**Display.py**

# \_\*\_ coding:utf-8 \_\*\_  
import db  
from flask import Flask, render\_template, request, jsonify, redirect, flash, url\_for  
  
app = Flask(\_\_name\_\_)  
  
  
@app.route('/') # 根路径  
def index():  
 items = db.getItem()  
 words = db.getWord()  
 count = len(items)  
 url = None  
 with open('E:\Python Code\Code\Crawler-0.1\urls', 'r') as f:  
 url = f.readline()  
  
 return render\_template(  
 'index.html',  
 items=items,  
 words=words,  
 url = url,  
 count = count,  
 data=str(db.getCount\_byMonth())  
 )  
  
  
@app.route('/search', methods=['POST'])  
def Search():  
 content = request.form['search\_content']  
 items = db.searchTitle(content)  
 words = db.getWord()  
 print '----------',type(items)  
 count = len(items)  
 url = None  
 with open('E:\Python Code\Code\Crawler-0.1\urls', 'r') as f:  
 url = f.readline()  
  
 return render\_template('index.html',  
 items=items,  
 words=words,  
 url = url,  
 count = count,  
 data=str(db.getCount\_byMonth\_byContent(content)))  
  
  
@app.route('/search/<word>', methods=['POST', 'GET'])  
def Search\_Word(word):  
 items = db.searchTitle(word)  
 words = db.getWord()  
 count = len(items)  
  
 url = None  
 with open('E:\Python Code\Code\Crawler-0.1\urls', 'r') as f:  
 url = f.readline()  
  
 return render\_template('index.html',  
 items=items,  
 words=words,  
 url = url,  
 count = count,  
 data=str(db.getCount\_byMonth\_byContent(word)))  
  
  
@app.route('/content.html/<int:id>', methods=['POST', 'GET'])  
def Content(id):  
 item = db.getUser\_byId(id)  
 return render\_template('content.html', item=item)  
  
  
@app.route('/add\_word', methods=['POST', 'GET'])  
def Add\_Word():  
 word = request.form.get('word')  
 db.addWord(word)  
 return jsonify(word=word)  
  
  
@app.route('/delete\_word', methods=['POST', 'GET'])  
def Delete\_Word():  
 word = request.form.get('word')  
 db.deleteWord(word)  
 return jsonify(word=word)  
  
  
@app.route('/set\_url', methods=['POST'])  
def set\_url():  
 items = db.getItem()  
 count = len(items)  
 data = request.form  
 url = data.get('url', None)  
 # if url is None:  
 # flash('URL为空', 'danger')  
  
  
 # 写入文件  
 with open('E:\Python Code\Code\Crawler\urls', 'w') as f:  
 f.write(url)  
  
 items = db.getItem()  
 words = db.getWord()  
 return render\_template(  
 'index.html',  
 items=items,  
 words=words,  
 url = url,  
 count = count,  
 data=str(db.getCount\_byMonth())  
 )  
  
@app.route('/empty\_sql', methods=['POST'])  
def empty\_sql():  
 items = db.getItem()  
 words = db.getWord()  
 count = len(items)  
 url = None  
 with open('E:\Python Code\Code\Crawler-0.1\urls', 'r') as f:  
 url = f.readline()  
  
 db.emptySql();  
  
 return render\_template(  
 'index.html',  
 items=items,  
 words=words,  
 url = url,  
 count = count,  
 data=str(db.getCount\_byMonth())  
 )  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

**Db.py**

# -\*- coding:utf-8 -\*-  
import sqlalchemy  
  
from sqlalchemy import create\_engine  
from sqlalchemy.ext.declarative import declarative\_base  
from sqlalchemy import Column, Integer, String, DateTime  
from sqlalchemy.orm import sessionmaker  
from sqlalchemy.sql import text  
from sqlalchemy import desc  
  
