## 回顾java+jdbc

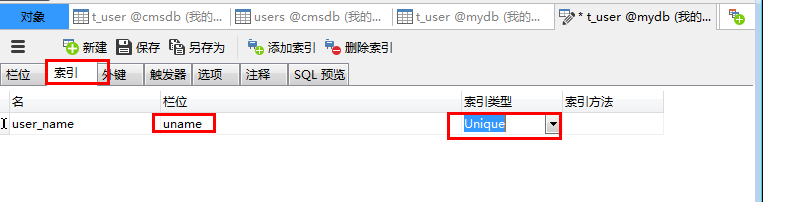
# 1.分层思想:

1.实体层 2.持久层 3.业务层 4.表示层

**持久层:** 数据库的访问:例如dao 返回值给业务层

**业务层:** 业务逻辑的处理 调用持久层代码,传递数据给持久层,返回数据给表示层

例如:用户注册:获取要注册的用户信息,然后去数据库表中判断这个用户信息是否存在,如果存在,提示用户:已经存在用户名了,请更换一个重新注册



例如:用户上机 下机

上机:用户是否存在?密码是否正确?余额是否足够,判断要使用的机器是否空闲…

我们dao单纯一些,将这些逻辑独立分成一个层.

**表示层:**和用户交互,swing的界面,用户登录,注册窗口; 调用业务层(要传递数据给业务层) 得到业务层数据,响应给用户

调用关系: **表示层<🡪业务层<->持久层<->访问数据库**

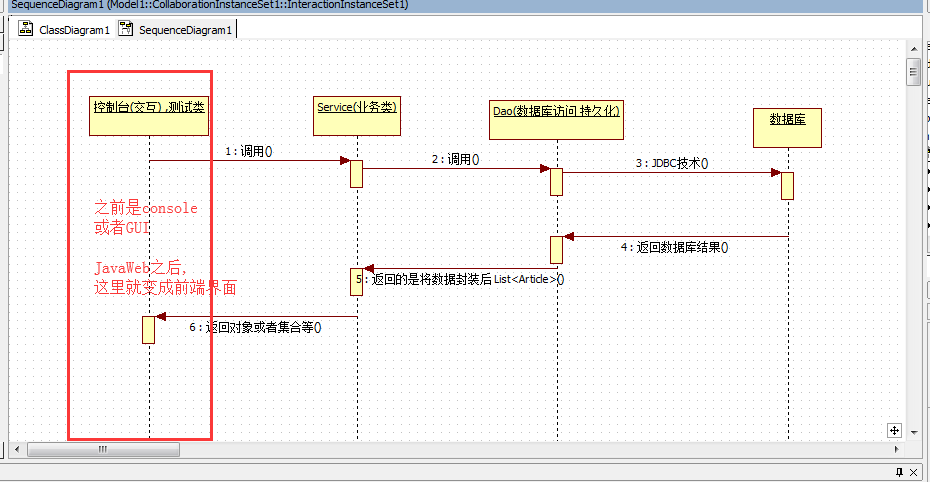
比喻:

持久层:生产商

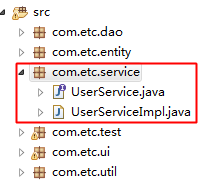
业务层:经销商(商家)

表示层:用户

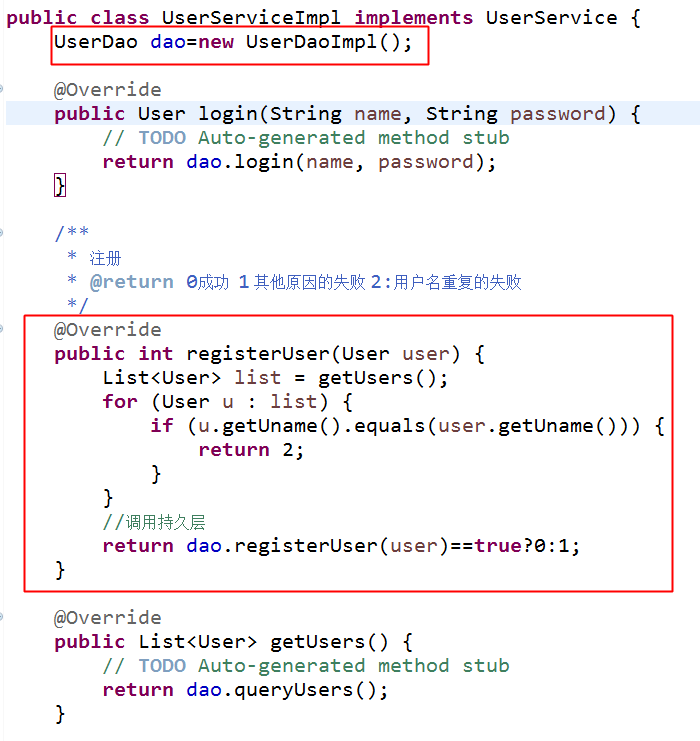
表示层->业务层->持久层



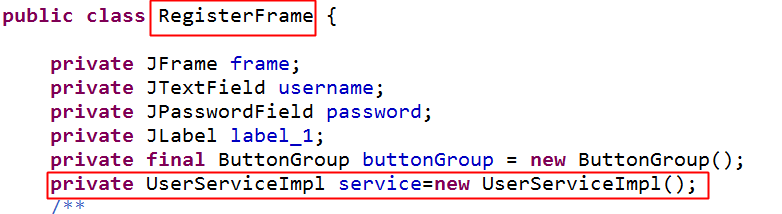
UserService:







注册页面:





我们增加一个需求,除了Mysql以外,再支持一种数据库:Oracle;

我们同时希望考虑程序的扩展问题,后期维护的问题,所以

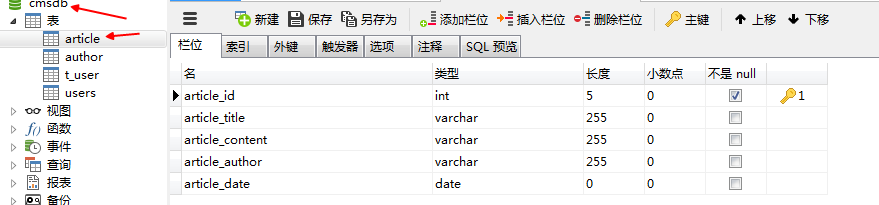


将Dao->IDao接口和一个Dao的一个实现类;

将Service->Service接口和一个Service实现类;

