* **scrapy创建项目：**

Pycharm中用alt+F12切换到命令行，在命令行输入：

(venv2) E:\>**scrapy startproject peilv**

**就会生成Scrapy项目，项目名称是peilv，结构如下：主要改写2个文件：“items、settings”，新增2个文件：“爬虫主程序”、itemcsvexporter。**

|  |
| --- |
| **peilv**  scrapy.cfg #创建项目时自动生成，项目的配置文件  **peilv**/  \_\_init\_\_.py #创建项目时自动生成，无需任何改动  items.py #创建项目时自动生成，**定义爬取的字段**  pipelines.py #创建项目时自动生成，如存入文件，无需任何改动  settings.py #创建项目时自动生成，**将爬取字段按顺序输出**  middlewares.py #创建项目时自动生成，无需任何改动  **spiders**/  \_\_init\_\_.py #创建项目时自动生成，无需任何改动  itemcsvexporter.py #**需自己编写，代码固定**  爬虫主程序.py #**需自己编写，爬虫的主程序** |

* items.py：

|  |
| --- |
| *# -\*- coding: utf-8 -\*-*  **import** scrapy  **class** PeilvItem(scrapy.Item):  *# define the fields for your item here like:* cc = scrapy.Field()*#changci* li = scrapy.Field()*#libo* b5 = scrapy.Field()*#bet365* |

* settings.py：

|  |
| --- |
| *# -\*- coding: utf-8 -\*-*  BOT\_NAME = **'peilv'**  SPIDER\_MODULES = [**'peilv.spiders'**] NEWSPIDER\_MODULE = **'peilv.spiders'** FEED\_EXPORT\_ENCODING = **"gb18030"** *#解决导出的Excel文件中文乱码问题*  user\_agent = **"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/22.0.1207.1 Safari/537.1"**  FEED\_URI = **'file:///e:/PythonLearn/Python learning/peilv/BaseData.csv'** FEED\_FORMAT = **'csv'** FEED\_EXPORTERS = {  **'csv'**: **'peilv.spiders.itemcsvexporter.itemcsvexporter'**, } *# 这里你的project名字为peilv*  FIELDS\_TO\_EXPORT = [  **'cc'**,  **'li'**,  **'b5'**,  ]  ROBOTSTXT\_OBEY = **False** *#当用cookies时候要设置为false*  DOWNLOADER\_MIDDLEWARES = { **'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware'**: **None**, } HTTPERROR\_ALLOWED\_CODES = [403] |

* itemcsvexporter.py

|  |
| --- |
| **from** scrapy.conf **import** settings *# from scrapy.contrib.exporter import CsvItemExporter* **from** scrapy.exporters **import** CsvItemExporter *#指定输出到csv文件中字段的顺序，结合setting.py* **class** itemcsvexporter(CsvItemExporter):   **def** \_\_init\_\_(self, \*args, \*\*kwargs):  delimiter = settings.get(**'CSV\_DELIMITER'**, **','**)  kwargs[**'delimiter'**] = delimiter   fields\_to\_export = settings.get(**'FIELDS\_TO\_EXPORT'**, [])  **if** fields\_to\_export:  kwargs[**'fields\_to\_export'**] = fields\_to\_export   super(itemcsvexporter, self).\_\_init\_\_(\*args, \*\*kwargs) |

