## XML-Tomcat-Servlet-HTTP

### 1，xml

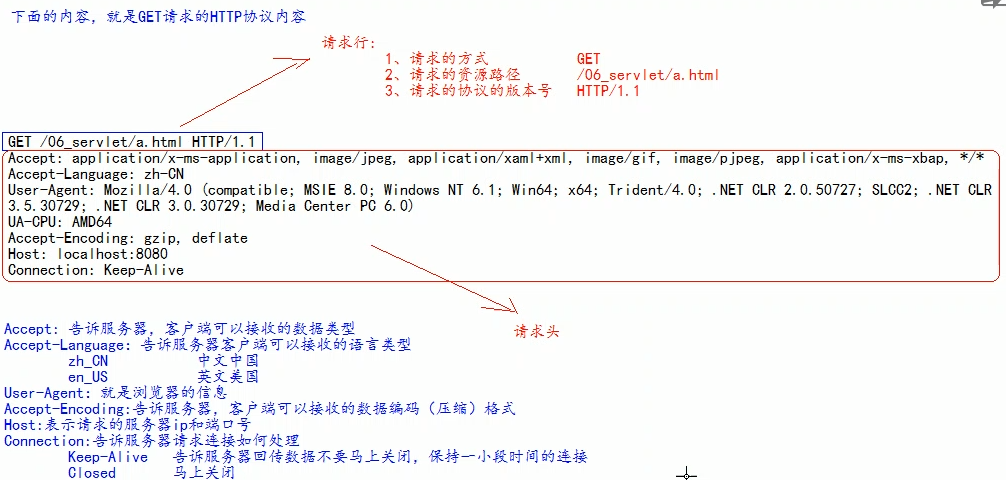
1. Xml是可扩展的标记性语言
2. Xml的作用：
   1. 用来保存数据，而且这些数据具有自我描述性
   2. 还可以作为项目的配置文件
   3. 可以作为网络传输的格式（现在json为主）

### 2，Tomcat

1. 请求和响应是成对出现的，有请求就有响应
2. web资源的分配
   1. 静态资源：html，css，js，txt，MP4，jpg
   2. 动态资源：jsp页面，servlet程序
3. 常用服务器（javaweb容器）
   1. Tomcat
   2. Jboss
   3. Glassfish
   4. Resin
   5. Weblogic
4. Tomcat目录
   1. Bin：专门用来存放tomcat服务器的可执行 程序
   2. Conf：专门用来存放tomcat服务器的配置文件
   3. Lib：专门用来存放tomcat服务器的jar包
   4. Logs：专门用来存放tomcat服务器运行时输出的日志信息
   5. Temp：专门用来存放tomcat运行时产生的临时数据
   6. Webapps：专门用来存放部署的web工程
   7. Work：是tomcat工作时的目录，用来存放tomcat运行时jsp翻译为servlet的源码，和session钝化的目录
5. 部署方式
   1. 找到tomcat下的conf目录\catalina\localhost\创建一个xml配置文件
      1. 内容：<Context path=”/book” docBase=”E:\yangyangli\Desktop\IDEA-workspace”>
      2. Context:表示一个工程的上下文
      3. Path：表示工程的访问路径
      4. docBase表示工程目录在哪里
6. java.net.BindException: Address already in use: bind
   1. 这个错误是端口被占用
7. 如果地址栏没有输入工程名默认访问html文件
8. 默认访问root项目
9. Web工程目录介绍
   1. Web-inf : 是一个手服务器保护的目录，浏览器无法直接访问到此目录的内容
   2. Web.xml: 它是整个动态web工程的配置部署描述文件，可以在这里配置很多web工程的组件，比如：servlet程序，filter过滤器，listen监听器，session超时等等
10. 热部署：修改on frame 为：update classes and resources