# 创建对象的基类  
Base = declarative\_base()  
# 初始化数据库连接 防止中文乱码  
engine = create\_engine('mysql://root:1234@localhost/spider?charset=utf8')  
  
  
class Post(Base):  
 \_\_tablename\_\_ = 'jihemei' # 寻找table  
  
 id = Column(Integer, primary\_key=True)  
 postMan = Column(String)  
 firstTime = Column(DateTime)  
 title = Column(String)  
 content = Column(String)  
 readCount = Column(Integer)  
 replyCount = Column(Integer)  
 link = Column(String)  
 '''  
 def \_\_init\_\_(self, postMan, firstTime, title, content, readCount, replyCount, link):  
 self.postMan = postMan  
 self.firstTime = firstTime  
 self.title = title  
 self.content = content  
 self.readCount = readCount  
 self.replyCount = replyCount  
 self.link = link  
 '''  
  
  
# 获取月份，转化为int返回。  
def getMonth(datetime):  
 return int(str(datetime)[5:7])  
  
  
# 获取所有的item  
def getItem():  
 # 创建session对象  
 DBSession = sessionmaker(bind=engine)  
 session = DBSession()  
  
 sql = text("select \* from jihemei ORDER BY firstTime desc")  
  
 # 获取所有的item 这里使用字典的方式绑定参数  
 items = session.execute(sql).fetchall()  
  
 session.close()  
  
 return items  
  
# 按照月份获取  
def getCount\_byMonth():  
 DBSession = sessionmaker(bind=engine)  
 session = DBSession()  
  
 items = session.query(Post).all()  
  
 # 字典-按照key-value存储  
 data = {}  
 for item in items:  
 month = getMonth(item.firstTime)  
 if data.get(month):  
 data[month] = data[month] + 1  
 else:  
 data[month] = 1  
  
 return data  
  
  
# 按照月份获取——搜索版  
def getCount\_byMonth\_byContent(content):  
 sql = "select \* from jihemei where title like :Content"  
  
 items = engine.execute(text(sql), Content='%' + content + '%')  
  
 data = {} # 字典  
 for item in items:  
 month = getMonth(item.firstTime)  
 if data.get(month):  
 data[month] = data[month] + 1  
 else:  
 data[month] = 1  
  
 return data  
  
  
# 按照用户获取  
def getCount\_byUser():  
 DBSession = sessionmaker(bind=engine)  
 session = DBSession()  
  
 items = session.query(Post).all()  
  
 session.close()  
  
  
# 按照标题搜索  
def searchTitle(content):  
 # 创建session对象  
 DBSession = sessionmaker(bind=engine)  
 session = DBSession()  
  
 sql = text("select \* from jihemei where title like :Title")  
  
 # 获取所有的item 这里使用字典的方式绑定参数  
 items = session.execute(sql, {'Title': '%' + content + '%'}).fetchall()  
  
 session.close()  
  
 return items  
  
  
# 按照内容搜索  
def searchContent(content):  
 sql = "select \* from jihemei where content like :Content"  
  
 items = engine.execute(text(sql), Content='%' + content + '%')  
  
 return items  
  
  
# 按照id搜索  
def getUser\_byId(id):  
 sql = "select \* from jihemei where id = :Id"  
 sql\_enter = "UPDATE jihemei SET content = REPLACE(content, '\n', '<br\\>');"  
  
 item = engine.execute(text(sql), Id=int(id)).fetchone() # fetchone() 去一个数据。  
 # item.content = item.content.replace("\n","<br/>")  
 # engine.execute(text(sql))  
  
 return item  
  
  
# 按照热词搜索  
def getWord():  
 sql = "select \* from Hot\_Words"  
  
 words = engine.execute(text(sql))  
  
 return words  
  
  
# 添加热词  
def addWord(word):  
 sql = "insert into Hot\_Words(word) values(:Word)"  
 items = engine.execute(text(sql), Word=word)  
  
