

一款极简的流媒体Web服务器 (Streaming Media Web Server) , 提供视频音乐的在线播放功能

文章分类: Project; 作者: Hackyle;

更新时间: Tue Dec 20 12:09:24 CST 2022

1. Table of Contents

2. 快速预览

3. 环境检查与配置

1. 设置系统编码

2. Java环境

3. 下载Tomcat

4. 配置Tomcat

4. 部署

5. 代码设计

1. index.jsp

2. delete.jsp

3. rename.jsp

一款极简的流媒体Web服务器 (Streaming Media Web Server) , 提供视频音乐的在线播放功能

A extremely simple web server of streaming media, which can provide online play of video and audio

项目源码: <https://github.com/HackyleShawe/StreamingMediaWebServer>

项目背景

- 在硬盘上的高清电影, 难道就只能在电脑上才能看吗? 我想在手机上看行不行呢?
- 类似NAS (Network Attached Storage, 网络附属存储) 一样, 搭建一个局域网私有视频云服务器
- 本项目的核心目的是做一个类似于视频NAS, 可以在局域网下提供视频音乐、在线观看服务

市场上已有的局域网流媒体播放器平台

- flex: 收费
- jellyfin: 免费; 坑多, 可用性较差
- emby: 收费; 可用性还行

Features

- 以文件夹为单位, 递归收集指定目录下的所有文件
- 虽然可收集所有类型的文件, 但目前只可以在线观看视频、音乐, 其他类型文件暂不支在线观看
- 视频文件的两种预览模式
 - 列表模式: 依次展示每个文件夹下的所有文件
 - 宫格模式: 以384px * 216px (是1920*1080的同比缩放) 的格式展示视频、音乐文件
- 文件管理: 文件删除、文件重命名

内容导航

Table of Contents

- [快速预览](#)
- [环境检查与配置](#)
 - [设置系统编码](#)
 - [Java环境](#)
 - [下载Tomcat](#)
 - [配置Tomcat](#)
- [部署](#)
- [代码设计](#)
 - [index.jsp](#)
 - [delete.jsp](#)
 - [rename.jsp](#)

快速预览

Home - Streaming Media Web S... X +

localhost:8080/ms/index.jsp?mode=list

Total: 19

列表模式

图标模式

No.1

删除

重命名

 Alder feat. Mia Pfirman - Maybe We'll Get There.mp4

No.2

删除

重命名

 Promise (承诺) - 山冈晃 (AKIRA YAMAOKA).mp3

No.3

删除

重命名

 中国人-刘德华.mp4

No.4

删除

重命名

 中岛美雪-骑在银龙的背上.mp4

/测试文件夹A

No.1

删除

重命名

 测试文件夹A-Alder feat. Mia Pfirman - Maybe We'll Get There.mp4

No.2

删除

重命名

 测试文件夹A-中国人-刘德华.mp4

No.3

删除

重命名

 测试文件夹A-中岛美雪-骑在银龙的背上.mp4

/测试文件夹B

No.1

删除

重命名

 测试文件夹B-Alder feat. Mia Pfirman - Maybe We'll Get There.mp4

No.2

删除

重命名

 测试文件夹B-中国人-刘德华.mp4

No.3

删除

重命名

 测试文件夹B-中岛美雪-骑在银龙的背上.mp4

HACKYLE

列表模式

Delete - Streaming Media Web S... X +

localhost:8080/ms/delete.jsp?dir=/测试文件夹A&name=测试文件夹A-Alder%20feat.%20Mia%20Pfir

D:/A03-Program/A-Programming/Media

/测试文件夹A/测试文件夹A-Alder feat. Mia Pfirman - Maybe We'll Get There.mp4

是否删除?

确定

返回

HACKYLE

在列表模式下删除文件

Rename - Streaming Media Web S... X +

localhost:8080/ms/rename.jsp?dir=/测试文件夹A&name=测试文件夹A-Alder%20feat.%20Mia%

D:/A03-Program/A-Programming/Media

/测试文件夹A/测试文件夹A-Alder feat. Mia Pfirman - Maybe We'll Get There.m

是否重命名?

