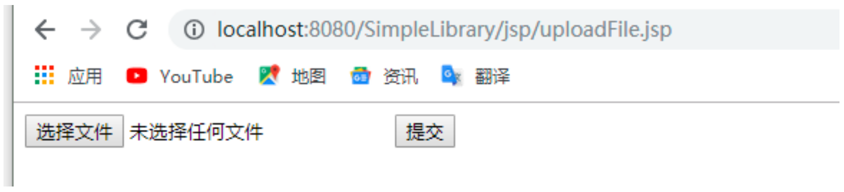
javaweb文件上传基础

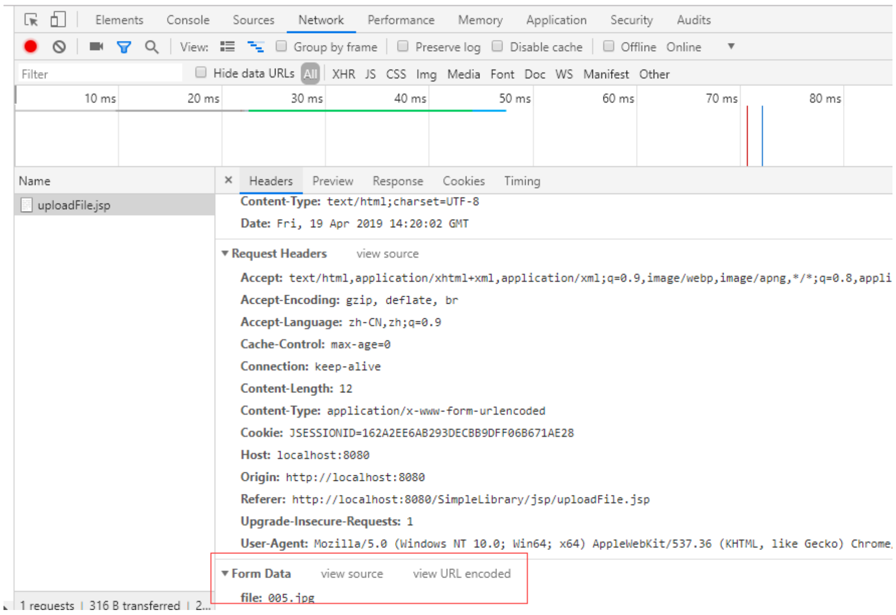
创建jsp，创建文件上传的选择框

```
<form method="post" action="uploadFile.jsp">
    <input type="file" name="file">
    <input type="submit">
</form>
```

显示效果如图



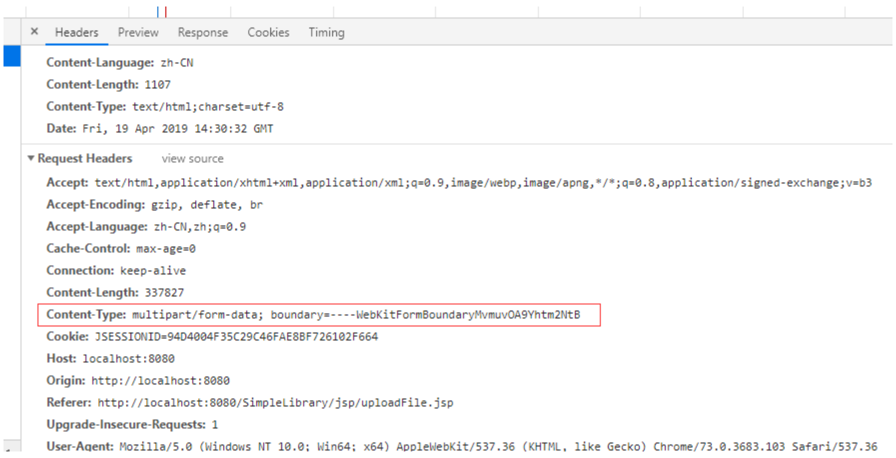
选择文件后，点击提交，选择检查，查看提价类型



```
Content-Type: text/html;charset=UTF-8
Date: Fri, 19 Apr 2019 14:20:02 GMT
▼ Request Headers     view source
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,appli
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Cache-Control: max-age=0
Connection: keep-alive
Content-Length: 12
Content-Type: application/x-www-form-urlencoded
Cookie: JSESSIONID=162A2EE6AB293DECBB9DFF06B671AE28
Host: localhost:8080
Origin: http://localhost:8080
Referer: http://localhost:8080/SimpleLibrary/jsp/uploadFile.jsp
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
▼ Form Data     view source     view URL encoded
file: 005.jpg
```

我们可以看到form data 为字符串类型，所以我们要修改提交的类型。

```
<form method="post" action="uploadFile.jsp" enctype="multipart/form-data">
    <input type="file" name="file">
    <input type="submit">
</form>
```



```
Content-Language: zh-CN
Content-Length: 1107
Content-Type: text/html;charset=utf-8
Date: Fri, 19 Apr 2019 14:30:32 GMT

▼ Request Headers     view source
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Cache-Control: max-age=0
Connection: keep-alive
Content-Length: 337827
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryMvmuvOA9Yhtm2NtB
Cookie: JSESSIONID=94D4004F35C29C46FAE8BF726102F664
Host: localhost:8080
Origin: http://localhost:8080
Referer: http://localhost:8080/SimpleLibrary/jsp/uploadFile.jsp
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36
```

