

# webdav文件的上传与下载

作者：徐康尊

创建日期：20220301

修订日期：

审批人：

审批日期：

版本号：V 1.0

## 一、概述

1. 简单来说，webdav就像一个存储服务，各种应用都可以连接到它，允许应用直接访问我们存储的内容，对其进行读写操作。这里使用curl以及jar包来测试webdav文件的上传下载功能。

## 二、webdav的搭建

### 2.1. 在Tomcat的webapps目录下新建webdav文件夹，并在此文件夹下新建WEB-INF\web.xml文件

进入/opt/RHGL/wlh/wlh\_srv/wlh3inservice/tomcat7/webapps 文件夹，此文件夹为tomcat安装的文件夹

命令如下：

```
mkdir -p webdav/WEB-INF
cd webdav/WEB-INF
touch web.xml
```

### 2.2. 在新建web.xml下添加配置

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  id="webApp_ID" version="3.0">
  <display-name>webdav</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
```

```

<servlet>
  <servlet-name>webdav</servlet-name>
  <servlet-class>org.apache.catalina.servlets.WebdavServlet</servlet-
class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>
    <param-value>true</param-value>
  </init-param>
  <!-- Read-Write Access Settings -->
  <init-param>
    <param-name>readonly</param-name>
    <param-value>false</param-value>
  </init-param>
</servlet>
<!-- URL Mapping -->
<servlet-mapping>
  <servlet-name>webdav</servlet-name>
  <url-pattern>/*</url-pattern>
</servlet-mapping>
<security-constraint>
  <web-resource-collection>
    <web-resource-name>webdav</web-resource-name>
    <!-- Detect WebDAV Methods in URL For whole Application -->
    <url-pattern>/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
    <!--
    <http-method>GET</http-method>
    <http-method>PUT</http-method>
    <http-method>HEAD</http-method>
    <http-method>TRACE</http-method>
    <http-method>POST</http-method>
    <http-method>DELETE</http-method>
    <http-method>OPTIONS</http-method>
    -->
    <http-method>PROPFIND</http-method>
    <http-method>PROPPATCH</http-method>
    <http-method>COPY</http-method>
    <http-method>MOVE</http-method>
    <http-method>LOCK</http-method>
    <http-method>UNLOCK</http-method>
  </web-resource-collection>
  <!-- Restrict access by role -->
  <auth-constraint>
    <role-name>*</role-name>
  </auth-constraint>
</security-constraint>
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>webdav</realm-name>
</login-config>
<security-role>
  <description>WebDAV User</description>
  <role-name>webdav</role-name>

```

```
</security-role>
</web-app>
```

## 2.3. 如果通过浏览器访问时不需要用户名密码，将web.xml中以下配置删除后重启服务即可

```
<http-method>GET</http-method>
<http-method>POST</http-method>
```

## 2.4. 添加用户

### 2.4.1 为自定义权限名称。

根据上面权限名称，在Tomcat账号体系中增加账号密码，配置如下：

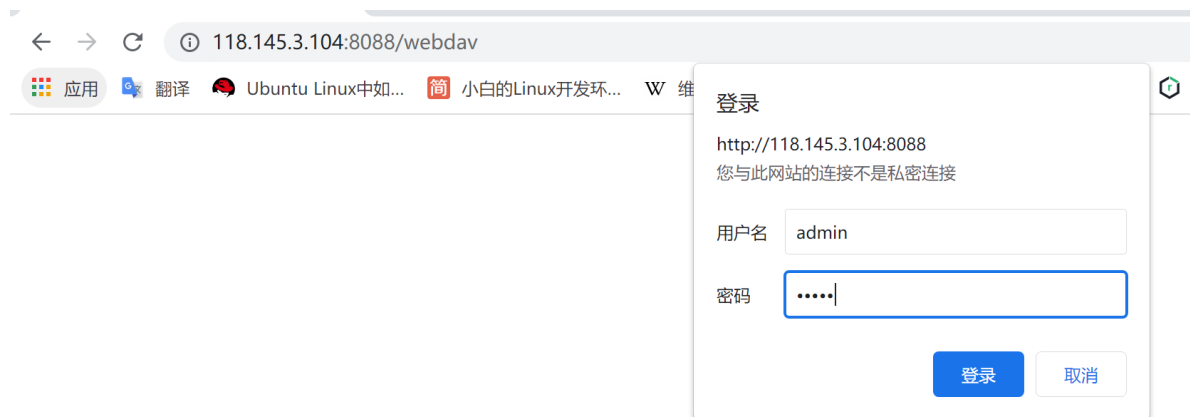
编辑/opt/RHGL/wlh/wlh\_srv/wlh3inserver/tomcat7/conf/tomcat-users.xml

```
<role rolename="webdav"/>
<user username="admin" password="admin" roles="webdav"/>
```

重启tomcat（按照不同服务器，tomcat的重启方式不同，请按需重启）：

```
systemctl restart wlh3inserver
```

### 2.4.2 访问http:118.145.3.104:8088/webdav，出现以下信息表示成功：



此时118.145.3.104为安装webdav的服务器IP，端口号8088为此服务器中tomcat设置的默认端口号，账号密码为2.4.1tomcat账号体系中增加的账号密码

输入账号密码后，显示如下：



### 三、CURL的使用

几乎所有具有相对较新操作系统的人都可以使用cURL，因为cURL在Windows，MacOS和大多数Linux发行版中作为默认设置提供。对于较早的系统，例如10之前的任何Windows操作系统，可能需要下载并安装cURL。

要使用cURL，只需打开终端并输入“curl”。正常情况下，“curl -help”会自动跳出，用户可以选择是否执行“curl -help”命令行。如前所述，“帮助”将列出所有命令可能性。