旧文件夹名: /测试文件夹A

新文件夹名: /测试文件夹A

旧文件名: 测试文件夹A-Alder feat. Mia Pfirman - Maybe We

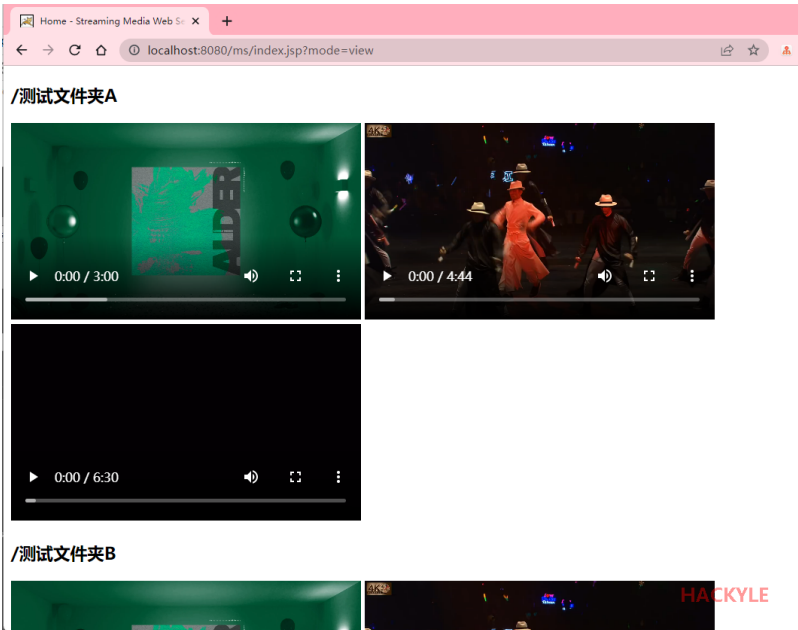
新文件名: 测试文件夹A-Alder feat. Mia Pfirman - Maybe We

提交

返回

HACKYLE

列表模式下重命名文件



宫格模式



宫格模式下的播放

宫格模式下的实时播放

环境检查与配置

主要步骤

- 1. 设置系统编码
- 2. 安装Java以及配置环境变量
- 3. 安装以及配置Tomcat

设置系统编码

前言：Windows环境无需配置此项

检查Linux系统编码环境：

```
root@kali:~# locale
LANG=
LANGUAGE=
LC_CTYPE="POSIX"
LC_NUMERIC="POSIX"
LC_TIME="POSIX"
LC_COLLATE="POSIX"
LC_MONETARY="POSIX"
LC_MESSAGES="POSIX"
LC_PAPER="POSIX"
LC_NAME="POSIX"
LC_ADDRESS="POSIX"
LC_TELEPHONE="POSIX"
LC_MEASUREMENT="POSIX"
LC_IDENTIFICATION="POSIX"
LC_ALL=
root@kali:~#
```

为什么要检查系统编码：使得支持中文，可以正常显示中文的视频文件，以及解析URL中的中文

修改为UTF-8：

进入在root目录，在.bashrc和.profile文件中追加以下数据：

```
1 | export LC_ALL=en_US.UTF-8
2 | export LANG=en_US.UTF-8
3 | export LANGUAGE=en_US.UTF-8
```

使生效：

```
1 | source .bashrc
2 | source .profile
```

再检查：

```
root@kali:~# locale
LANG=en_US.UTF-8
LANGUAGE=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
LC_ADDRESS="en_US.UTF-8"
LC_TELEPHONE="en_US.UTF-8"
LC_MEASUREMENT="en_US.UTF-8"
LC_IDENTIFICATION="en_US.UTF-8"
LC_ALL=en_US.UTF-8
root@kali:~#
```

Java环境

这里以aarch64的树莓派为例，演示安装过程。其他Linux和Windows平台同理。

查看系统平台：uname -a

```
root@kali:~# uname -a
Linux kali 4.19.93-Re4son-v8+ #1 SMP PREEMPT Tue Jan 7 13:59:37 UTC 2020 aarch64 GNU/Linux
```

