1:手动编写第一个Servlet

1：继承HttpServlet类;

2：重写doGet（）或doPost（）方法；

3：在web.xml中注册Servlet；

1：Index.jsp文件代码：

<body>

<h1>servlet小例子</h1>

<hr>

<a href=*"servlet/HelloServlet"*>Get方式请求HelloServlet</a>

<form action=*"servlet/HelloServlet"* method=*"post"*>

<input type=*"submit"* value=*"post方式请求HelloServlet"*>

</form>

</body>

2：Web.xml文件代码：

<servlet>

<servlet-name>HelloServlet</servlet-name>

<servlet-class>servlet.HelloServlet</servlet-class>//HelloServlet所在的类名，包括包名。

</servlet>

<servlet-mapping>

<servlet-name>HelloServlet</servlet-name>

<url-pattern>/servlet/HelloServlet</url-pattern>//HelloServlet访问的路径，与jsp上超连接要一致。

</servlet-mapping>

3：servlet.HelloServlet文件代码：

**public** **class** HelloServlet **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

System.***out***.println("处理Get请求。。。");

PrintWriter out = response.getWriter();

response.setContentType("text/html;charset=utf-8");

out.println("<Strong>HelloServlet</Strong><br>");

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

System.***out***.println("处理Post请求。。。");

PrintWriter out = response.getWriter();

response.setContentType("text/html;charset=utf-8");

out.println("<Strong>HelloServlet</Strong><br>");

}

}

4:通过web.xml文件来获取初始化参数：

Web.xml代码：

<servlet>

<servlet-name>HelloServlet</servlet-name>

<servlet-class>servlet.GetInitParameterServlet</servlet-class>

<init-param>

<param-name>username</param-name>

<param-value>admin</param-value>

</init-param>

<init-param>

<param-name>password</param-name>

<param-value>admin</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>HelloServlet</servlet-name>

<url-pattern>/servlet/GetInitParameterServlet</url-pattern>

</servlet-mapping>

GetInitParameterServlet代码：

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

out.println(**this**.getUsername());

out.println(**this**.getPassword());

}

**public** **void** init() **throws** ServletException {

**this**.setUsername(**this**.getInitParameter("username"));

**this**.setPassword(**this**.getInitParameter("password"));

}

2：用Eclipse编写servlet：

1：创建index.jsp文件；

2：在src下创建Servlet文件，重写doGet和doPost方法。eclipse+tomcat8不需要生成web.xml直接在servlet类里使用注解。

注：创建连接等一定要用 包名.类名，否则报错。

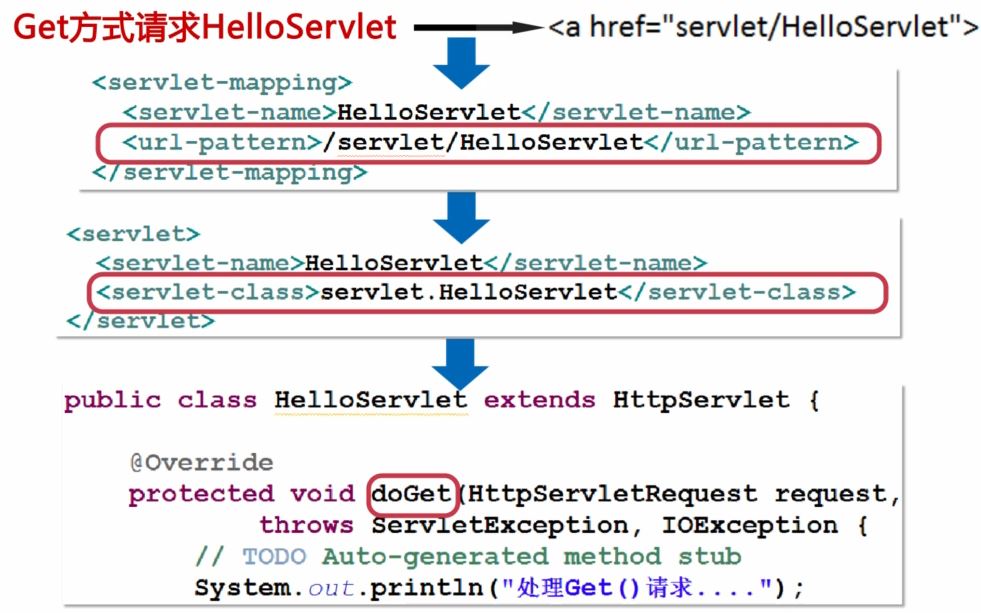
PrintWriter out = response.getWriter();

response.setContentType("text/html;charset=utf-8");

out.println("<Strong>HelloServlet</Strong><br>");