```
命令提示符

Microsoft Windows [版本 10.0.19044.1526]
(c) Microsoft Corporation。保留所有权利。

C:\Users\wlh>curl
curl: try 'curl --help' for more information

C:\Users\wlh>curl --help
Usage: curl [options...] <url>
  -d, --data <data>      HTTP POST data
  -f, --fail              Fail silently (no output at all) on HTTP errors
  -h, --help <category> Get help for commands
  -i, --include           Include protocol response headers in the output
  -o, --output <file>    Write to file instead of stdout
  -O, --remote-name       Write output to a file named as the remote file
  -s, --silent           Silent mode
  -T, --upload-file <file> Transfer local FILE to destination
  -u, --user <user:password> Server user and password
  -A, --user-agent <name> Send User-Agent <name> to server
  -v, --verbose           Make the operation more talkative
  -V, --version           Show version number and quit

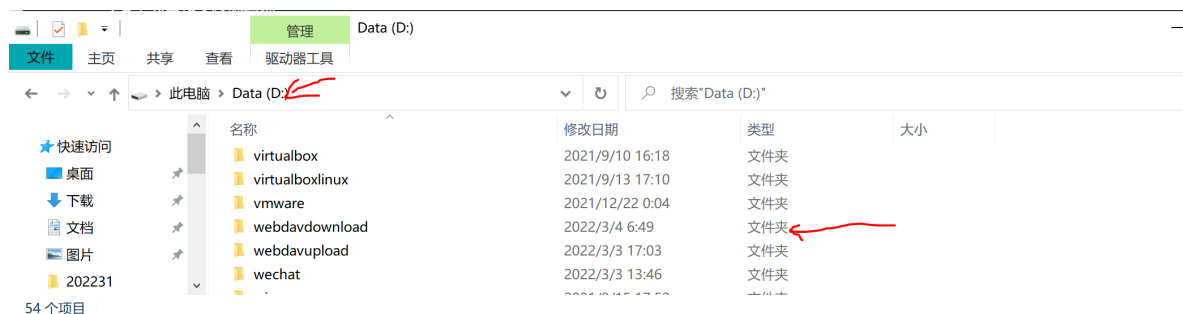
This is not the full help, this menu is stripped into categories.
Use "--help category" to get an overview of all categories.
For all options use the manual or "--help all".

C:\Users\wlh>
```

若系统中没有curl，请自行搜索安装curl。

### 四、使用curl测试webdav文件的上传与下载

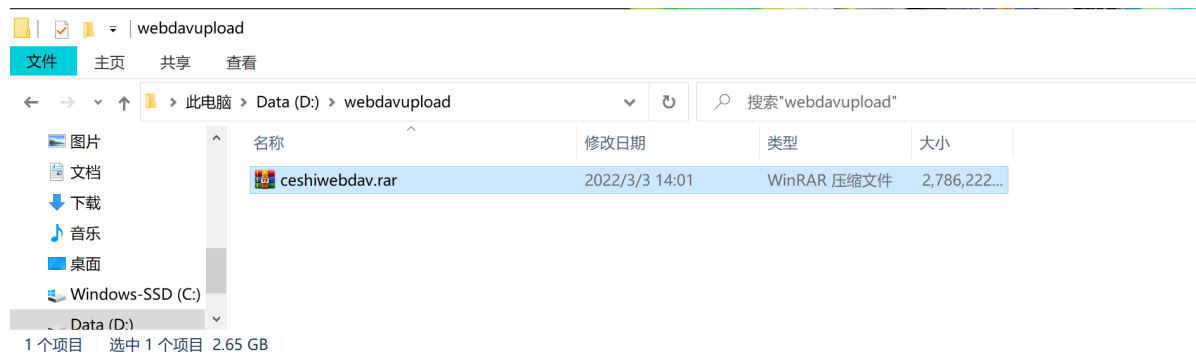
准备工作：在本机任意目录下创建webdavupload文件夹：



#### 4.1 http协议下webdav文件的上传与下载

## 4.1.2上传

webdavupload文件夹下事先放入一个接近3G的大文件，如下图所示



在webdavupload文件下打开终端cmd,输入以下命令

