XML 大作业

20203206222 唐权威

### 需求

1. 请使用xpath技术爬取某html方式的网站资料。

（注： 1、不允许爬取json方式的网站；2、若发现爬取网站数据、代码相同，判为雷同作业。)

1. 分别将爬取的数据存取到Xml文件与Json文件。
2. 对上面得到的Xml文件编写相应的DTD 或者Schema验证文件.
3. 使用纯JavaScript的方式对该Xml进行解析（如：表格展示）。
4. 使用Xslt技术对该Xml文件进行转换并展示。

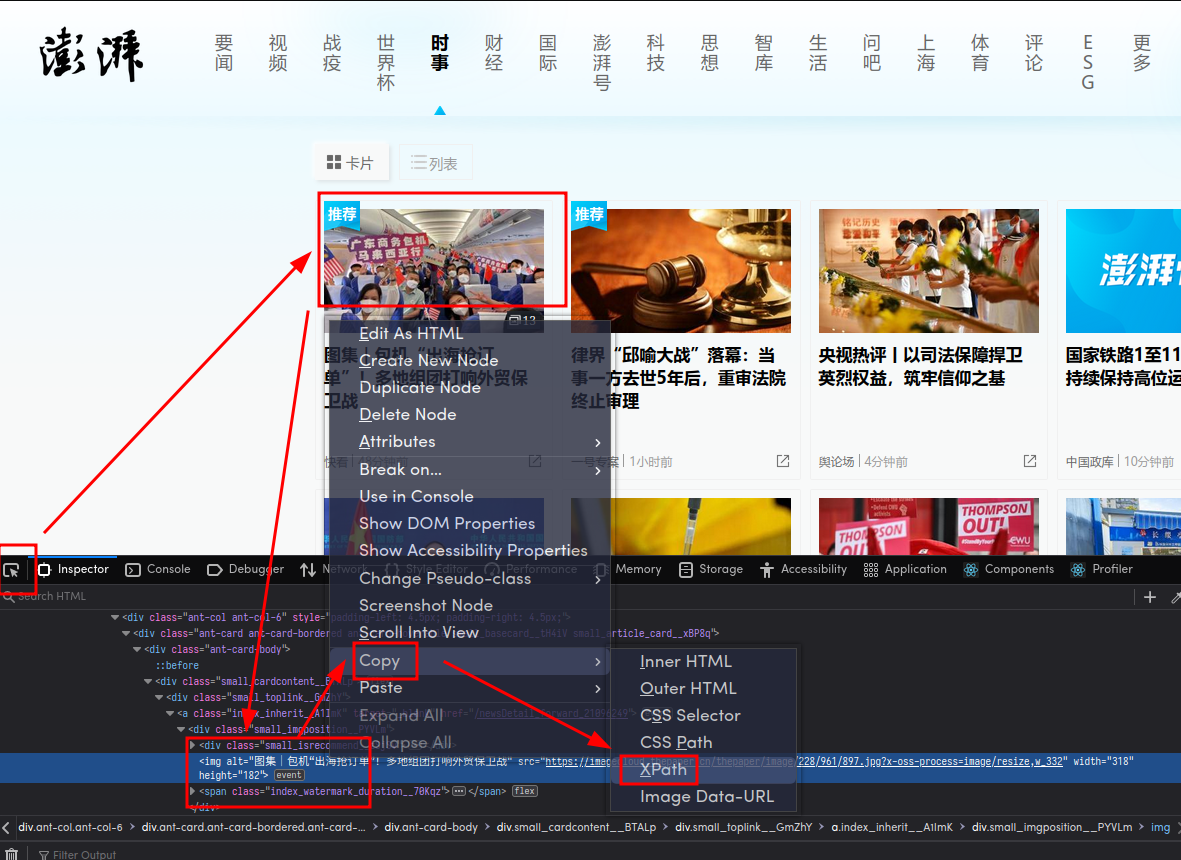
6. 编写JavaScript代码对所得到的Json文件进行解析并展示。

### 分析

1. 使用xpath提取html页面的内容，这里我选择澎湃新闻的时事标签下的卡片：



1. 复制xpath：



### 设计和实现

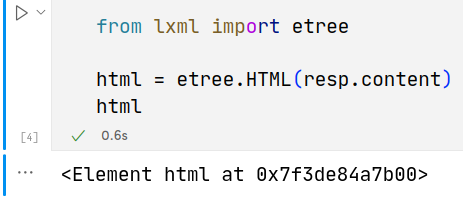
#### 1.使用xpath技术爬取html方式的网站资料

##### 请求网页



可以看到请求成功

##### 将网页内容转换成对象



##### 使用xpath提取信息



我发现图片的链接不能直接打开可能是因为有防盗链的设置，下面我将把图片下载到本地：



#### 分别将爬取的数据存取到Xml文件与Json文件

##### 将提取的信息持久化存储到xml文件



info.xml如图：



##### 将提取的信息持久化存储到json文件

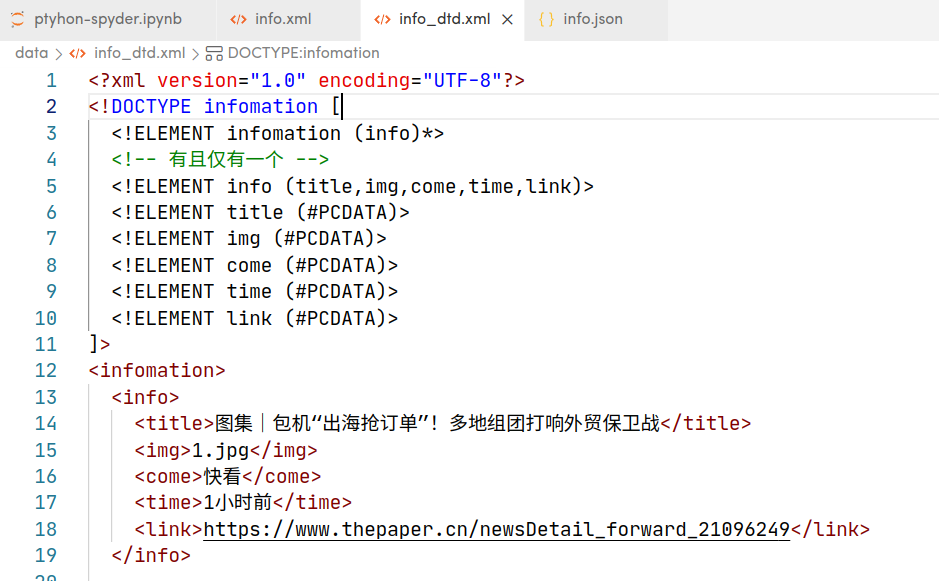


info.json文件如下图：

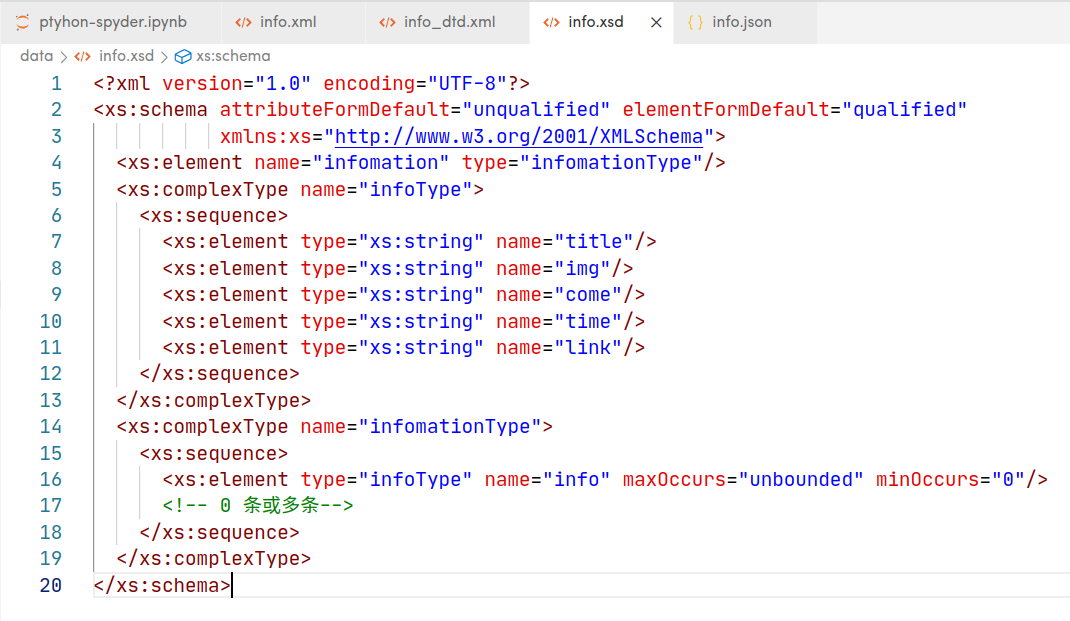


#### 对上面得到的Xml文件编写相应的DTD 或者Schema验证文件

##### 编写DTD文件



##### 编写Schema验证的.xsd文件



下面使用DTD验证

#### 使用纯JavaScript的方式对该Xml和JSON进行解析

##### 4.1编写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=1.0">

<title>XML 大作业</title>

</head>

<body>

<h2>获取最新消息</h2>

<div>

<button onclick="loadXML(event)">获取最新消息 XML</button>

