**IO的路径问题**

**0. 无论是抽象路径名还是路径名字符串，都可以是绝对 路径名或相对 路径名。**绝对路径名是完整的路径名，不需要任何其他信息就可以定位它所表示的文件。相反，相对路径名必须使用取自其他路径名的信息进行解释。默认情况下，java.io 包中的类总是根据当前用户目录来解析相对路径名。此目录由系统属性 user.dir 指定，通常是 Java 虚拟机的调用目录。

调用此类的 [getParent()](mk:@MSITStore:C:\Users\Administrator\Desktop\API_JDK_1_6.CHM::/java/io/File.html#getParent()) 方法可以获取抽象路径名的父 路径名，它由路径名前缀以及路径名名称序列中的每个名称（最后一个除外）组成。对于任何具有绝对抽象路径名的 File 对象，如果其绝对抽象路径名以某个目录的绝对路径名开头，那么该目录的绝对路径名是该 File 对象的祖先。例如，抽象路径名 "/usr" 表示的目录是路径名 "/usr/local/bin" 所表示目录的一个祖先。

**1.绝对路径**：

使用包括系统盘符的路径

如：

String jsonFilePath = “D:/eclipse-jee-mars-2-win32-x86\_64/workspace/LiveInYouth/WebContent/WEB-INF/hitSongData.json”;

new BufferedReader(new InputStreamReader(new FileInputStream(new File(jsonFilePath))))

**2.相对路径:**

**1）在开发Web方面**的应用时, 经常需要获取 服务器中当前WebRoot的物理路径。

**如果是Servlet , Action , Controller, 或则Filter , Listener , 拦截器等相关类时， 我们只需要获得ServletContext**， 然后通过**ServletContext.getRealPath("/")**来获取当前应用在服务器上的物理地址。

**2）如果在类中取不到ServletContext时， 有两种方式可以做到：**

1. 利用Java的类加载机制 调用 XXX.class.getClassLoader().getResource(""); 方法来获取到ClassPath , 然后处理获得WebRoot目录，这种方式只能是该class在WebRoot/WEB-INF/classes下才能生效， 如果该class被打包到一个jar文件中， 则该方法失效。这时就应该用下面一种方式。

2. spring框架的思路, 在WEB-INF/web.xml中 ， 创建一个webAppRootKey的param, 指定一个值（默认为webapp.root）作为键值， 然后通过Listener , 或者Filter , 或者Servlet 执行String webAppRootKey = getServletContext().getRealPath("/"); 并将webAppRootKey对应的webapp.root 分别作为Key , Value写到System Properties系统属性中。之后在程序中通过System.getProperty("webapp.root")来获得WebRoot的物理路 径。

**ServletContextListener获取相对路径实例**：

**public** **void** contextInitialized(ServletContextEvent sce) {

ServletContext ctx = sce.getServletContext();

/\*

\* 独家MV模块数据（源文件、未经更新判断）

\*/

//**TODO** 系统找不到指定的路径

SoleMvData soleMvData = (SoleMvData) LvyJsonUtil.*jsonToObj*("/LiveInYouth/WEB-INF/soleMvData.json", SoleMvData.**class**);// 通JSON文件的路径和class构建bean

ctx.setAttribute("soleMvData", soleMvData);

/\*

\* 热门单曲模块数据（源文件、未经更新判断）

\*/

HitSongData hitSongData = (HitSongData) LvyJsonUtil.*jsonToObj*("/LiveInYouth/WEB-INF/hitSongData.json", HitSongData.**class**);// 通JSON文件的路径和class构建bean

System.***out***.println(hitSongData==**null**?"null":"not Null");

ctx.setAttribute("hitSongData", hitSongData);

//通过ctx获取组装相对路径

String p = ctx.getRealPath("/");

System.***out***.println("realPath is : " + ctx.getRealPath("/"));

//realPath is : D:\eclipse-jee-mars-2-win32-x86\_64\workspace\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\LiveInYouth\

p.replace("\\","\\\\");

p+="WEB-INF\\";

p+="hitSongData.json";

hitSongData = (HitSongData) LvyJsonUtil.*jsonToObj*(p, HitSongData.**class**);

System.***out***.println(hitSongData==**null**?"p is null":"p is not Null");

//通过contextclassloader获取

//file:/D:/JavaWeb/.metadata/.me\_tcat/webapps/TestBeanUtils/WEB-INF/classes/

String path=Thread.*currentThread*().getContextClassLoader().getResource("").toString();

path=path.replace('/', '\\'); // 将/换成\

path=path.replace("file:", ""); //去掉file:

path=path.replace("classes\\", ""); //去掉class\

path=path.substring(1); //去掉第一个\,如 \D:\JavaWeb...

path+="hitSongData.json";

hitSongData = (HitSongData) LvyJsonUtil.*jsonToObj*(path, HitSongData.**class**);

System.***out***.println(hitSongData==**null**?"path is null":"path is not Null");

/\*

\* null

realPath is : D:\eclipse-jee-mars-2-win32-x86\_64\workspace\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\LiveInYouth\

p is not Null

path is not Null

\*/

}