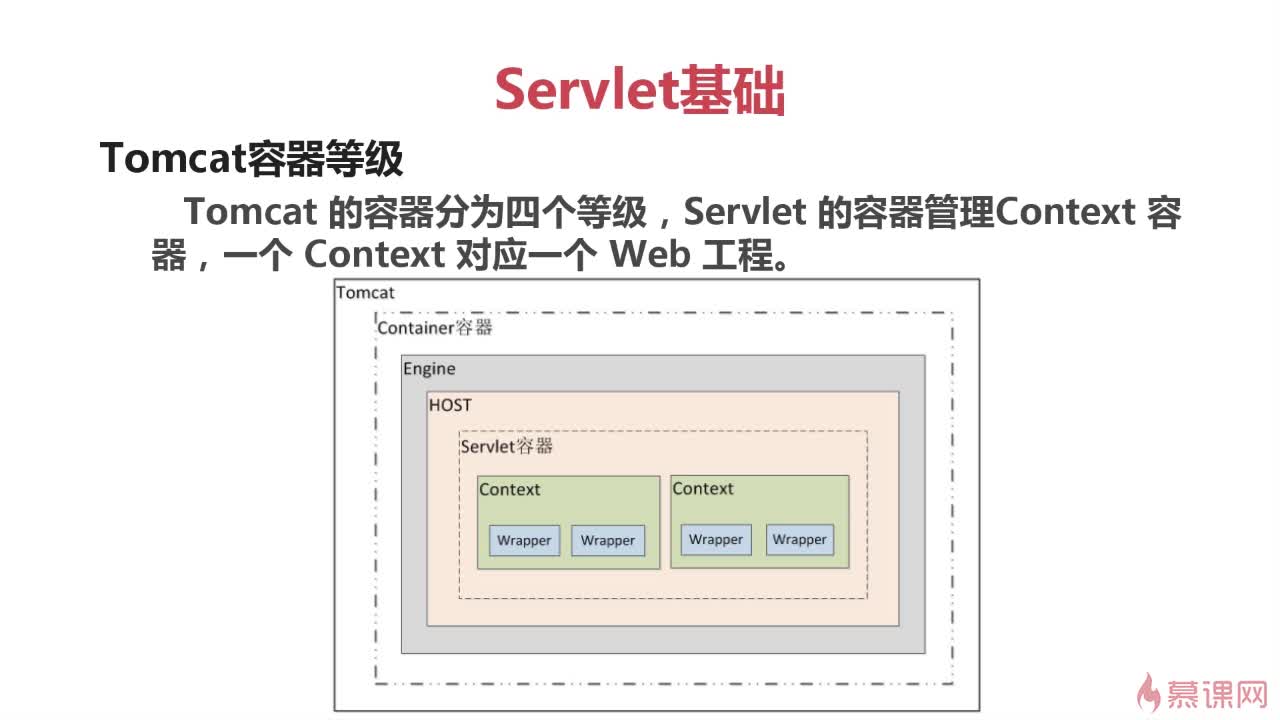
5一、Servlet基础

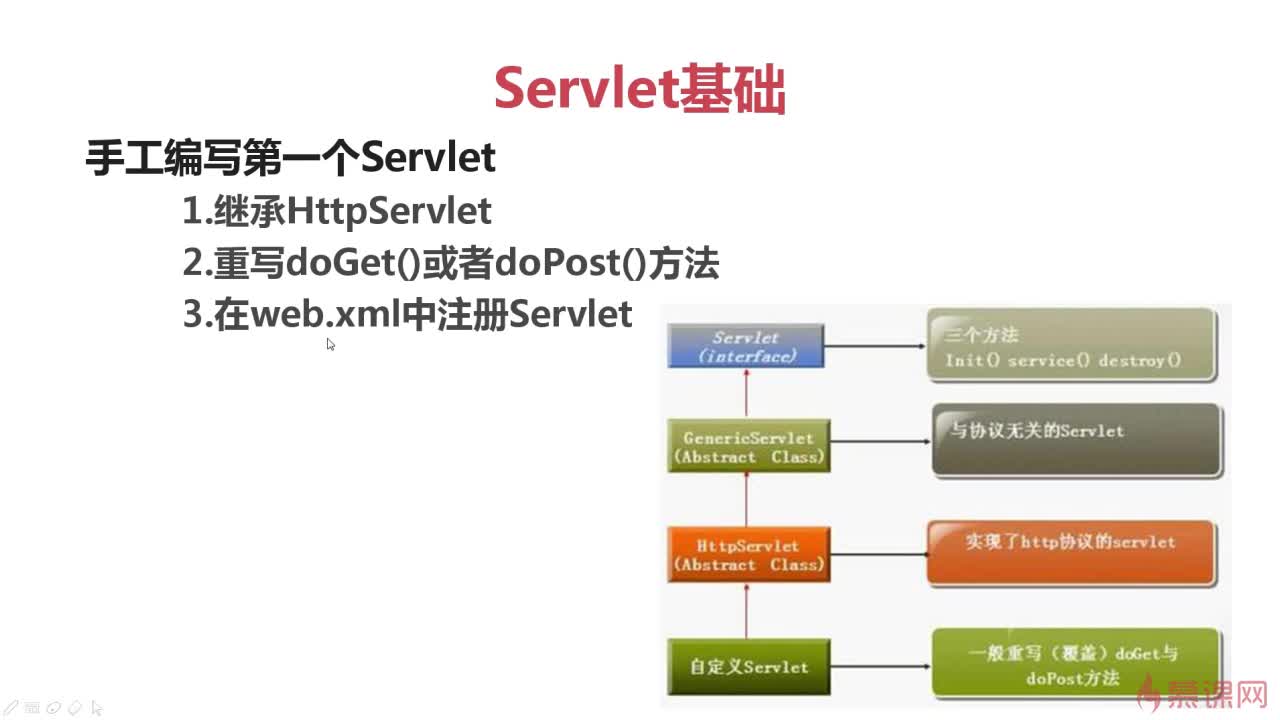
1、Servlet简介

* servlet是在服务器上运行的小程序。一个servlet就是一个java类，并且可以通过“请求－响应”编程模型来访问的这个驻留在服务器内存里的servlet程序。

2、Tomcat容器等级



3、手工编写Servlet



* 超链接使用的就是get方法。
* 表单提交时可以是post或者get方法。
* 在web.xml中注册：

<servlet>

