Web中的文件上传与下载

1 文件上传

1-1 file表单

文件选择域表单：<input type=“file”/>

1-2 设置enctype属性

》enctype属性

enctype=“multipart、form-data”

* 在<form>标签中设置enctype属性，修改表单以二进制编码的方式提交数据。
* 因为文件的数据量比较大，因此要使用POST方式提交。
* 若提交表单，则不能用request.getParameter()的方式获取请求参数。

1-3读取请求参数

》request.getInputStream()

使用request.getInputStream()方法，可以获取一个包含请求参数的二进制输入流，但使用该方法读出的文件数据会添加有对参数的一些说明信息，不便对数据进行处理。

》commons-fileupload+commons-io

2 Apach-commons-fileupload

2-1 工作原理

fileupload组件是通过解析request，将请求中每个请求的表单封装为一个FileItem类的对象，FileItem类封装了判断是文件表单还是文本表单的方法，以及获取表单信息的方法。同时，获取FileItem对象需要通过DiskFileItemFactory这个工厂类和ServletFileUpload文件上传处理器，通过这两个部件可以添加对上传的文件的约束。

2-2 操作流程

》创建DiskFileItemFactory工厂类的对象：

DiskFileItemFactory factory = new DiskFileItemFactory();

》设置工厂对象的约束：

setSizeThreshold(int)：设置单个FileItem对象的最大容量，超过这个大小的，数据会先被写到磁盘上，以缓解内存压力。

setRepository(File repository)：设置数据写到磁盘上的路径。

》创建文件上传控制器ServletFileUpload对象：

ServletFileUpload uploader = new ServletFileUpload(FileItemFactory factory);

setFileSizeMax(long fileSizeMax): 设置上传的所有文件的总大小，若文件的总大小超过这个值，则会抛出异常。

》解析request获取FileItem列表

List<FileItem> items = uploader.parseRequest(request);

通过遍历该列表，可以逐个访问FileItem对象。

》FileItem的API

|  |  |  |
| --- | --- | --- |
| 方法名 | 返回值 | 说明 |
| isFormField | boolean | 判断该FileItem是否为一个单纯的文本表单域，否则就是一个文件表单域 |
| getFieldName | String | 返回表单域的名称，即表单的name属性值 |
| getString | String | 若是一个文本表单，则返回表单的value值 |
| getName | String | 若是一个文件表单，返回表单所选文件的文件名（含扩展名） |
| isInMemory | boolean | 若是一个文件表单，判断该文件是否在内存中 |
| getSize | long | 若是一个文件表单，返回文件的大小 |
| getContentType | String | 返回表单的内容类型 |
| getInputStream | InputStream | 获取文件表单域的一个输入流，以读取文件的内容 |
| getOutputStream | OutputStream |  |

\*\* getString方法可以指定编码方式，通常表单改用二进制编码方式传输数据时，需要在此处指定解码方式。

2 文件下载

2-1 简单下载

将超链接的地址指向要下载的文件，通过就可以直接浏览器打开文件，或右键另存为。

2-2 设置response的响应参数

》设置contentType响应头

response.setContentType(“application/x-msdownload”);

设置响应的内容类型，以告知浏览器要开启一个下载工作：

》设置Content-Disposition响应头

response.setHeader(“Content-Disposition”, “attachment;filename=文件名”)；

通知浏览器不要自行打开文件或链接，而由用户手动完成：

此处的filename是指下载下来的文件名称，设置时，使用URLEncoder.encode()设置编码类型。

》使用输出流输出文件

response.getOutputStream();

使用response的输出流对象，从服务器端读取文件后输出到浏览器客户端。

3 国际化

使软件能够支持不同国家的语言、数字符号表示方式等习惯

* 本地化
* 国际化（i18n，internationlization）

》基本的思想

* 时间、货币、数值等可以在程序中根据不同的地区选用不同的标识方式
* 文本内容存储在一个资源文件中，根据地区读取相应的资源文件

3-1 Locale类

Locale（java.util.Locale）是一个代表国家地区和语言的类，用于提供与地区相关的信息，与国际化的格式和日期、数字等类的格式使用相关。

》创建对象

* Locale类中封装了许多常量的Locale对象代表不同的国家和地区，以方便获取对象。
* 使用构造器关键Locale对象，指定语言的缩写字符和国家的缩写字符，构造器会从封装好的Locale对象中找出相应的对象返回。