这里注意一下，当表单上传文件时，需要制定表单的enctype的值为multipart/form-data

表单enctype默认值为 application/x-www-form-urlencoded。对于大容量的二进制数据或包非ASCII字符的文本来说，这种编码不能满足要求。

当设置为enctype=multipart/form-data后，表示表单以二进制传输数据，

编写servlet，由于编码方式改变了，我不能再用request.getParamater()的方式来获取数据。

我们可以使用输入流的方式来获取，但是比较麻烦。

```java
public  class uploadServlet extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws Servle
        InputStream in = req.getInputStream();

        Reader reader = new InputStreamReader(in);
        BufferedReader bufferedReader = new BufferedReader(reader);
        String str = null;

         while ((str = bufferedReader.readLine()) != null) {
             System.out.println(str);
         }
     }
 }

 //结果
 ------WebKitFormBoundaryQ42Mm58FWqxqBrfM
 Content-Disposition: form-data; name="file"; filename="Hello.txt"
 Content-Type: text/plain

 Helloworld
 what's your name .hhhhhh!
 ------WebKitFormBoundaryQ42Mm58FWqxqBrfM
 Content-Disposition: form-data; name="desc"

 this is text
```

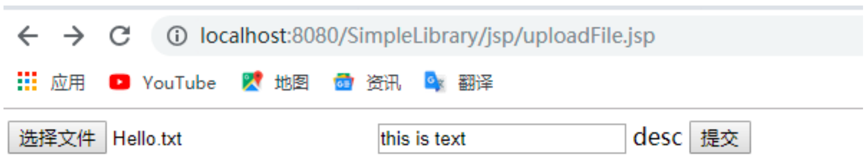
```
------WebKitFormBoundaryQ42Mm58FWqxqBrfM--

```

这个是jsp布局

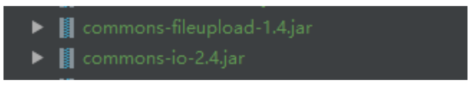
```
<form method="post" action="<%=request.getContextPath()%>/uploadServlet" enctype="multipa
    <input type="file" name="file">
    <input type="text" name="desc"> desc
    <input type="submit">
</form>
```

配置文件

```
<servlet>
    <servlet-name>upload</servlet-name>
    <servlet-class>servlet.uploadServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>upload</servlet-name>
    <url-pattern>/uploadServlet</url-pattern>
</servlet-mapping>
```



使用下面两个jar包


commons-fileupload-1.4.jar
commons-io-2.4.jar

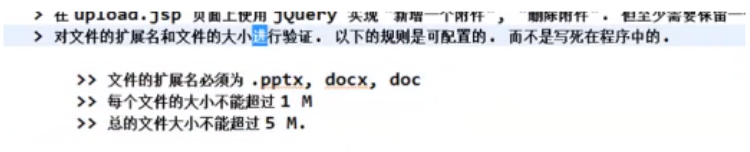
common fileupload 可以解析请求，得到一个Fileitem对象组成的List，无论是一般的文本域还是一个文件域，

```
       //1.得到FileItem的集合items
        // Create a factory for disk-based file items
        DiskFileItemFactory factory = new DiskFileItemFactory();

// Set factory constraints
        factory.setSizeThreshold(1024 * 500);
        //临时目录
        File yourTempDirectory = new File("d:\\temp");
        factory.setRepository(yourTempDirectory);

// Create a new file upload handler
        ServletFileUpload upload = new ServletFileUpload(factory);

// Set overall request size constraint
        //设置总的大小
        upload.setSizeMax(1024 * 1024 * 5);

```

```
// Parse the request
        try {
            List<FileItem> items = upload.parseRequest(req);

            //遍历items
            Iterator<FileItem> iter = items.iterator();
            while (iter.hasNext()) {
                FileItem item = iter.next();

                //如果是表单域
                if (item.isFormField()) {
                    String name = item.getFieldName();
                    String value = item.getString();
                    System.out.println(name + "----" + value);
                } else {
                    //若是文件域，就保存到tempDirectory中
                    String fieldName = item.getFieldName();
                    String fileName = item.getName();
                    String contentType = item.getContentType();
                    boolean isInMemory = item.isInMemory();
                    long sizeInBytes = item.getSize();
                    System.out.println("fieldName----" + fieldName);
                    System.out.println("fileName----" + fileName);
                    System.out.println("contentType---" + contentType);
                    System.out.println("isInMemory---" + isInMemory);
                    System.out.println("sizeInBytes---" + sizeInBytes);

                    InputStream inputStream = item.getInputStream();
                    byte[] bytes = new byte[1024];
                    int len = 0;

                    fileName = yourTempDirectory + "\\" + fileName;
                    System.out.println("temp" + fileName);
                    OutputStream outputStream = new FileOutputStream(fileName);

                    while ((len = inputStream.read(bytes)) != -1) {
                        System.out.println("len ----" + len);
                        outputStream.write(bytes, 0, len);
                    }
                    inputStream.close();
                    outputStream.close();
                }
            }

        } catch (FileUploadException e) {
            e.printStackTrace();
        }
```