### 3，Servlet

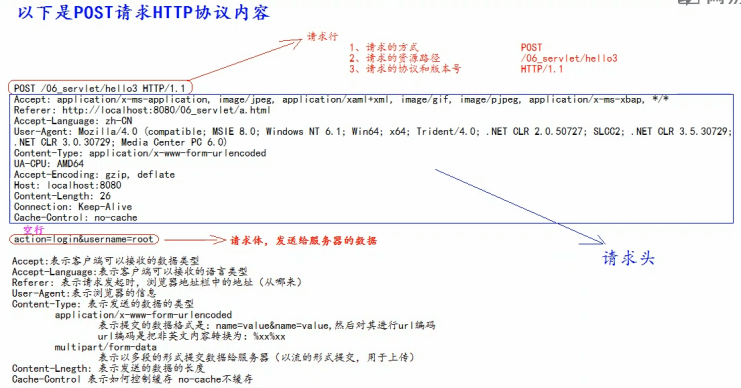
1. 什么是servlet：
   1. Servlet是javaee规范之一，规范就是接口
   2. 三大组件之一，三大组件分别是：servlet程序，filter过滤器，listener监听器
   3. Servlet是运行在服务器上的一个java小程序，它可以接收客户端发送过来的请求，并相应数据给客户端
2. 约定大于配置
3. 常见错误
   1. Url-pattern:没有加斜杠/
   2. Servlet-name：配置的值不存在
   3. Servlet-class：全类名配置错误
4. Servlet声明周期
   1. 执行servlet构造器方法，第一次访问时会被调用
   2. 执行init初始化方法，第一次访问时会被调用
   3. 执行service方法，每次访问时都会调用
   4. 执行destroy销毁方法，停止web工程时调用
5. ServletConfig程序的配置信息类的三大作用
   1. 可以获取servlet程序的别名servlet-name 的值
   2. 获取初始化参数 init-param
   3. 获取servletContext对象
6. ServletContext对象
   1. Servletcontext是一个接口，它表示servlet上下文对象
   2. 一个web工程只有一个servletcontext对象实例，单例
   3. Servletcontext是一个域对象，整个项目工程
      1. 什么是域对象
         1. 域对象就是可以像map一样存取数据的对象，叫域对象
         2. 这里的域指的是存取数据的操作范围
   4. Servletcontext是在web工程部署启动的时候创建，在web工程停止的时候销毁
   5. 四个作用
      1. 获取web.xml中配置的上下文参数 context-param
      2. 获取当前的工程路径，格式：/工程路径
      3. 获取工程部署后在服务器硬盘上的绝对路径
      4. 像map一样存取数据
7. HTTP协议
   1. 协议是指双方，或多方，相互约定好大家都需要遵守的规则，叫协议
   2. 所谓HTTP协议就是指，客户端和服务器之间通信时发送的数据，需要遵守的协议叫HTTP协议
   3. HTTP协议中的数据又叫 报文
8. Get请求
   1. 请求行
      1. 请求的方式 get
      2. 请求的资源路径[+?+请求参数]
      3. 请求的协议的版本号 HTTP/1.1
   2. 请求头
      1. Key:value 组成 不同的键值对表示不同的含义



1. Post请求
   1. 请求行
      1. 请求的方式 post
      2. 请求的资源路径[+?+请求参数]
      3. 请求的协议的版本号 HTTP/1.1
   2. 请求头
      1. Key:value 组成 不同的请求头有不同的含义

