# 回顾

1. tomcat:web服务器软件,也可以称其为容器,解释jsp代码;

tomcat安装和使用(启动,修改端口号,修改部署目录,修改启动时间 )

几个目录:bin:可执行文件 startup.bat(启动->window)

webapps:存放的是部署的项目文件

work:存放的是部署后生成的可执行文件(eg:java编译后的字节码文件等)

conf:tomcat的配置文件:server.xml->修改对外的端口号<Connector port=”9090”>

1. jsp:java服务端页面 :java+html;动态的网站技术,动态的数据交互
2. 处理数据:①设置请求的编码格式 :request.setCharacterEncoding(“utf-8”)

②:获取请求参数:String value=request.getParameter(“name”);

# 1.jsp

## 1.1 jsp的执行过程

当我们部署了一个项目的时候,这个项目会出现在:webapps

部署完成后,我们通过浏览器访问页面的时候:

此时会在tomcat的work目录下生成对应的xx.java和xxx.class,我们以下面为例:

访问: index.jsp

在work生成了index\_jsp.class index\_jsp.java

由以上过程我们可以分析得出:

Index.jsp(翻译 tomcat)-> index\_jsp.java(java编译器)-> index\_jsp.class执行结果响应给浏览器,你看到的是字节码执行的结果

## 1.2 jsp的运行原理

web容器处理jsp文件请求需要经过3个阶段

1. 翻译阶段 (把jsp翻译成.java的源文件)
2. 编译阶段 (把.java源文件编译成.class字节码文件)
3. 执行阶段 (解释执行.class字节码文件)

## 1.3 jsp的组成部分

|  |  |  |
| --- | --- | --- |
| 组成部分 | 说明 | 备注 |
| 指令 |  |  |
| 小脚本 |  |  |
| 表达式 |  |  |
| Jsp中动作(jsp:) |  | Inlude表示包含其他页面;  forward表示的是转发,直接转发到某个特定的页面 |
| Jsp的注释 |  | Html注释在查看源代码中看得到,jsp的注释看不到,就这个区别 |

# 2. servlet

概念：Servlet:运行在服务器端一段java代码;用来处理web端发送的请求;Servlet是运行时的jsp; MVC中的角色是控制器; 本质上就是创建一个Java类,这个类要继承 HttpServlet;重写doGet/doPost方法,声明Servlet的访问url : @WebServlet("/ArticleController")

Servlet是.java的源文件,文件后缀名就是xx.java;

## Servlet与jsp区别:

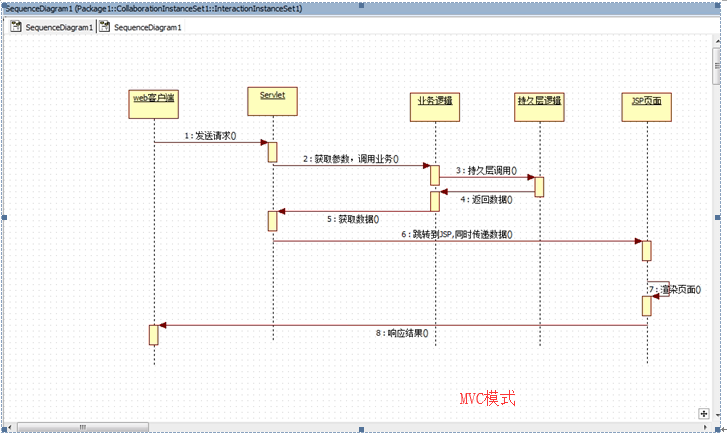
servlet和jsp是一样的,都可以做数据的展示,但是如果servlet做数据的展示,代码量非常的大;

jsp程序->tomcat翻译之后,形成一个xxx.java文件,其实就是一个servlet文件

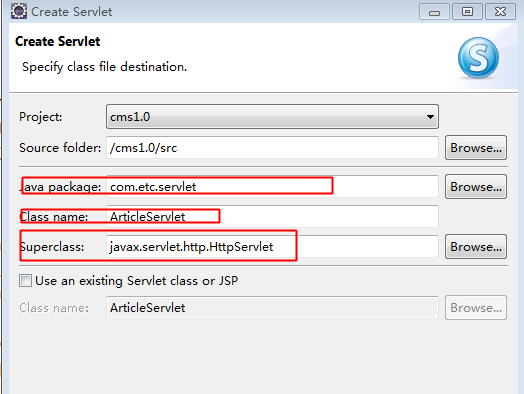
所以jsp程序运行起来之后就是一个servlet文件,本质上jsp=servlet