根据平台下载JDK：<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

oracle.com/java/technologies/downloads/#java8

ORACLE

Products Industries Resources Support Events Developer

View Accounts Contact Us

Documentation Download

Linux macOS Solaris Windows

| Product/file description | File size | Download |
|---------------------------|-----------|---|
| ARM 64 RPM Package | 59.15 MB | jdk-8u301-linux-aarch64.rpm |
| ARM 64 Compressed Archive | 70.84 MB | jdk-8u301-linux-aarch64.tar.gz |
| ARM 32 Hard Float ABI | 73.55 MB | jdk-8u301-linux-arm32-vfp-hflt.tar.gz |
| x86 RPM Package | 109.49 MB | jdk-8u301-linux-i586.rpm |
| x86 Compressed Archive | 138.48 MB | jdk-8u301-linux-i586.tar.gz |
| x64 RPM Package | 109.24 MB | jdk-8u301-linux-x64.rpm |
| x64 Compressed Archive | 138.78 MB | jdk-8u301-linux-x64.tar.gz HACKYLE |

上传解压：

- 在opt目录下创建Java文件夹，专门存放jdk数据： /opt/java
- 将jdk数据包上传并解压： tar -zxvf jdk-8u301-linux-aarch64.tar.gz

注意：usr存放的是系统自带的程序数据包，opt存放的是用户的数据包。

添加环境变量：

- 编辑： vim /etc/profile
- 追加以下数据：

```
1 export JAVA_HOME=/opt/jdk1.8.0_301
2 export PATH=$JAVA_HOME/bin/:$PATH
3 export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

- 使得配置的环境变量立即生效： source /etc/profile
- 查看是否配置成功： java -version

下载Tomcat

下载：<https://tomcat.apache.org/download-90.cgi>

tor Jakarta EE
Tomcat Connectors
Tomcat Native
Taglibs
Archives

Documentation
Tomcat 10.1 (alpha)
Tomcat 10.0
Tomcat 9.0
Tomcat 8.5
Tomcat Connectors
Tomcat Native
Wiki
Migration Guide
Presentations
Specifications

Problems?
Security Reports
Find help
FAQ
Mailing Lists
Bug Database
IRC

Get Involved
Overview

Mirrors

You are currently using **https://dldcn.apache.org/**. If you encounter a problem with this mirror failing, there are *backup* mirrors (at the end of the mirrors list) that should be available.

Other mirrors:

9.0.54

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha512\)](#)

上传解压：

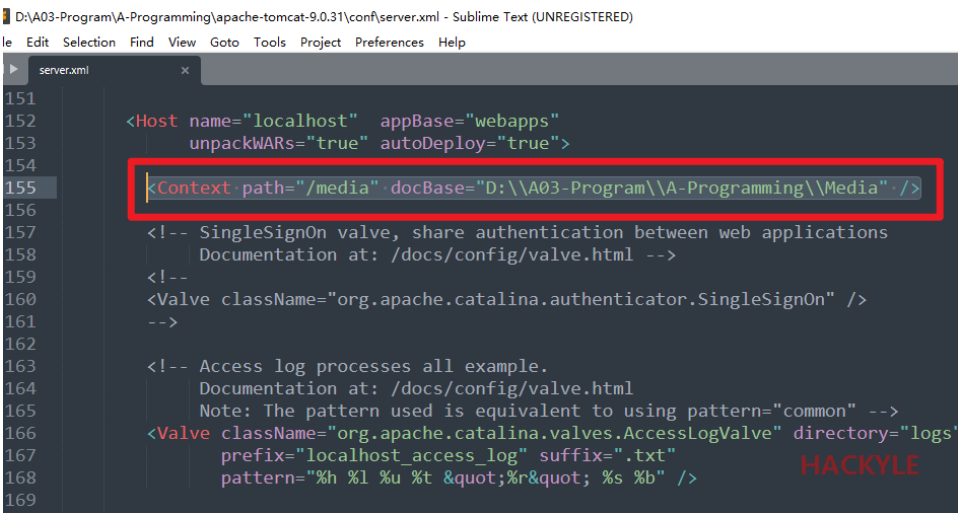
- 在opt目录下创建Tomcat文件夹，专门存放jdk数据： /opt/tomcat
- 将jdk数据包上传并解压： tar -zxvf apache-tomcat-9.0.54.tar.gz
- 解压后将文件夹名字改短一点，方便以后使用： mv apache-tomcat-9.0.54 tomcat9

一款极简的流媒体Web服务器（Streaming Media Web Server），提供视频音乐的在线播放功能
修改配置文件conf/server.xml，自定义端口和URI编码为UTF-8：