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

<servlet-class>com.servlet.HelloServlet</servlet-class>

</servlet>

<servlet-mapping>

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

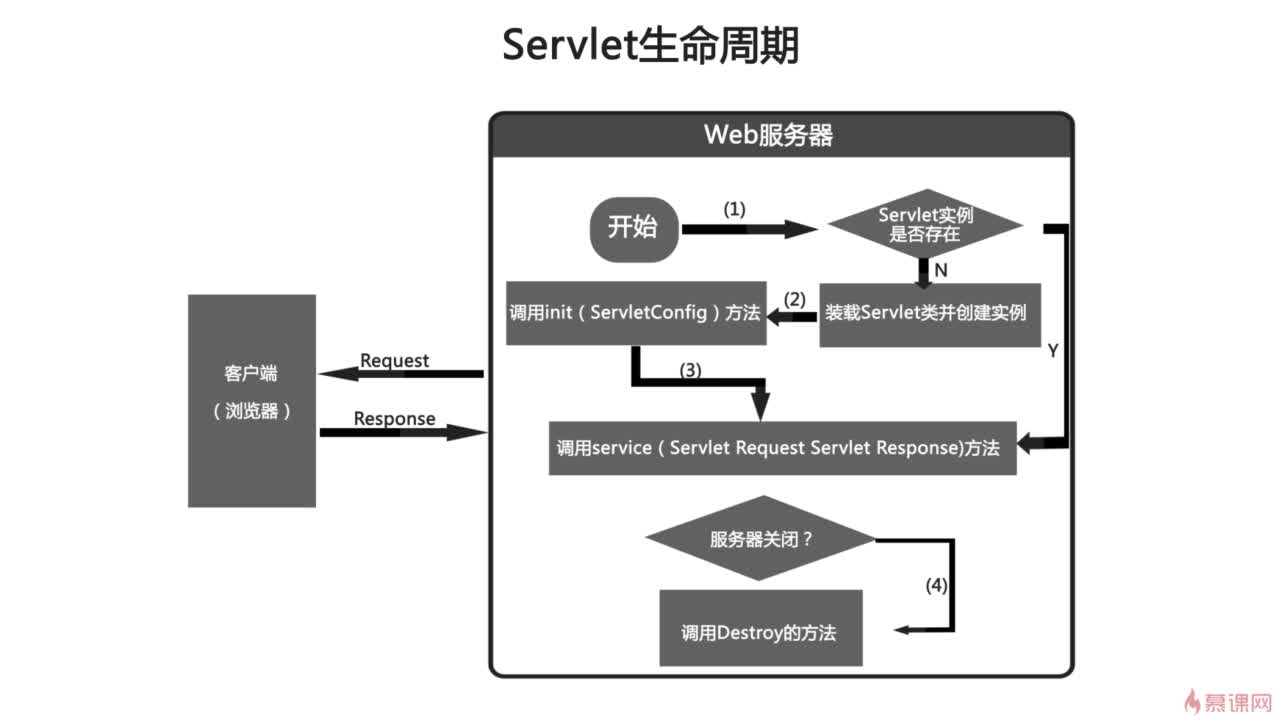
<url-pattern>/servlet/HelloServlet</url-pattern>