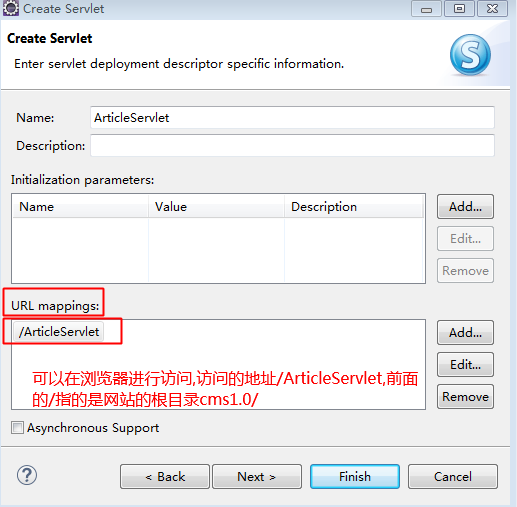
Sun:先有servlet(做接收用户请求与展示页面->展示太复杂),后面又推出了一个新的程序jsp->数据的展示(用户交互的页面)

以后,我们的servlet主要用来接收请求,做页面的转发(定向)处理,不做页面的显示,作用相当于一个调度器



### 1.6.1如何新建一个servlet

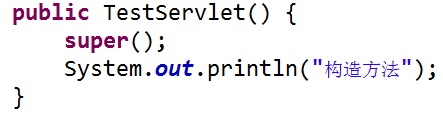




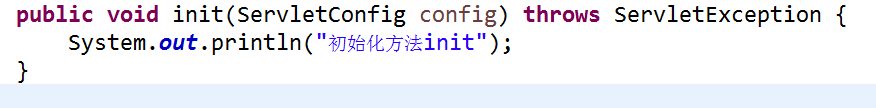


### 1.6.2 servlet的生命周期

实例化servlet对象

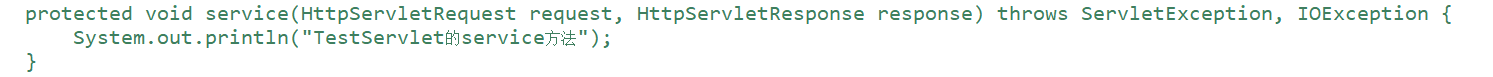


初始化(实例化servlet对象后就只初始化一次)



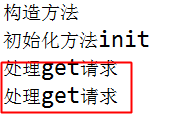
执行服务(多次),请求一次就执行一次

(访问<http://localhost:9999/cms1.0/ArticleServlet>)

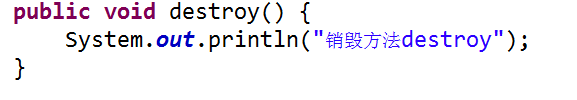


当我们的service方法被注释掉后,servlet根据具体的请求执行具体的get或post方法(如果是浏览器直接访问,这种方式为get请求)





