



链滴

# 扫描文件目录并树形展示，利用jQueryTree Table实现，怎么将同目录文件夹升序，文件降序呢？正文贴代码

作者：[Angonger](#)

原文链接：<https://ld246.com/article/1461745241797>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

以下是代码，如果太长不想看，可以看图 [cry](#)

```
public static Map traverseDirectory(String DirPath, int level, String rootDir, Map map) {

    // 根据目录构造file对象
    File dirFile = new File(DirPath);
    // 列出该目录下的文件和目录
    String[] fileArr = dirFile.list();

    // 目录非空
    if (fileArr != null && fileArr.length > 0) {
        List<UnFileTab> fileList = (List<UnFileTab>) map.get(level);
        if (fileList == null) {
            fileList = new ArrayList<UnFileTab>();
        }
        // 遍历保存该目录下的文件和目录至Map中，以level为key，所有文件（目录）封装为UnFileTab
        // 象存入list作为value
        for (String filePath : fileArr) {
            // Map ChildFileMap = new HashMap();// 每层一个
            File file = new File(DirPath + File.separator + filePath);
            UnFileTab unfile = null;
            unfile = assembleFile(file, level, rootDir);
            fileList.add(unfile);
            // ChildFileMap.put(rootDir, fileList);// 储存方式 父节点文件名为Key 值为父节点下的list
            map.put(rootDir, fileList);
            if (file.isDirectory()) {
                int nowLevel = level + 1;
                traverseDirectory(DirPath + File.separator + filePath, level + 1, filePath, map);
                map.put("MAXLEVEL", nowLevel);
            }
        }
    }

    return map;
}

public static StringBuilder parseDirectory(List<UnFileTab> list,
    Map<String, List<UnFileTab>> map, StringBuilder sb, String rootDir, String level) {
    if (list != null && list.size() > 0) {
        for (int i = 0; i < list.size(); i++) {
            UnFileTab unf = list.get(i);
            if ((i & 1) != 0) { // 奇数
                sb.append("<tr class=\"even branch ui-droppable\" data-tt-id=\""
                    + assembleVersion(level, i) + "\" data-tt-parent-id=\"" + level + "\">");// 第一次 根目录
            } else {
                sb.append("<tr class=\"odd branch ui-droppable\" data-tt-id=\""
                    + assembleVersion(level, i) + "\" data-tt-parent-id=\"" + level + "\">");
            }
            sb.append("<td align=\"center\"> <span " + getTypeClass(unf.getType()) + " >"
                + unf.getFilename() + "</span></td>");
            sb.append("<td align=\"center\" class=\"indenter\">" + unf.getParent() + "</td>");
        }
    }
}
```

```

sb.append("<td align=\"center\" class=\"indenter\">" + getType(unf.getType()) + "</td>");
sb.append("<td align=\"center\" class=\"indenter\"> 修改 删除 新增</td>");
sb.append("<td style=\"display:none\">" + unf.getPath() + "</td>");
sb.append("</tr> \n");
List<UnFileTab> childList = map.get(unf.getFilename());
if (childList != null && childList.size() > 0) {
    parseDirectory(childList, map, sb, unf.getFilename(), assembleVersion(level, i));
}
}
}
}

return sb;
}

```

```

public static Map traverseDirectory(String DirPath, int level, String rootDir, Map map) {
    // 根据目录构造file对象
    File dirFile = new File(DirPath);
    // 列出该目录下的文件和目录
    String[] fileArr = dirFile.list();

    // 目录非空
    if (fileArr != null && fileArr.length > 0) {
        List<UnFileTab> fileList = (List<UnFileTab>) map.get(level);
        if (fileList == null) {
            fileList = new ArrayList<UnFileTab>();
        }
        // 遍历保存该目录下的文件和目录至Map中, 以level为key, 所有文件(目录)封装为UnFileTab对象存入list作为value
        for (String filePath : fileArr) {
            // Map ChildFileMap = new HashMap();// 每层一个
            File file = new File(DirPath + File.separator + filePath);
            UnFileTab unfile = null;
            unfile = assembleFile(file, level, rootDir);
            fileList.add(unfile);
            // ChildFileMap.put(rootDir, fileList);// 储存方式 父节点文件名为Key 值为父节点下的list
            map.put(rootDir, fileList);
            if (file.isDirectory()) {
                int nowLevel = level + 1;
                traverseDirectory(DirPath + File.separator + filePath, level + 1, filePath, map);
                map.put("MAXLEVEL", nowLevel);
            }
        }
    }
}

return map;
}

```

```

public static StringBuilder parseDirectory(List<UnFileTab> list,
    Map<String, List<UnFileTab>> map, StringBuilder sb, String rootDir, String level) {
    if (list != null && list.size() > 0) {
        for (int i = 0; i < list.size(); i++) {
            UnFileTab unf = list.get(i);
            if ((i & 1) != 0) { // 奇数
                sb.append("<tr class=\"even branch ui-droppable\" data-tt-id=\""
                    + assembleVersion(level, i) + "\" data-tt-parent-id=\"" + level + "\">");// 第一次 根目录
            } else {
                sb.append("<tr class=\"odd branch ui-droppable\" data-tt-id=\""
                    + assembleVersion(level, i) + "\" data-tt-parent-id=\"" + level + "\">");
            }
            sb.append("<td align=\"center\"> <span " + getTypeClass(unf.getType()) + " >"
                + unf.getFilename() + "</span></td>");
            sb.append("<td align=\"center\" class=\"indenter\">" + unf.getParent() + "</td>");
            sb.append("<td align=\"center\" class=\"indenter\">" + getType(unf.getType()) + "</td>");
            sb.append("<td align=\"center\" class=\"indenter\"> 修改 删除 新增</td>");
            sb.append("<td style=\"display:none\">" + unf.getPath() + "</td>");
            sb.append("</tr> \n");
            List<UnFileTab> childList = map.get(unf.getFilename());
            if (childList != null && childList.size() > 0) {
                parseDirectory(childList, map, sb, unf.getFilename(), assembleVersion(level, i));
            }
        }
    }
    return sb;
}

```

感觉用QQ截图发出来也是不错的啊