servlet_part1.txt

笔记本:	杰普实训三_笔记		
创建时间:	2018/8/15 14:56	**更新时间:**	2018/8/18 16:57
作者:	家龙		
URL:	file://D:\资料\Desktop\servlet资料1\servlet_part1.txt		

对于简单的html页面，我们只需要双击就可以看到其效果。怎么样可以让别人像平时上网一样能够访问到我们的页面呢，需要借助Web服务器与Web项目（应用程序）。

WEB，在英语中web即表示网页的意思，它用于表示Internet主机上供外界访问的资源。

web是英文网的意思，WEB引申为“环球网”，而且，在不同的领域，有不同的含义。对于普通的用户来说，web仅仅只是一种环境——互联网的使用环境、氛围、内容等；而对于网站制作、设计者来说，它是一系列技术的复合总称（包括网站的前台布局、后台程序、美工、数据库领域等等的技术概括性的总称）

Internet上供外界访问的Web资源分为：

1）静态web资源（如html 页面）：指web页面中供人们浏览的数据始终是不变。

2）动态web资源：指web页面中供人们浏览的数据是由程序产生的，不同时间点访问web页面看到的内容各不相同。

静态web资源开发技术：Html

常用动态web资源开发技术：JSP/Servlet、ASP、PHP等

在Java中，动态web资源开发技术统称为Javaweb。

1.什么是WEB应用程序：

可以通过互联网（web）访问的应用程序，需要用户有网，用户只需要有浏览器即可，不需要再安装其他软件。如电子商务网站，网络银行，搜索引擎，门户网站，论坛等。WEB应用程序的一个最大好处是用户很容易就可以访问应用程序。

一个web应用由多个静态web资源和动态web资源组成，如:html、css、js文件，Jsp文件、java程序、支持jar包、配置文件等等。

WEB应用程序开发，是目前软件开发领域的主流方向之一。Web应用开发好后，若想供外界访问，需要把web应用交给web服务器管理。

2.搭建JavaWEB开发环境

使用javaEE开发Web项目

开发环境：JDK,Tomcat服务器，eclipse

技术：java,html&css，servlet/jsp，js，xml..

3.Tomcat服务器简介：

Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由Apache、Sun 和其他一些公司及个人共同开发而成。由于有了Sun 的参与和支持，javaEE最新的Servlet 和JSP 规范总是能在Tomcat 中得到体现，Tomcat 7实现了对于Servlet 3.0 和JSP 2.2 规范包括注解。因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。

4.Tomcat目录结构：

/bin：存放各种平台下用于启动和停止Tomcat的文件。startup.bat用来启动Tomcat，但

需要先配置JAVA_HOME环境变量才能启动，shutdawn.bat用来停止Tomcat

windows：

startup.bat

Shutdown.bat

Linux：

Startup.sh

Shutdown.sh

/conf：存放Tomcat服务器的各种配置文件。编码、端口…

/lib： 存放Tomcat服务器所需的各种jar文件，在tomcat中可以部署多个web项目，共享此目录下的jar包。我们也可以自己把一些其他jar包放到这里。