  
# 删除热词  
def deleteWord(word):  
 sql = "delete from Hot\_Words WHERE word = :Word"  
 engine.execute(text(sql), Word=word)  
  
  
# 清空数据库  
def emptySql():  
 sql = "truncate table jihemei";  
 engine.execute(text(sql));  
  
getItem()

**index.html**

<!DOCTYPE html>  
<html lang="en">  
  
<head>  
  
 <meta charset="utf-8">  
 <meta http-equiv="X-UA-Compatible" content="IE=edge">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <meta name="description" content="">  
 <meta name="author" content="">  
  
 <title>网上舆情爬取系统</title>  
  
 <!-- Bootstrap Core CSS -->  
 <link href="/static/bower\_components/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">  
  
 <!-- MetisMenu CSS -->  
 <link href="/static/bower\_components/metisMenu/dist/metisMenu.min.css" rel="stylesheet">  
  
 <!-- Timeline CSS -->  
 <link href="/static/css/timeline.css" rel="stylesheet">  
  
 <!-- Custom CSS -->  
 <link href="/static/css/sb-admin-2.css" rel="stylesheet">  
  
 <!-- Morris Charts CSS -->  
 <link href="/static/bower\_components/morrisjs/morris.css" rel="stylesheet">  
  
 <!-- Custom Fonts -->  
 <link href="/static/bower\_components/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">  
  
 <!---->  
 <link href="/static/mypackage/css/jquery.bdt.css" type="text/css" rel="stylesheet">  
 <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->  
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->  
 <!--[if lt IE 9]>  
 <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>  
 <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>  
 <![endif]-->  
  
</head>  
  
<body>  
<a name="top" id="top"></a>  
<div id="wrapper">  
  
 <!-- Navigation -->  
 <nav class="navbar navbar-default navbar-static-top" role="navigation" >  
 <div class="navbar-header">  
 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">  
 <span class="sr-only">Toggle navigation</span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 </button>  
 <a class="navbar-brand" href="index.html">李林</a>  
  
 <form action="{{ url\_for('set\_url') }}" method="post" class="navbar-form navbar-right setUrl row" role="search">  
 <div class="form-group col-md-10 padding-right-0">  
 <input id="urlValue" name="url" type="text" class="form-control width-full" placeholder="http://www.cxjhm.com/forum-46-1.html" value="{{ url }}">  
 </div>  
 <div class="text-left col-md-2">  
 <button id="setUrl" type="submit" class="btn btn-default">设置</button>  
 </div>  
  
 </form>  
 </div>  
 <!-- /.navbar-header -->  
  
  
 <!-- /.navbar-top-links -->  
 <div class="navbar-default sidebar" role="navigation">  
 <div class="sidebar-nav navbar-collapse" id="mymenu">  
 <ul class="nav" id="side-menu">  
  
 <li>  
 <a href="#top" target="\_self"><i class="fa fa-bar-chart-o fa-fw"></i> Data </a>  
 </li>  
 <li>  
 <a href="#buttom" target="\_self"><i class="fa fa-dashboard fa-fw"></i> Charts </a>  
 </li>  
  
 <li>  
 <a href="#"><i class="fa fa-sitemap fa-fw"></i> Hot Words <span class="fa arrow"></span></a>  
 <ul class="nav nav-second-level" id="hot\_words">  
 {% for word in words %}  
 <li>  
 <a href="http://127.0.0.1:5000/search/{{ word.word }}"  
 target="\_self">{{ word.word }}</a>  
 </li>  
 {% endfor %}  
 <li>  
 <a href="javascript:;" onclick="add\_word()" id="add\_word"><strong>添加敏感词</strong></a>  
 </li>  
 <li>  
 <a href="javascript:;" onclick="delete\_word()" id="delete\_word"  
 style="font-weight:bold; color:red">删除敏感词</a>  
 </li>  
 </ul>  
 <!-- /.nav-second-level -->  
 </li>  
 <!-- /.dropdown -->  
  
