项目二 爬虫编程

一、实验目的

1. 了解爬虫编程技术的应用前景

2. 掌握爬虫的概念

3. 懂得编写爬虫程序

4. 懂得使用框架编写爬虫

5. 懂得如何提升爬虫运行速度

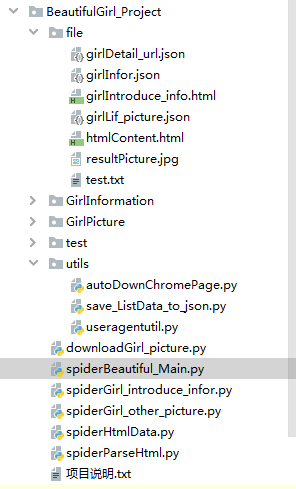
6. 掌握应对反爬虫网页的相关技术

7. 具备项目分析能力

8. 懂得项目的创新与应用

二、实验原理、内容与步骤要求

整体项目结构如下：



1. 实验原理

1. 通过分析网页的结构，判断是否是动态网页，然后采取相应的爬虫技术。
2. 获取网页内容所需数据，将爬取的数据保存在本地，如json格式的文件
3. 将保存的数据进行相应的数据分析

2 实验内容

爬取校花美女网的美女图片，个人资料信息，点赞数，大学，详情地址，图片，网址链接<http://www.xueshengmai.com/list-1-0.html>

3. 实验步骤

1. 、分析网页结构，判断所爬取的数据是否是动态结构，图如下：

网页数据：



Network数据：



分析得：所需数据不是采取动态网页渲染的效果，所以可以采取常规手段爬取数据

1. 、爬取美女图片链接地址、点赞数、美女名称

a. 获取网页内容，并保存首页数据

代码如下：

**import** requests  
**import** os  
**from** BeautifulGirl\_Project.utils **import** useragentutil  
  
**"获取待爬取的网页信息"  
def** getHtml(url):  
 html = requests.get(url,verify=**False**,headers=useragentutil.get\_headers())  
 *# 处理乱码问题* htmlContent = html.content.decode(**"gbk"**)  
 **return** htmlContent  
  
**"""先将该首页的（第一页)数据保存起来"""  
def** save\_Html\_data(htmlContent):  
 *# 创建一个file文件夹* dirName = **"file"  
 if not** os.path.exists(dirName):  
 os.makedirs(dirName)  
 *# 打开一个文件，没有就创建  
 # file = open("./file/htmlContent.html","w",encoding="utf-8")* file = open(**"./file/girlIntroduce\_info.html"**, **"w"**, encoding=**"utf-8"**)  
 *# 写入数据* file.write(htmlContent)  
 *# 关闭文件* file.close()  
 print(**"文件保存成功"**)  
  
**def** main():  
 *# url = "http://www.xueshengmai.com/list-1-0.html"* details\_url = **"http://www.xueshengmai.com/p-1-2092.html"** htmlContent = getHtml(details\_url)  
 save\_Html\_data(htmlContent)  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

b. 采取的python模块是requests模块，运用lxml中的xpath技术分析网页所需数据结构,并把数据保存为json格式的数据

代码如下：

**import** lxml.html  
**from** BeautifulGirl\_Project.utils **import** save\_ListData\_to\_json **as** save  
**from** BeautifulGirl\_Project **import** spiderHtmlData  
  
**"""该.py是用来解析网页，然后获取自己想要的数据信息"""  
  
  
"""解析网页数据，返回美女姓名、图片地址、点赞数、详细地址四个列表"""  
def** get\_BeautifulGirl\_info(htmlContent):  
 *# 获取解析器* metree = lxml.html.etree  
 *# 获取解析器对象* parse = metree.HTML(htmlContent)  
 *# 开始解析  
 # 美女姓名* girlName\_list = parse.xpath(**"//div[@class='img']/span/text()"**)  
 *# 美女图片地址 http://www.xueshengmai.com/d/file/20200116/smallf29ef026905f40d34814b251c0c85d761579167601.jpg* girlImg\_url\_list = parse.xpath(**"//div[@class='img']/a/img/@src"**)  
 *# 美女点赞数* girlZan\_count\_list = parse.xpath(**"//div[@class='items\_likes fl']/em/text()"**)  
 *# 美女详细信息地址* girlDetails\_url\_list = parse.xpath(**"//div[@class='img']/a/@href"**)  
 **return** girlName\_list, girlImg\_url\_list, girlZan\_count\_list, girlDetails\_url\_list  
  
