第 5 章 Servlet

5.1 Servlet 基础

1.什么是 Servlet？

（1）服务器端的 Java 应用程序

Servlet = Server + Applet

（2）JSP 页面的本质

执行一个 JSP 页面，生成文件所在服务器端位置：

工作空间\.metadata\.plugins\org.eclipse.wst.server.core\tmp1\work\Catalina\localhost\ 项 目 名 \org\apache\jsp 文件夹

2.解释.java 文件

（1）final 关键字：终结者

-类遇见 final，类无后

-方法遇见 final，方法不重写

-变量遇见 final，变量变常量

（2）三个主要方法：_jspInit、_jspDestroy、_jspService

（3）内置对象：pageContext、session、application、config、out、page

（4）response.setContentType

（5）try-catch

（6）显示内容

3.演示.class 文档

5.2 在 Eclipse 中创建 Servlet（模板）

1.创建 Web 工程（web2.5，为什么选择这个版本？）

2.右键单击项目：New -> Servlet

3.java package：tom.jiafei

Class name：MyServlet -> Next

4.URL Mappings：/aaa -> Next

http://localhost:8080/项目名/aaa 回车

5.选择方法 √init √destroy √service

==========

5.3 Servlet 文档

5.3.1 主要方法

1.init 方法（执行次数=1）

完成类似于构造方法的初始化功能，参数为 ServletConfig 的实例。

2.service 方法（执行次数>=1）

（1）作用：该方法用来响应客户端发出的请求。

（2)执行过程:执行时会检查 HTTP 请求的类型,并相应地调用 doGet()、doPost()等方法。

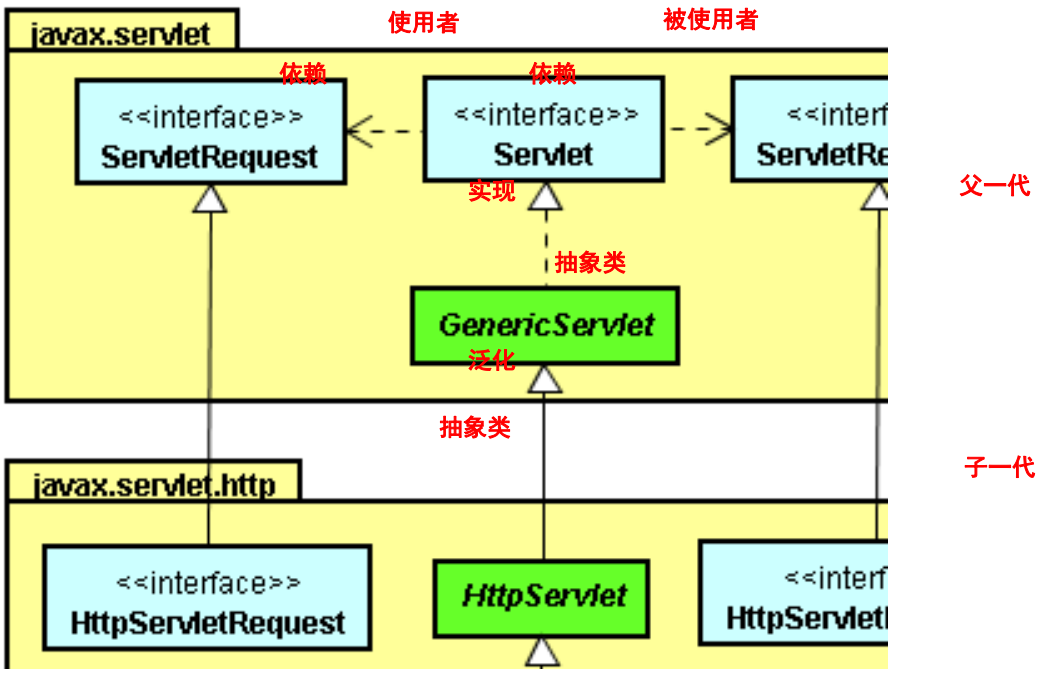
（3）通常做法：不使用 service() 方法，直接使用 doGet()、doPost()等方法。

3.destroy 方法（执行次数=1）

当不再需要 Servlet 实例或重新装入时。

使用 destroy() 方法可以释放掉所有在 init 方法中申请的资源。

5.3.2 Servlet 常用类及接口（**红楼梦四大家族图**）



Servlet **体系-继承-依赖结构图**

5.3.3 生命周期

1. Tomcat 创建 Servlet 的一个实例

MyServlet ms = new MyServlet();

