

web相关概念回顾

1. 软件架构

1. C/S: 客户端/服务器端
2. B/S: 浏览器/服务器端

2. 资源分类

1. 静态资源: 所有用户访问后, 得到的结果都是一样的, 称为静态资源. 静态资源可以直接被浏览器解析
 - * 如: html,css,JavaScript
2. 动态资源: 每个用户访问相同资源后, 得到的结果可能不一样. 称为动态资源. 动态资源被访问后, 需要先转换为静态资源, 在返回给浏览器
 - * 如: servlet/jsp,php,asp....

3. 网络通信三要素

1. IP: 电子设备(计算机)在网络中的唯一标识。
2. 端口: 应用程序在计算机中的唯一标识。 0~65536
3. 传输协议: 规定了数据传输的规则
 1. 基础协议:
 1. tcp:安全协议, 三次握手。 速度稍慢
 2. udp: 不安全协议。 速度快

web服务器软件:

- * 服务器: 安装了服务器软件的计算机
- * 服务器软件: 接收用户的请求, 处理请求, 做出响应
- * web服务器软件: 接收用户的请求, 处理请求, 做出响应。
 - * 在web服务器软件中, 可以部署web项目, 让用户通过浏览器来访问这些项目
 - * web容器

* 常见的java相关的web服务器软件:

- * webLogic: oracle公司, 大型的JavaEE服务器, 支持所有的JavaEE规范, 收费的。
- * webSphere: IBM公司, 大型的JavaEE服务器, 支持所有的JavaEE规范, 收费的。
- * JBOSS: JBOSS公司的, 大型的JavaEE服务器, 支持所有的JavaEE规范, 收费的。
- * Tomcat: Apache基金组织, 中小型的JavaEE服务器, 仅仅支持少量的JavaEE规范servlet/jsp。开源的, 免费的。

- * JavaEE: Java语言在企业级开发中使用的技术规范的总和, 一共规定了13项大的规范

* Tomcat: web服务器软件

1. 下载: <http://tomcat.apache.org/>
2. 安装: 解压压缩包即可。
 - * 注意: 安装目录建议不要有中文和空格
3. 卸载: 删除目录就行了
4. 启动:
 - * bin/startup.bat , 双击运行该文件即可
 - * 访问: 浏览器输入: <http://localhost:8080> 回车访问自己

http://别人的ip:8080 访问别人

* 可能遇到的问题:

1. 黑窗口一闪而过:

- * 原因: 没有正确配置JAVA_HOME环境变量
- * 解决方案: 正确配置JAVA_HOME环境变量

2. 启动报错:

1. 暴力: 找到占用的端口号, 并且找到对应的进程, 杀死该进程
 - * netstat -ano
2. 温柔: 修改自身的端口号
 - * conf/server.xml
 - * <Connector port="8888" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8445" />
 - * 一般会将tomcat的默认端口号修改为80。80端口号是http协议的默认端口号。
 - * 好处: 在访问时, 就不用输入端口号

5. 关闭:

1. 正常关闭:

- * bin/shutdown.bat
- * ctrl+c

2. 强制关闭:

- * 点击启动窗口的×

6. 配置:

* 部署项目的方式:

1. 直接将项目放到webapps目录下即可。
 - * /hello: 项目的访问路径-->虚拟目录
 - * 简化部署: 将项目打成一个war包, 再将war包放置到webapps目录下。
 - * war包会自动解压缩
2. 配置conf/server.xml文件
 - 在<Host>标签体中配置
 - <Context docBase="D:\hello" path="/hehe" />
 - * docBase:项目存放的路径
 - * path: 虚拟目录
3. 在conf\Catalina\localhost创建任意名称的xml文件。在文件中编写
 - <Context docBase="D:\hello" />
 - * 虚拟目录: xml文件的名称

* 静态项目和动态项目:

* 目录结构

- * java动态项目的目录结构:
 - 项目的根目录
 - WEB-INF目录:
 - web.xml: web项目的核心配置文件
 - classes目录: 放置字节码文件的目录
 - lib目录: 放置依赖的jar包