当容器停止(重新启动),对象:执行销毁的方法



### [面试题]servlet的生命周期

1当用户第一次访问某个servlet程序的时候，tomcat会调用servlet的构造方法生成servlet对象。

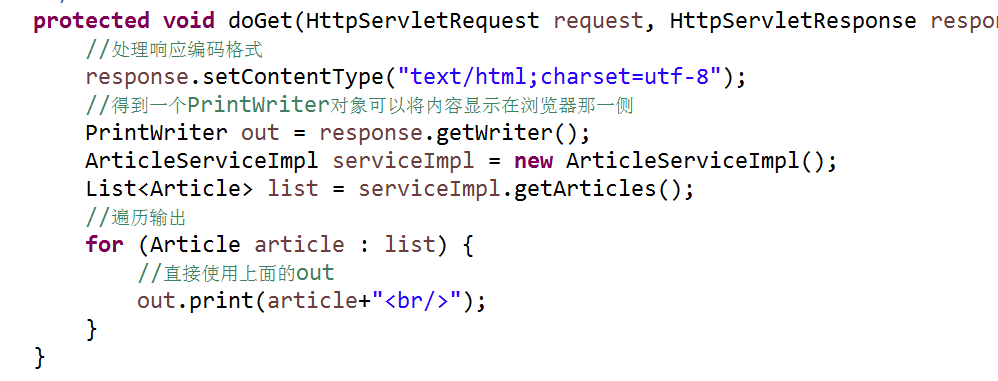
2生成对象以后，tomcat容器又调用init()进行初始化。

3调用service方法，由service根据请求的类型是get、还是post类型.post类型的调用doPost方法。get类型的调用doGet方法。

4当tomcat容器关闭后，调用destory()销毁对象。

当以后调用时，直接调用service方法即可。即从步骤3开始即可。

### 1.6.3使用Servlet显示数据



访问的页面:

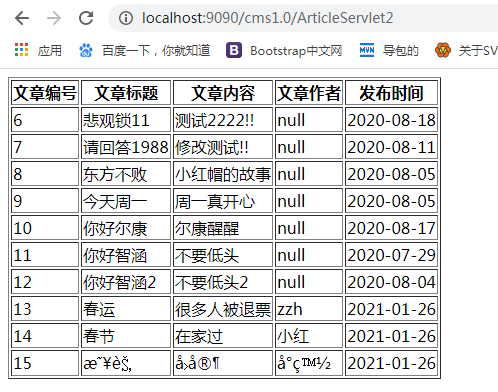


## 课堂任务:

1. 熟悉servlet的生命周期,能描述清楚
2. 在servlet中展示article表的记录,要求显示为表格格式(行 列 标题)

我们发现,上面的servlet只是实现了一个表格的数据展示而已,但是代码量非常多,界面的效果也不是很好

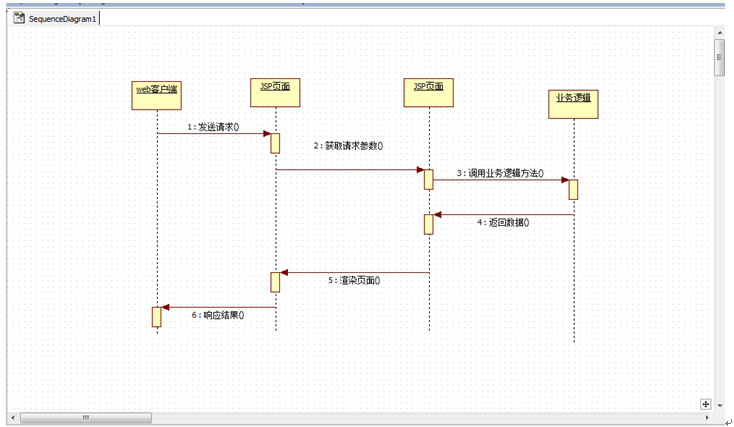




## JSP(Model)

### Model1模式：

Model1模式中，分为两层，视图层和模型层。其中jsp就充当了项目中的业务流程，所有的业务逻辑都是由jsp来处理的，JSP页面身兼View和Controller两种角色，将控制逻辑和表现逻辑混在一起，通常是通过jsp直接调用模型来处理相关业务，model1以jsp为中心。



Model1架构模式的优缺点

(1) 优点：架构简单，比较适合小型项目的开发。基本上使用JSP+持久层就可以完成一个小型项目的开发。

(2) 缺点：JSP页面有控制页面显示的代码又有与后台业务逻辑交互的Java代码，职责不单一，页面负担很重，影响页面运行速度；而且不方便后期的维护。如果不加选择地随意运用Model 1，会导致JSP页面内被嵌入大量的脚本片段或Java代码，特别是当需要处理的请求量很大时，情况会更为糟糕。

Servlet和jsp最主要不同点在于:servlet的应用逻辑是在java文件中,并且完全从表示层中的html里分离开来.而jsp是java和html组合成一个扩展名为.jsp文件