结果显示

```
fieldName----file
fileName----Hello.txt
contentType---text/plain
isInMemory---true
sizeInBytes---37
tempd:\temp\Hello.txt
len ----37
desc----this is text

```



> 在 upload.jsp 页面上使用 jQuery 实现 "新增一个附件"， "删除附件"． 但至少需要保留一

> 对文件的扩展名和文件的大小进行验证． 以下的规则是可配置的． 而不是写死在程序中的．

>> 文件的扩展名必须为 .pptx, docx, doc
>> 每个文件的大小不能超过 1 M
>> 总的文件大小不能超过 5 M.

添加配置文件

```
exts = pptx,docx,doc
file.max.size=1048576
total.file.max.size=52442880
```

利用监听器在初始化时，获取配置文件的限制信息

工具栏

```java
package utils;

import java.util.HashMap;
import java.util.Map;

public class FileUploadAppProperties {
    private Map<String, String> properties = new HashMap<>();

    private FileUploadAppProperties(){}

     private  static FileUploadAppProperties instance = new FileUploadAppProperties();

     public static FileUploadAppProperties getInstance(){
         return  instance;
     }

     public void addProperty(String propertyName, String propertyValue) {
         properties.put(propertyName, propertyValue);
     }

     public String getProperty(String propertyName) {
         return properties.get(propertyName);
     }
 }
```

```java
package utils;

```

```java
import java.util.HashMap;
import java.util.Map;

public class FileUploadAppProperties {
    private Map<String, String> properties = new HashMap<>();

    private FileUploadAppProperties(){}

     private  static FileUploadAppProperties instance = new FileUploadAppProperties();

     public static FileUploadAppProperties getInstance(){
         return  instance;
     }

     public void addProperty(String propertyName, String propertyValue) {
         properties.put(propertyName, propertyValue);
     }

     public String getProperty(String propertyName) {
         return properties.get(propertyName);
     }
 }
```

查看结果

```java
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletExc
    String exts = FileUploadAppProperties.getInstance().getProperty("exts");
    String maxsize = FileUploadAppProperties.getInstance().getProperty("file.max.size");
    String totalMaxSize = FileUploadAppProperties.getInstance().getProperty("total.file.m

    System.out.println(exts);
    System.out.println(maxsize);
    System.out.println(totalMaxSize);
 }

 //
 pptx,docx,doc
 1048576
 52442880



```

上传文件的主要步骤如下

```java
//把需要上传的FileItem放入Map中
Map<String, FileItem> uploadFiles = new HashMap<>();


//解析请求，得到FileItem的集合
List<FileItem> items = upload.parseRequest(req);
```

```

//构建FileUploadBean的集合，同事填充uploadFiles
List<FileUploadBean> beans = buildFileUploadBeans(items, uploadFiles);
 //校验扩展名
 validateExtName(beans);

 //校验文件大小，解析的时候已经校验，我们只需要通过异常的到结果

 //进行文件的上传操作
 upload(uploadFiles);

 //把上传的信息保存到数据库中
 saveBeans(beans);
```

文件下载

1.设置contentType 响应头：设置响应的类型是什么？通知浏览器是个下载的文件，通过这个方式，浏览器会自动下载内容，下载后文件名为error.jsp

```
<body>
$END$
<a href="jsp/error.jsp"> error.jsp</a>
</body>

<body>

<%
    response.setContentType("application/x-msdownload");%>
 <h1>error</h1> ${requestScope.message}
 <a href="<%=request.getContextPath()%>/jsp/uploadFile.jsp">return</a>
 </body>
```

2.设置content-Disposition 响应头，通知浏览器不再有浏览器自行处理(或打开)要下载的文件，而由用户手工完成,下载后名字为abc.txt

```
response.setHeader("Content-Disposition","attachment;filename=abc.txt");
```

3.具体文件，可以调用response.getOutputStream的方式，以IO流的方式发给客户端。

```
<body>
$END$

<a href="jsp/error.jsp"> error.jsp</a>
<a href="<%=request.getContextPath()%>/download"> hello.txt</a>
</body>
```

添加Servlet

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletExc
    resp.setContentType("application/x-msdownload");

    String fileName = "hello.txt";
```

```java
    resp.setHeader("Content-Disposition", "attachment;filename=" + URLEncoder.encode(file

    OutputStream outputStream = resp.getOutputStream();

     //mac
     String file = "/Users/rhm/temp/hello.txt";

     InputStream in = new FileInputStream(file);

     byte[] bytes = new byte[1024];
     int len = 0;

     while ((len = in.read(bytes)) != -1) {
         outputStream.write(bytes, 0, len);
     }

     in.close();
     outputStream.close();

}
```