空行

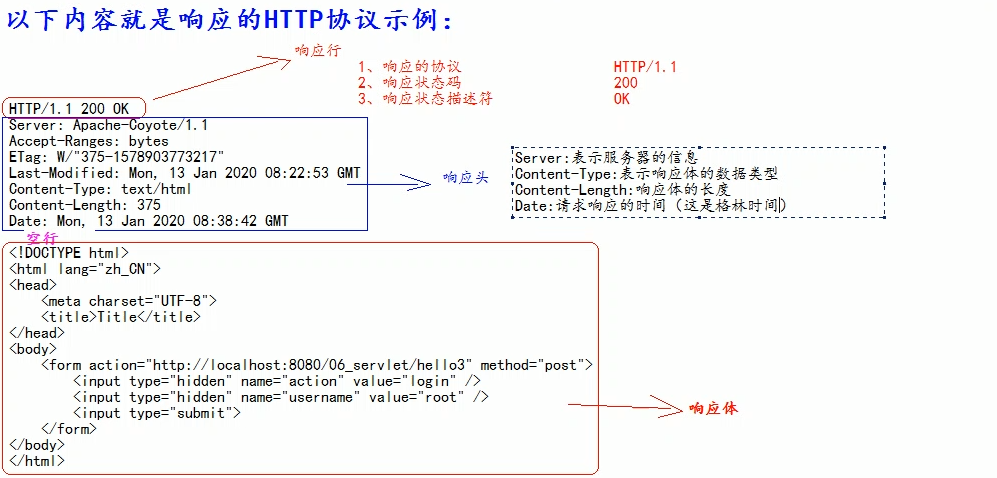
* 1. 请求体 ===》 就是发送给服务器的数据



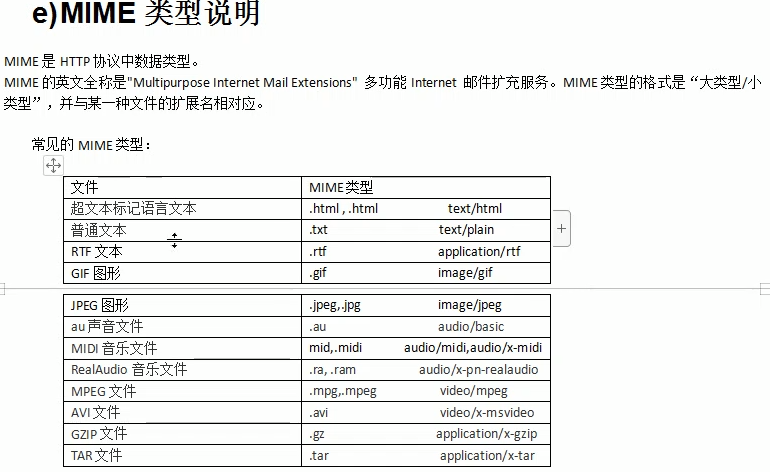
1. 常用请求头
   1. Accept：表示客户端可以接收的数据类型
   2. Accept-Languguese：表示客户端可以接收的语言类型
   3. User-Agent：表示客户端浏览器的信息
   4. Host：表示请求时的服务器ip和端口号
2. 哪些是get请求，哪些是post请求
   1. Get
      1. Form标签method=get
      2. a标签
      3. Link标签引入css
      4. Script标签引入js
      5. Img标签加载图片
      6. Iframe引入html页面
      7. 在浏览器地址栏中输入地址后敲回车
   2. Post
      1. Form标签method=post
3. 相应的HTTP协议
   1. 响应行
      1. 响应的协议和版本号
      2. 响应状态码
      3. 响应状态描述符
   2. 响应头
      1. Key：value 不同的响应头有不同的含义

空行

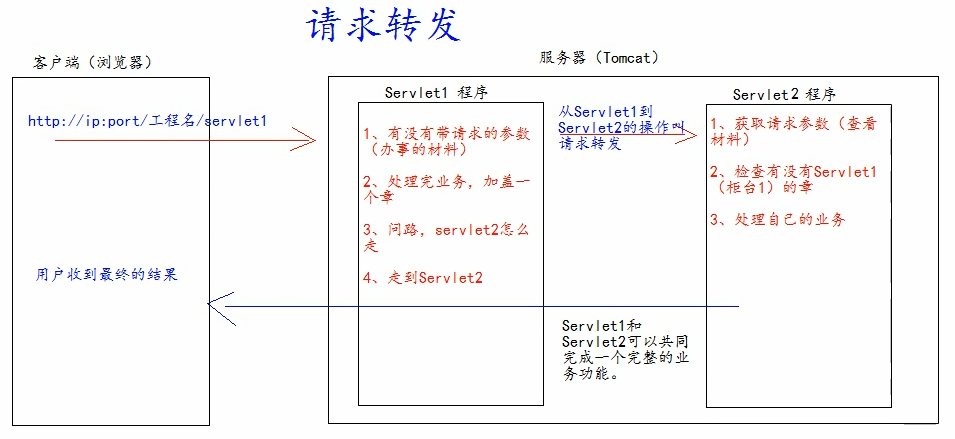
* 1. 响应体 ==》回传给客户端的数据



1. 常用的响应码
   1. 200 ： 表示请求成功
   2. 302 ：表示请求重定向
   3. 404 ：表示请求服务器已经收到了，但是你要的数据不存在（请求地址错误）
   4. 500 ：表示服务器已经收到了请求，但是服务器内部错误（代码）
2. MIME类型