```
Java HTTP Connector: /docs/config/http.html
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="9999" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" URIEncoding="UTF-8" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
           port="8080" protocol="HTTP/1.1"
```

配置Tomcat

在Tomcat安装目录下的conf/server.xml中指定静态资源的目录，以及解析地址



凡是以/media开头的请求，全部都会解析到docBase指定的目录下

docBase指定的目录是视频文件的存放地点

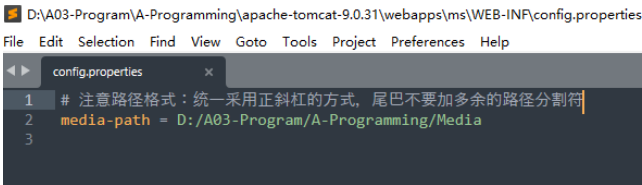


部署

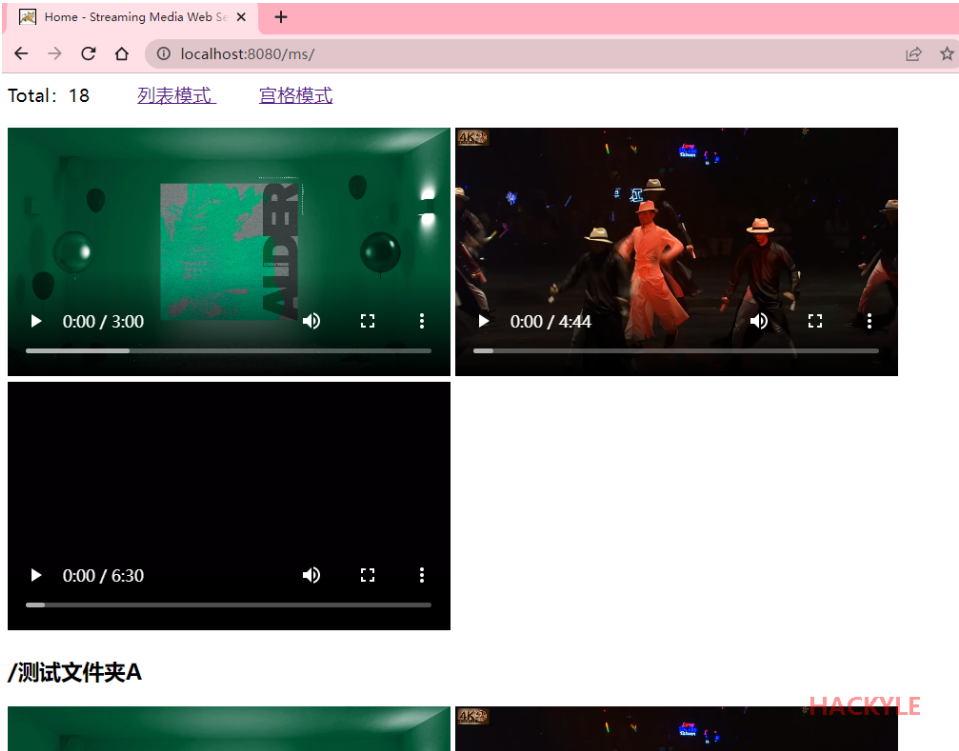
- 1. 下载源码包：<https://github.com/HackyleShawe/StreamingMediaWebServer>
- 2. 将其解压，重命名为一个简单的名字，我重命名为：ms，意为Media Server
- 3. 复制ms到到%Tomcat%/webapps/



- 4. 编辑WEB-INF/config.properties文件，指定为在配置Tomcat事conf/server.xml中指定静态资源的目录，也即docBase的值



- 5. 启动Tomcat，指定%Tomcat%/bin/shartup.bat
- 6. 访问项目：http://localhost:8080/ms/



首次访问

代码设计

设计思想

- 1. 创建一个专门存放视频文件的目录，例如：D:\A03-Program\A-Programming\Media
- 2. 配置Tomcat将所有的静态资源请求全部解析到该个文件夹下
- 3. 写一个JSP来读取该目录下的所有媒体文件
- 4. 点击媒体文件后，经过Tomcat就解析到了媒体文件的真实路径

index.jsp

功能

- 读取配置文件中指定路径下的所有视频文件，以文件夹为单位暂存
- 两种呈现方式：列表、宫格

读取配置文件主要逻辑