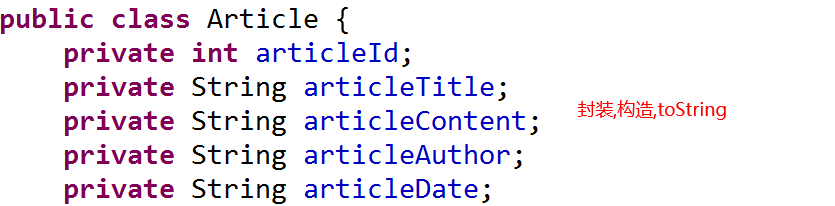
# 2.文章管理为例 cms1.0

1.建立数据库,建立表,插入一些测试数据

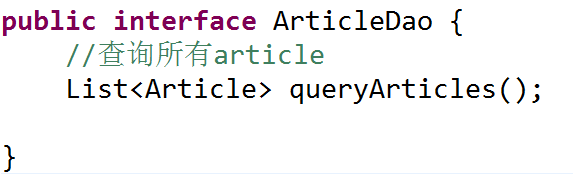


2.Java访问数据库代码:copy DBUtil与导入mysql-connection…jar包

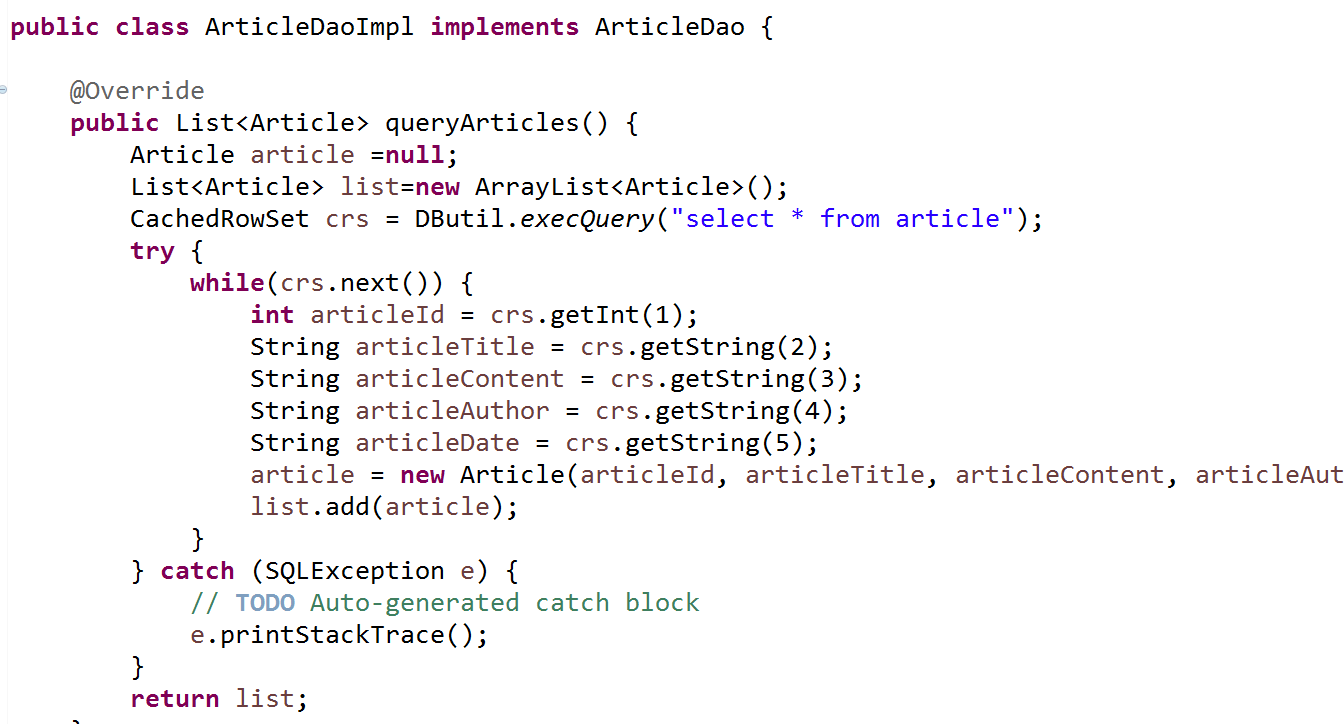
3.根据表结构创建实体类



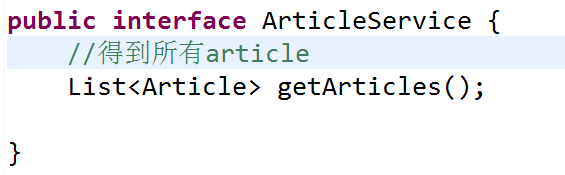
4.根据实体类创建dao接口与dao的实现类 查询所有



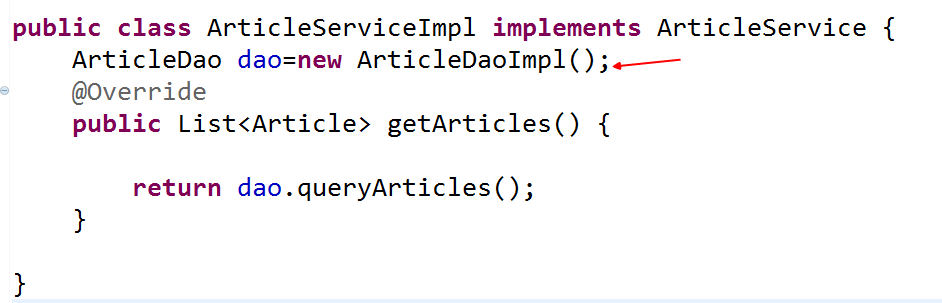
实现类:



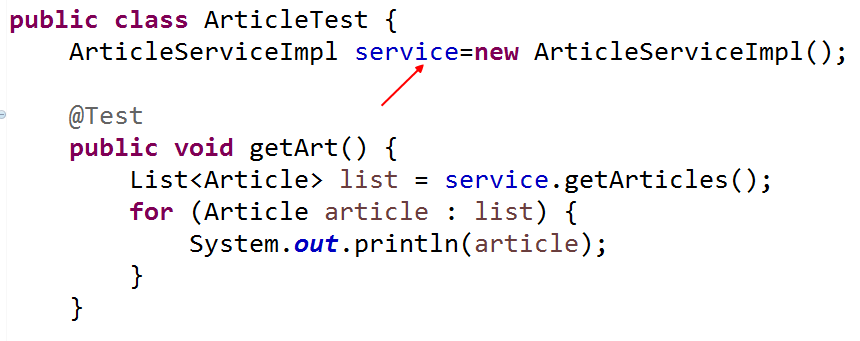
5.根据dao创建service业务层与实现类,调用dao实现类



实现类:

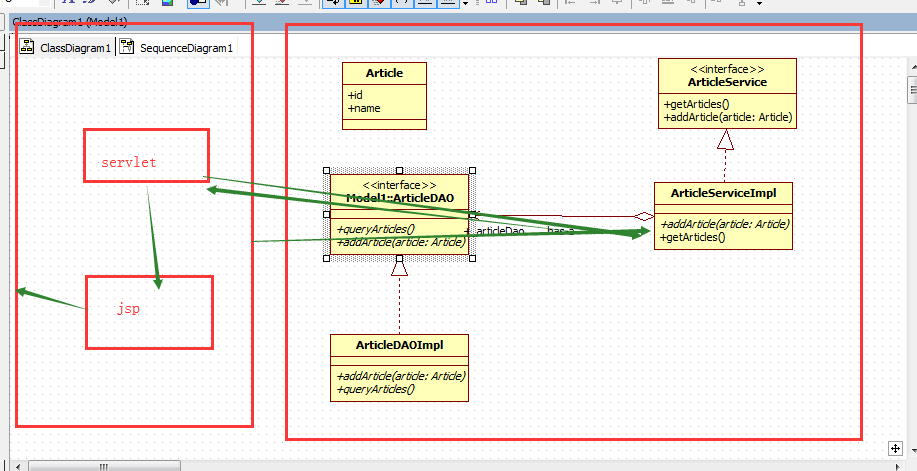


6.创建测试类,调用业务层实现类



代码包结构:





## 2.1 JavaWeb工程

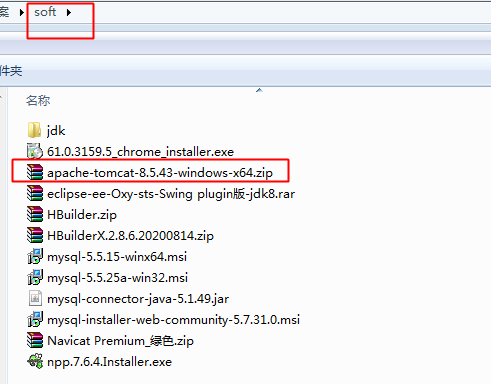
HTML页面:静态的网站

动态的网站:可交互,直接:可以和数据库交互

JavaWeb工程是B/S(Browser/Server )浏览器/服务器架构:包含前端+后端

Runtime:就是一个运行在主机(windows Linux)系统 上的对外提供服务的软件

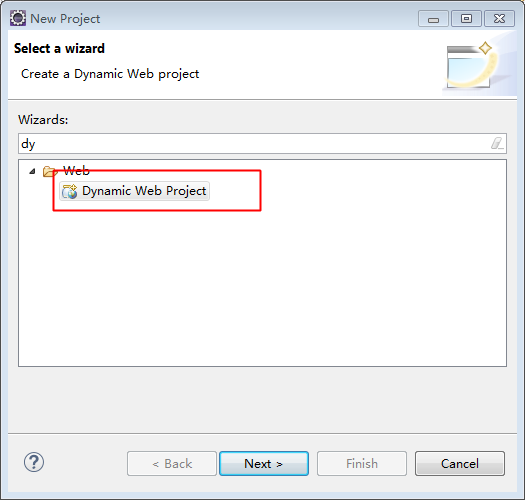
TomCat:



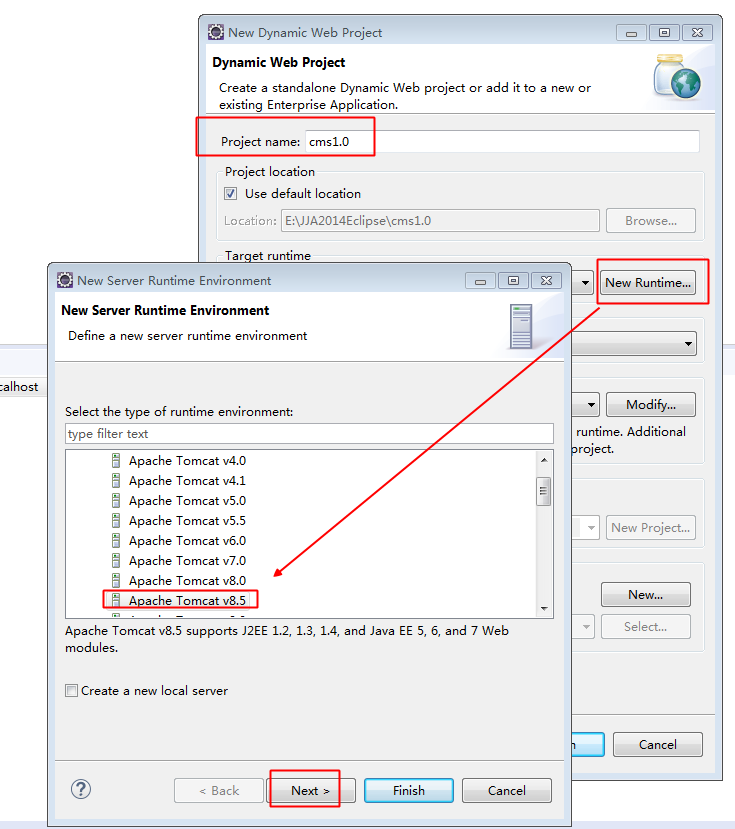
直接解压就可以了,

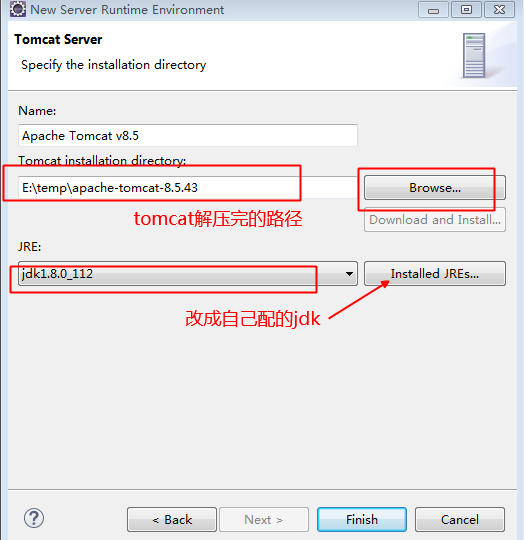
创建web项目:

直接搜web/dy选择

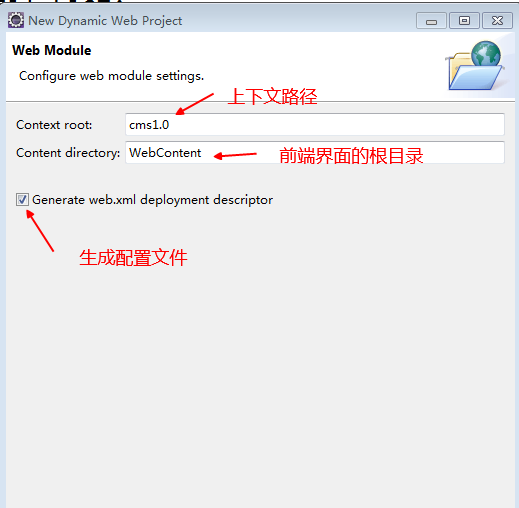


## 2.2 在javaWeb工程中配置tomcat

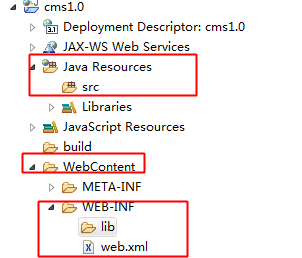




finish后一直点next,直到:



### 目录结构:



以上的目录结构:

java resources:java的源代码,通常将java代码存在在src目录下

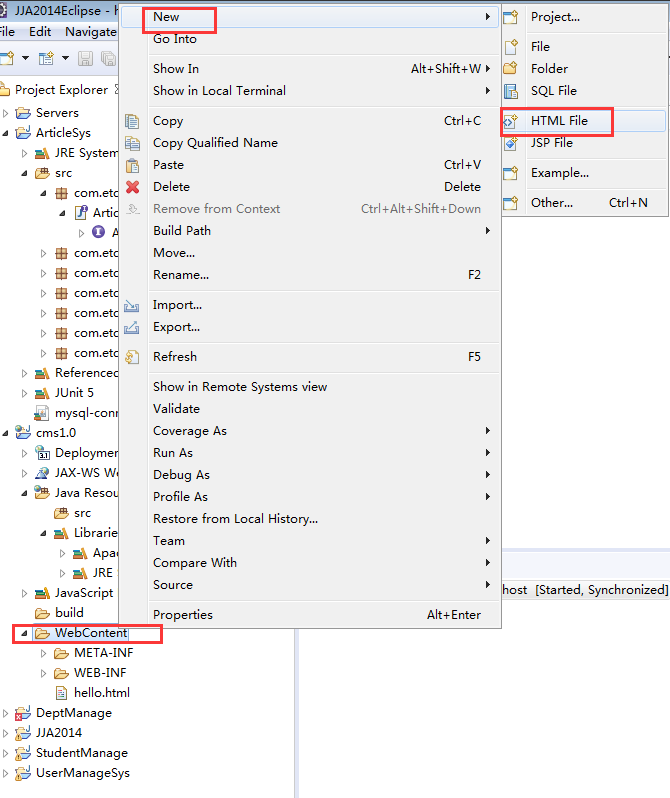
Webcontent:网站的根目录,主要包含了xx.jsp/xx.html/x.js/xx.css….(静态网页,动态网页…..)

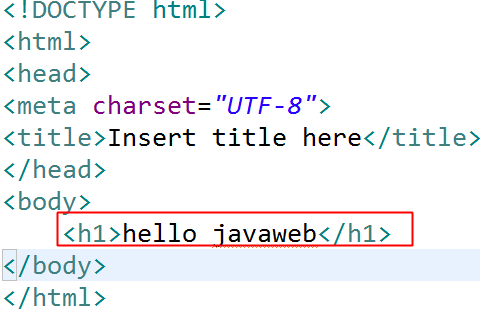
Web-inf:这个目录比较特殊,这里的文件不能通过浏览器直接访问

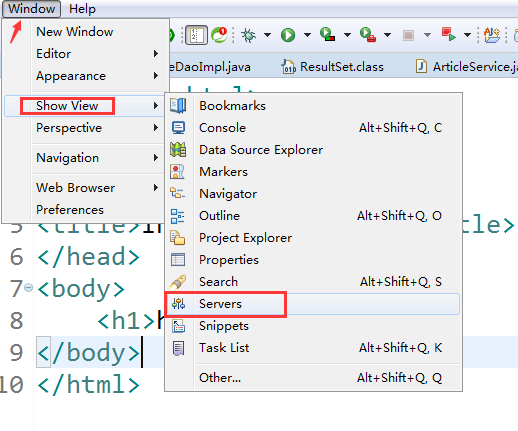
lib目录:存放jar包,直接复制外部的jar到该目录下,自动添加到构建路径中(最好手动添加->add to build )

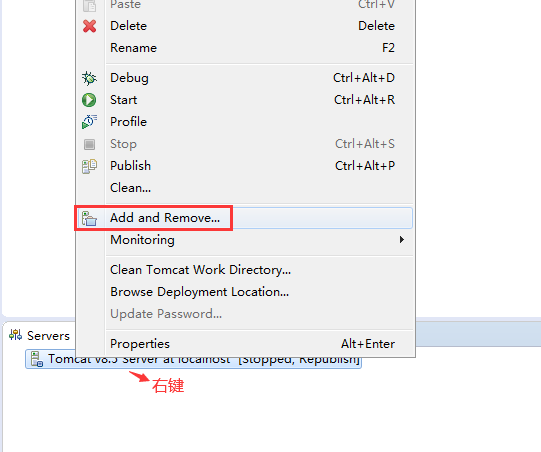
Web.xml:配置文件

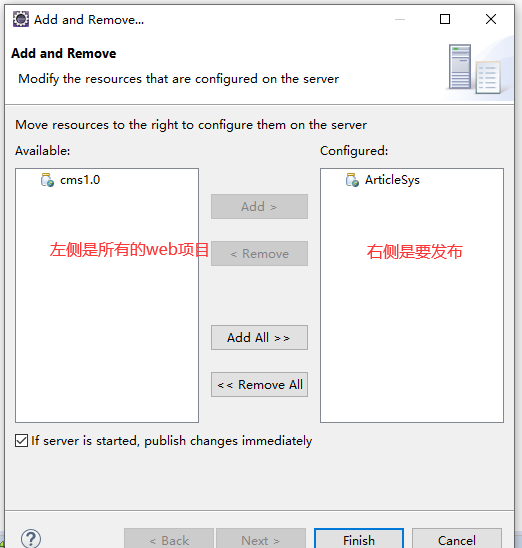
## 2.3运行web项目











http://localhost:8081/cms1.0/hello.html

协议://ip地址:端口号/上下文路径名/文件名



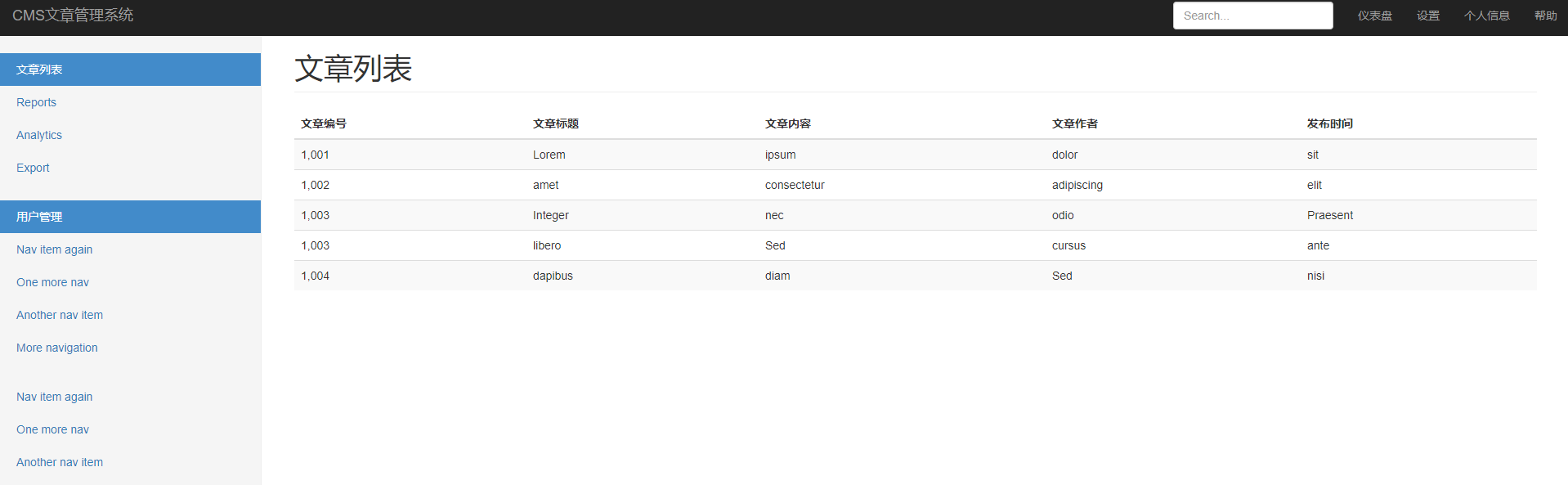
任务:创建一个html页面,写个简单的自我介绍,发布后,截图到群里,建议你们用好看一点的模板