```
curl --user admin:admin -T ceshiwebdav.rar  
http://118.145.3.104:8088/webdav/ceshiwebdav.rar
```

如下图：



其中admin:admin为2.4.1中tomcat账号体系中添加的账号密码；

118.145.3.104为安装webdav的服务器IP，8088为服务器上tomcat默认端口号；

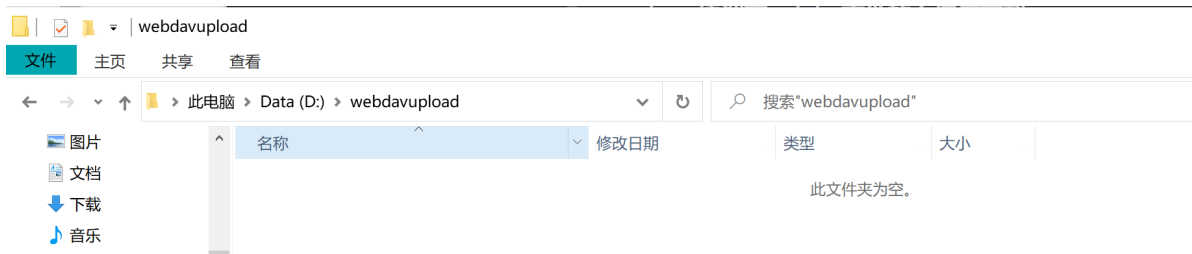
ceshiwebdav.rar为要上传的文件

以下为上传效果图，可以看到ceshiwebdav.rar已经位于目录列表中：

文件名	大小	上次修改时间。
<a href="#">ceshiwebdav.rar</a>	2786221.2 kb	Thu, 03 Mar 2022 06:04:58 GMT

## 4.1.3下载

在任意目录下创建一个下载目录，这里使用的还是D:\webdavupload文件夹，可以看到当前目录下并没有将要下载的ceshiwebdav.rar文件



在此文件夹下打开终端cmd,输入以下命令

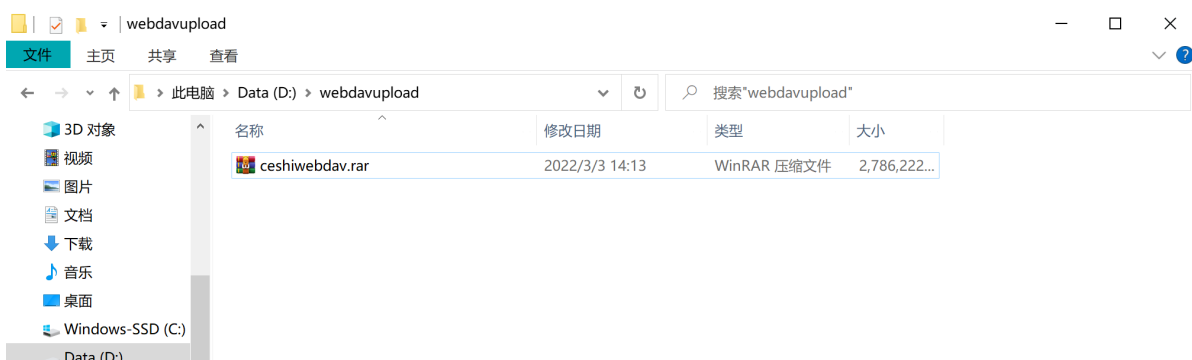
```
curl --user admin:admin http://118.145.3.104:8088/webdav/ceshiwebdav.rar>ceshiwebdav.rar
```

如下图:

```
D:\webdavupload>curl --user admin:admin http://118.145.3.104:8088/webdav/ceshiwebdav.rar>ceshiwebdav.rar
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 2720M  100 2720M    0     0  112M      0  0:00:24  0:00:24 --:--:-- 113M
```

ceshiwebdav.rar为将要下载的文件, 其他参数如4.1.2中所诉。

下载效果图如下:



可以看到该目录下已经存在名为ceshiwebdav.rar的文件

注意: 由于不同系统识别文件的方式不同, 所以文件在windows和Linux的文件大小可能不一致, 这是正常的

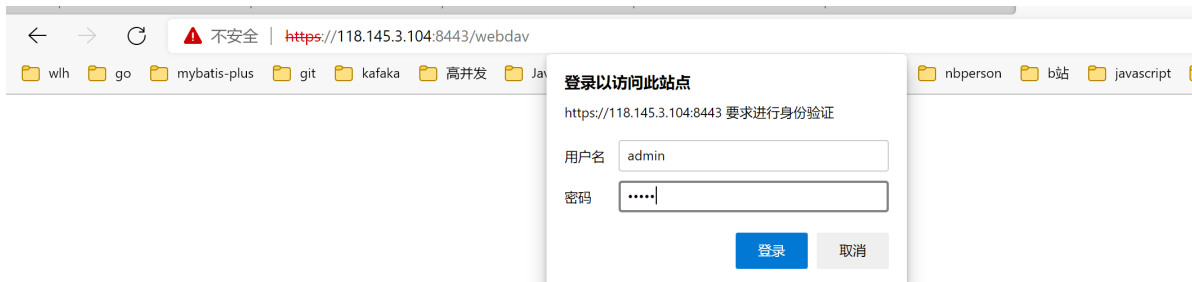
## 4.2 https协议下webdav文件的上传与下载

若tomcat里的https协议是已经配置好的, 那么无需额外配置, 此时tomcat里面https协议默认的端口号是8445, 在浏览器中输入<https://118.145.3.104:8445/webdav>若出现以下弹窗:



点击继续前往

若出现下图所示：



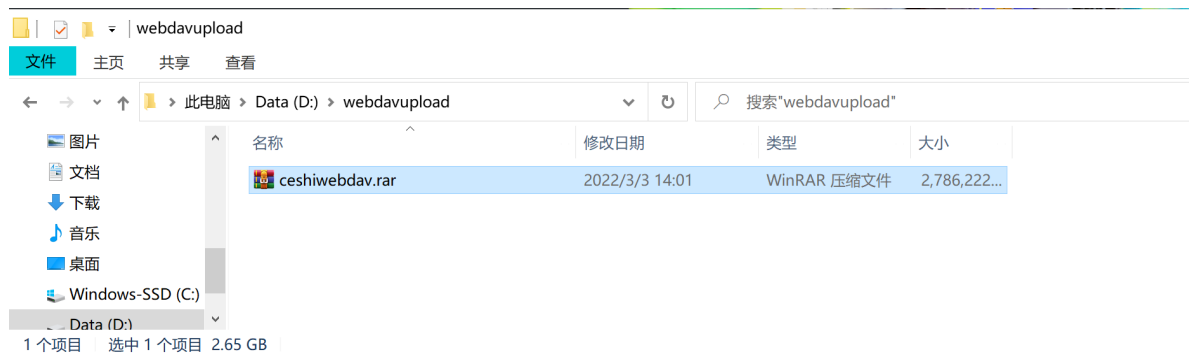
则证明tomcat已经配置https协议，不然参照目录三、常见问题-->6.1tomcat中https协议配置来配置https协议。

之后输入2.4.1tomcat账号体系添加的账号密码，登录之后的页面如下图所示：



## 4.2.1上传

首先如4.1.2中一样在windows任意目录创建一个名为webdavupload的文件夹，里面放入将要上传的文件，这里设置的是一个接近3G的大文件，如下图所示：



在该文件夹下打开终端cmd,输入一下上传命令：

