## 学习Servlet背景

原理：之前开发web，是使用Servlet开发，进行接受请求参数，数据共享，页面转跳的，但是使用Servlet比较麻烦，那么就想mybatis对JDBC做封装一样，SpringMVC就是对Servlet进行封装。

怎么实现的：Spring MVC是一个spring的内置MVC框架。使用（Model-View-Controller的模式）：解决页面代码和后台代码的分离，实现解耦。

## Servlet接受GET请求

### 项目目录和部署的物理绝对路径和里面文件结构和内容

项目目录和Web项目部署到tomcat服务器上的目录是两个不同的概念

项目目录相当于是个生成汽车的车间，tomcat目录相当于是马路。

项目的名称是Web项目的虚拟目录，会映射到Web部署后的绝对物理路径

将车间里的汽车部署到马路上使用：实际上是将编译好的文件部署到绝对物理路径当中，通过虚拟目录映射绝对物理路径

### War和war：exploded

在idea中有工作目录结构，和target结构，

工作目录是我们生成车间，target是成品。

如果是在项目的根目录下，有一个out目录，这个目录可以看作是一个试车场，热部署的时候，可以不用重启tomcat而直接运行这个目录中文件，随时往里面添加的文件，不用重启可以随时生效（前提是部署exploded方式部署的）

如果不是exploded方式部署的，那么就是直接将项目打包，发布到tomcat下的ROOT目录下。而且还是已经打包好了的。那么每次添加修改新的文件就需要重新启动tomcat了。

我们配置的url-name相当于是一个虚拟路径。当我们在url里面访问的时候，如果是热部署，会映射到out文件夹中的.class类。如果不是热部署，会映射到ROOT文件夹下的.class

War:exploded(分解模式) 是在开发的时候使用，如果我们修改了文件，不需要重启，就能直接生效，这个时候url中的虚拟路径对应的资源文件的物理路径为，idea项目下out文件夹

War：（archive）模式，就是发布模式了，直接值把项目打包，放到tomcat的ROOT文件夹下了，这个时候，url中的虚拟路径对应的绝对物理路径就是tomcat ROOT的物理路径了

使用方法HTTPServletRequest.getServletContext().getRealPath()获得绝对物理路径。或者是项目运行时候在的路径。

## Tomcat中文乱码

Tomcat默认使用的是utf-8编码，但是日志输出到控制台，我们的控制台使用的是gbk编码，为了统一编码，将tomcat日志输出编码改成gbk，在config/log.propertities中将utf-8改成gbk

## doGet请求和doPost

HTTP主要有两种请求：doGet和doPost 请求。

从浏览器角度：单击超链接和直接在地址栏输入网址都被认为是doGet请求。

从Servlet角度：每个Servlet程序都有两个入口，doGet()方法用于处理GET请求，doPost用于处理POST请求。

初次执行Servlet时候，会触发构造方法和init()方法，doGet()方法，但是再次执行的时候，只会出发doGet()方法。

一般的Java程序，每次执行的时候，都会从新从新加载字节码文件，执行构造方法，初始化对象。当时Servlet程序是在tomcat启动的时候（或者重新部署的时候），再加上第一次访问后，才会执行构造方法和initial()方法。之后每访问一次都不再执行，只有tomcat服务器重新启动，将项目从新部署，这时候在url中访问Servlet，才会执行构造方法和initial（）方法。

## Servlet程序和JSP程序（动态资源程序）

都包含动态程序，JSP是以HTML为主，里面包含了一点点Java程序，主要是以展示为主，充当视图View层。Servlet是一个Java程序为主，但是也包含了HTML，主要是侧重逻辑控制，从当Controller。。但是都含有动态代码，所以执行的时候会被Servler容器翻译成字节码文件执行。Servlet容器中还内置了两个Java对象，request和response对象。

## JSP文件字符编码和pageEncoding

<%@ **page** contentType="text/html;charset=utf-8" language="java" %>

Jsp文件只能使用一种编码格式，但是浏览器页面为了应对不同的jsp程序，可以使用各种编码来处理jsp程序。这句代码的意思就是告诉浏览器页面，用什么编码来处理这个jsp程序。（jsp编码和pageEncoding之间没有直接的关联，同个设置pageEncoding，将两者编码同一，）

HTTP请求和request和HTTP相应respond

请求的格式application/x-www-form-unrlencoded

Multipart/form-data

响应的格式text/html

<%

Page contextType=”text/html; charset= “utf-8””

%>作用是向HTTP响应头中添加Content-Type响应头

Chartset=”utf-8”1.将响应数据按着utf-8编码，告诉浏览器页面一utf-8进行解码。