IMG_256http://192.168.17.71:8081/cms1.0/hello.html

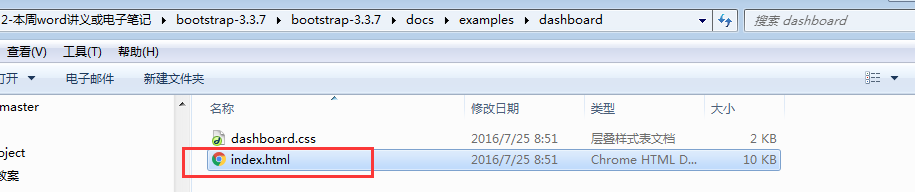
# 作业

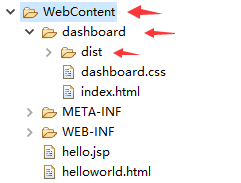
1.修改页面





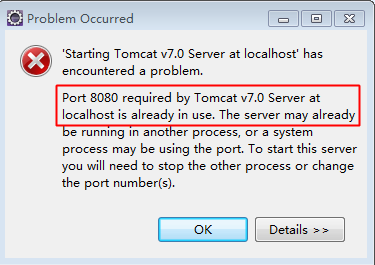
模板:

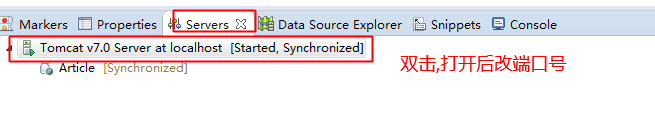


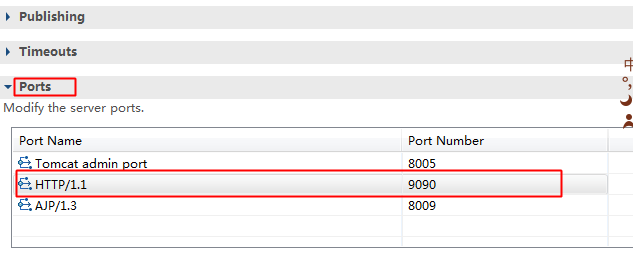


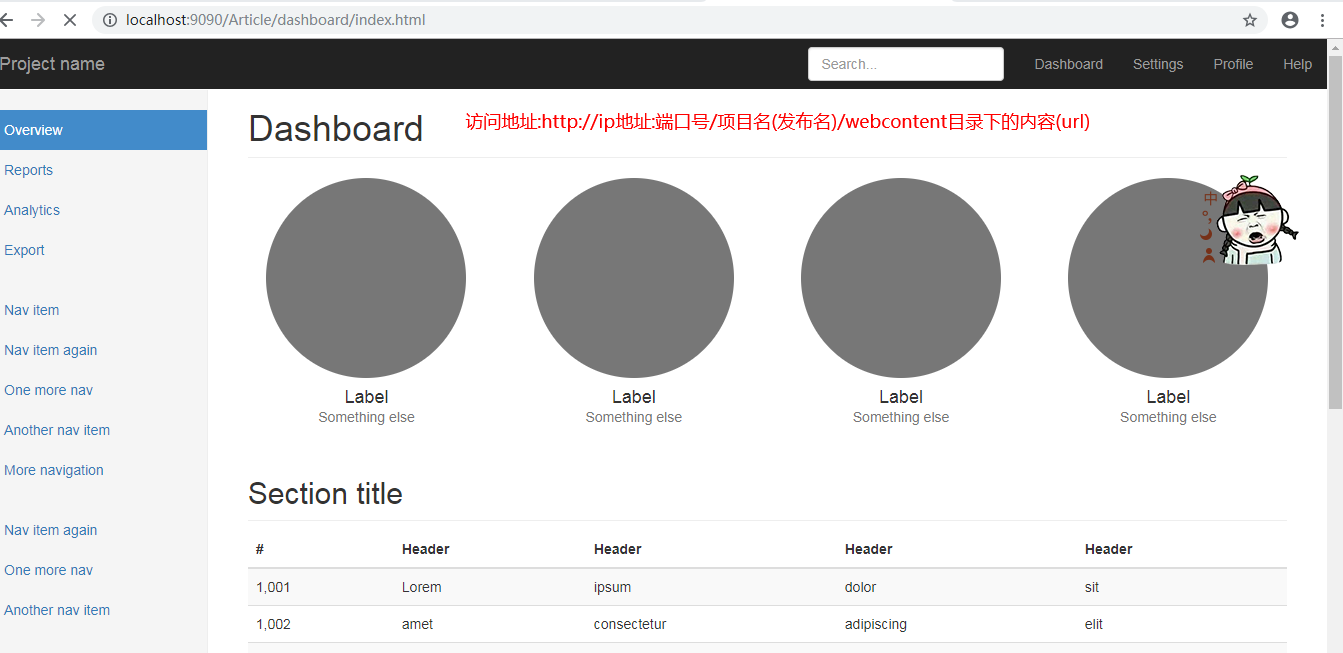
2.实现增加功能(后端)

### 如果出现端口号冲突:









弄出来的这个地址就是提供给别人用

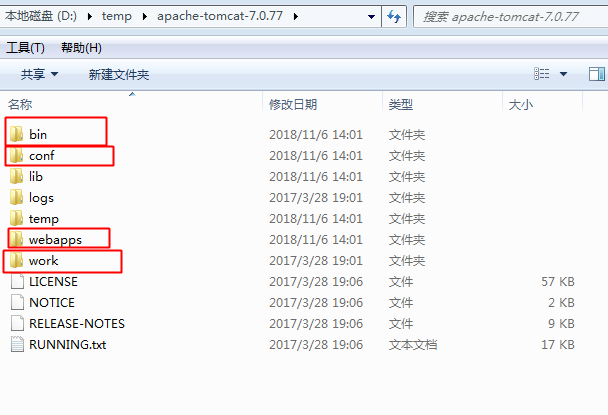
<http://localhost:8888/ArticleSys/dashboard/index.html>

任务:

在eclipse中新建动态的web工程,同时配置tomcat,必须要熟练

## 2.4 Tomcat

是一只猫,是一个软件,这个软件能够对外提供服务,web服务器软件,Tomcat单独使用?