```
curl -k --user admin:admin -T ceshiwebdav.rar https://118.145.3.104:8445/webdav/ceshiwebdav.rar
```

注意：上述-k命令为允许不使用证书到SSL站点，因为https协议是使用的自签的SSL证书，所以在协议下上传与下载时要检查该自签证书，导致只执行-T指令显示不安全，无法上传与下载。

-k上传命令如下图

```
D:\webdavupload>curl -k --user admin:admin -T ceshiwebdav.rar https://118.145.3.104:8445/webdav/ceshiwebdav.rar
D:\webdavupload>
```

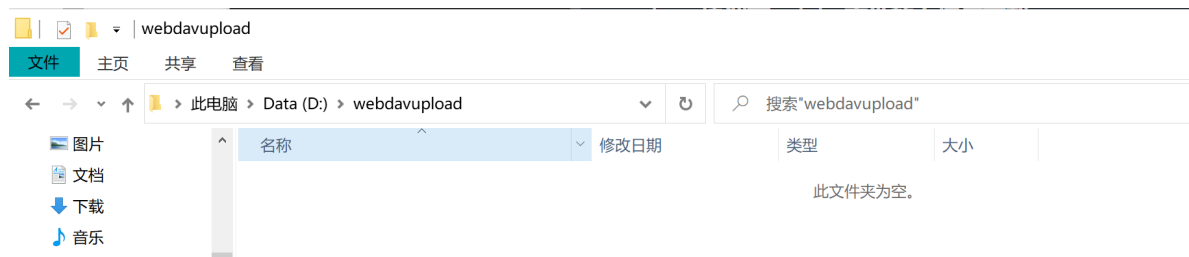
-k上传命令效果图如下：



对于https协议下指定协议的上传与下载命令，并不可行，具体请参照6.4curl下指定协议无法实现webdav文件的上传与下载

## 4.2.2下载

在任意目录下创建一个下载目录，这里使用的还是D:\webdavupload文件夹，可以看到当前目录下并没有将要下载的ceshiwebdav.rar文件：



在此文件夹下打开终端cmd,输入如下命令：

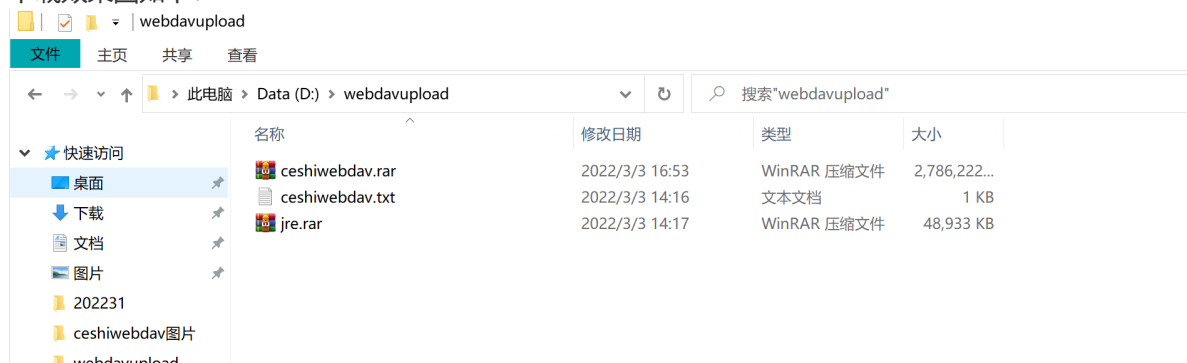


```
curl -k --user admin:admin  
https://118.145.3.104:8445/webdav/ceshiwebdav.rar>ceshiwebdav.rar
```

命令如下图:

```
D:\webdavupload>curl -k --user admin:admin https://118.145.3.104:8445/webdav/ceshiwebdav.rar>ceshiwebdav.rar  
% Total % Received % Xferd Average Speed Time Time Time Current  
Dload Upload Total Spent Left Speed  
100 2720M 100 2720M 0 0 71.2M 0 0:00:38 0:00:38 --:--:-- 72.8M  
D:\webdavupload>
```

下载效果图如下:



## 五、使用jar包测试webdav文件的上传与下载

上传与下载的jar包: **wedavupload.jar**, 可以联系研发人员获取

将jar包放到任意目录下, 执行如下5.1与5.2命令

**注意:** 若要用此jar包来执行上传与下载, webdav的账户密码必须是admin:admin, 若不是, 请参照2.4.1修改账户密码

### 5.1 http协议下webdav文件的上传与下载

