回顾：

### ****什么是servlet？****

什么是servlet？

**我们后台具备接收请求、响应数据功能就是servlet！！！**

服务程序或服务连接器，用Java编写的[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8/100571)端程序，具有独立于平台和[协议](https://baike.baidu.com/item/%E5%8D%8F%E8%AE%AE/13020269)的特性，主要功能在于**交互式地浏览和生成数据，生成动态**[Web](https://baike.baidu.com/item/Web/150564)**内容**。

### servlet如何创建？

继承HttpServlet

添加版本id

重写继承HttpServlet中的service方法

将方法内部代码删除，并将参数改为request、response

创建类继承HttpServlet

添加版本id

重写service方法(HttpServletRequest, HttpServletResponse)

删除重写方法中的代码，并修改方法参数

**为servlet类添加注解@WebServlet(“/访问路径”) 或者 在web.xml配置注册**

servlet的访问流程

**action---url-patter---servlet-name---- servlet-name---- servlet-class—servlet类—service重写**

### servlet的原理

**服务器会将请求移交给servlet容器，servlet容器首先对所请求的URL进行解析并根据web.xml配置文件找到相应的处理servlet，同时将request、response对象传递给它，servlet通过request对象可知客户端的请求者、请求信息以及其他的信息等，servlet在处理完请求后回把所需要返回的信息放入response对象中并返回客户端，servlet一旦处理完请求，servlet容器就会刷新respons对象，并把控制权重新返还给web服务器**

### servlet的生命周期

**init()初始化，创建servlet**

第一次访问servlet时，会执行init()方法来创建servlet对象，而且整个生命周期中只会执行一次，所以是单例对象！

**在访问到达servlet之前就创建servlet对象！以空间换时间！在servlet配置时，添加一个loadOnStartUp属性、参数，取值范围为1~5，值越小优先级越高！**

**service()只要servlet被访问，也就是接收到请求，此方法就会执行**

**而且多个请求同时到达servlet时，也不会出现卡顿的现象，验证了servlet是支持多线程！**

**destroy()销毁**

**正常关闭tomcat服务器，右键stop**

**重新启动tomcat服务器**

**修改servlet代码也会自动执行destroy**

**服务器超时时间一到(30分钟)，也会执行destroy()**

### servlet参数获取

get自动处理了中文参数乱码，但是是在tomcat8.0之后，post需要指定请求编码集为utf-8

request.setCharacterEncoding(“utf-8”)

request.getParameter(“参数关键字”)(name)

servlet中响应数据

设置响应数据的类型及编码 response.setContentType(“text/html;charset=utf-8”)

通过自带的打印流对象输出内容 response.getWriter.write(“内容”)

### 什么是XML？

**可扩展**[标记语言](https://baike.baidu.com/item/%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80)，[标准通用标记语言](https://baike.baidu.com/item/%E6%A0%87%E5%87%86%E9%80%9A%E7%94%A8%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80/6805073)的子集，简称XML。是一种用于标记电子文件使其具有结构性的[标记语言](https://baike.baidu.com/item/%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80/5964436)。

**标记最典型的特点就是代码形式为标签，也就是说xml文件也可以书写标签！**

**可扩展的意思：支持自定义标签，我们之前学过的html中的标签都是内定的、固定的。**

可扩展标记语言(xml)是一种很像html的语言

**它的设计宗旨并不是像html一样为了展示数据，而是用来存储数据、传输数据！**

### xml 与 html的区别：

它不是超文本标记语言的替代

它是对超文本标记语言的扩展(标签可以自定义)

它与超文本标记语言的设计初衷是完全不一样的：

**xml用来存储数据、传输数据，其焦点在于数据的内容**

**html用来展示数据，其焦点在于数据展示的外观形式**

**html旨在显示信息，而xml旨在传输、存储数据！**

## xml文件在以后的开发中主要做什么？

xml文件在项目中主要用于做**配置文件！！！**

配置：配置项目的启动信息，配置框架的功能，配置框架的连接

web.xml：配置欢迎列表、注册servlet、为servlet绑定路径！

#### xml文档的书写规范：

自定义的标签名，可以包含数字、字母、符号(\_-)，但是不能以数字开头，而且为了让标签具备含义，建议采用全英文加-连接符 servlet-class servlet-name url-pattern

名称内部不能包含空格

标签名区分大小写，自定义内容严格区分大小写

**文档内有且只有一个根节点，根节点标签!!!**

标签属性的值必须写在双引号内部

标签必须严格的嵌套，标签的关标签必须书写/

#### 解析：读取内容的过程叫做解析！

两种：dom解析、sax解析

dom解析：

首先将整个文档加载到内存中，然后**一层一层**解析，类似于树状结构！效率高，但是内存占用率也高！

sax解析：

不需要将整个文档加载到内存中，而是**一行一行**对文档进行解析。它将解析的每一个步骤或者时间点都转换成了对应的解析方法(开始解析文档、结束解析文档、遇到开标签、遇到关标签、遇到标签内的内容)。内存占用率低，但是效率也低。

主流的解析方式：**dom4j // xPath，属于dom4j的衍生版本，支持路径表达式，可以更加快速的定位、查找到需要得到的内容！**

1. ***使用dom4j解析xml文档的步骤：***

**创建一个SAXReader的解析器**

**使用解析器去加载xml文档**

**获取文档的根节点**

**一层一层开始解析，找到需要获取的内容，只要获取到根节点，就相当于所有内容都可以获取！**

//创建SAXReader解析器

SAXReader saxReader = **new** SAXReader();

//使用解析器去加载xml文档 Document就是整个xml文档对象

Document document = saxReader.read(**new** File("src/book.xml"));

//从文档对象中获取到文档的根节点

**return** document.getRootElement();

什么是XPath呢？

XPath即为[XML](https://baike.baidu.com/item/XML)路径语言（XML Path Language），它是一种用来确定XML文档中某部分位置的语言。

它支持**路径表达式**，来**精确、快速**的定位到某一个层级下的节点，从而减轻了逐层遍历的压力！

1. XPath的使用步骤：

**创建SAXReader解析器**

**通过解析器去加载文档**

**直接使用第二步得到的文档对象，并配合路径表达式，直接查找节点，获取内容！**

## 总结dom4j与XPath的方法区别

dom4j:

获取节点属性的方法：attributes() ==> List<Attribute>

获取属性名称的方法：getName()

获取属性值的方法：getValue()

获取所有子节点集合的方法：elements() ==> List<Element>

获取节点名称的方法：getName()

获取节点文本内容的方法：getText()

XPath:

配合路径表达式来获取节点集合的方法：document.selectNodes(“路径表达式”)

获取节点名称的方法：getName()

获取节点文本内容的方法：getText()

dom4j中的节点类型为Element XPath中的节点类型为Node