  
  
**"""将每个美女数据拼接成一个小字典，最后用一个大列表包住，最后返回这个大列表"""  
def** splicing\_data\_to\_dict(girlName\_list, girlImg\_url\_list, girlZan\_count\_list, girlDetails\_url\_list):  
 *# 定义一个大列表* girlList = []  
 **for** index **in** range(0,len(girlName\_list)):  
 *# 定义一个字典，用来封装美女信息* girDict = {}  
 girDict[**"girlName"**] = girlName\_list[index]  
 girDict[**"girlImg\_url"**] = **"http://www.xueshengmai.com"**+girlImg\_url\_list[index]  
 girDict[**"girlDetails\_url"**] = girlDetails\_url\_list[index]  
 girDict[**"点赞数"**] = girlZan\_count\_list[index]  
 *# 将每个美女字典添加到大列表中* girlList.append(girDict)  
 **return** girlList  
  
  
**"""爬取指定页数的数据"""  
def** spiderDataLimt(num):  
 *# 存放所有美女的信息列表* girlAllInfo = []  
 *# 用于递增url的页数* **for** i **in** range(0,num):  
 url = **"http://www.xueshengmai.com/list-1-"** + str(i) + **".html"** htmlContent = spiderHtmlData.getHtml(url)  
 *# 解析网页数据，并返回所需信息* girlName\_list, girlImg\_url\_list, girlZan\_count\_list, girlDetails\_url\_list = get\_BeautifulGirl\_info(htmlContent)  
 *# 将每个美女数据拼接成一个小字典，最后用一个大列表包住，最后返回这个大列表* girlList = splicing\_data\_to\_dict(girlName\_list, girlImg\_url\_list, girlZan\_count\_list, girlDetails\_url\_list)  
 girlAllInfo.append(girlList)  
 *# 将列表数据保存为json数据* filePath = **"D:/python练习/校花美女图片爬取/BeautifulGirl\_Project/file/girlInfor.json"** save.save\_Data\_to\_json(filePath, girlAllInfo,**"w"**)  
  
**def** main():  
 *# 这里可以修改爬取页数* spiderDataLimt(4)  
 print(**"程序已经走完"**)  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

保存的数据：

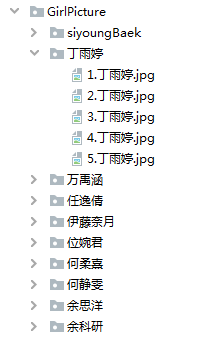


c. 下载美女图片，并保存在本地用到的技术是os模块，已及对文件的操作

代码如下：

*"""此.py用于下载美女图片"""***import** json  
**import** os  
**import** requests  
**from** BeautifulGirl\_Project.utils **import** useragentutil  
  
  
**"""将json数据转成python基本数据类型,并返回 """  
def** load\_Json\_to\_data(filePath):  
 *# 将json数据转成python基本数据类型,并返回  
 # 打开文件* file = open(filePath,**"r"**,encoding=**"utf-8"**)  
 *# 加载数据* girlAll\_list = json.load(file)  
 *# 关闭文件* file.close()  
 **return** girlAll\_list  
  
  
**"""为每个美女创建一个文件夹，名就是她的名字"""  
def** createGirlDir(girlAll\_list):  
 *# 创建一个Picture文件夹* pictureDir = **"./GirlPicture"  
 if not** os.path.exists(pictureDir):  
 os.makedirs(pictureDir)  
 **for** list **in** girlAll\_list:  
 **for** elements\_dict **in** list:  
 *# 美女名字* girlName = elements\_dict[**"girlName"**]  
 **if not** os.path.exists((pictureDir+**"/"**+girlName)):  
 os.makedirs((pictureDir+**"/"**+girlName))  
 print(**"文件夹创建成功"**)  
  