```
上传: java -jar wedavupload.jar 32.txt D:\webdavupload\  
http://localhost:8080/webdav/ null null null  
下载: java -jar wedavupload.jar null null null 17.rar D:\webdavupload\  
http://localhost:8080/webdav/
```

上传与下载使用的同一个jar包, **上传的时候使用前三个参数, 其他设置为null; 下载的时候使用后三个参数, 其他设置为null;** 命令中6个参数设置如下:

第一个参数如**16.txt**为本地将要上传的文件名

第二个参数如 **D:\webdavupload\** 为本地将要上传的文件的位置

第三个参数如 <http://localhost:8080/webdav/> 为上传webdav目的路径, 此处为http协议, 设置为http:localhost为目的路径IP; 端口号为tomcat中设置的端口号如下图:



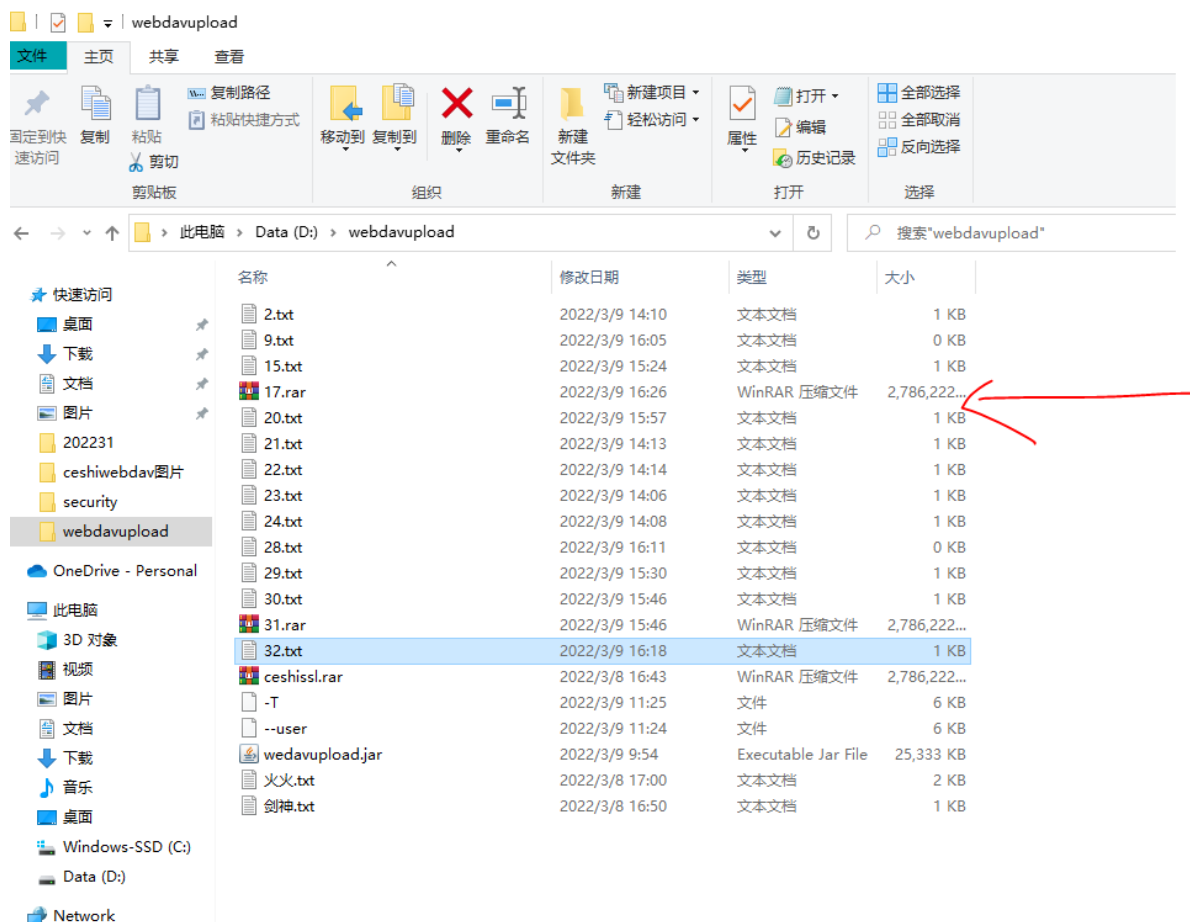
## 5.1.2 下载

执行下载命令，如下图：

```
D:\webdavupload>java -jar wedavupload.jar null null null 17.rar D:\webdavupload\ http://localhost:8080/webdav/
-----文件上传开始-----
getContentType, File ContentType is : application/octet-stream
16:25:54.915 [main] ERROR com.superred.wedavupload.utils.FileUploadUtil - 上传文件失败:
java.lang.IllegalArgumentException: Expected URL scheme 'http' or 'https' but no colon was found
    at okhttp3.HttpUrl$Builder.parse$okhttp(HttpUrl.kt:1260)
    at okhttp3.HttpUrl$Companion.get(HttpUrl.kt:1633)
    at okhttp3.Request$Builder.url(Request.kt:184)
    at com.superred.wedavupload.utils.FileUploadUtil.UploadFileByHttpClient(FileUploadUtil.java:46)
    at com.superred.wedavupload.utils.UploadAndDownloadMain.main(UploadAndDownloadMain.java:27)
-----文件上传结束-----
-----文件下载开始-----
6:26:13.737 [main] INFO com.superred.wedavupload.utils.FileDownloadUtil - 文件下载成功,文件路径[D:\webdavupload\17.rar]
-----文件下载结束-----
D:\webdavupload>
```

对于下载命令，只需要关注上图红框里面的日志即可，上方上传异常日志是正常的，无需关注；

下载成功，下载效果图如下：



## 5.2 https协议下webdav文件的上传与下载

```
上传: java -jar wedavupload.jar 33.txt D:\webdavupload\
https://localhost:8443/webdav/ null null null
下载: java -jar wedavupload.jar null null null 31.rar D:\webdavupload\
https://localhost:8443/webdav/
```

上传与下载使用的同一个jar包，上传的时候使用前三个参数，其他设置为null；下载的时候使用后三个参数，其他设置为null；命令中只有url和5.1里面不一样，如<https://localhost:8443/webdav/>，此处协议为https，所以设置为https；端口号为tomcat里面配置的端口号如下图：

