智联封装

```
import requests
from lxml import etree
import json
import MySQLdb
import com.baizhi.AI145.爬虫.urllib.utils.解析为字典的工具 as tools

# 1. 提高重用性
# 2. 列表页工具  详情页工具  数据库管理工具
# 3. 封装的程度 根据业务而定
# 4. 封装的质量： 1. 可读性  2. 解耦合性  3. 高可用  4. 高灵活性
# 5. 如果封装成类：1. 实现的代码，可以通过继承的方式传承下去  2. 如果有需要更改的方法，直接方法覆盖，不会影响原有代码 3. 可扩展性
# 6. 如果封装成函数：1. 如果该任务不需要修改（有且只有一种任务）可以封装成函数
# 7. 封装成类： 劣势：消耗的内存更大  优势：参考以上内容    解决方案：如果工具类中的方法不需要当前对象，尽量设置成静态方法（@staticmethod）
#    封装成函数：劣势：不够灵活，扩展性不够强    优势：占用资源少， 业务分明，耦合性更低
# 8. 代码质量要求：1. 耦合性小,2. 使用设计模式（23种）装饰器，插件，单例模式（懒汉式，饿汉式，完全懒汉式，完全饿汉式）。
#                  3. 资源消耗尽量少    4. 效率尽量高  5. 考虑时间复杂度，空间复杂度    6. 技巧

class URLListTools: # 列表页工具
    def __init__(self):
        pass
    # 1. 发送请求
    @ staticmethod
    def __sendRequest(url,requestType='get',data={},params={},headers={},cookies={},proxies={},**kwargs):

 res=requests.request(requestType,url,data=data,params=params,headers=headers,cookies=cookies,proxies=proxies,**kwargs)
        res.encoding=res.apparent_encoding
        return res.text

    # 2. 解析列表页中的详情url
    def  analysisUrls(self,url,**kwargs):
        data=self.__sendRequest(url,**kwargs)
        rData=json.loads(data)
        return [i['positionURL'] for i in rData['data']['results']]

    # 3. 对外提供调用接口---解耦合
    def getURLs(self,url,**kwargs):
        return self.analysisUrls(url,**kwargs)

class DetailTools: # 详情页工具
    # 解析数据
    def analysisData(self,urls,**kwargs):
```

```
        for url in urls:
            # 发送请求 并初始化
            e_html=etree.HTML(requests.get(url,**kwargs).text)
            # 解析html
            try:
                # 职位
                jobName = e_html.xpath('//h1[@class="l info-h3"]/text()')[0]  # 因为一个网站可能有多个模板，可能取到的数据为空--try-except
                # 公司名称
                company = e_html.xpath('//div[@class="company l"]/a/text()')[0]
                # 薪资
                salary = ''
                # 基本要求
                requirement = ''
                # 职位信息：岗位职责，岗位需求
                message = str(e_html.xpath('//div[@class="pos-ul"]//p/text()'))
                # 公司简介
                desc = ''
                # 公司规模
                companySize = ''
                # 联系方式
                tel = ''
            except:
                # 如果数据为空，重新给数据
                try:
                    # 提供第二套模板（解析规则）
                    message = str(e_html.xpath('//div[@class="pos-ul"]//div/text()'))
                except:
                    message = ''
            yield [jobName,company,salary,requirement,message,desc,companySize,tel]

class DataBaseTools: # 数据库操作工具
    def __init__(self,host,port,user,password,db,charset):