</servlet- mapping>

4、使用Myeclipse编写servlet



5、Servlet生命周期

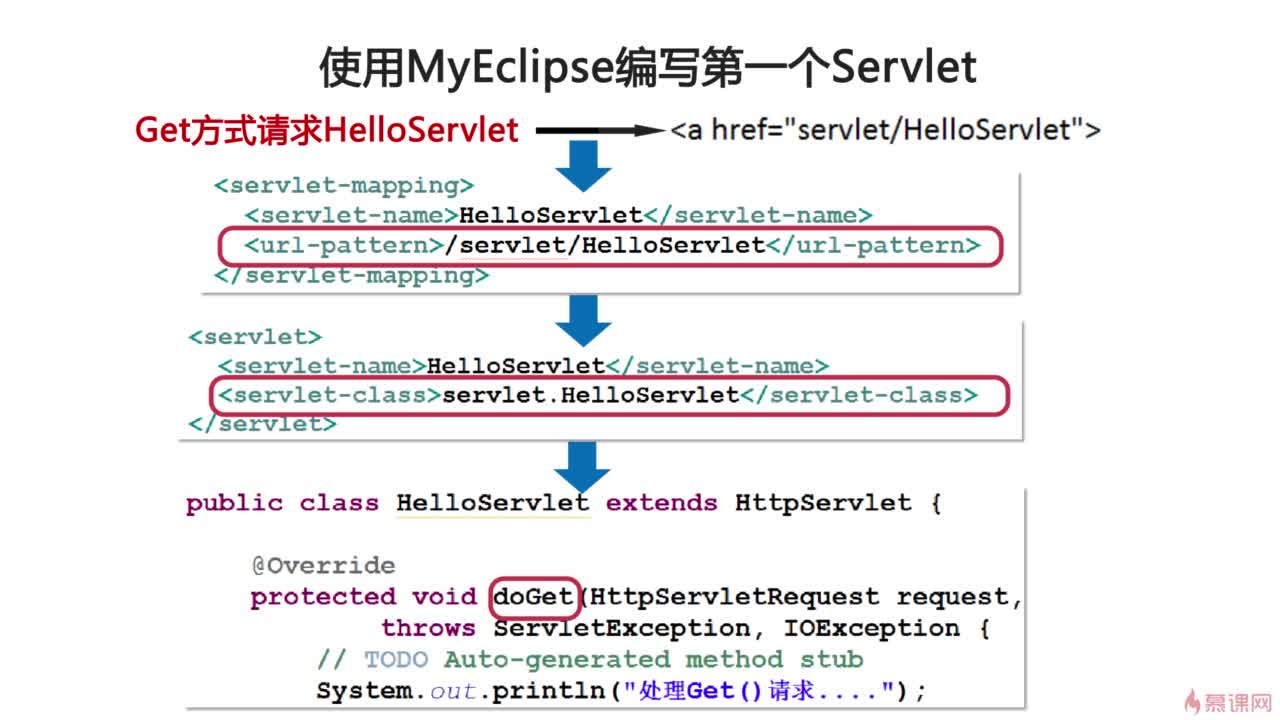


1.初始化阶段，调用init()方法。

2.响应客户请求阶段，调用service()方法。由service()方法根据提交方式选择执行doGet()或者doPost()方法。

3.终止阶段，调用destroy()方法。

* Servlet处理流程



6、Tomcat加载Servlet-- servlet是放置在服务器内存当中的。

* Servlet容器装载Servlet的三种情况

1. **自动装载**，在web.xml <Servlet></Servlet>之间添加<loadon-startup>1</loadon-startup>，数字越小，优先级别越高。