```
D:\apache-tomcat-8.5.64-windows-x64\apache-tomcat-8.5.64\conf\server.xml - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

web.xml — webapps\webdav\WEB-INF — WebappUploadApplication.class — tomcat-users.xml — server.xml — web.xml — conf

100 configuration is used below.
101 --
102 <!--
103 <Connector port="8443" protocol="org.apache.coyote.http11.Http11AprProtocol"
104 maxThreads="150" SSLEnabled="true" >
105 <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
106 <SSLHostConfig>
107 <Certificate certificateKeyFile="conf/localhost-rsa-key.pem"
108 certificateFile="conf/localhost-rsa-cert.pem"
109 certificateChainFile="conf/localhost-rsa-chain.pem"
110 type="RSA" />
111 </SSLHostConfig>
112 </Connector>
113 --
114 <Connector port="8443" protocol="org.apache.coyote.http11.Http11AprProtocol"
115 maxThreads="150" SSLEnabled="true" scheme="https" secure="true" clientAuth="false" sslProtocol="TLS" keystoreFile="
D:\apache-tomcat-8.5.64-windows-x64\apache-tomcat-8.5.64\conf\tomcat.keystore" keystorePass="123456" />
116
117 <!-- Define an AJP 1.3 Connector on port 8009 -->
118 <!--
119 <Connector protocol="AJP/1.3"
120 address=":::1"
121 port="8009"
122 redirectPort="8443" />
123 -->
124
125 <!-- An Engine represents the entry point (within Catalina) that processes
126 every request. The Engine implementation for Tomcat stand alone
127 analyzes the HTTP headers included with the request, and passes them
128 on to the appropriate Host (virtual host).
129 Documentation at /docs/config/engine.html -->
130
131 <!-- You should set jvmRoute to support load-balancing via AJP ie :
132 <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
133 -->
134 <Engine name="Catalina" defaultHost="localhost">
135
136 <!--For clustering, please take a look at documentation at:
137 /docs/cluster-howto.html (simple how to)
138 /docs/config/cluster.html (reference documentation) -->
139
140 <!--
141 <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
142 -->
143
144 <!-- Use the LockOutRealm to prevent attempts to guess user passwords
```

此https端口号配置按需配置，只需要以下两处端口号一致即可，此文件路径为tomcat安装目录下conf文件夹下的server.xml

```
D:\apache-tomcat-8.5.64-windows-x64\apache-tomcat-8.5.64\conf\server.xml - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

web.xml — webapps\webdav\WEB-INF — WebappUploadApplication.class — tomcat-users.xml — web.xml — conf — server.xml — web.xml — conf

67 Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
68 --
69 <!-- <Connector port="8080" protocol="HTTP/1.1"
70 connectionTimeout="20000"
71 URIEncoding="UTF-8"/> -->
72 <Connector port="8080" protocol="HTTP/1.1"
73 connectionTimeout="20000"
74 redirectPort="8443" />
75 URIEncoding="UTF-8"/>
76
77 <!-- A "Connector" using the shared thread pool-->
78 <!--
79 <Connector executor="tomcatThreadPool"
80 port="8080" protocol="HTTP/1.1"
81 connectionTimeout="20000"
82 redirectPort="8443" />
83 -->
84 <!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
85 This connector uses the NIO implementation. The default
86 SSLImplementation will depend on the presence of the APR/native
87 library and the useOpenSSL attribute of the
88 AprLifecycleListener.
89 Either JSSE or OpenSSL style configuration may be used regardless of
90 the SSLImplementation selected. JSSE style configuration is used below.
91 -->
92 <!--
93 <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
94 maxThreads="150" SSLEnabled="true">
95 <SSLHostConfig>
96 <Certificate certificateKeyFile="conf/localhost-rsa.jks"
97 type="RSA" />
98 </SSLHostConfig>
99 </Connector>
100 -->
101 <!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443 with HTTP/2
102 This connector uses the APR/native implementation which always uses
103 OpenSSL for TLS.
104 Either JSSE or OpenSSL style configuration may be used. OpenSSL style
105 configuration is used below.
106 -->
107 <!--
108 <Connector port="8443" protocol="org.apache.coyote.http11.Http11AprProtocol"
109 maxThreads="150" SSLEnabled="true" >
110 <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
111 <SSLHostConfig>
112 <Certificate certificateKeyFile="conf/localhost-rsa-key.pem"
113 certificateFile="conf/localhost-rsa-cert.pem"
114 certificateChainFile="conf/localhost-rsa-chain.pem"
115 type="RSA" />
116 </SSLHostConfig>
117 </Connector>
118 -->
119 <Connector port="8443" protocol="org.apache.coyote.http11.Http11AprProtocol"
120 maxThreads="150" SSLEnabled="true" scheme="https" secure="true" clientAuth="false" sslProtocol="TLS" keystoreFile="
D:\apache-tomcat-8.5.64-windows-x64\apache-tomcat-8.5.64\conf\tomcat.keystore" keystorePass="123456" />
121 -->
122
```

## 5.2.1上传

执行上传命令，如下图：