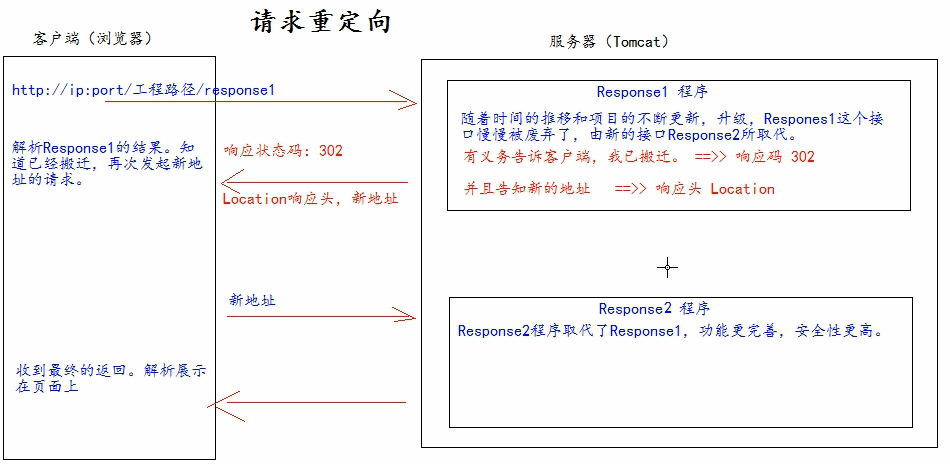


1. HttpServletRequest类的作用
   1. 每次只要有请求进入tomcat服务器，tomcat服务器就会把请求过来的HTTP协议信息解析好封装到request中
   2. 然后传递到service方法（doGet或doPost）中给我们使用，我们可以使用HttpServletRequest对象，获取到所有请求的信息
   3. 常用方法
2. 请求转发

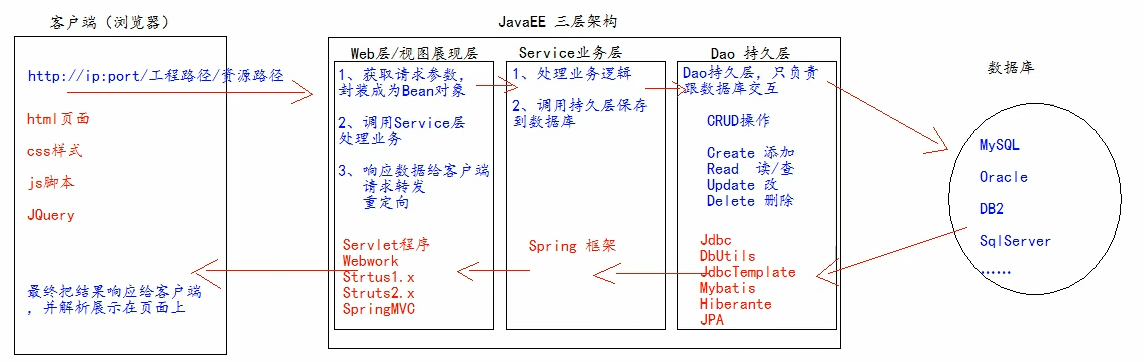


* 1. 特点：
     1. 浏览器地址栏没有变化
     2. 它们是一次请求
     3. 它们共享request域中的数据
     4. 可以转发到WEB-INF目录下
     5. 不可以访问工程以外的资源

1. Web中斜杠 / 的不同意义
   1. 在web中斜杠是一种绝对路径
   2. / 如果被浏览器解析，得到的地址是：<http://ip:port/>
   3. / 如果被服务器解析，得到的地址是：<http://ip:port/>工程路径
   4. 特殊情况：response.sendRediect(“/”); 将斜杠发送给浏览器解析得到：<http://ip:port/>
2. HttpServletResponse类的作用
   1. HttpServletResponse类和HttpServletRequest类一样，每次请求进来，tomcat服务器都会常见一个response对象传递给servlet程序去使用。HttpServletRequest表示请求过来的信息，HttpServletResponse表示所有的响应信息。
   2. 我们如果需要设置返回给客户端的信息，都可以通过HttpServletResponse对象来进行设置
3. 两个响应流只能使用其中一个，否则会报错
4. 请求重定向



1. 请求重定向特点
   1. 浏览器地址栏会发生变化
   2. 两次请求
   3. 不能共享response域中的数据
   4. 不能访问WEB-INF下的资源
   5. 可以访问工程外的资源
2. JavaEE三层架构



包



23，