2. Tomcat 用实例调用 init()方法初始化

ms.init();

说明：主要用于一次性的初始化

3. 若 Tomcat 对该实例有请求，则调用 service()方法

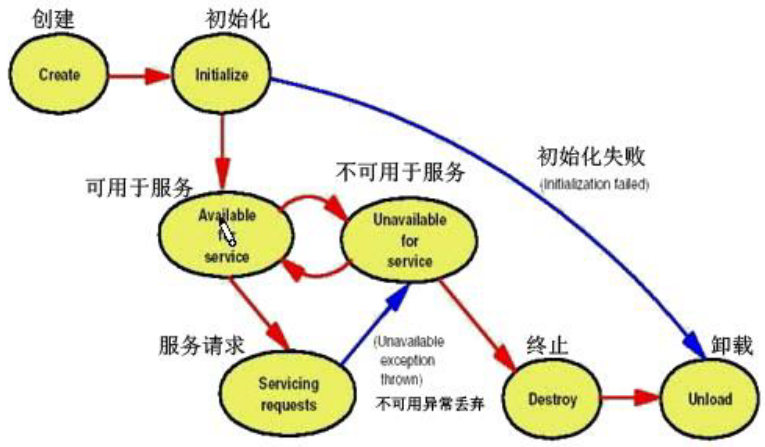
ms.service();

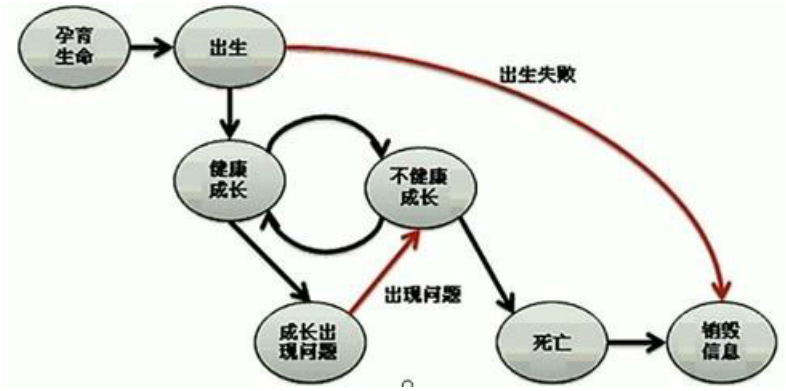
4. 实例被销毁前，调用 destroy()方法

ms.destroy();

5. Tomcat 销毁 ms 并标记 ms 为垃圾

6. 挂图





5.3.4 演示：在三个方法中写

System.out.println("init");

System.out.println("destroy");

System.out.println("service");

