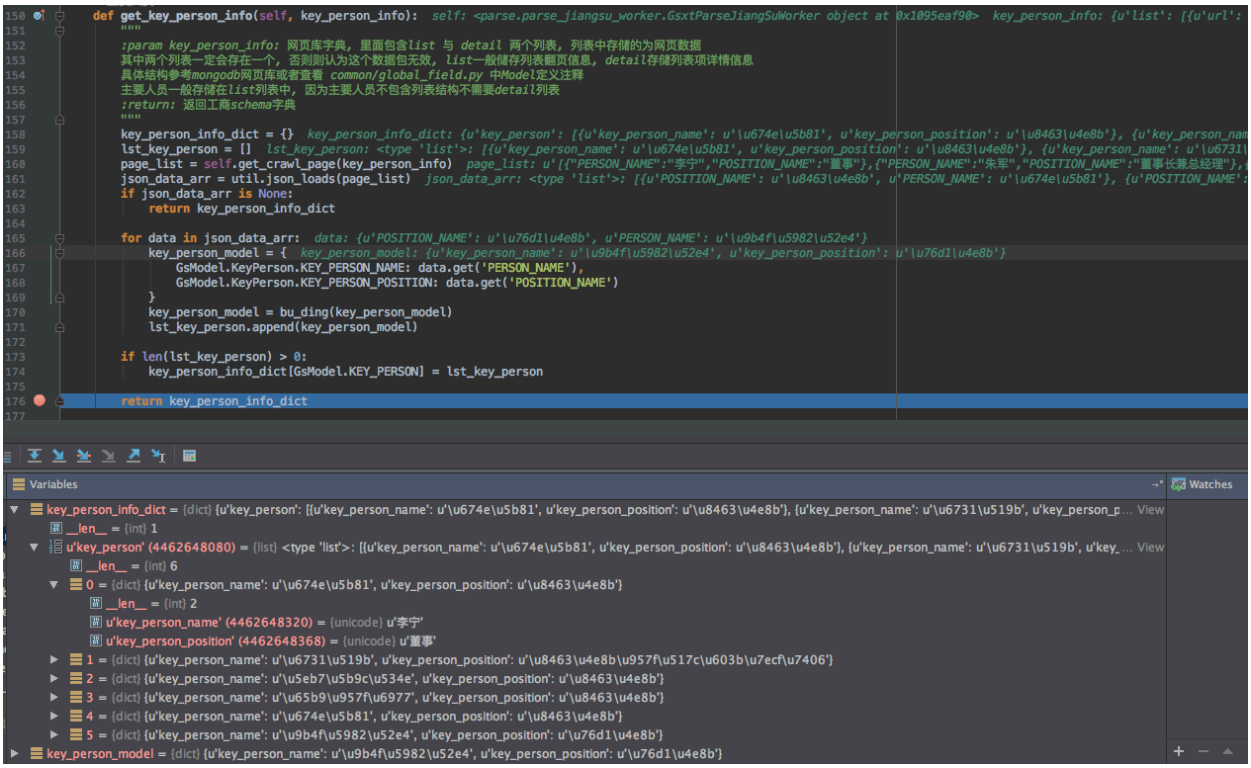
crawl_online：捉取状态

> '_id':'伊诗夏兰薇时装（南京）有限公司' # 搜索公司名为:伊诗夏兰薇时装（南京）有限公司
>
> 'crawl_online':{'$ne':100} # 搜索crawl_online 不等于100的数据