/logs：存放Tomcat的日志文件，记录Tomcat的启动和关闭等信息。

/temp：存放Tomcat在运行过程中，产生的临时文件。

/webapps：存放部署到Tomcat中的Web项目，部署项目后，默认将web应用的文件发布

到此目录下。一个文件夹代表一个项目，如Tomcat自带的Root项目。

/work：与web项目中的JSP文件有关。当客户端用户（浏览器）访问一个JSP文件时

Tomcat会通过JSP生成java文件，然后编译成class文件，存放在此目录下。

5.手动编写第一个Web项目。

对于java项目，我们只需要写好类，写好程序访问入口（main方法），即可运行。

运行环境：JDK

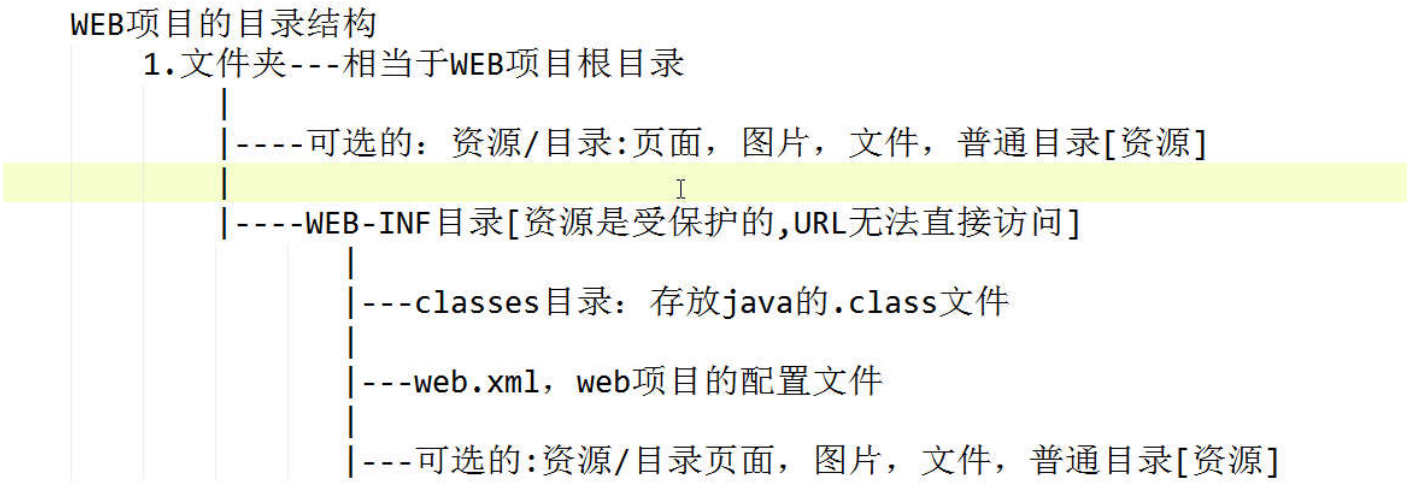
运行方式：main方法。

对于Web项目，我们需要有一定的目录结构，需要借助于服务器运行。

运行环境：JDK，JavaEE环境（相关jar包，web服务器中有）

同时Web项目与java项目的目录结构也是不一样的，Web项目的项目结构是固定的。

5.1 Web项目结构：



classes目录 （存放将来web项目中的java文件编译后的class）

lib目录（存放该web项目要使用到的第三方jar包）

WEB-INF目录---web.xml（web项目中，最重要的一个配置文件）

项目名-----

自定义目录、自定义文件（图片/css/html）

1）建一个文件夹，表示整个Web项目，如FirstWebDemo。

2）在FirstWebDemo目录下建立文件夹，名为WEB-INF，WEB-INF目录下新建两个文件夹

一个叫classes，一个叫lib，新建一个文档，名字为web.xml。

注：tomcat服务器在启动的时候会读取webapps下的所有Web项目的web.xml配置文件

web.xml的书写格式可以参考webapps下的example项目中的web.xml。

3）在FirstWebDemo目录下，除了WEB-INF外，还可以创建任意的文件或者文件夹

例如：在项目中要用到的html页面，图片，css样式，js脚本等。

5.2 FirstWebDemo

根据上述项目结构，可以建立如右所示FirstWebDemo项目，FirstWebDemo目录中除了必要的WEB-INF目录外，还有一个html页面，一个txt文档，images文件夹中有图片，css文件夹中有css文件。

注：web.xml参考范例即可，如：

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
```

```
                        http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
    version="3.0"
    metadata-complete="true">

      <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.xhtml</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
      </welcome-file-list>

  </web-app>
```

5.21.修改Tomcat的默认端口号

Tomcat在启动时默认使用8080端口号接受http协议请求，但是Oracle数据库默认使用的是

8080端口，所以一般把Tomcat的端口号修改一下。

conf/server.xml文件中:

```
<Connector connectionTimeout="20000" port="修改此处"
protocol="HTTP/1.1" redirectPort="8443"/>
```

修改port属性值，如8888。

5.22 Web项目的部署和访问

部署：把写好的FirstWebDemo文件夹放到webapps目录下即可

启动服务器：双击bin目录下的startup.bat，启动不报错，项目部署成功，可以通过浏览器访问服务器的方式访问web项目。

访问：访问服务器中的Web应用程序，按照http协议的访问规则访问即可

http://ip:port/项目名/资源路径

如：http://localhost:8888/FirstWebDemo/hello.html

http://localhost:8888/FirstWebDemo/first.txt

http://localhost:8888/FirstWebDemo/images/sn.jpg

以上为访问web项目中的静态资源（html、css、图片）的方法步骤，注：WEB-INF中的资源不能直接访问，需要经过特殊的方法访问，将在后面的章节介绍。

5.3 访问Web项目中的动态资源，如java代码。访问页面后提交了一些数据给服务器，要与

数据库进行交互，或者发送某个请求，请求数据要在页面展示出来，比如电子商务网站的商品信息，商品的数量，价格等不是一成不变的，数据保存在数据库中。

使用浏览器通过url的方式访问服务器中应用程序下的java代码，使其调用对应方法，完成

相应功能，处理或者返回数据，需要借助于javaEE规范中的Servlet。

5.4 对于一个普通的java类，我们要调用其中的方法，需要有程序访问入口（main方法），创建对象，然后调用其方法，对于web项目来说，tomcat负责处理程序入

口的问题，使用反射的方式创建对象，也会调用对应对象的对应方法（service方法）。

6.Servlet
简介：
允许浏览器通过url的形式来访问的一种java类，叫做servlet，也是java类。

参考资料：关于servlet与servlet容器
Java Servlet（Java服务器小程序）是一个基于Java技术的Web组件，运行在服务器端，由Servlet容器（Tomcat提供）所管理，用于生成动态的内容。Servlet是平台独立的Java类，编写一个Servlet，实际上就是按照Servlet规范（JAVAEE规范之一）编写一个Java类。Servlet被编译为平台独立的字节码，可以被动态地加载到支持Java技术的Web服务器中运行。
Servlet容器也叫做Servlet引擎，是Web服务器或应用程序服务器的一部分，用于在发送的请求和响应之上提供网络服务，解码请求，格式化响应。Servlet没有main方法，不能独立运行，它必须被部署到Servlet容器（tomcat）中，由容器来实例化和调用 Servlet的对应方法（如doGet()和doPost()），Servlet容器在Servlet的生命周期内包容和管理Servlet。在JSP技术 推出后，管理和运行Servlet/JSP的容器也称为Web容器。
Web 容器的例子有: Tomcat, Resin, Weblogic、WebSphere、JBoss等。
有了servlet之后，用户通过单击某个链接或者直接在浏览器的地址栏中输入URL来访问Servlet，Web服务器接收到该请求后，并不是将 请求直接交给Servlet，而是交给Servlet容器。Servlet容器实例化Servlet，调用Servlet的一个特定方法对请求进行处理， 并产生一个响应。这个响应由Servlet容器返回给Web服务器，Web服务器包装这个响应，以HTTP响应的形式发送给Web浏览器。

- servlet容器能提供什么：

1）通信支持：利用容器提供的方法，你能轻松的让servlet与web服务器对话，而不用自己建立serversocket、监听某个端口、创建流等。只需要考 虑如何在servlet中实现业务逻辑（如处理一个订单）。
2）生命周期管理：servlet容器控制着servlet的生与死，它负责加载类、实例化和初始化servlet，调用servlet方法，以及使servlet实例被垃圾回收，有了servlet容器，你不需要太多的考虑资源管理。
3）多线程支持：容器会自动为它所接收的每个servlet请求创建一个新的java线程。针对用户的请求，如果servlet已经运行完相应的http服务方法，这个线程就会结束。
4）声明方式实现安全：利用servlet容器，你可以使用xml部署描述文件来配置和修改安全性，而不必将其硬编码写到servlet类代码中。
5）JSP支持：servlet容器负责将jsp代码翻译为真正的java代码。

6.1 如何编写一个servlet


```
package servlet;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.ServletException;
import java.io.IOException;
import java.io.PrintWriter;

//第一个servlet类;继承抽象父类HttpServlet
public class FirstServlet extends HttpServlet{

    //请求URL--执行一段java代码(作用:处理 这个请求)(哪段:service方法)
    //service-->service-->doXxx;
    //方法重写:方法名一样,参数列表一样,访问权限不能被缩小,异常不能被扩大'

    protected void doGet(HttpServletRequest request,HttpServletResponse response) throws
servletException,IOException{
        //这个URL请求的处理过程就在这里写了
        //获取请求中的相关信息[参数],进行一系列处理[接收参数]
        //注册:...数据库交互....
        //返回响应
        PrintWriter pw = response.getWriter();
        pw.print("hello world");
        pw.flush();
        pw.close();
    }
}
```