* 服务器启动时，执行Servlet构造方法和初始化方法。

1. 在Servlet容器启动后，客户首次向Servlet发送请求时。
   * Servlet构造方法→初始化方法→doGet方法→服务器关闭执行销毁方法
2. Servlet类文件被更新后，重新加载Servlet（长期驻留在内存当中）。

* 重新加载就是再执行一遍构造方法和初始化方法。

7、Servlet和JSP九大内置对象的对比



8、Servlet路径跳转

* 在JSP页面跳转到Servlet页面中
* 相对路径：

<!-- /servlet/HelloServlet 第一个/表示服务器的根目录 -->

<a href="servlet/HelloServlet">访问HelloServlet!</a><br>

* 绝对路径：

<!-- 使用绝对路径 访问HelloServlet,可以使用path变量:path变量表示项目的根目录-->

<a href="<%=path%>/servlet/HelloServlet">访问HelloServlet!</a><br>

* 在Servlet页面跳转到JSP页面中
* 请求重定向response：

//请求重定向方式跳转到test.jsp,当前路径是ServletPathDirection/servlet/

//response.sendRedirect("test.jsp");

//使用request.getContextPath获得上下文对象

//response.sendRedirect(request.getContextPath()+"/test.jsp");

* 服务器内部跳转request：

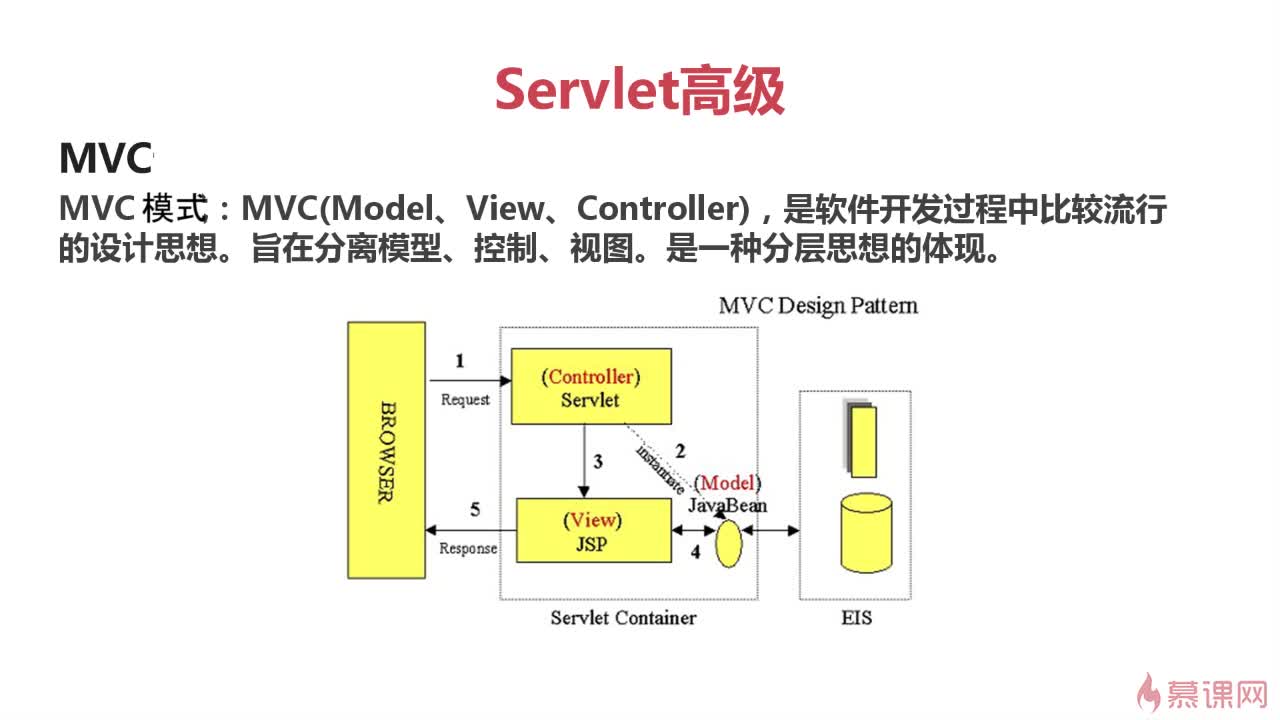
//服务器内部跳转,这里的斜线表示项目的根目录

//request.getRequestDispatcher("/test.jsp").forward(request, response);

//request.getRequestDispatcher("../test.jsp").forward(request, response);