<button onclick="loadJSON(event)">获取最新消息JSON</button>

<button onclick="document.location='info\_xsl.xml'">获取最新消息XSL</button>

<br><br>

<table id="demo" border="sold"></table>

</div>

<script>

/\*

获取XML文件

\*/

const loadXML = (e) => {

const xhr = new XMLHttpRequest();

xhr.onreadystatechange = function () {

if (this.readyState === 4 && this.status === 200) {

handleXML(this.responseXML);

}

};

htm = e.target.innerHTML;

if (htm !== "更新新闻") {

e.target.innerHTML = "更新新闻";

}

xhr.open("GET", "info\_dtd.xml", true);

xhr.send();

}

/\*

处理xml文件

参数 xml ： xml文件

\*/

const handleXML = (xml) => {

let table = "<tr><th>通过XML</th><th>标题</th><th>来源</th><th>时间</th></tr>";

const x = xml.getElementsByTagName("info");

for (let i = 0; i < x.length; ++i) {

const title = x[i].getElementsByTagName("title")[0].childNodes[0].nodeValue;

const img = x[i].getElementsByTagName("img")[0].childNodes[0].nodeValue;

const come = x[i].getElementsByTagName("come")[0].childNodes[0].nodeValue;

const time = x[i].getElementsByTagName("time")[0].childNodes[0].nodeValue;

const link = x[i].getElementsByTagName("link")[0].childNodes[0].nodeValue;