  
**def** downloadPicture(girlAll\_list):  
 *# 记录图片的张数* i = 0  
 **for** list **in** girlAll\_list:  
 **for** elements\_dict **in** list:  
 *# 美女名字* girlName = elements\_dict[**"girlName"**]  
 *# 获取美女图片链接* girlImg\_url = elements\_dict[**"girlImg\_url"**]  
 *# 获取美女图片* ImgContent = requests.get(girlImg\_url,verify=**False**,headers=useragentutil.get\_headers())  
 *# 打开文件夹* filePath = **"./GirlPicture"**+**"/"**+girlName+**"/1."**+girlName+**".jpg"** file = open(filePath,**"wb"**)  
 file.write(ImgContent.content)  
 *# 关闭文件夹* file.close()  
 i = i+1  
 print(**"第%d张图片已经下载成功"**%i)  
  
  
**def** main():  
 filePath = **"./file/girlInfor.json"** *# 读取数据* girlAll\_list = load\_Json\_to\_data(filePath)  
 *# 创建文件夹* createGirlDir(girlAll\_list)  
 *# 下载图片* downloadPicture(girlAll\_list)  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

图片截图：



1. 、下载美女详细资料，并保存为txt文本

a. 首先获取详细资料的地址，并保存为json格式

代码如下：

*"""此.py文件是用来爬取美女的个人信息的"""***from** BeautifulGirl\_Project **import** downloadGirl\_picture  
**from** BeautifulGirl\_Project.utils **import** save\_ListData\_to\_json **as** save  
**import** os  
**import** requests  
**from** BeautifulGirl\_Project.utils **import** useragentutil  
**import** lxml.html  
  
  
**"""该函数：用于获取美女的详情链接，并于列表形式返回"""  
def** getGirl\_detatils\_url\_list(girlAll\_info):  
 *# 定义一个大列表* girlDetails\_url\_list = []  
 **for** list **in** girlAll\_info:  
 **for** elements\_dict **in** list:  
 *# 定义一个字典，用来存放美女信息* girlDetails\_url\_dict = {}  
 *# 美女名字* girlName = elements\_dict[**"girlName"**]  
 *# 详情链接* girlDetails\_url = elements\_dict[**"girlDetails\_url"**]  
 girlDetails\_url\_dict[**"girlName"**] = girlName  
 girlDetails\_url\_dict[**"Details\_link"**] = girlDetails\_url  
 *# 加入到列表中* girlDetails\_url\_list.append(girlDetails\_url\_dict)  
 **return** girlDetails\_url\_list  
  
  
**"""该函数：用于获取每个校花的详细资料"""  
  
  
def** create\_Introduce\_txt(girlName,detailsContent\_list):  
 fileDir = **"./GirlInformation"  
 if not** os.path.exists(fileDir):  
 os.makedirs(fileDir)  
 fileName = girlName+**"个人资料"** *# 打开文件* file = open(fileDir+**"/"**+fileName+**".txt"**,**"a"**,encoding=**"utf-8"**)  
 **for** str **in** detailsContent\_list:  
 *# 写入数据* file.writelines(str+**"\n"**)  
 *# 关闭文件* file.close()  
 *# print("数据写入成功")***def** getGirl\_introduce(girlDetail\_url\_list):  
 i = 0  
 **for** dict **in** girlDetail\_url\_list:  
 *# 美女名字* girlName = dict[**"girlName"**]  
 *# 美女详情链接* girlDetail\_link = dict[**"Details\_link"**]  
 *# 获取详情的网页内容* htmlContent = requests.get(girlDetail\_link,verify=**False**,headers=useragentutil.get\_headers()).content.decode(**"gbk"**)  
 *# 获取解析器* metree = lxml.html.etree  
 *# 获取解析器对象* parse = metree.HTML(htmlContent)  
 *# 开始解析网页内容，获取美女详情信息* detailsContent\_list = parse.xpath(**"//div[@class='infocontent']/p/span/text()"**)  
 *# 用于填充没有爬取到的美女信息* lessContent\_list = parse.xpath(**"//\*[ @ id = 'post']/div[1]/div[1]/div[4]/table/tbody/tr[5]/td[2]/text()"**)  
 *# 以下三句，只是用来填充无法获取到详细信息时的文本信息* schoolName = **"学校:"**+lessContent\_list[0]  
 lessContent\_list[0] = girlName  
 lessContent\_list.append(schoolName)  
 *# 判断detailsContent\_list是否为空* **if** detailsContent\_list:  
 *# 创建文本* create\_Introduce\_txt(girlName, detailsContent\_list)  
 **else**:  
 *# 为空，填充基本信息  
 # 创建文本* create\_Introduce\_txt(girlName, lessContent\_list)  
 *# 用来记录下载数* i = i + 1  
 print(i)  
  