 self.conn=MySQLdb.Connection(host=host,port=port,user=user,password=password,db=db,charset=charset)
        self.cursor=self.conn.cursor()

    def saveDB(self,data,dbName):
        n=len(data)
        sql='insert into '+dbName+' values('+'%s,'*n[:-1]+')'
        self.cursor.execute(sql,data)
        self.conn.commit()
    def close(self):
        self.cursor.close()
        self.conn.close()

    @classmethod
    def dbFrom(cls,host='localhost',port=3306,user='root',password='123456',db='crawler',charset='utf8'):
        # 可以做前置工作，并且不需要消耗当前对象资源
        # 类方法不需要创建实例对象就可以使用
```

```
        return cls(
            host=host,
            port=port,
            user=user,
            password=password,
            db=db,
            charset=charset )

def main(cityIDs=[],kws=[],url='',cookies={},cityIdex=0,kwIndex=0,start=0,dbName=''):
    while 1:
        listUrls=URLListTools().getURLs(url,cookies=cookies)
        detailDatas=detailTools.analysisData(listUrls,cookies=cookies) # 是个生成器：是个
可迭代对象
        for i in detailDatas: # i:每一次的所有数据
[jobName,company,salary,requirement,message,desc,companySize,tel]
            dbTools.saveDB(i,dbName)
        if len(listUrls) <= 0:  # 80%
            # 此时视为该条件已经爬取完毕
            # 更换条件继续查询--本质更换的是下标
            if cityIndex == len(cityIDs) - 1:  # city已满
                cityIndex = 0
                if kwIndex == len(kws) - 1:  # kw已满
                    # 彻底执行完毕
                    dbTools.close()
                    break
                else:
                    kwIndex += 1
            else:
                cityIndex += 1

        else:
            # 分页
            start += 90


if __name__ == '__main__':
    # 创建工具
    dbTools=DataBaseTools.dbFrom() # 通过方法，或函数创建的对象--- 工厂模式
    listTools=URLListTools()
    detailTools=DetailTools()
    url = 'https://fe-api.zhaopin.com/c/i/sou?
pageSize=90&cityId=635&salary=0,0&workExperience=-1&education=-1&companyType=-1&employm
entType=-1&jobWelfareTag=-1&kw=python&kt=3&=0&at=5933c31f4f42494ca7b2d11c81476227&rt=c7
a0066831414df2afa815f3f425d175&_v=0.53217532&userCode=635249182&x-zp-page-request-
id=6c20e277dcd4427d970961139e1e65b6-1548236569547-481583'
    cookies = tools.analysisByEqual(
```

```
        'adfbid2=0; sts_deviceid=16855a20d8443d-0786d15818d698-5d1f3b1c-1044480-
16855a20d85754; JSloginnamecookie=18730231911; JSShowname=%E8%B5%B5%E9%B9%8F%E9%A3%9E;
ZP_OLD_FLAG=false; adfcid2=none;
__xsptplus30=30.3.1548122438.1548122438.1%232%7Csp0.baidu.com%7C%7C%7C%25E6%2599%25BA%2
5E8%2581%2594%25E6%258B%259B%25E8%2581%2598%7C%23%23BgfSyvgiMvhv_Brk_gA4aHUvi9FoYpJJ%23
;
_jzqy=1.1548057849.1548122439.1.jzqsr=baidu|jzqct=%E6%99%BA%E8%81%94%E6%8B%9B%E8%81%98.