 </ul>  
 </div>  
 <!-- /.sidebar-collapse -->  
 </div>  
 <!-- /.navbar-static-side -->  
 </nav>  
  
 <div id="page-wrapper">  
 <div class="row">  
 <div class='col-lg-9'>  
 <h1 class="page-header" style="font-family:Microsoft Yahei">  
 网上舆情爬取系统  
 </h1>  
 </div>  
 <form action="/search" method="post">  
 <div class="input-group page-header col-lg-3">  
 <input name="search\_content" type="text" class="form-control" placeholder="Search...">  
 <span class="input-group-btn">  
 <button class="btn btn-default" type="submit">  
 <i class="glyphicon glyphicon-search"></i>  
 </button>  
 </span>  
 </div>  
 </form>  
 </div>  
  
 <div class="row">  
 <div class="col-lg-9">  
  
 <div class="panel panel-default">  
 <div class="panel-heading">  
 <i class="fa fa-bar-chart-o fa-fw"></i>Data Show  
 </div>  
 <!-- /.panel-heading -->  
 <div class="panel-body">  
 <table border="0" class="table table-bordered" >  
 <thead>  
 <tr>  
 <th>id</th>  
 <th>postMan</th>  
 <th>title</th>  
 <th>link</th>  
 </tr>  
 </thead>  
 <!-- 只要渲染一下部分 -->  
 <tbody id="myTable">  
 {% for item in items %}  
 <tr>  
 <td id="getId">{{ loop.index }}</td>  
 <td>{{ item.postMan }}</td>  
 <td>{{ item.title }}</td>  
 <td><a href="http://127.0.0.1:5000/content.html/{{ item.id }}"  
 target="\_blank">文章链接</a></td>  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
 <P>&nbsp;&nbsp;总共爬取了{{ count }}个帖子</P>  
 <div class="col-md-12 text-right">  
 <ul class="pagination" id="myPager"></ul>  
 </div>  
 </div>  
 <!-- /.panel-body -->  
 </div>  
  
 <div class="panel panel-default">  
 <div class="panel-heading">  
 <i class="fa fa-bar-chart-o fa-fw"></i>Chart by month  
 </div>  
 <!-- /.panel-heading -->  
 <div class="panel-body">  
 <div id="morris-area-chart"></div>  
 </div>  
 <!-- /.panel-body -->  
 </div>  
 <!-- /.panel -->  
  
  
 </div>  
  
 </div>  
 <!-- /.row -->  
 </div>  
 <!-- /#page-wrapper -->  
  
</div>  
<a name="buttom" id="buttom"></a>  
<!-- /#wrapper -->  
<!--Top-->  
<script src="http://libs.baidu.com/jquery/1.10.2/jquery.min.js"></script>  
  
<!-- jQuery -->  
<script src="/static/bower\_components/jquery/dist/jquery.min.js"></script>  
  
<script src="http://code.jquery.com/jquery-2.1.1.min.js" type="text/javascript"></script>  
<script src="/static/mypackage/js/vendor/jquery.sortelements.js" type="text/javascript"></script>  
<script src="/static/mypackage/js/jquery.bdt.js" type="text/javascript"></script>  
  
<!-- Bootstrap Core JavaScript -->  
<script src="/static/bower\_components/bootstrap/dist/js/bootstrap.min.js"></script>  
  
<!-- Metis Menu Plugin JavaScript -->  
<script src="/static/bower\_components/metisMenu/dist/metisMenu.min.js"></script>  
  
<!-- Morris Charts JavaScript -->  
<script src="/static/bower\_components/raphael/raphael-min.js"></script>  
<script src="/static/bower\_components/morrisjs/morris.min.js"></script>  
  
<!--Powered by me-->  
<script type="text/javascript">  
 function getWord(obj) {  
 return "http://127.0.0.1:5000/search/" + (obj.innerText);  
 }  
</script>  
  