5.4 web.xml 文档

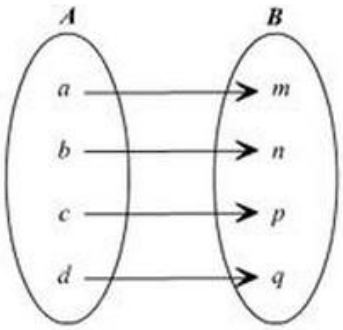
1.文档位置：WebContent/WEB-INF/web.xml

2.文档标记结构及意义

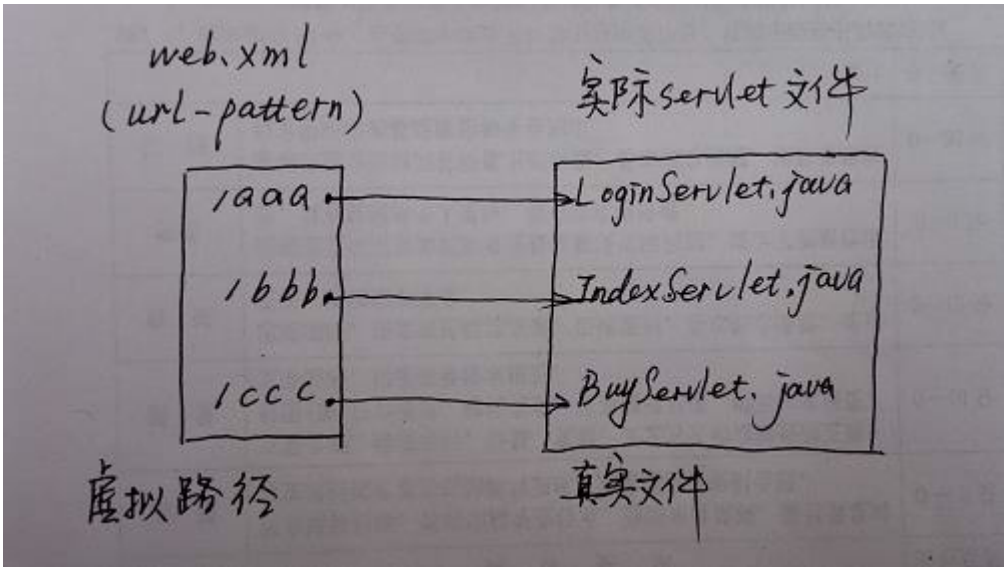
```
<servlet>  －－Servlet 基本信息
   <servlet-name>Servlet 名称</servlet-name>
   <servlet-class>类包路径</servlet-class>
</servlet>
<servlet-mapping>  －－  Servlet 映射
   <servlet-name>Servlet 名称</servlet-name>
   <url-pattern>映射路径（或 Servlet 地址）</ url-pattern >
</servlet-mapping>
```

3.url-pattern：虚拟目录/地址映射

（1）函数映射 B = f(A),m=f(a),n=f(b)…



（2）地址映射



==========

5.5 doPost 和 doGet 方法

1.创建 Servlet（添加三个初始化参数 x,y,z）选择方法

√ doGet　　√ doPost

在 Eclipse 中输入

doGet(){System.out.println(“doGet()”);}

doPost(){System.out.println(“doPost ()”);}

右键 Run As　->说明：默认调用 doGet()方法

2.在 Servlet 页面上显示内容

```
out.print(“doGet()”);//错误
//利用 PrintWriter 实例将数据发送到客户端
PrintWriter out = response.getWriter();
out.print(“doGet()”);//正确
```

3.在 Servlet 页面上显示内容

```
response.setContentType(“text/html;charset=gb2312”);
```

4.两方法互相调用

```
doGet(){doPost(request,response);}
doPost(){doGet(request,response);}
```

5.使用超链接执行 Servlet

```
<a href= " aaa">ActionServlet</a>
```

6.提交表单执行 Servlet

```
<form action="aaa">
   <input type=text    name="txt"/>
   <input type=submit    value=ok/>
</form>
```

7.在 Servlet 中获取表单参数

```
String txt = request.getParameter("txt");
因为 servlet 是 JSP 的本质，又因为在 JSP 中使用 getParameter 方法获取参数，
所以在 servlet 中也使用 getParameter 获取参数。
```

==========

8.Servlet 响应表单提交方式

<form action="aaa" method="get/post"></form>

get 方式：调用 doGet 方法

post 方式：调用 doPost 方法

9.获取单个初始化参数

（1）添加初始化参数（x，y，z）

（2）初始化参数在 web.xml 文件中位置

```
<servlet>
   <init-param>
        <param-name>x</param-name>
        <param-value>参数 x</param-value>
</init-param>
</servlet>
```

（3）调用方法：String x = getInitParameter("x");

==========

9.获取所有初始化参数

```
Enumeration e = getInitParameterNames();
while(e.hasMoreElements()){
```

```
    String param = e.nextElement().toString();
    String value = getInitParameter(param);
    out.print(param+":"+value+"<br/>");
}
```

10.获取 ServletConfig 对象（配置对象）

```
ServletConfig config = getServletConfig();
//使用 config 对象获取初始化参数
String y = config.getInitParameter("");
```

11.获取 Servlet 名称

```
String name = getServletName();
```

12.重定向和 Servlet 转发

（1）重定向 response.sendRedirect("地址");

（2）转发

```
//获取 RequestDispatcher 对象
RequestDispatcher dis = request.getRequestDispatcher("地址");
//调用 forward 方法
dis.forward(request,response);
```

==========

（3）转发与重定向区别

1)跳转后页面地址（安全性）：重定向显示 = 转发不显示

2)获取前一页面数据：重定向获取不到 = 转发可获取

（4）实例：

a.jsp		servlet		b.jsp
name="num"	-->	**request.getParameter("num");**	--->	**request.getParameter("num");**