**def** main():  
 filePath = **"./file/girlInfor.json"** *# 读取数据* girlAll\_info = downloadGirl\_picture.load\_Json\_to\_data(filePath)  
 *# 从美女信息中获取详情地址* girlDetails\_url\_list = getGirl\_detatils\_url\_list(girlAll\_info)  
 *# 将美女详情地址的数据保存在json中* girlDetail\_Json\_path = **"D:/python练习/校花美女图片爬取/BeautifulGirl\_Project/file/girlDetail\_url.json"** save.save\_Data\_to\_json(girlDetail\_Json\_path,girlDetails\_url\_list,**"w"**)  
 girlDetail\_url\_path = **"./file/girlDetail\_url.json"** *# 读取数据* girlDetail\_url\_list = downloadGirl\_picture.load\_Json\_to\_data(girlDetail\_url\_path)  
 *# 获取美女资料* getGirl\_introduce(girlDetail\_url\_list)  
 print(**"程序已走完"**)  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

数据截图：

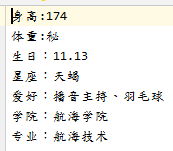


b. 获取详细资料，并保存，代码同上：

效果截图：



资料详情：



1. 、爬取美女生活照，并保存至相应的文件夹中

a. 首先同样需要获取美女生活照的图片地址，并保存数据

代码如下：

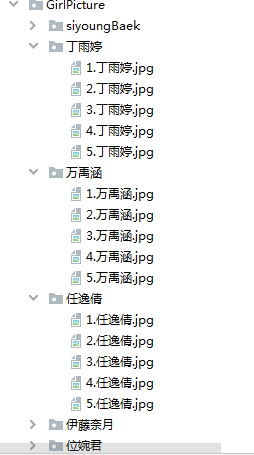
*""""该.py是爬取每个美女的生活照"""***from** BeautifulGirl\_Project **import** downloadGirl\_picture  
**import** requests  
**from** BeautifulGirl\_Project.utils **import** useragentutil  
**import** lxml.html  
**from** BeautifulGirl\_Project.utils **import** save\_ListData\_to\_json **as** save  
  
  
  
**"""用于获取美女生活照的链接，并返回列表"""  
def** getGirl\_life\_picture\_url\_list(girlDetail\_list):  
 *# 定义一个大列表，用于存放美女生活照信息 格式如[{"name":[……]},{"name2":[……]}……]* Allgirl\_life\_picture\_url\_list = []  
 **for** dict **in** girlDetail\_list:  
 *# 定义一个字典，用于封装美女生活照信息* girLif\_picture\_link\_dict = {}  
 *# 获取美女名称* girlName = dict[**"girlName"**]  
 *# 美女详情链接* girlDetail\_link = dict[**"Details\_link"**]  
 *# 获取详情的网页内容* htmlContent = requests.get(girlDetail\_link,verify=**False**,headers=useragentutil.get\_headers()).content.decode(**"gbk"**)  
 *# 获取解析器* metree = lxml.html.etree  
 *# 获取解析器对象* parse = metree.HTML(htmlContent)  
 *# 开始解析网页内容，获取美女详情信息* girl\_life\_picture\_url\_list = parse.xpath(**"//div[@class='p-tmb']/img/@src"**)  
 girLif\_picture\_link\_dict[**"girlName"**] = girlName  
 girLif\_picture\_link\_dict[**"life\_picture\_link"**] = girl\_life\_picture\_url\_list  
 *# 将这个大字典添加到大列表中* Allgirl\_life\_picture\_url\_list.append(girLif\_picture\_link\_dict)  
 **return** Allgirl\_life\_picture\_url\_list  
  