<!--Ajax动态刷新-->  
<!--添加热词-->  
<script>  
 function add\_word() {  
 var word = prompt("添加你想要的敏感词~");  
 console.log(word);  
 $.ajax({  
 method: "POST",  
 url: "/add\_word",  
 data: {word: word},  
 success: function () {  
 /\*  
 var li = document.createElement('li');  
 var a = document.createElement('a');  
 a.textContent = word;  
 li.appendChild(a); //构造li标签  
 var menu = document.querySelector("#hot\_words") //选择ul  
 menu.insertBefore(li, menu.firstChild); //添加  
 \*/  
 location.reload();  
 }  
 }  
 );  
 }  
</script>  
<!--删除热词-->  
<script>  
 function delete\_word() {  
 var word = prompt("请输入你想要删除的敏感词~");  
 var choice = window.confirm("你确定要删除么~")  
 if (choice == 1) {  
 $.ajax({  
 method: "POST",  
 url: "/delete\_word",  
 data: {word: word},  
 success: function () {  
 location.reload();  
 }  
 }  
 );  
 }  
 }  
</script>  
  
  
<script>  
 var data = {{ data }};  
 var chartData = [];  
 //加强版的for循环  
 for (var key in data) {  
 if(key<=5){  
 chartData.push({  
 month: '2016-'+key,  
 count: data[key]  
 })  
 }else{  
 chartData.push({  
 month: '2015-'+key,  
 count: data[key]  
 })  
 }  
  
 console.log(key)  
 }  
 Morris.Area({  
 element: 'morris-area-chart',  
 data: chartData,  
 xkey: 'month',  
 ykeys: ['count'],  
 labels: ['Count'],  
 pointSize: 2,  
 hideHover: 'auto',  
 resize: true  
 });  
  
</script>  
<!-- Custom Theme JavaScript -->  
<script src="/static/js/sb-admin-2.js"></script>  
  
<!-- 分页 -->  
<script type="text/javascript" src="/static/bower\_components/jquery/dist/jquery.goup.min.js"></script>  
<script>  
 $(document).ready(function(){  
 $('#myTable').pageMe({pagerSelector:'#myPager',showPrevNext:true,hidePageNumbers:false,perPage:10});  
 });  
</script>  
  
<!-- 返回顶部 -->  
<script type="text/javascript" src="/static/mypackage/js/pagination.js"></script>  
<script type="text/javascript">  
 $(document).ready(function () {  
 $.goup({  
 trigger: 100,  
 bottomOffset: 100,  
 locationOffset: 100,  
 title: 'Top',  
 titleAsText: true  
 });  
 });  
</script>  
  
<!-- 清空数据库 -->  
<!--  
<script type="text/javascript">  
 function emptySql(){  
 var choice = window.confirm("你确定要清空数据库么~")  
 if (choice == 1) {  
 $.ajax({  
 method: "POST",  
 url: "/empty\_sql",  
 success: function () {  
 parent.location.reload();  
 }  
 }  
 );  
 }  
 }  
</script>  
-->  
  
</body>  
  
  
</html>

**content.html**

<!DOCTYPE html>  
<html lang="en">  
  
<head>  
  
 <meta charset="utf-8">  
 <meta http-equiv="X-UA-Compatible" content="IE=edge">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <meta name="description" content="">  
 <meta name="author" content="">  
  
 <title>{{ item.title }}</title>  
  
 <!-- Bootstrap Core CSS -->  
 <link href="/static/bower\_components/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">  
  
 <!-- MetisMenu CSS -->  
 <link href="/static/bower\_components/metisMenu/dist/metisMenu.min.css" rel="stylesheet">  
  
 <!-- Timeline CSS -->  
 <link href="/static/css/timeline.css" rel="stylesheet">  
  
 <!-- Custom CSS -->  
 <link href="/static/css/sb-admin-2.css" rel="stylesheet">  
  
 <!-- Morris Charts CSS -->  
 <link href="/static/bower\_components/morrisjs/morris.css" rel="stylesheet">  
  