》Locale的API

|  |  |
| --- | --- |
| 方法 | 说明 |
| getDisplayCountry() | 获取Locale对象所代表的国家，根据本地语言返回国家名称 |
| getDisplayLanguage() | 获取Locale对象所代表的国家的语言，根据本地语言返回语言名称 |
| getCountry() | 获取Locale对象所代表的国家，返回国家的英文缩写 |
| getLanguage() | 获取Locale对象所代表的国家的语言，返回语言的英文缩写 |

3-2 DateFormat类

DateFormat（java.text.DateFormat）是一个日期格式的类，用于将日期以不同的字符串格式输出，以及将日期字符串转化为日期类型。

》获取对象

* 静态方法：getDateInstance()、getDateTimeInstance()、getInstance()、getTimeInstance()等静态方法可以获取对象；DateFormat是一个抽象类，通过此方法可以直接获得一个该类的对象，同时可以传入Locale类指定地区、日期和时间格式的字符串（DateFormat以封装好相关的格式常量）。
* SimpleDateFormat实现类：创建子类对象，同时指定时间的字符串格式。

》DateFormat类的API

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数 | 返回值 | 说明 |
| format | Date | String | 将Date日期按照DateFormat对象所指定测格式转化为字符串 |
| parse | String | Date | 将代表时间的字符串转化Date类型的时间 |

3-3 NumberFormat类

NumberFormat（java.text.NumberFormat）与DateFormat相似，用于数字与格式化数字字符串之间的转化。

》获取对象

* getCurrencyInstance()：获取对象，该对象能够实现数字与货币字符串之间的转化。
* getNumberInstance()：获取一个普通的数字与格式化字符串的转化对象。

\*\* 在获取对象时，通过传入Locale对象指定地区的数字格式。

》NumberFormat类的API

与DateFormat相同，有format和parse两个常用方法。

3-4 MessageFormat类

MessageFormat（java.text.MessageFormat）用于字符串的格式化输出，使用占位符，再传入目标的字符串。

》使用方法

* 静态方法：format(String pattern,Object…argument)，pattern的字符串中要带有占位符，使用argument的参数填充占位符。
* 占位符：{n}，n为从0开始的整数。

3-5 ResourceBundle类

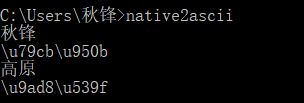
ResourceBundle用于根据Locale指定的地区地位资源文件，即不同的国家有专门的资源文件，而ResourceBundle则是根据传入的Locale对象找到对应的资源文件，读取其中的数据。

》资源文件

properties文件：以键值对的形式存储数据（key=value）。

* 对于一个properties文件（原始资源文件的文件名为基名city.properties），创建不同地区语言版本的properties文件，以“基名\_语言代码\_国家代码”的方式命名（city\_en\_US.properties或city\_zh\_CN.properties）。
* 所有资源文件的key必须与原始文件的key保持一致。
* 对于中文的value，要使用其ASCII码。

》native2ascii程序



* Java的native2ascii程序可以获取中文的ASCII码。
* 该程序在JDK安装目录的bin文件夹中。
* 打开命令行，输入native2ascii启动程序，
* 再输入中文，敲回车即可得到对应的ASCII码

》ResourceBundle的API

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | 说明 |
| static getBundle | String 原始资源文件的基名，不带扩展名 | ResourceBundle对象 | 获取ResourceBundle对象，指定原始资源文件和地区，用于读取资源文件的数据 |
| Locale 指定国家地区 |
| getString | String 资源文件中的key | String 指定key对应的value | 根据键获取资源文件中的数据 |