Bin:有很多可执行文件或者批处理文件



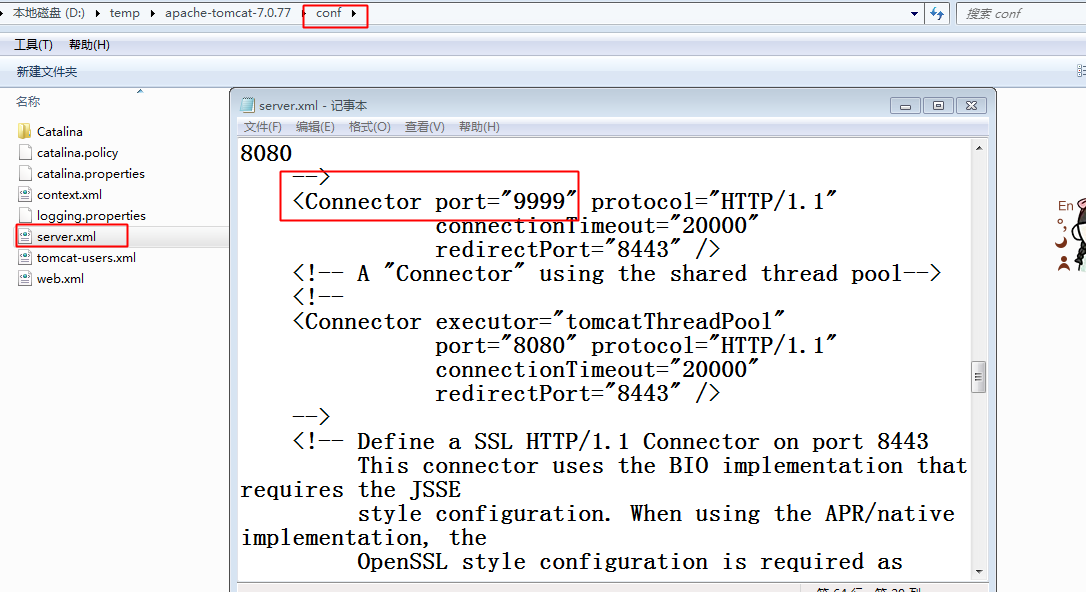
如果启动startup.bat,一闪而过,我们可以使用命令来测试->(Eclipse的服务关掉/JAVA\_HOME)

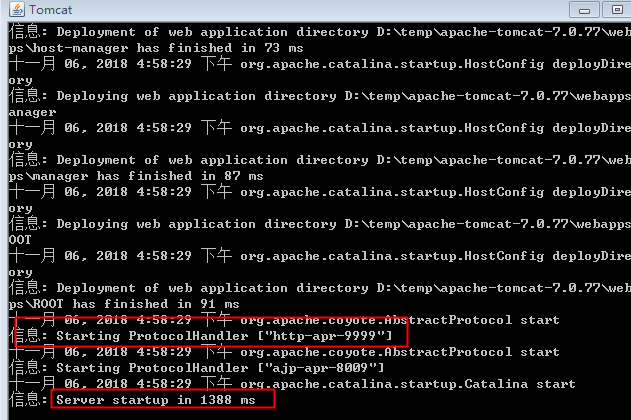




如何改端口号:

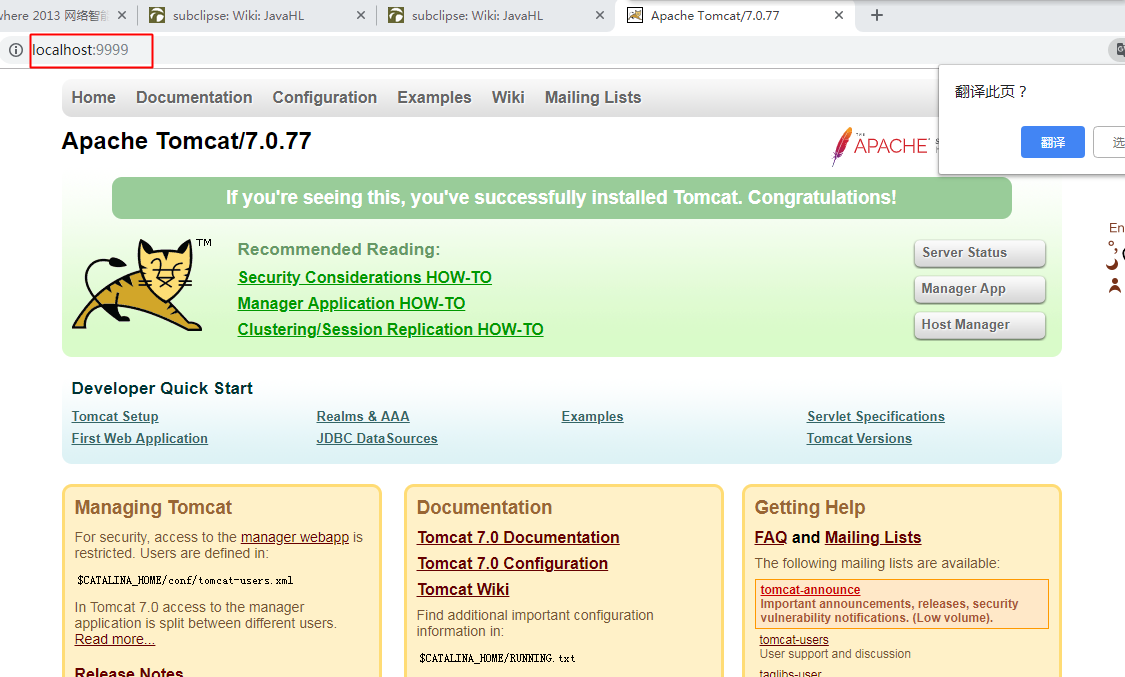
<Conf:配置文件的目录>



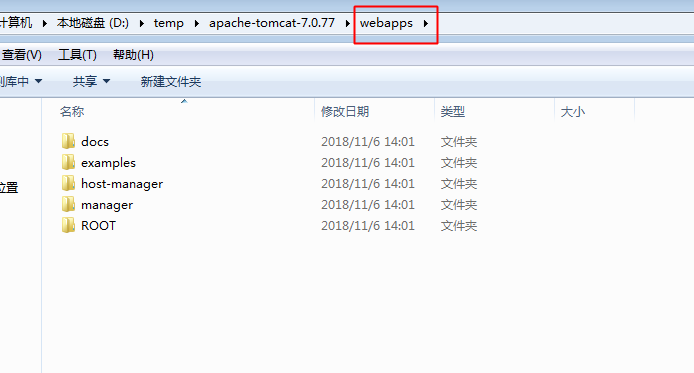


窗口要留着

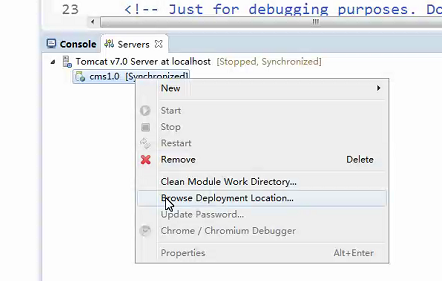
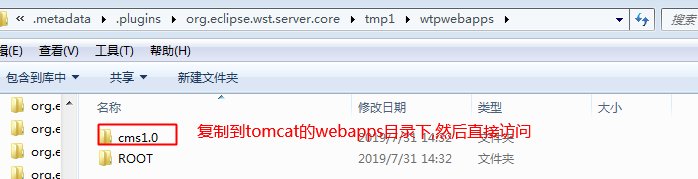
Localhost:9999



Webapps:存放的是部署之后的项目代码



拷贝编译后的工程,直接访问: 编译后的工程(eclipse中servers下面已经部署后的项目)