 <!-- Custom Fonts -->  
 <link href="/static/bower\_components/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">  
  
 <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->  
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->  
 <!--[if lt IE 9]>  
 <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>  
 <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>  
 <![endif]-->  
  
</head>  
  
<body>  
 <a name="top" id="top"></a>  
 <div id="wrapper">  
  
 <!-- Navigation -->  
 <nav class="navbar navbar-default navbar-static-top" role="navigation" style="margin-bottom: 0">  
 <div class="navbar-header">  
 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">  
 <span class="sr-only">Toggle navigation</span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 </button>  
 <a class="navbar-brand" href="index.html">李林</a>  
 </div>  
 <!-- /.navbar-header -->  
 </nav>  
  
 <div id="page-wrapper">  
 <div class="row">  
 <div class='col-lg-9'>  
 <h1 class="page-header" style="font-family:Times New Roman">{{ item.title }}</h1>  
 <div style="float:right;">  
 <p style="font-family:'Microsoft JhengHei'; font-size:20px">Author:{{ item.postMan }}</p>  
 </div>  
 <br><br>  
 <div style="float:right;">  
 <p style="font-family:'Microsoft JhengHei'; font-size:20px">Time:{{ item.firstTime }}</p>  
 </div>  
 </div>  
 <form action="/search" method="post">  
 <div class="input-group page-header col-lg-3">  
 <input name = "search\_content" type="text" class="form-control" placeholder="Search...">  
 <span class="input-group-btn">  
 <button class="btn btn-default" type="submit" >  
 <i class="glyphicon glyphicon-search"></i>  
 </button>  
 </span>  
 </div>  
 </form>  
 </div>  
  
 <div class="row">  
 <div class="col-lg-9">  
  
 <div class="panel panel-default">  
 <div class="panel-heading">  
 <i class="fa fa-bar-chart-o fa-fw"></i>Content Show  
 </div>  
 <!-- /.panel-heading -->  
 <div class="panel-body">  
 {% if item.content %}  
 <pre style="white-space:pre-wrap;">{{ item.content }}</pre>  
 {% else %}  
 <pre style="font-weight:bold; font-size:40px; white-space:pre-wrap; color:red;">作者被禁止或删除 内容自动屏蔽</pre>  
 {% endif %}  
 </div>  
 <!-- /.panel-body -->  
 </div>  
  
 </div>  
  
 </div>  
 <!-- /.row -->  
 </div>  
 <!-- /#page-wrapper -->  
  
 </div>  
 <a name="buttom" id="buttom"></a>  
 <!-- /#wrapper -->  
  
 <!-- jQuery -->  
 <script src="/static/bower\_components/jquery/dist/jquery.min.js"></script>  
 <script src="http://libs.baidu.com/jquery/1.10.2/jquery.min.js"></script>  
 <script type="text/javascript" src="/static/bower\_components/jquery/dist/jquery.goup.min.js"></script>  
 <script type="text/javascript">  
 $(document).ready(function () {  
 $.goup({  
 trigger: 100,  
 bottomOffset: 100,  
 locationOffset: 100,  
 title: 'Top',  
 titleAsText: true  
 });  
  
 });  
 </script>  
  
 <!-- Bootstrap Core JavaScript -->  
 <script src="/static/bower\_components/bootstrap/dist/js/bootstrap.min.js"></script>  
  
 <!-- Metis Menu Plugin JavaScript -->  
 <script src="/static/bower\_components/metisMenu/dist/metisMenu.min.js"></script>  
  
 <!-- Morris Charts JavaScript -->  
 <script src="/static/bower\_components/raphael/raphael-min.js"></script>  
 <script src="/static/bower\_components/morrisjs/morris.min.js"></script>  
  