-; smidV2=201901221042410a679e4110dac47f058085068857e79a00de975d4f456b2a0;
urlfrom2=121113803; LastCity=%E5%8D%97%E4%BA%AC; LastCity%5Fid=635; urlfrom=121113803;
adfbid=0;
sensorsdata2015jssdkcross=%7B%22distinct_id%22%3A%2263524918 2%22%2C%22%24device_id%22%3
A%2216855a1ffee7a6-030b7ae9fefb92-5d1f3b1c-1044480-
16855a1fff035a%22%2C%22props%22%3A%7B%22%24latest_traffic_source_type%22%3A%22%E4%BB%98
%E8%B4%B9%E5%B9%BF%E5%91%8A%E6%B5%81%E9%87%8F%22%2C%22%24latest_referrer%22%3A%22https%
3A%2F%2Fsp0.baidu.com%2F9q9JcDHa2gU2pMbgoY3K%2Fadrc.php%3Ft%3D06KL00c00fZmx9C05x-
6OKqiAsjlnD9T00000FtgOdC000000ToeAW.THLyktAJ0A3qrH6dPW04n1wxpA7EgLKM0ZnqmHfsPHD1ujfsnj0
1uWmkmfKd5Rn1fYmvwWckPWnzPHT1wDRvPbPjn1D%22%2C%22%24latest_referrer_host%22%3A%22sp0.ba
idu.com%22%2C%22%24latest_search_keyword%22%3A%22%E6%99%BA%E8%81%94%22%2C%22%24latest_u
tm_source%22%3A%22baidupcpz%22%2C%22%24latest_utm_medium%22%3A%22cpt%22%7D%2C%22first_i
d%22%3A%2216855a1ffee7a6-030b7ae9fefb92-5d1f3b1c-1044480-16855a1fff035a%22%7D;
dywea=95841923.1897540460572125700.1547624907.1548140439.1548234826.6; dywec=95841923;
dywez=95841923.1548234826.6.5.dywecsr=baidupcpz|dyweccn=
(not%20set)|dywecmd=cpt|dywectr=%E6%99%BA%E8%81%94;
Hm_lvt_38ba284938d5eddca645bb5e02a02006=1547624961,1548057833,1548122438,1548234826;
sts_sg=1; sts_sid=16879fca278923-09c2750d84ac15-5d1f3b1c-1044480-16879fca2791de;
sts_chnlsid=121113803;
zp_src_url=https%3A%2F%2Fsp0.baidu.com%2F9q9JcDHa2gU2pMbgoY3K%2Fadrc.php%3Ft%3D06KL00c0
0fZmx9C05x-
6OKqiAsjlnD9T00000FtgOdC000000ToeAW.THLyktAJ0A3qrH6dPW04n1wxpA7EgLKM0ZnqmHfsPHD1ujfsnj0
1uWmkmfKd5Rn1fYmvwWckPWnzPHT1wDRvPbPjn1DknjNDnRmYP1cd0ADqI1YhUyPGujY1nHT1P1mLrHczFMKzUv
wGujYkP6K-5y9YIZK1rBtEILILQMGCmyqspy38mvqV5LPGujYknWDknHn3njnhIgwVgLPEIgFWuHdBmy-
bIgKWTZChIgwVgvd-uA-
dUHdWTZf0mLFW5HfkPwfz%26tpl%3Dtpl_11535_18778_14772%26l%3D1510152095%26attach%3Dlocatio
n%253D%2526linkName%253D%2525E6%2525A0%252587%2525E5%252587%252586%2525E5%2525A4%2525B4
%2525E9%252583%2525A8-%2525E6%2525A0%252587%2525E9%2525A2%252598-
%2525E4%2525B8%2525BB%2525E6%2525A0%252587%2525E9%2525A2%252598%2526linkText%253D%2525E
3%252580%252590%2525E6%252599%2525BA%2525E8%252581%252594%2525E6%25258B%25259B%2525E8%2
52581%252598%2525E3%252580%252591%2525E5%2525AE%252598%2525E6%252596%2525B9%2525E7%2525
BD%252591%2525E7%2525AB%252599%252520%2525E2%252580%252593%252520%2525E5%2525A5%2525BD%
2525E5%2525B7%2525A5%2525E4%2525BD%25259C%2525EF%2525BC%25258C%2525E4%2525B8%25258A%252
5E6%252599%2525BA%2525E8%252581%252594%2525E6%25258B%25259B%2525E8%252581%252598%2525EF
%2525BC%252581%2526xp%253Did(%252522m3173767922_canvas%252522)%25252FDIV%25255B1%25255D