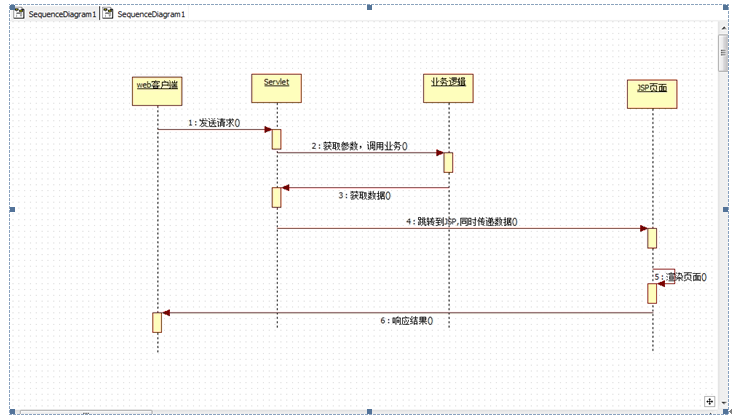
### Model2(重要)

Model 2架构模式是一种把JSP与Servlet联合使用来实现动态内容服务的方法，所以Model2就是JSP+Servlet+JavaBean模式。它吸取了两种技术各自的突出优点，使用JSP生成表达层的内容，使用Servlet完成深层次的处理任务。在Model 2中，Servlet充当控制者的角色，负责管理对请求的处理，创建JSP页面所需的JavaBean和对象，同时根据用户的动作决定把哪个JSP页面传给用户。特别要注意，在JSP页面内没有处理逻辑，它仅负责检索原先由Servlet创建的对象或JavaBean，然后从Servlet中提取动态内容插入到静态模板中进行页面显示。所以简单来说，在Model 2中，JSP进行页面显示，JavaBean进行逻辑处理，Servlet完成逻辑控制。

Model2架构模式的优缺点

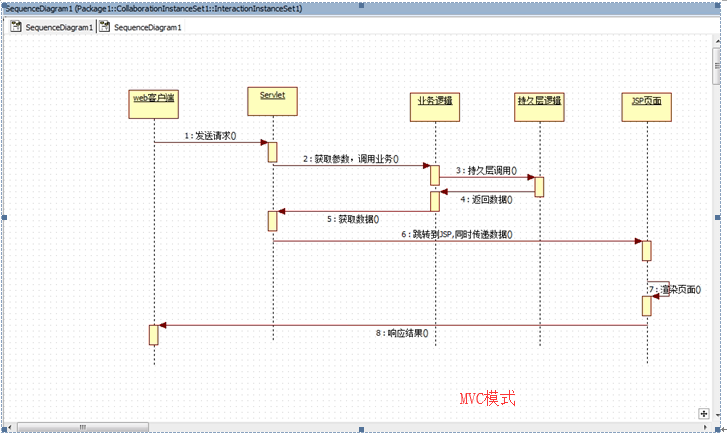
(1) 优点：清晰地分离了表达和内容，明确了角色的定义以及开发者与网页设计者的分工，适合大型项目的开发。(前后端分离)