 <script>  
 var data = {{ data }};  
 var chartData = [];  
 var months = [];  
 //加强版的for循环  
 for (var key in data){  
 chartData.push({  
 month:key,  
 count:data[key]  
 })  
 months.push(key)  
 }  
 //console.log(chartData)  
 console.log(months)  
 Morris.Area({  
 element: 'morris-area-chart',  
 data: chartData,  
 xkey: 'month',  
 ykeys: ['count'],  
 labels: ['Count'],  
 pointSize: 2,  
 hideHover: 'auto',  
 resize: true  
 });  
  
 </script>  
  
 <!-- Custom Theme JavaScript -->  
 <script src="/static/js/sb-admin-2.js"></script>  
  
</body>  
  
  
</html>

**items.py**

# -\*- coding: utf-8 -\*-  
  
# Define here the models for your scraped items  
#  
# See documentation in:  
# http://doc.scrapy.org/en/latest/topics/items.html  
  
import scrapy  
  
from scrapy.item import Item, Field  
  
  
class CrawlerItem(scrapy.Item):  
 # define the fields for your item here like:  
 # name = scrapy.Field()  
 pass  
  
class PostItem(Item):  
 postMan = Field()  
 firstTime = Field()  
 title = Field()  
 content = Field()  
 readCount = Field()  
 replyCount = Field()  
 link = Field()

**jihemei\_spider.py**

# -\*- coding:utf-8 -\*-  
  
from scrapy.spider import Spider  
from scrapy.selector import Selector  
from Crawler.items import PostItem  
from bs4 import BeautifulSoup  
  
import scrapy  
import urllib2  
import random  
  
# 预备用头部  
my\_headers = [  
 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.87 Safari/537.36',  
 'Mozilla/5.0 (Windows; U; Windows NT 6.1; en-us) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50',  
 'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0',  
 'Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1'  
]  
  
  
# 模拟头部  
def get\_content(url, headers):  
 '''  
 @获取403禁止访问网页  
 '''  
 random\_header = random.choice(headers) # 从传入参数headers随机挑选一个  
  
 # print random\_header  
  
 req = urllib2.Request(url)  
 req.add\_header('User-Agent', random\_header)  
 req.add\_header('Host', 'www.cxjhm.com')  
 req.add\_header('GET', url)  
 req.add\_header('Referer', 'http://www.cxjhm.com/forum.php')  
  
 content = urllib2.urlopen(req).read()  
 return content  
  
# 去掉HTML代码  
def delete\_html(str):  
 # 先去除第一个br前面的HTML代码  
 index1 = str.index('<br>')  
 index2 = len(str)  
 str = str[index1+4:index2]  
 # 删除所有的br和font  
 str = str.replace('<br>', '')  
 str = str.replace('</font>', '')  
  
 return str  
  
class JihemeiSpider(Spider):  
 name = "jihemei"  
 allowed\_domains = ["cxjhm.com"] # 该域名范围内爬去  
 start\_urls = [  
 # "http://www.cxjhm.com/forum-46-1.html" # 起始地址  
 ]  
  
 def \_\_init\_\_(self):  
 # self.start\_urls.append(url)  
 with open("E:\Python Code\Code\Crawler-0.1\urls", "r") as f:  
 self.start\_urls.append(f.readline())  
  
 # 入口方法  
 def parse(self, response):  
 # 爬取底部网址  
 for href in response.css("#fd\_page\_bottom a::attr('href')"): # 抓取button所有的链接  
 url = response.urljoin(href.extract()) # 加入队列  
 yield scrapy.Request(url, callback=self.parse) # 再次调用该函数  
  
 # 开始解析爬取的网页  
 sel = Selector(response)  
 sites = sel.css('.bm\_c tr')  
  
 # 选取文章链接，调用parse\_content\_page方法  
 for site in sites:  
 link = site.css('.s.xst::attr(href)').extract\_first() # 选择href  
 if link:  
 link = "http://www.cxjhm.com/" + link  
 yield scrapy.Request(link, callback=self.parse\_content\_page)  
  