**"""函数：用于下载美女生活照，并且保存到相应的文件夹中"""  
def** download\_life\_picture(girlLife\_link\_list):  
 k = 0  
 **for** dict **in** girlLife\_link\_list:  
 *# 用于标记图片序号* i = 2  
 *# 获取美女名字* girlName = dict[**"girlName"**]  
 *# 获取美女生活照链接列表* life\_picture\_link\_list = dict[**"life\_picture\_link"**]  
 *# 遍历链接，下载图片* **for** link **in** life\_picture\_link\_list:  
 *# 拼接好图片网址* url = **"http://www.xueshengmai.com"**+link  
 *# 获取图片内容* **try**:  
 ImgContent = requests.get(url,verify=**False**,headers=useragentutil.get\_headers()).content  
 **except** Exception **as** e:  
 *# 打印报错信息* print(e)  
 print(girlName)  
 print(url)  
 *# 打开文件* filePath = **"./GirlPicture/"**+girlName+**"/"**+str(i)+**"."**+girlName+**".jpg"** file = open(filePath,**"wb"**)  
 file.write(ImgContent)  
 *# 关闭文件* file.close()  
 i = i + 1  
 k = k + 1  
 print(**"第%d个美女的生活照下载完毕"**%k)  
  
**def** main():  
 *# 加载json数据* filePath = **"./file/girlDetail\_url.json"** girlDetail\_list = downloadGirl\_picture.load\_Json\_to\_data(filePath)  
 *# 获取每个美女生活照的地址* girl\_life\_picture\_url\_list = getGirl\_life\_picture\_url\_list(girlDetail\_list)  
 *# 将它保存为json格式的数据* girl\_life\_picturePath = **"D:/python练习/校花美女图片爬取/BeautifulGirl\_Project/file/girlLif\_picture.json"** save.save\_Data\_to\_json(girl\_life\_picturePath,girl\_life\_picture\_url\_list,**"w"**)  
 *# 加载美女生活图片链接数据* girlLife\_linkPath = **"./file/girlLif\_picture.json"** girlLife\_link\_list = downloadGirl\_picture.load\_Json\_to\_data(girlLife\_linkPath)  
 *# 下载美女生活照* download\_life\_picture(girlLife\_link\_list)  
 print(**"程序已经走完"**)  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

数据保存截图：



b. 爬取美女生活照图片，代码同上

效果如下：



其中一位展示：



1. 、最后封装主函数

代码如下：

**import** requests  
**from** BeautifulGirl\_Project **import** spiderHtmlData  
**from** BeautifulGirl\_Project **import** spiderParseHtml  
**from** BeautifulGirl\_Project **import** downloadGirl\_picture  
**from** BeautifulGirl\_Project **import** spiderGirl\_introduce\_infor  
**from** BeautifulGirl\_Project **import** spiderGirl\_other\_picture  
  
  
**"""主函数集群"""  
def** main():  
 spiderHtmlData.main()  
 spiderParseHtml.main()  
 downloadGirl\_picture.main()  
 spiderGirl\_introduce\_infor.main()  
 spiderGirl\_other\_picture.main()  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

三、实验心得

学会了如何爬取动态和静态网页，如何利用自动化工具来操纵浏览器，这是最深有体会的一点。其次就是如何整理项目的结构化，这次的项目让我体会到：做项目，如果是想一点就写一点，那真的很浪费时间，然后也会出现很多的错误，正确的做法应该是：先做好需求分析，设计好整体的架构，然后在把大概思路走一遍，最后才是写代码

如果是这样的话，那效果就快很多，培训老师说过一句话，写代码是最容易的，思路才是最难得的。

最后一点心得就是：做一个爬虫项目，必须时常把数据保存好，如果不保存，下次就得重新爬，那很浪费时间，还有在写一个功能代码时就得养成测试的习惯，不然到最后就会很乱，无法排错。