```
62      javac -d WEB-INF/classes -cp ../servlet-api.jar -encoding utf-8
        FirstServlet.java
63
64      1.请求URL很多【查询学生，查询教师，查询课程信息，登录，注册，删除，更改个人
        信息】
65      --执行的一段java代码(service--daGet--servlet)
66      怎么样让一个URL---serlvet[配置]
67      很多个请求--很多个servlet
68
69    http://localhost:8888/firstDemo/login.servlet  -- servlet1
70    http://localhost:8888/firstDemo/images/register  -- serlvet2
71    http://localhost:8888/firstDemo/findStudent      --servlet3
72    2. 执行方法：Servlet---多态
73       .Servlet接口中定义的5个方法
74       .service(ServletReuqst)--HttpServletservice()--service--doGet
75
76      部署
77      启动
78      访问
```

1）编写一个普通的java类，然后这个类实现或者继承javaEE规范中所提供的指定接口或者父类，并重写一些指定方法。这个指定的父类或者指定的接口，可以在javaEE的API中找到，在Tomcat的lib下的servlet-api.jar中有。

Servlet接口--

GenericServlet抽象类

--HttpServlet抽象类

--自定义servlet

自定义servlet可以实现Servlet接口或者，继承GenericServlet或者HttpServlet。

实际上最重要的是我们需要service方法，继承HttpServlet功能最丰富。

2）继承HttpServlet后，重写其doGet()与doPost()方法，tomcat在获取到get或者post请求后分别调用。编译后，将包以及类文件放入webapps/FirstWebDemo/WEB-INF/classes中。

javac -d ../ -cp ../servlet-api.jar FirstServlet.java

3）配置web.xml

由于客户端是通过URL地址访问web服务器中的资源，所以Servlet程序若想被外界访问，必须把servlet程序映射到一个URL地址上，这个工作在web.xml文件中使

用<servlet>元素和<servlet-mapping>元素完成。
<servlet>元素用于注册Servlet，它包含有两个主要的子元素：<servlet-name>和<servlet-class>，分别用于设置Servlet的注册名称和Servlet的完整类名。
一个<servlet-mapping>元素用于映射一个已注册的Servlet的一个对外访问路径，它包含有两个子元素：<servlet-name>和<url-pattern>，分别用于指定Servlet的注册名称和Servlet的对外访问路径

```
<servlet>
<servlet-name>firstServlet</servlat-name>
<servlet-class>com.briup.servlet.FirstServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>firstServlet</srvlet-name>
<url-pattern>/firstServlet</url-pattern>
</servlet-mapping>
```

url-pattern标签指定访问此servlet的请求路径，/不能省略，代表项目根目录。
4）通过浏览器url访问servlet
localhost:8888/FirstWebDemo/FirstServlet

6.2 使用eclipse编写servlet
1）ecplise编码设置
Window--preferences--输入encoding，把下面的选项编码都设置为UTF-8。
2）ecplise配置Tomcat
Servers窗口下new server选择解压好的Tomcat，配置成功后需更改配置。
服务器路径与部署路径选择Tomcat安装目录，webapps
端口号为了避免冲突，进行更改如8888.
3）使用IDE工具创建web项目
New--Dynamic Web Project--创建web项目，修改Target runtime为配置好的Tomcat，Dynamic web module version版本为2.5。
4）编程实现servlet的访问。
Index.html:

```
<a href="servlet/firstServlet">使用GET方式请求servlet</a>
<form action="servlet/firstServlet" method="post">
<input type="submit" value="使用Post方式请求" />
</form>
```

FirstServletDemo的doGet方法中:

```
System.out.println("接收到get请求");
resp.setContentType("text/html;charset=utf-8");
PrintWriter pw = resp.getWriter();
pw.print("<center><Strong>处理Get请求</Strong></center>");
```

```
        pw.flush();
```

5）浏览器中输入localhost:8888/ServletDemo/index.html查看。