 # 爬取指定网页  
 def parse\_content\_page(self, response):  
 item = PostItem()  
  
 item['title'] = response.css('#thread\_subject::text').extract\_first()  
 item['postMan'] = response.css('.authi .xw1::text').extract\_first()  
  
 time = response.css('.plhin .authi em span::attr(title)').extract\_first()  
 if not time:  
 time = response.css('.plhin .authi em::text').extract\_first().split(' ')[1] # 按空格分隔，取第二个。  
 else:  
 time = time.split(' ')[0]  
 item['firstTime'] = time  
  
 readCount, replyCount = [int(count) for count in response.css('.xi1::text').extract()]  
 item['readCount'] = readCount  
 item['replyCount'] = replyCount  
  
 # 先爬取正文模块，去除回帖模块，获取正文，用空字符串连接所有的元组。  
 content = ''.join(response.css('.plhin')[0].css('.t\_f').extract())  
 link = response.url # response的属性，dir(response)  
 if 'font' in content:  
 html = get\_content(link, my\_headers)  
 soup = BeautifulSoup(html)  
 item['content'] = soup.find(attrs={'class': 't\_f'}).get\_text()  
 else:  
 item['content'] = ''.join(response.css('.plhin')[0].css('.t\_f::text').extract())  
 item['link'] = link  
  
 yield item

**pipelines.py**

# -\*- coding: utf-8 -\*-  
  
# Define your item pipelines here  
#  
# Don't forget to add your pipeline to the ITEM\_PIPELINES setting  
# See: http://doc.scrapy.org/en/latest/topics/item-pipeline.html  
  
import sqlalchemy  
  
from sqlalchemy import create\_engine  
from sqlalchemy.ext.declarative import declarative\_base  
from sqlalchemy import Column, Integer, String, DateTime  
from sqlalchemy.orm import sessionmaker  
from sqlalchemy.sql import text  
from sqlalchemy.exc import IntegrityError  
  
Base = declarative\_base()  
# 初始化数据库连接 防止中文乱码  
engine = create\_engine('mysql://root:1234@localhost/spider?charset=utf8')  
  
  
class Post(Base):  
 \_\_tablename\_\_ = 'jihemei' # 寻找table  
  
 id = Column(Integer, primary\_key=True)  
 postMan = Column(String)  
 firstTime = Column(DateTime)  
 title = Column(String)  
 content = Column(String)  
 readCount = Column(Integer)  
 replyCount = Column(Integer)  
 link = Column(String)  
  
  
class CrawlerPipeline(object):  
 def process\_item(self, item, spider):  
 # 创建Session类型  
 DBSession = sessionmaker(bind=engine)  
 # 创建session对象  
 session = DBSession()  
  
 # 从item中获取帖子的属性，创建Post对象。  
 admin = Post(postMan=item['postMan'],  
 firstTime=item['firstTime'],  
 title=item['title'],  
 content=item['content'],  
 readCount=item['readCount'],  
 replyCount=item['replyCount'],  
 link=item['link'])  
  
 # 去掉重复  
 flag = 0 # 标记flag，初始值为0，找到相同的帖子置1。  
 items = session.query(Post).all() # query方法需要加Post  
 for item in items: # 在已经存在的数据库里面查找当前帖子。  
 if item.link == admin.link:  
 flag = 1  
  
 if flag == 0:  
 try:  
 session.add(admin) # 加入数据库  
 session.commit()  
 except IntegrityError, e:  
 session.rollback()  
  
 session.close() # 提交并关闭  
  
 return item  
  
  
def deleteWord(id):  
 sql = "delete from jihemei WHERE id = :Id"  
 engine.execute(text(sql), Id=id)  
  
# deleteWord(16)