

marquee实现一行文件闪过屏幕

```
<?xml version = "1.0" encoding = "utf-8">
```

在web项目中，查看自己项目的编码方式：右键需要查看的文件名--->properties就可以查看

dtd约束不严谨

xml用于编写配置文件

语法严格，区分大小写，用于存储数据，xml的标签都是自定义的标记语言

如果出现JVM_bind的话，表明是8080端口被占

WEB-INF下面的文件不能被直接访问，

发布的时候不会看到webroot目录，发布的时候只需要将webroot里面的东西放在Tomcat中的webapps下面新建的文件夹下就可以啦

lib: 该web项目所需要的架包全部放在这里

web.xml: 该web项目的配置文件，配置一下默认主页

classes: src下java文件生成的字节码文件

5、部署应用到 Tomcat 服务器

> 开放目录部署方式

把应用直接复制到 F:\apache-tomcat-7.0.52\webapps 下。

> 把应用打成 war 包。

打 war 包命令: jar -cvf MyApp.war .

把 war 包直接复制到 F:\apache-tomcat-7.0.52\webapps 下，应用自动解压

注: webapps 目录下有几个目录就代表有几个应用。

Tomcat 服务器与 MyEclipse 集成(一定要掌握)

1、Tomcat 集成

2、创建 web 应用

3、部署

URL: 统一资源定位符 (网址)

URI: 统一资源标识符

URL里面包含有URI(当前应用的资源路径)

http://主机IP: 端口号/URI

端口被占用问题:

解决方案1: 在Tomcat服务器的conf中的servers.xml中修改connector端口号

解决方案2: 在命令行输入netstat -ano, 找到端口为8080的进程, 启动任务管理器, 在查看选项中添

加pid列, 找到对应8080端口所在进程的应用, 结束任务即可

当项目过多时候, 需要将一些项目放在其他的磁盘路径下, 这时候就需要配置虚拟路径:

虚拟路径配置方式1: (不建议使用, 需要重启服务器)

在conf下的servers中的Host标签中写入一个标签:

<Context path = "/访问时的虚拟目录" docBase = "项目的真实目录"

重启服务器后, 输入http://localhost:8080/虚拟目录名的访问文件名

虚拟路径配置方式2:

①新建myApp.xml文件, 文件名就代表应用的虚拟目录名

在其中写入:

<?xml version="1.0" encoding="utf-8"?>

<Context docBase="项目的真实路径">

②将新建的文件放在: Tomcat服务器路径下的

conf\Catalina\localhost目录下

访问时: <http://localhost:8080/myApp>

配置默认端口, 默认应用, 默认主页

①默认端口

把server.xml中

<Connector port = "80" protocol = "HTTP/1.1"

connection Timeout = "20000 "

redirectPort = "8443" />

浏览器http默认端口80

②默认应用:

把配置的虚拟目录的配置文件名改为ROOT.xml

③默认主页

修改当前应用的web.xml，添加一下内容

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="2.5"
3     xmlns="http://java.sun.com/xml/ns/javaee"
4     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5     xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
6     http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
7     <display-name></display-name>
8     <welcome-file-list>
9         <welcome-file>index.jsp</welcome-file>
10        <welcome-file>1.html</welcome-file>
11    </welcome-file-list>
12 </web-app>
13
```

http协议：超文本传输协议，传输HTML文件

用于定义web浏览器与web服务器之间交换数据的过程及数据本身的格式，包含请求和响应

1，请求部分：

①请求消息行

a， 请求方式：get（默认），post，delete，head等

get：明文传输，不安全，传输数量有限，一般不超过1kb

post：暗文传输，安全，传输数据量没有限制

b， URI：同一资源标识符，去掉协议和IP地址（主机加端口号）

c， 协议版本

抓包软件（适用浏览器：IE浏览器）httpwatch.zip，左侧是请求部分，右侧是响应部分

②请求消息头

第一行与空行之间都是请求消息头

content-type默认提交方式都是string（字符串）类型

③请求消息正文

空行以下的东西是请求正文

2，响应部分

①响应消息行

协议/版本 响应状态码 对响应码的描述（ok表示一切正常）

200（正常） 一切正常

302/307（临时重定向）

304（未修改，表示从客户机缓存的版本是最新的，客户机可以继续使用它，无需到服务器去请求）

304表示从本地获取数据，查看lastmodify的时间，如果时间没发生变化，则表示从本地获取数据

404（找不到）服务器上不存在客户机所请求的资源

500（服务器内部错误）

②响应消息头http

③响应正文

和网页右键查看源码所看到的内容一样