%25252FDIV%25255B1%25255D%25252FDIV%25255B1%25255D%25252FDIV%25255B1%25255D%25252FDIV%2
5255B1%25255D%25252FH2%25255B1%25255D%25252FA%25255B1%25255D%2526linkType%253D%2526chec
ksum%253D147%26ie%3Dutf-
8%26f%3D8%26srcqid%3D2779934700804399157%26tn%3D98560934_hao_pg%26wd%3D%25E6%2599%25BA%
25E8%2581%2594%26oq%3D%25E6%2599%25BA%25E8%2581%2594%26rqlang%3Dcn%26sc%3DUWY4rjRvnjb1P
7qCmyqxTAThIjYkPHf3nW04nHf4n1fzFhnqpA7EnHc1Fh7w5Hn3PHDvnjn1PHR%26ssl_sample%3Ds_103%26H
123Tmp%3Dnu; __utma=269921210.1814771033.1547624911.1548140439.1548234849.6;
__utmc=269921210; __utmz=269921210.1548234849.6.5.utmcsr=ts.zhaopin.com|utmccn=
(referral)|utmcmd=referral|utmcct=/jump/index_new.html;
_jzqa=1.1153422773997826000.1547624940.1548140448.1548234849.5; _jzqc=1;
_jzqx=1.1547624940.1548234849.1.jzqsr=ts%2Ezhaopin%2Ecom|jzqct=/jump/index_new%2Ehtml.-
```

```
; _jzqckmp=1; _jzqb=1.1.10.1548234849.1;
firstchannelurl=https%3A//passport.zhaopin.com/login;
lastchannelurl=https%3A//ts.zhaopin.com/jump/index_new.html%3Futm_source%3Dbaidupcpz%26
utm_medium%3Dcpt%26utm_provider%3Dpartner%26sid%3D121113803%26site%3Dnull;
JsNewlogin=1905747548; at=5933c31f4f42494ca7b2d11c81476227;
Token=5933c31f4f42494ca7b2d11c81476227; rt=c7a0066831414df2afa815f3f425d175;
JSpUserInfo=3e692f645b6a5f64406b5c6b56665e695d735a695364546a52643f6b276b59665b695c735c6
95864566a5b64406b586b5d665b69557350693e64286a546406e626f5bafe5169217326695664576a58644b
6b5b6b526653695c735d695a64526a2964026b186b4a6609690b7306695064356a3d644e6b586b5f662b693
07356695b644b6a5a64536b586b5366506956735a695064276a25644e6b586b5f663f692573566921642b6a
5a64466b5b6b5066536955735369596 45f6a5264266b3d6b59665b695f73386922645b6a5964486b4;
uiioit=3d79306c4d735265576442640332506840745d704b6457645f77263176645579446c4b732;
jobRiskWarning=true; dyweb=95841923.7.6.1548235082260;
__utmb=269921210.6.6.1548235082263;
acw_tc=3ccdc15b15482352958395118e3f765b02e3f56286d2b01cb2ad5779a3368e;
ZL_REPORT_GLOBAL={%22sou%22:{%22actionid%22:%22e75d2c7a-cea8-433c-816f-d49b797f9efd-
sou%22}}; sts_evtseq=9; Hm_lpvt_38ba284938d5eddca645bb5e02a02006=1548236576')


    main([530, 801, 854, 635],['爬虫', '大数据', 'python',
'AI'],kwIndex=0,start=0,cityIndex=0,url=url,cookies=cookies
```

IP 代理池

1. IP被封禁
   - 换IP---找代理
   - 找代理IP：
     1. 不知道是否可用
     2. 不知道何时失效
     3. 不知道从哪里获取代理
     4. 失效之后的ip未来是否还能用
     5. 要获取有用的IP需要较长的时间
2. 提前准备好有用的IP
   - 异步的动态的给IP代理池补充可用的IP（多线程）
3. 对IP进行管理
   - 池：进程池 串池 常量池
   - 池：1. 是个容器 2. 应该有容量 3. 批量处理任务
4. 流程：
   1. IP数据源
   2. 检测可用性
   3. 入库存储
   4. 对外提供获取IP接口
   5. 出库
   6. 原则上：出库的IP应该丢弃
      修改状态值
   7. 定期检查所有入库的IP的可用性
5. 策略:
   - 超市 货物 --- 快没有货时，进行补货
   - 设置：阈值 ---一个上限值
   - 如果可用IP数量减少到了阈值，此时开始补充IP，补满（补充到最大值）