* 爬虫主程序.py：

|  |
| --- |
| *# -\*- coding: utf-8 -\*- #项目源码地址：github.com/acredjb/FBP #作者：acredjb # scrapy crawl FBP -o BaseData.csv* **import** datetime **import** sys **import** requests **import** scrapy **import** time **import** json **import** scrapy.http **from** peilv.items **import** PeilvItem **from** lxml **import** etree *#获取当天或未来某天数据的地址* wl\_url = **'https://live.leisu.com/saicheng?date='***#wl历史https://live.leisu.com/saicheng?date=20190620 #获取历史数据的地址* ls\_url = **'https://live.leisu.com/wanchang?date='***#ls历史https://live.leisu.com/wanchang?date=20190606* **class** LiveJiangSpider(scrapy.Spider):  name = **'FBP'** allowed\_domains = [**'leisu.com'**]  **def** start\_requests(self):  d1=**'20190606'** *#历史的比赛  # d1='20190914' #未来的比赛* request = scrapy.http.FormRequest(ls\_url + d1,callback=self.parseLs, meta={**'d1'**: d1}) *#历史的比赛  # request = scrapy.http.FormRequest(wl\_url + d1,callback=self.parseWl, meta={'d1': d1})#未来的比赛* **yield** request  **def** parseLs(self,response):  d2=response.meta[**'d1'**]  sel=response.xpath  racelist=[e5.split(**"'"**) **for** e5 **in** sel(**'//li[@data-status="8"]/@data-id'**).extract()]  **for** raceid **in** racelist:*#raceid=['2674547'];raceid[0]=2674547* item = PeilvItem()  sel\_div=sel(**'//li[@data-id='**+str(raceid[0])+**']/div[@class="find-table layout-grid-tbody hide"]/div[@class="clearfix-row"]'**)  **if** str(sel\_div.xpath(**'span[@class="lab-lottery"]/span[@class="text-jc"]/text()'**).extract()) == **"[]"**:  item[**'cc'**]=**""  else**:  item[**'cc'**]=str(d2) + str(sel\_div.xpath(**'span[@class="lab-lottery"]/span[@class="text-jc"]/text()'**).extract()[0])  **if "周" in** item[**'cc'**]:*#取竞彩-周一001等* plurl=**'https://live.leisu.com/3in1-'**+raceid[0]  request = scrapy.http.FormRequest(plurl,callback=self.parse,meta={**'item'**:item})  **yield** request *#并非return，yield压队列，parse函数将会被当做一个生成器使用。scrapy会逐一获取parse方法中生成的结果，并没有直接执行parse，循环完成后，再执行parse* **def** parseWl(self,response):  d2=response.meta[**'d1'**] *#接收参数* sel=response.xpath  racelist=[e5.split(**"'"**) **for** e5 **in** sel(**'//li[@data-status="1"]/@data-id'**).extract()]  **for** raceid **in** racelist:*#raceid=['2674547'];raceid[0]=2674547* item = PeilvItem()  sel\_div=sel(**'//\*[@data-id='** + str(raceid[0]) + **']/div[@class="find-table layout-grid-tbody hide"]/div[@class="clearfix-row"]'**)  **if** str(sel\_div.xpath(**'span[@class="lab-lottery"]/span[@class="text-jc"]/text()'**).extract()) == **"[]"**:  changci=**""  else**:  changci =str(sel\_div.xpath(**'span[@class="lab-lottery"]/span[@class="text-jc"]/text()'**).extract()[0])   **if "周" in** changci:*#取竞彩-周一001等* item[**'cc'**]=str(d2) + changci  plurl=**'https://live.leisu.com/3in1-'**+raceid[0]  request = scrapy.http.FormRequest(plurl, callback=self.parse,meta={**'item'**:item})  **yield** request *#并非return，yield压队列，parse函数将会被当做一个生成器使用。scrapy会逐一获取parse方法中生成的结果，并没有直接执行parse，循环完成后，再执行parse* **def** parse(self, response):  print(**'--------------into parse----------------------'**)  item = response.meta[**'item'**]  pv=response.xpath  pl\_str = **'/td[@class="bd-left"]/div[@class="begin float-left w-bar-100 bd-bottom p-b-8 color-999 m-b-8"]/span[@class="float-left col-3"]/text()'   if** str(pv(**'//\*[@data-id="5"]'**+pl\_str).extract())==**"[]"**:  item[**'li'**] = **''  else**:  item[**'li'**]=pv(**'//\*[@data-id="5"]'** + pl\_str).extract()[0]   **if** str(pv(**'//\*[@data-id="2"]'**+pl\_str).extract())==**"[]"**:  item[**'b5'**] = **''  else**:  item[**'b5'**]=pv(**'//\*[@data-id="2"]'** + pl\_str).extract()[0]   **yield** item*#程序在取得各个页面的items前，会先处理完之前所有的request队列里的请求，然后再提取items* |

* **改写完程序后，最终执行命令：**

|  |
| --- |
| **Pycharm用alt+F12切换到命令行在项目peilv路径上执行：**  **(venv2) E:\>cd peilv\peilv**  **(venv2) E:\peilv\peilv>scrapy crawl FBP -o BaseData.csv**  **其中FBP是在“爬虫主程序.py”定义的——name = 'FBP'，**  **“-o BaseData.csv”是将爬取的数据输出到csv文件中。** |