(2) 缺点：分层过多，不适合小型项目的开发。



# 3三层架构(MVC):

将Model2中的数据访问部分再次抽离出来得到持久化逻辑层，可以得到经典的三层架构模型。如下图所示。



(1) 以上为非常经典的三层结构模型：

表示层：JSP/Servlet

业务逻辑层：业务规则 service

持久化层：主要是包装持久化逻辑 Dao

(2) 抽取出持久化层的好处：主要是为了持久化层的改变不会影响业务逻辑层，从而更好地划分职责。

(3) 表示层、业务逻辑层、持久化层是自上而下的依赖关系，而且通常是单向依赖，最好依赖于抽象的东西。

### mvc

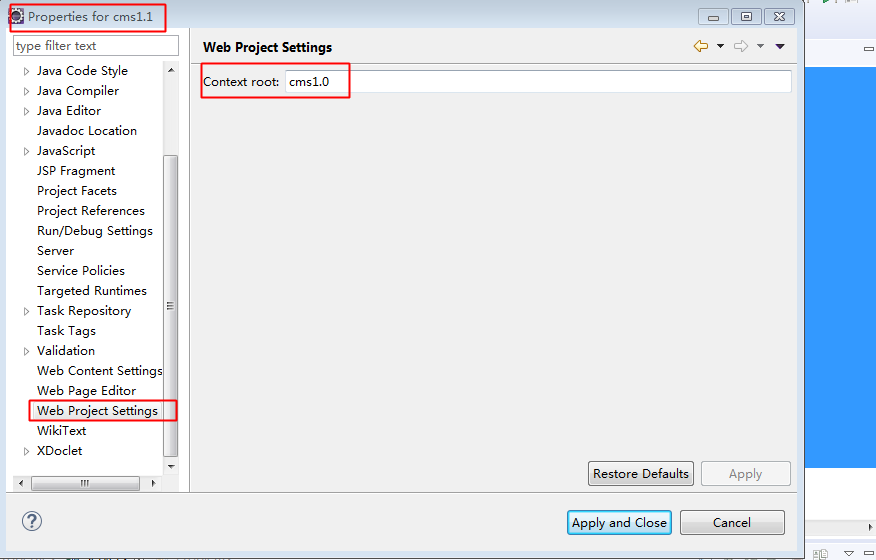
model模型: 业务,持久化等部分

view视图: jsp以及前端的一些技术等等

controller:控制器 servlet(第二阶段) Springmvc(第三阶段) Struts2

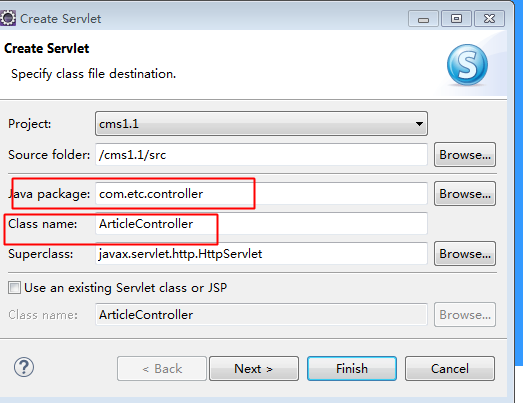
## 1.9 cms1.1实现mvc显示Article列表

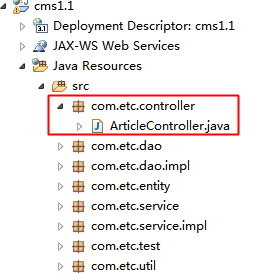
当复制一份工程的时候,最好把上下文路径改成跟新的工程名一致,因为访问的时候还是按以前的工程名访问,不同的tomcat版本可能会引起404错误.



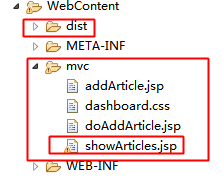
改成cms1.1

新建一个控制器









showArticles.jsp:





之前是直接调用service,现在是从请求范围内获取文章集合对象信息

当我们访问的时候:

<http://localhost:9999/cms1.1/ArticleController>

**是控制器的地址,控制器调用业务层,业务层调用dao层,然后获取数据,再将数据结果存储到request的作用域中,然后转发到showArticles.jsp页面,最后jsp页面从request作用域中获取对象集,做显示.**

## 1.10 cms1.1实现mvc新增文章

问题1:之前提交表单给jsp页面,而现在应该是提交给控制器servlet

问题2:提交到servlet控制器中,控制器又有查询代码,现在又要做增加业务?

解决:跳转的时候,传递一个参数op=add

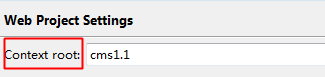
Servlet中有所改动,通过传递参数判断该做什么业务

此时的请求过程是:

<http://localhost:9999/cms1.1/ArticleController?op=query>

## 1.12上下文路径

关于上下文路径:



获取:

addArticle.jsp

## 总结:

1. jsp的执行过程:翻译(.jsp->.java) 编译(.java->.class文件)
2. Jsp的组成部分:指令 小脚本 注释 表达式 动作
3. Servlet是什么?Servlet的生命周期
4. Mvc架构:: m->model(service dao) v->view(jsp等前端技术) c->controller(servlet)
5. 实例和应用:cms系统 1.1版本:采用mvc架构实现新增和查询

## 作业:

1. 完成cms系统 1.1版本:采用mvc架构实现新增和查询用户
2. 完成login的controller