```
1 String projectPath = request.getSession().getServletContext().getRealPath(""); //3
2 String path = projectPath + "/WEB-INF/config.properties";
3
4 //从配置文件中读取媒体文件的所在目录
5 FileInputStream is = new FileInputStream(path);
6 Properties pro = new Properties();
7 pro.load(is);
8 String mediaPath = pro.getProperty("media-path");
```

以文件夹为单位，收集该路径下的所有文件，暂存与Map

```

1 //递归收集路径(mediaPath)下的所有文件存于fileMap中
2 public void collectFile(String mediaPath, File file, Map<String, List<String>> fileMap) {
3     if(!file.exists()) {
4         return;
5     }
6
7     //收集文件
8     File[] files = file.listFiles(File::isFile);
9     if(files != null && files.length > 0) {
10         String name = file.getAbsolutePath().replace(File.separator, "/");
11         List<String> collect = Arrays.stream(files).map(ff -> ff.getName()).collect(Collectors.toList());
12         fileMap.put(name.replace(mediaPath, ""), collect);
13     }
14
15     //收集文件夹
16     File[] dirs = file.listFiles(File::isDirectory);
17     if(dirs == null || dirs.length < 1) {
18         return;
19     }
20     //递归收集文件夹下的子文件、文件夹
21     for (File dir : dirs) {
22         collectFile(mediaPath, dir, fileMap);
23     }
24 }

```

两种呈现模式，基于JSTL的choose-when标签，类似于Java的switch-case

```

1 <c:choose>
2     <c:when test="${!empty mode && mode == 'list' }"> <!-- else if 的意思 --%>
3         <p>No.${varSta.count}</p>
4         <a href="delete.jsp?dir=${entry.key}&name=${name}"><b>删除</b></a>
5         <a href="rename.jsp?dir=${entry.key}&name=${name}"><b>重命名</b></a>
6         <a target="_blank" href="/media/${entry.key}/${name}">${name}</a>
7     </p>
8 </c:when>
9
10 <c:otherwise> <!-- else的意思 --%>
11     <!-- 视频的高度、宽度是按照1920*1080同比例缩小 -->
12     <video src="/media/${entry.key}/${name}" width="384px" height="216px" controls>
13 </c:otherwise>
14 </c:choose>

```

delete.jsp

删除文件：调用java.util.File的delete方法

```

1 <!-- 删除确认 -->
2 <c:if test="${!empty deleted && deleted == 'yes' }">
3     <%
4         String rootname = mediaRoot+"/"+dir+"/"+name;
5         // out.println(rootname);
6         File file = new File(rootname);
7         boolean del = file.delete();
8         // out.println(del);
9         response.sendRedirect("index.jsp?mode=list");
10     %>
11

```


</c:if>

rename.jsp

重命名文件：调用java.util.File的renameTo方法

```
1 <!-- 确认重命名 -->
2 <c:if test="${!empty renamed && renamed == 'yes' }">
3   <%
4     String oldDir = request.getParameter("oldDir");
5     String newDir = request.getParameter("newDir");
6
7     String oldName = request.getParameter("oldName");
8     String newName = request.getParameter("newName");
9
10    String oldname = mediaRoot + oldDir + "/" + oldName;
11    String newname = mediaRoot + newDir + "/" + newName;
12
13    if(!oldname.equals(newname)) {
14      File oldFile = new File(oldname);
15      File newFile = new File(newname);
16
17      if(oldFile.exists()) {
18        oldFile.renameTo(newFile);
19        response.sendRedirect("index.jsp?mode=list");
20      } else {
21        out.println("路径不存在: " + oldname);
22        out.println("<h1><a href='index.jsp?mode=list'>返回</a></h1>");
23      }
24    }
25  %>
26 </c:if>
```

版权声明：非明确标注皆为原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上本文链接及此声明。
原文链接：<https://blog.hackyle.com/article/project/streaming-media-web-server>

留下你的评论

Name:

Email:

Link:

File Edit View Format Tools Table Help

↶

↷

B

I

U

~~S~~

≡

▼

≡

▼

A

I

{ }

Ω

😊

≡

≡

≡

≡

≡

≡

...

Input comment, please

p

0 words tiny

SUBMIT

RESET

© Copy Right: 2022 HACKYLE. All Rights Reserved

Designed and Created by HACKYLE SHAW

备案号: 浙ICP备20001706号-2