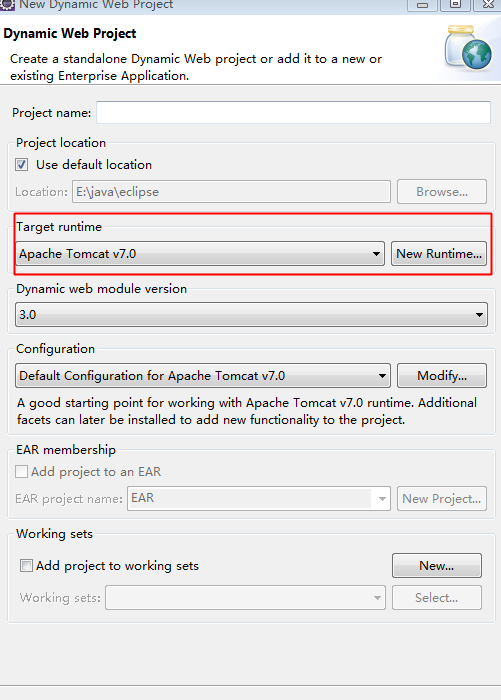
 

<http://localhost:9999/cms1.0/dashboard/index.html>

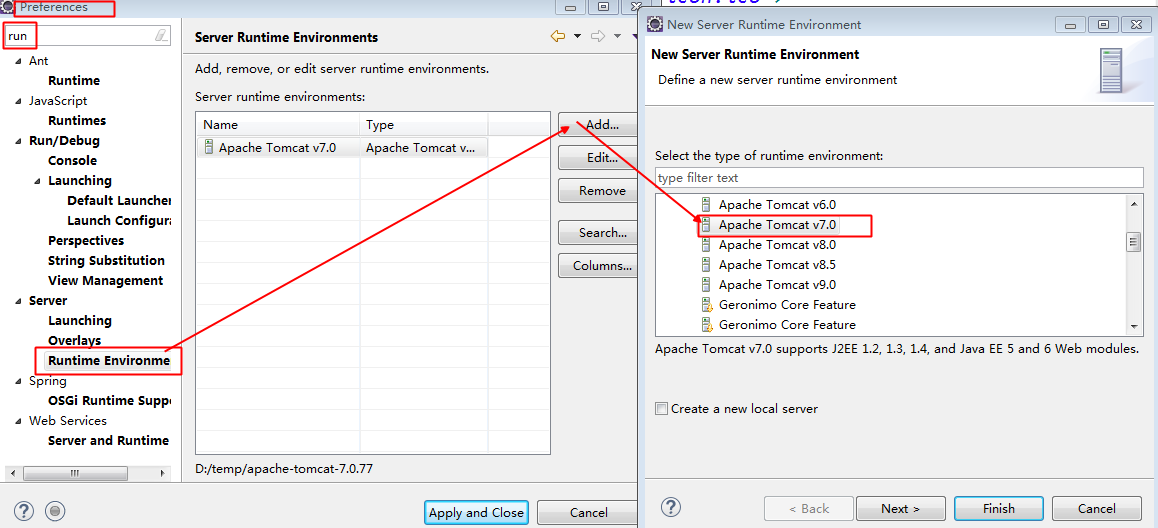
## 2. 5 Eclipse中配置tomcat

有很多种:

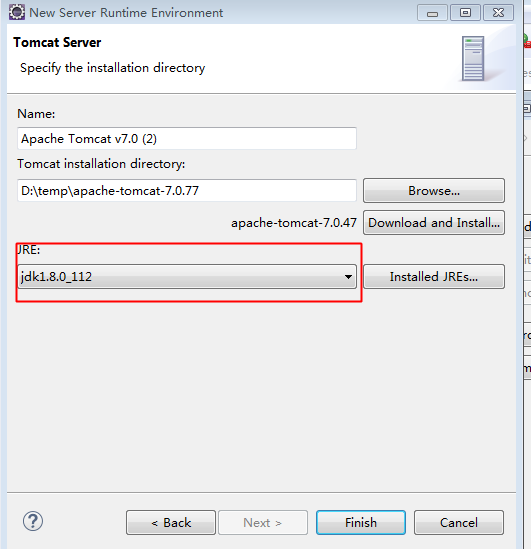
1. 新建web工程时直接配置tomcat



1. 在菜单项目:window->preferences->runtime



配置tomcat的时候最好指定一下jdk

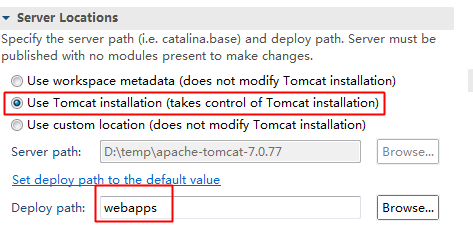


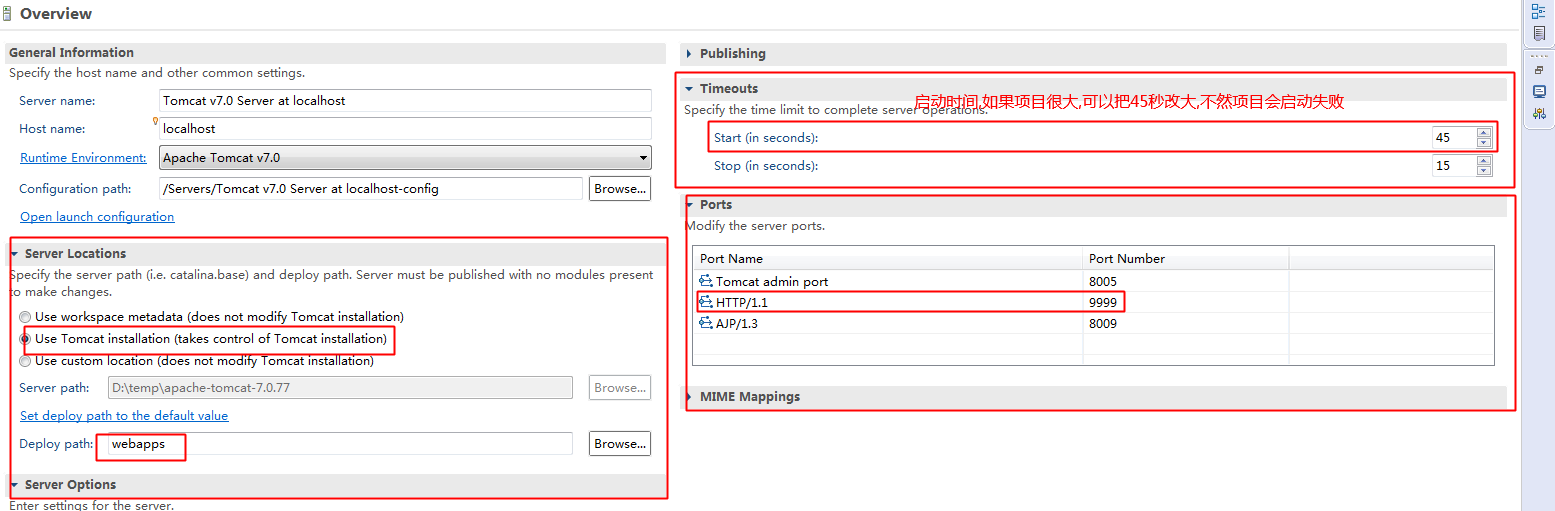
## 2.6管理tomcat

window->show view->servers

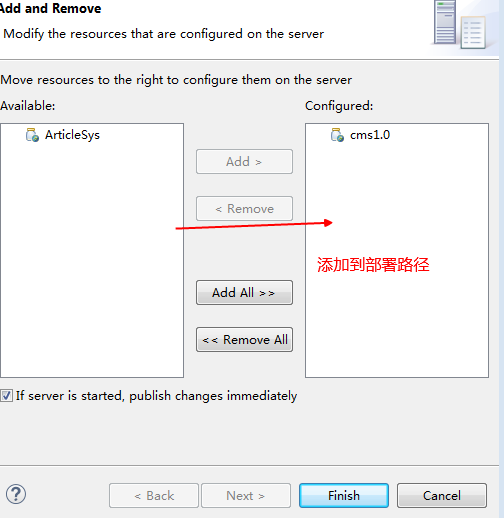
管理端口,启动,停止,设置部署的目录等等

server locations:部署项目存储在哪个目录,默认选中第一个,我们需要改成第二个:tomcat的路径,如果显示灰色无法修改,就移出项目,再delete,再新建就可以修改了





如果我想要在tomcat中跑这个项目的话,就要复制一份编译好的工程到tomcat的webapps里面去,虽然我们自己复制黏贴很快,但是eclipse中,他可以自动帮你做这件事,这个动作叫做部署



2种方法:

1. 直接找到.html页面右键 run as on server
2. 管理tomcat的地方:servers下面,右键add and remmove

## 小结:

Tomcat:web服务器软件,也可以称其为容器,解释jsp文件,熟悉tomcat的安装(直接解压)和使用

Bin:可执行文件 (startup.bat->windows系统) startup.sh(linux,mac)

Webapps:存放的是部署的项目文件

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

Conf:tomcat的配置文件:server.xml:修改tomcat对外服务的端口

<connector port=”9090”>

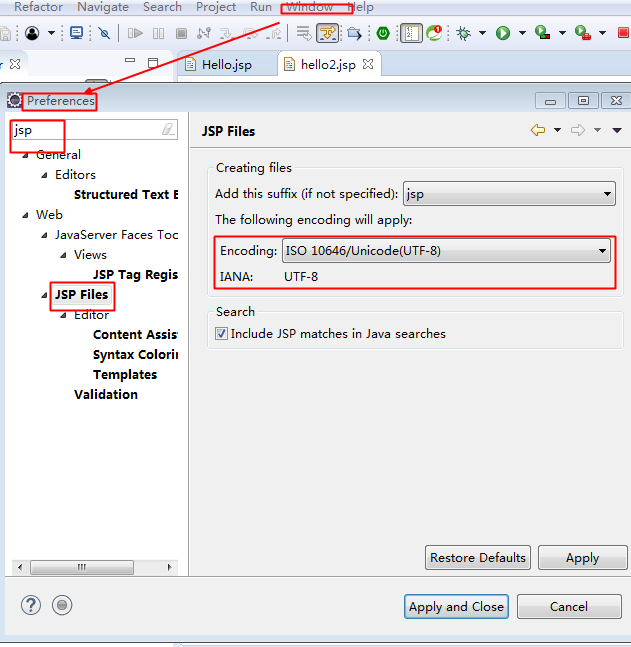
# 3.jsp: java Server Pages

java服务端页面:java+html(等静态页面部分),动态的网站技术,动态的数据交互(执行数据库的访问)

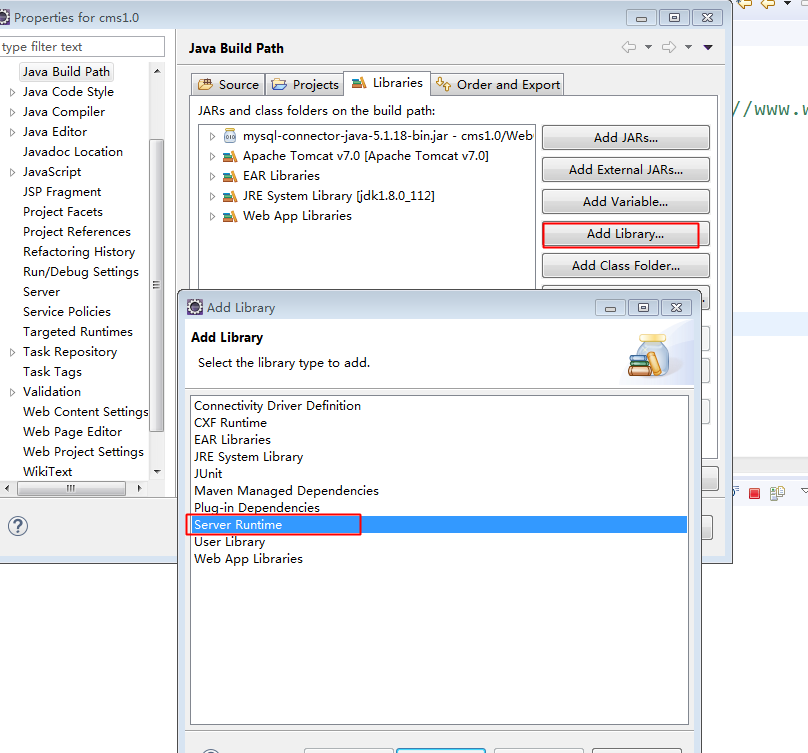
## 1指令

<%-- <%@ 指令名 属性="值" language="java":设置页面的语言 pageEncoding="ISO-8859-1":当前jsp页面的编码->改成支持中文 utf-8 %> --%>

## 2 修改默认编码



记得在项目中加入tomcat



## 3小脚本

<%这里写java代码 %>

## 4表达式:主要做页面输出

<%= %>

## 将article表中的记录在页面显示

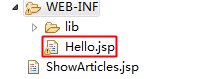
## 5.两个访问的http信息

1.404:url访问的资源不存在(找不到)

协议://域名:端口号/上下文路径(发布的工程名)/目录/某个文件



2.注意:web-inf下的文件不能直接访问





404还会有其他原因:路径不对,项目未部署

3.500服务器的内部错误,大多是因为java的代码引起,例如我们之前的java中看到各种异常(空指针…)都会引起500错误.

500错误的调试:大家要结合出错的页面和控制台

控制台看不懂,我希望大家能将错误信息记录下来,方便大家查阅

异常:需要加driver,编译器版本的问题:



## 将文章列表和之前的后台静态页面整合到一起



## 增加文章

首页:

addArticle.jsp:

doAddArticle.jsp:

## 总结:

1.分层:表现层<->业务层<->持久层 (ui<->service<->dao<->db)

2.新建web项目配置tomcat

3.tomcat的安装和配置,将项目部署到tomcat服务器上

4.在Eclipse中配置tomcat

5.管理Tomcat:默认的部署路径,端口,启动时间

6.jsp:java服务端页面:动态的页面技术:jsp常见的组成部分:指令;小脚本<%java代码%>,表达式:<%=%>:页面显示,直接输出

7.后台与前台结合

## 作业:

两个功能实现: 列表 新增->自己建立表 ,Dao接口->Dao实现类 ;Service接口->实现类->页面;