* 将Tomcat集成到IDEA中, 并且创建JavaEE的项目, 部署项目。

Servlet: server applet

- * 概念: 运行在服务器端的小程序
 - * Servlet就是一个接口, 定义了Java类被浏览器访问到(tomcat识别)的规则。
 - * 将来我们自定义一个类, 实现Servlet接口, 复写方法。

* 快速入门:

1. 创建JavaEE项目
2. 定义一个类, 实现Servlet接口
 - * `public class ServletDemo1 implements Servlet`
3. 实现接口中的抽象方法
4. 配置Servlet
 - 在web.xml中配置:
 - `<!--配置Servlet -->`
 - `<servlet>`
 - `<servlet-name>demo1</servlet-name>`
 - `<servlet-class>cn.itcast.web.servlet.ServletDemo1</servlet-class>`
 - `</servlet>`
 - `<servlet-mapping>`
 - `<servlet-name>demo1</servlet-name>`
 - `<url-pattern>/demo1</url-pattern>`
 - `</servlet-mapping>`

* 执行原理:

1. 当服务器接受到客户端浏览器的请求后, 会解析请求URL路径, 获取访问的Servlet的资源路径
2. 查找web.xml文件, 是否有对应的<url-pattern>标签体内容。
3. 如果有, 则在找到对应的<servlet-class>全类名
4. tomcat会将字节码文件加载进内存, 并且创建其对象
5. 调用其方法

* Servlet中的生命周期方法:

1. 被创建: 执行init方法, 只执行一次
 - * Servlet什么时候被创建?
 - * 默认情况下, 第一次被访问时, Servlet被创建
 - * 可以配置执行Servlet的创建时机。
 - * 在<servlet>标签下配置
 1. 第一次被访问时, 创建
 - * <load-on-startup>的值为负数
 2. 在服务器启动时, 创建
 - * <load-on-startup>的值为0或正整数

- * Servlet的init方法, 只执行一次, 说明一个Servlet在内存中只存在一个对象, Servlet是单例的
 - * 多个用户同时访问时, 可能存在线程安全问题。
 - * 解决: 尽量不要在Servlet中定义成员变量。即使定义了成员变量, 也不要对修改值

2. 提供服务: 执行service方法, 执行多次
 - * 每次访问Servlet时, Service方法都会被调用一次。
3. 被销毁: 执行destroy方法, 只执行一次

- * Servlet被销毁时执行。服务器关闭时，Servlet被销毁
 - * 只有服务器正常关闭时，才会执行destroy方法。
 - * destroy方法在Servlet被销毁之前执行，一般用于释放资源
- * Servlet3.0:
- * 好处:
 - * 支持注解配置。可以不需要web.xml了。
 - * 步骤:
 1. 创建JavaEE项目，选择Servlet的版本3.0以上，可以不创建web.xml
 2. 定义一个类，实现Servlet接口
 3. 复写方法
 4. 在类上使用@WebServlet注解，进行配置
 - * @WebServlet("资源路径")

```
@Target({ElementType.TYPE})
@Retention(RetentionPolicy.RUNTIME)
@Documented
public @interface WebServlet {
    String name() default ""; //相当于<Servlet-name>

    String[] value() default {}; //代表urlPatterns()属性配置

    String[] urlPatterns() default {}; //相当于<url-pattern>

    int loadOnStartup() default -1; //相当于<load-on-startup>

    WebInitParam[] initParams() default {};

    boolean asyncSupported() default false;

    String smallIcon() default "";

    String largeIcon() default "";

    String description() default "";

    String displayName() default "";
}
```

IDEA与tomcat的相关配置

1. IDEA会为每一个tomcat部署的项目单独建立一份配置文件

* 查看控制台的log: Using CATALINA_BASE:

"C:\Users\fqy\IntelliJ IDEA 2018.1\system\tomcat_itcast"

2. 工作空间项目 和 tomcat部署的web项目

* tomcat真正访问的是“tomcat部署的web项目”，"tomcat部署的web项目"对应着"工作空间项目" 的web目录下的所有资源

* WEB-INF目录下的资源不能被浏览器直接访问。

3. 断点调试: 使用"小虫子"启动 debug 启动