```

D:\webdavupload>java -jar wedavupload.jar 33.txt D:\webdavupload\ https://localhost:8443/webdav/ null null null
-----文件上传开始-----
getContentType, File ContentType is : text/plain
-----文件上传结束-----
-----文件下载开始-----
Exception in thread "main" java.lang.IllegalArgumentException: Expected URL scheme 'http' or 'https' but no colon was fo
und
    at okhttp3.HttpUrl$Builder.parse$okhttp(HttpUrl.kt:1260)
    at okhttp3.HttpUrl$Companion.get(HttpUrl.kt:1633)
    at okhttp3.Request$Builder.url(Request.kt:184)
    at com.superred.wedavupload.utils.FileDownloadUtil.downloadToServer(FileDownloadUtil.java:39)
    at com.superred.wedavupload.utils.UploadAndDownloadMain.main(UploadAndDownloadMain.java:32)
D:\webdavupload>

```

对于上传命令，只需要关注上图红框里面的日志即可，下方下载异常日志是正常的，无需关注；

上传成功，上传效果图如下：

21.txt	0.1 kb	Tue, 08 Mar 2022 0
22.txt	0.1 kb	Tue, 08 Mar 2022 0
23.txt	0.1 kb	Wed, 09 Mar 2022 0
24.txt	0.1 kb	Wed, 09 Mar 2022 0
25.txt	0.1 kb	Wed, 09 Mar 2022 0
26.txt	12231.5 kb	Wed, 09 Mar 2022 0
28.txt	0.1 kb	Wed, 09 Mar 2022 0
3.txt	0.1 kb	Wed, 09 Mar 2022 0
30.txt	0.1 kb	Wed, 09 Mar 2022 0
31.rar	2786221.2 kb	Wed, 09 Mar 2022 0
32.txt	0.1 kb	Wed, 09 Mar 2022 0
33.txt	0.1 kb	Wed, 09 Mar 2022 0
4.txt	0.1 kb	Sun, 06 Mar 2022 1
5.txt	0.2 kb	Sun, 06 Mar 2022 1
6.txt	0.2 kb	Sun, 06 Mar 2022 1
7.txt	0.2 kb	Mon, 07 Mar 2022 0
8.txt	0.2 kb	Mon, 07 Mar 2022 0
9.txt	0.2 kb	Mon, 07 Mar 2022 0

## 5.2.2下载

执行下载命令，如下图：

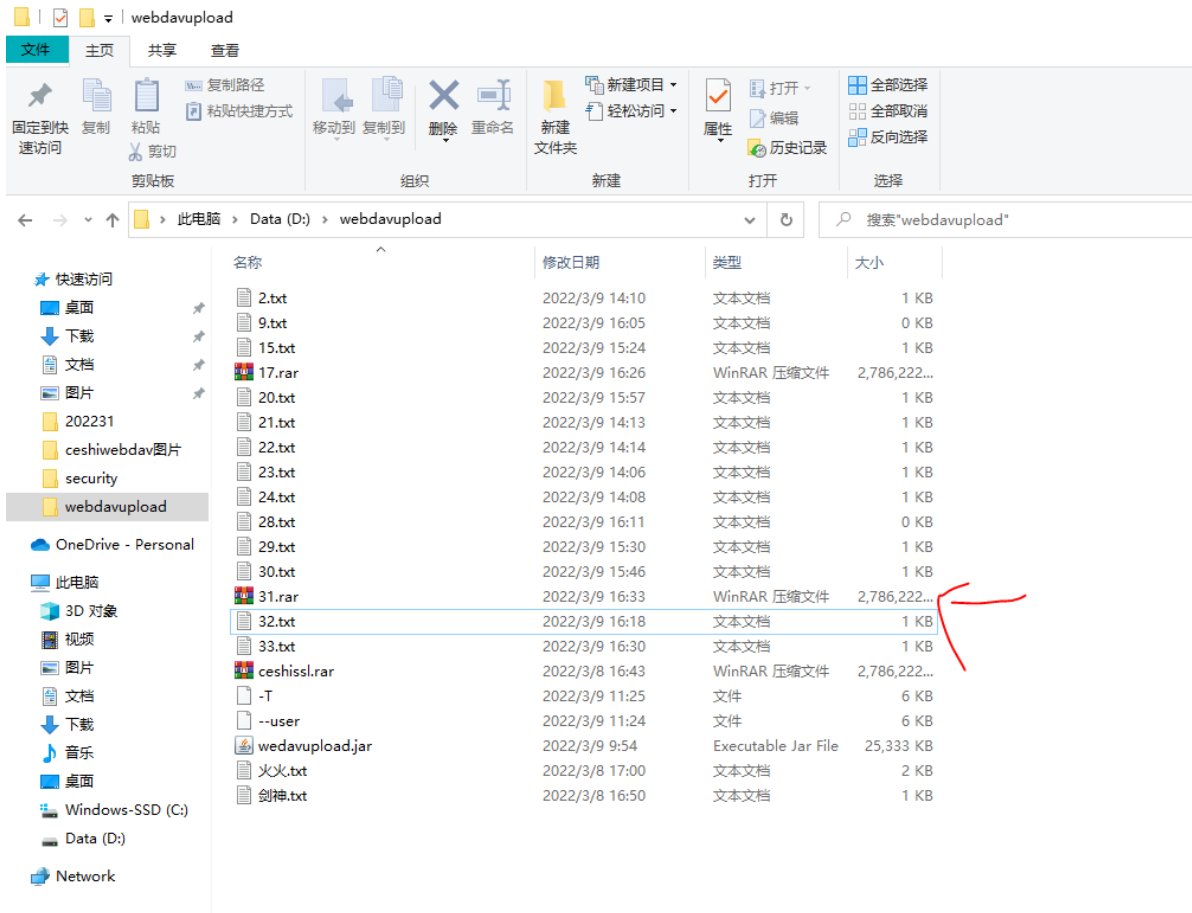
```