二、应用MVC模式架构项目

1、MVC模型介绍



2、获取servlet初始化参数：

* 在web.xml中配置Servlet时，可以配置一些初始化参数。而在Servlet中可以通过ServletConfig借口提供的方法来取得这些参数
* 在web.xml文档中配置init的相关参数：

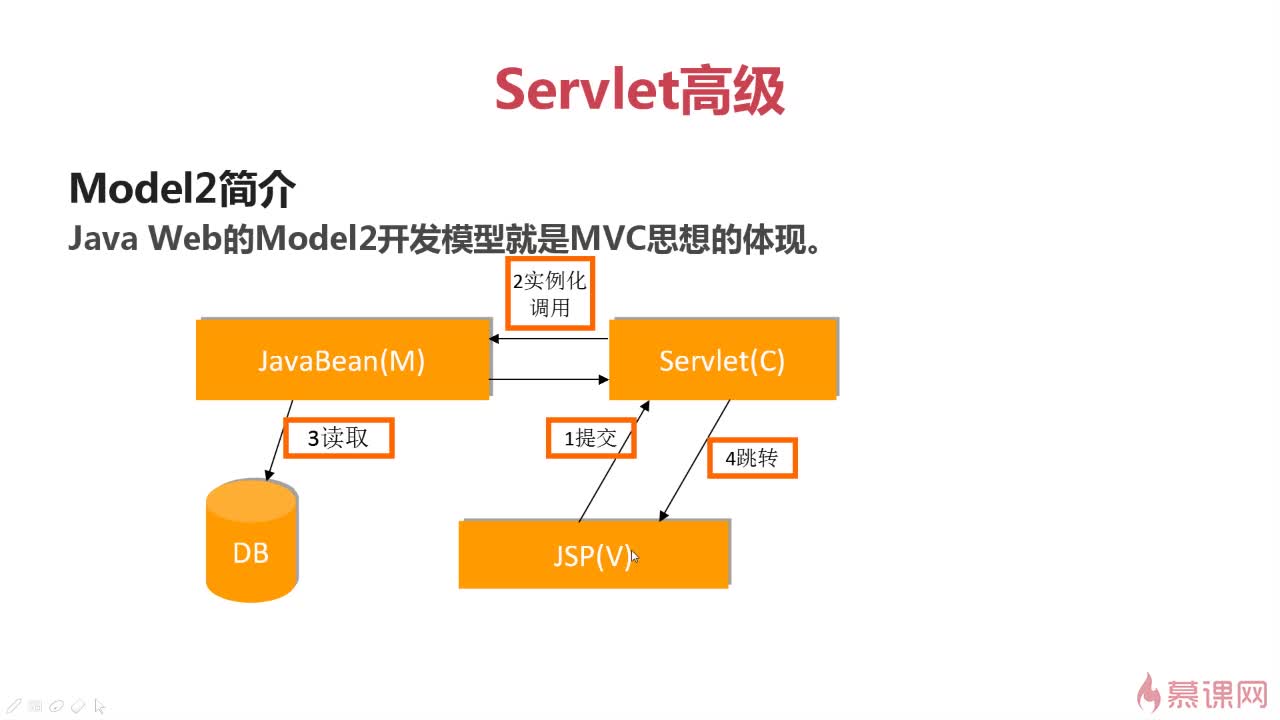
<init-param>

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

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

</init-param>

3、Model2模型介绍



* 该开发模型就是MVC思想的体现：

JavaBean（模型层）、Servlet（控制层）、JSP（视图层）、要访问的数据都封装在我们的数据库层，工作原理：

JSP页面给控制层Servlet提交一个请求 ，Servlet实例化模型层的对象 ，模型层访问读取数据库，得到结果反馈给控制层，控制层根据结果给用户呈现不同的页面

4、HashMap实现原理

put()方法实现：首先hash(key)得到key的hashcode()，hashmap根据获得的hashcode找到要插入的位置所在的链，在这个链里面放的都是hashcode相同的Entry键值对，在找到这个链之后，会通过equals()方法判断是否已经存在要插入的键值对，而这个equals比较的其实就是key。所以这里首先重写hashcode()方法找到相同的链，然后重写equals方法以保证key相同

* set/map中不重复添加相同的element:

解决方案:重写element的equals方法(hashcode方法)

原因:两个元素是否相同就是通过equals方法(hashcode方法)判断的，Object类继承的equals方法(hashcode方法)比较的是对象引用而不是对象内容，所以要重写equals方法(hashcode方法)

一般在设计类时，重写equals(hashcode)和toString,toString方法方便进行测试