4.3查看修改某个数据项捉取情况

文件parse/jiangsu_worker.py

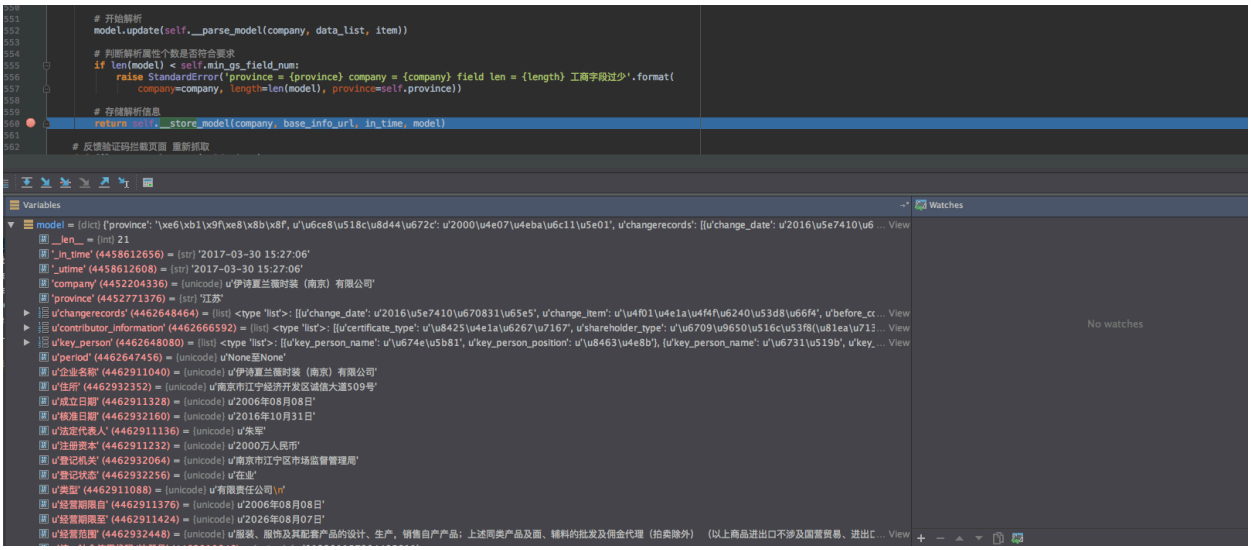
get_key_person_info函数 打断点



4.4查看解析基本信息入库前一刻

断点位置在文件parse_base_worker.py

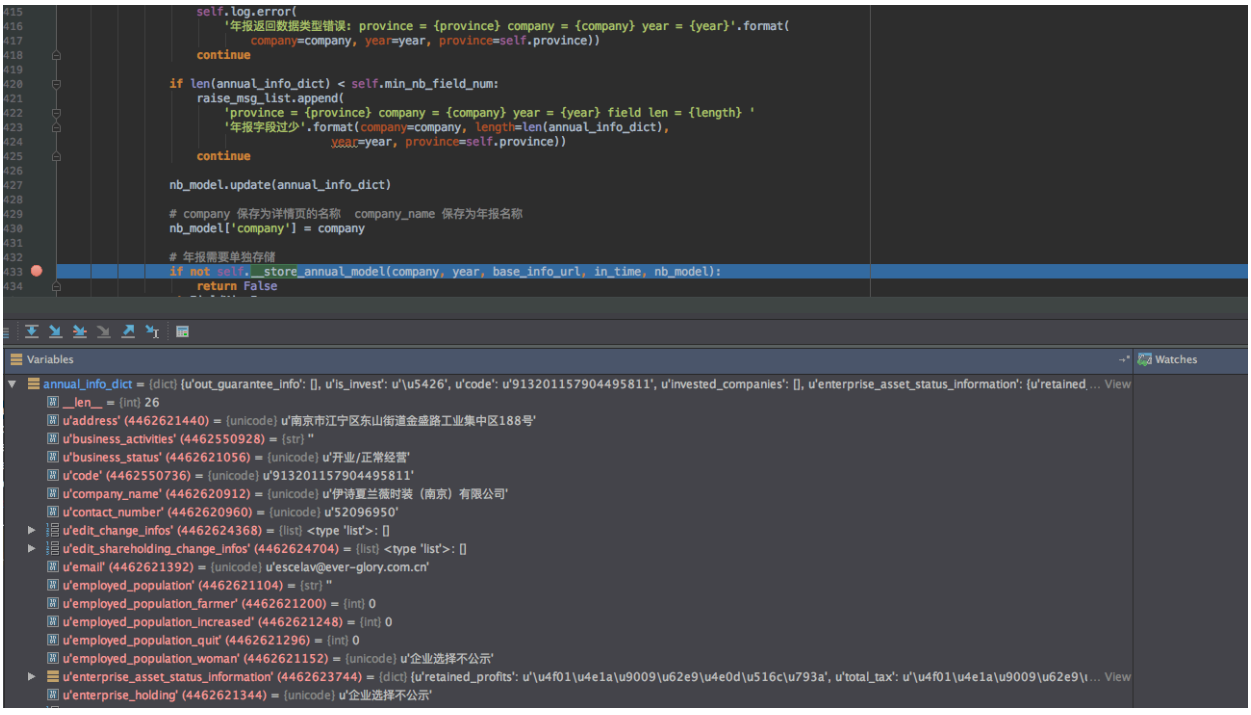
```
return self.__store_model(company, base_info_url, in_time, model)打断点
```