table += `<tr><td>

<img src="img/${img}" alt="img" width="100px">

</td><td>

<a href="${link}">${title}</a>

</td><td>

${come}

</td><td>

${time}

</td></tr>`;

}

document.getElementById("demo").innerHTML = table;

}

/\*

处理json文件

\*/

const loadJSON = (e) => {

const xhr = new XMLHttpRequest();

xhr.onreadystatechange = function () {

if (this.readyState === 4 && this.status === 200) {

handleJSON(JSON.parse(this.response));

}

};

htm = e.target.innerHTML;

if (htm !== "更新新闻") {

e.target.innerHTML = "更新新闻";

}

xhr.open("GET", "info.json", true);

xhr.send();

}

/\*

处理json对象

参数 json ：json对象

\*/

const handleJSON = (json) => {

let table = "<tr><th>通过JSON</th><th>标题</th><th>来源</th><th>时间</th></tr>";

const x = json.infomation;

for (let i = 0; i < x.length; ++i) {

const title = x[i].title;

const img = x[i].img;

const come = x[i].come;

const time = x[i].time;

const link = x[i].link;

table += `<tr><td>

<img src="img/${img}" alt="img" width="100px">

</td><td>

<a href="${link}">${title}</a>

</td><td>

${come}

</td><td>

${time}

</td></tr>`;