在doGet或doPost方法里这几句是必须的。设置编码，输出字体为加粗

3：Servlet执行流程：



4: Servlet执行顺序：

当Servlet首次被执行是生命周期：

TestServlet1构造方法被执行。

TestServlet1初始化方法被执行。

TestServlet1的doGet方法被执行；

TestServlet1销毁方法被执行。



数字越小优先级越高。

5：

6：





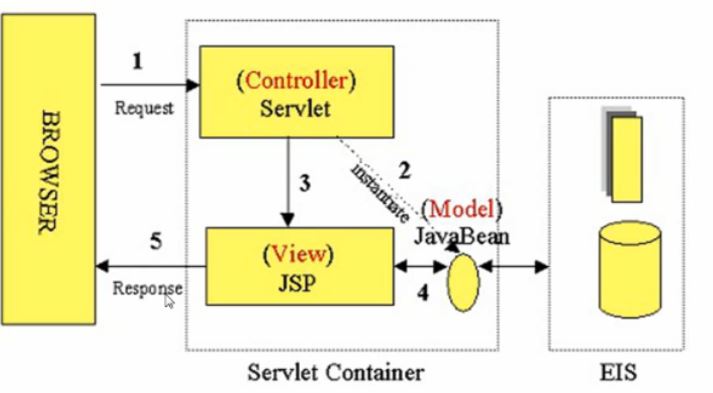


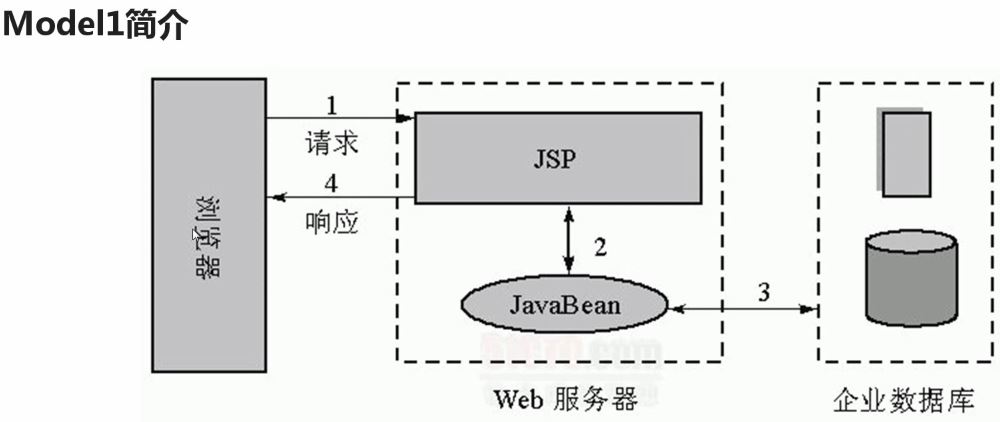


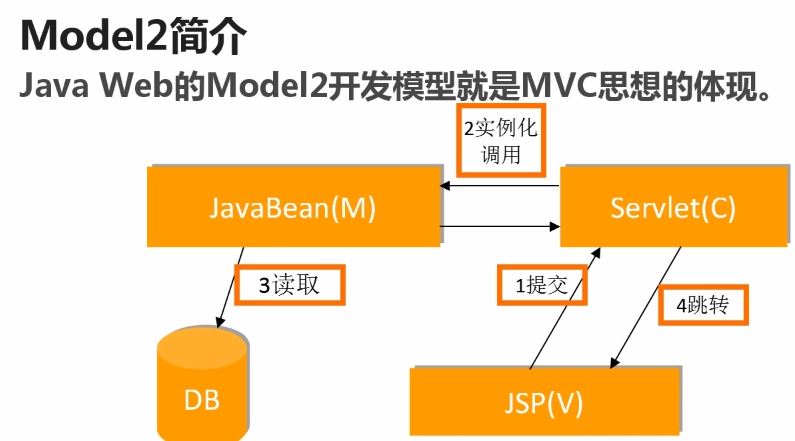
..表示返回上一层。页面中的/代表服务器根目录，web,xml中/代表项目根目录，请求重定向的/代表服务器根目录，请求转发的/代表项目根目录。

7:

Mvc模型：







8:在java项目中，只需要引入mysql-connector-java-5.1.7-bin.jar就可以运行java项目。在web项目中，当Class.forName("om.mysql.jdbc.Driver");时myeclipse是不会去查找字符串，不会去查找驱动的。所以只需要把mysql-connector-java-5.1.7-bin.jar拷贝到tomcat下lib目录就可以了。