4.5查看解析年报信息入库前一刻

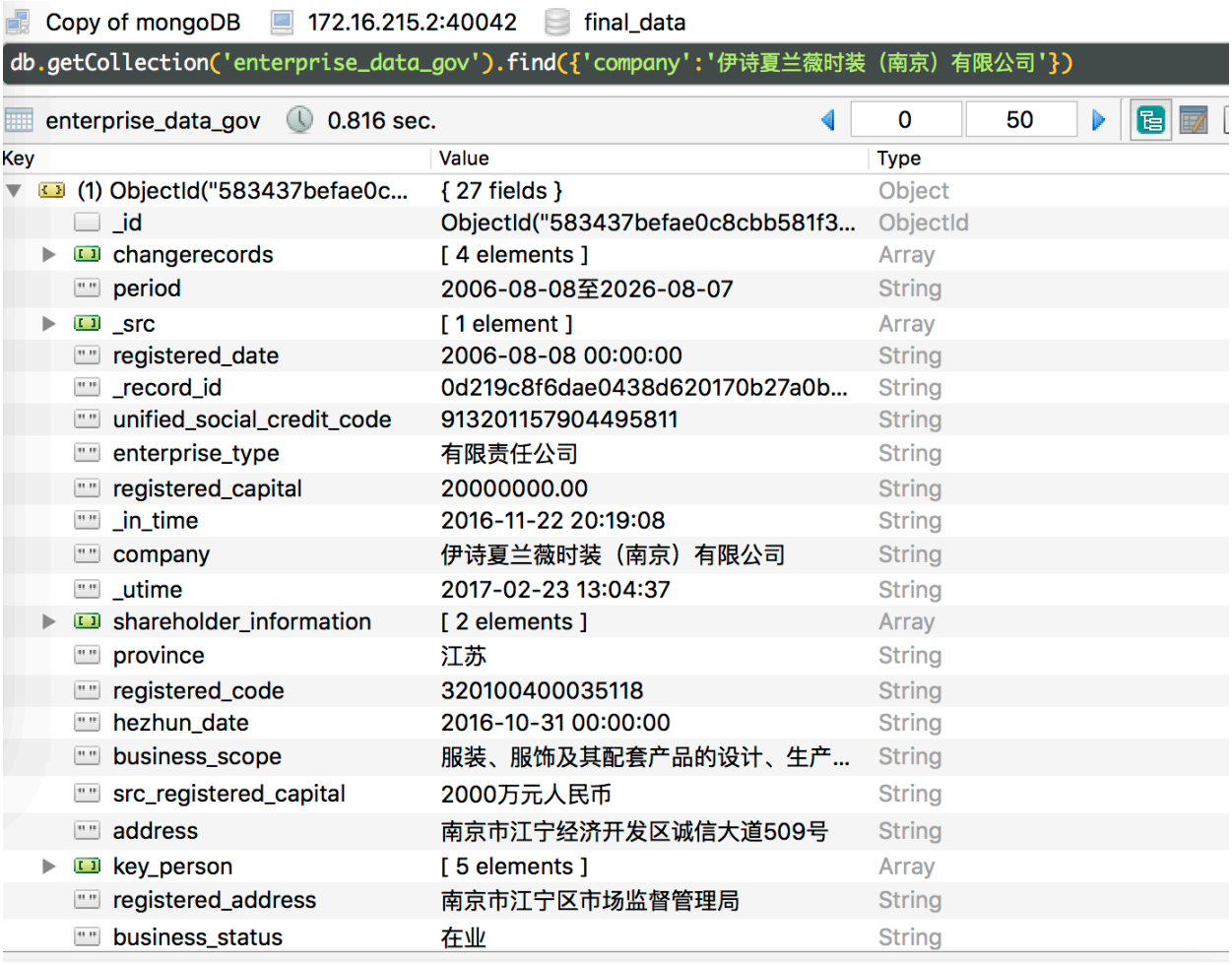
断点位置在文件parse_base_worker.py

```
if not self.__store_annual_model(company, year, base_info_url, in_time,
nb_model):
```



4.6数据库

前往mongoldb,final_data库/enterprise_data_gov 查看数据



如上图,数据已入库.

字段说明

mongoldb,crawl_data库/online_crawl*省份*new 字段说明

字段	对照官网字段
base_info	基本信息
branch_info	分支机构
change_info	变更信息
shareholder_info	股东及出资信息
contributive_info	股东及出资信息
annual_info	企业年报信息
_id	公司名字
province	省份
crawl_online	捉取状态