}

document.getElementById("demo").innerHTML = table;

}

</script>

</body>

</html>

##### 4.2 尝试打开：

点击之后没有反应，控制台只提示一个跨域问题



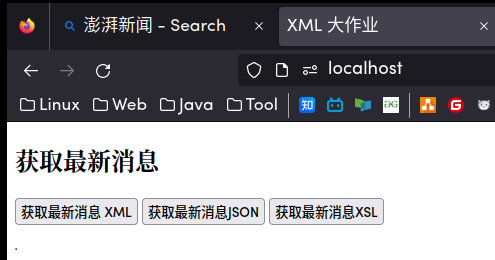
##### 4.3 解决：

配置启动nginx



配置一个监听和映射，上面的问题就不存在了。

##### 4.4效果：



点击“获取最新消息 XML”效果：



点击“获取最新消息 JSON”效果：



#### 使用Xslt技术对该Xml文件进行转换并展示

##### 5.1 编写info.xsl

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>

<body>

<h2>获取权威的新闻 BY XSLT</h2>

<button type="button" onclick="document.location='/'">返回</button>

<table border="1">

<tr>

<th>通过XML</th>

<th>标题</th>

<th>来源</th>

<th>时间</th>

</tr>

<xsl:for-each select="infomation/info">

<tr>

<td>

<img>

<xsl:attribute name="src">

img/<xsl:value-of select="img" />

</xsl:attribute>

<xsl:attribute name="width">

100px

</xsl:attribute>

</img>

</td>

<td>

<a>

<xsl:attribute name="href">

<xsl:value-of select="link" />

</xsl:attribute>

<xsl:value-of select="title" />

</a>

</td>

<td>

<xsl:value-of select="come" />

</td>

<td>

<xsl:value-of select="time" />

</td>

</tr>

</xsl:for-each>

</table>

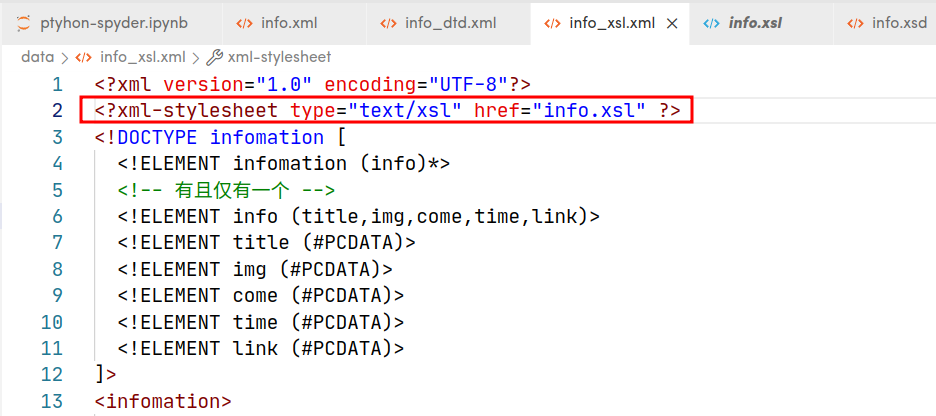
</body>

</html>

</xsl:template>

</xsl:stylesheet>

##### 5.2 引入xsl



##### 5.3效果：



### 总结

1. 此次作业用到了所学的大部分知识，包括xpath、爬虫、xml、json、dtd、xsd和xsl等等，也顺利的实现了所有的需求。
2. 此次作业在原来作业的基础上来说不算复杂，只能使用纯JavaScript反而降低了难度。
3. 以前我对xslt（可扩展样式转换语言）有一点小小的误解，以为它只能提供简单有限的功能，此次作业后我发现xslt其实是一种图灵完备的语言而且它的功能还算比较强大。
4. 此次作业大大的加深了我对所学知识的理解及运用能力，也大大提高了我的编码能力。