D:\webdavupload>java -jar wedavupload.jar null null null 31.rar D:\webdavupload\ https://localhost:8443/webdav/
-----文件上传开始-----
getContentType, File ContentType is : application/octet-stream
16:32:36.444 [main] ERROR com.superred.wedavupload.utils.FileUploadUtil - 上传文件失败:
java.lang.IllegalArgumentException: Expected URL scheme 'http' or 'https' but no colon was found
    at okhttp3.HttpUrl$Builder.parse$okhttp(HttpUrl.kt:1260)
    at okhttp3.HttpUrl$Companion.get(HttpUrl.kt:1633)
    at okhttp3.Request$Builder.url(Request.kt:184)
    at com.superred.wedavupload.utils.FileUploadUtil.uploadFileByHttpClient(FileUploadUtil.java:46)
    at com.superred.wedavupload.utils.UploadAndDownloadMain.main(UploadAndDownloadMain.java:27)
-----文件上传结束-----
-----文件下载开始-----
16:33:14.513 [main] INFO com.superred.wedavupload.utils.FileDownloadUtil - 文件下载成功,文件路径[D:\webdavupload\31.rar]
-----文件下载结束-----
D:\webdavupload>

```

对于下载命令，只需要关注上图红框里面的日志即可，上方上传异常日志是正常的，无需关注；

下载成功，下载效果图如下：



## 六、常见问题

### 6.1 tomcat中https协议配置

如果tomcat中https已经配置完成，可忽略此步

以下为Linux系统中tomcat的https协议配置

我们想要通过<https>访问程序，首先需要获得一个数字证书，自己给自己签发而来的证书也叫自签名ssl证书。

我们这里使用DK自带keytool工具，来创建本地[SSL](https)证书。

#### 6.1.1 找到服务器上面jdk的位置

命令行输入java -verbose可以从最后几行中找到jdk安装位置，如下图：

```
4. 118.145.3.104
[Loaded sun.util.locale.provider.LocaleServiceProviderPool$LocalizedObjectGetter
from /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.nfs.x86_64/jre/lib/rt.jar]
[Loaded java.util.Currency$CurrencyNameGetter from /usr/lib/jvm/java-1.8.0-openj
dk-1.8.0.242.b08-0.nfs.x86_64/jre/lib/rt.jar]
[Loaded sun.util.resources.OpenListResourceBundle from /usr/lib/jvm/java-1.8.0-o
penjdk-1.8.0.242.b08-0.nfs.x86_64/jre/lib/rt.jar]
[Loaded sun.util.resources.LocaleNamesBundle from /usr/lib/jvm/java-1.8.0-openjd
k-1.8.0.242.b08-0.nfs.x86_64/jre/lib/rt.jar]
[Loaded sun.util.resources.CurrencyNames from /usr/lib/jvm/java-1.8.0-openjdk-1.
8.0.242.b08-0.nfs.x86_64/jre/lib/rt.jar]
[Loaded sun.util.resources.zh.CurrencyNames_zh_CN from file:/usr/lib/jvm/java-1.
8.0-openjdk-1.8.0.242.b08-0.nfs.x86_64/jre/lib/ext/localedata.jar]
[Loaded java.util.HashMap$KeySet from /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.
b08-0.nfs.x86_64/jre/lib/rt.jar]
[Loaded java.text.DecimalFormat from /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b
08-0.nfs.x86_64/jre/lib/rt.jar]
[Loaded java.text.DigitList from /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0
.nfs.x86_64/jre/lib/rt.jar]
[Loaded java.math.RoundingMode from /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b0
8-0.nfs.x86_64/jre/lib/rt.jar]
[Loaded java.text.Format$FieldDelegate from /usr/lib/jvm/java-1.8.0-openjdk-1.8.
0.242.b08-0.nfs.x86_64/jre/lib/rt.jar]
[Loaded java.text.FieldPosition$Delegate from /usr/lib/jvm/java-1.8.0-openjdk-1.
8.0.242.b08-0.nfs.x86_64/jre/lib/rt.jar]
[Loaded java.text.NumberFormat$Field from /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.
242.b08-0.nfs.x86_64/jre/lib/rt.jar]
用法: java [-options] class [args...]
      (执行类)
或 java [-options] -jar jarfile [args...]
```

## 6.1.2进入jdk的bin目录，输入以下生成密钥命令：

```
keytool -genkey -v -alias keystorekey -keyalg RSA -validity 3650 -keystore
/opt/RHGL/wlh/wlh_srv/wlh3in1server/tomcat7/conf/tomcat.keystore
```

注意：keytool -genkey：自动使用默认的算法生成公钥和私钥

-alias 名称：给证书取个别名,这里设置的是keystoreKey

-keyalg：制定密钥的算法，如果需要制定密钥的长度，可以再加上keysize参数，密钥长度默认为1024位，使用DSA算法时，密钥长度必须在512到1024之间，并且是64的整数倍

-validity：证书的有效日期，默认是90天，这里设置的3650天

-keystore：参数可以指定密钥库的名称，密钥库其实是存放秘钥和证书文件，会将生成的证书存放到指定的目录下。一般放到tomcat安装路径的conf文件下

输入命令后如下图：

```
[root@localhost java-1.8.0-openjdk-1.8.0.242.b08-0.nfs.x86_64]# pwd
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.nfs.x86_64
[root@localhost java-1.8.0-openjdk-1.8.0.242.b08-0.nfs.x86_64]# keytool -genkey -v -alias keystorekey -keyalg RSA -validity 36
50 -keystore /opt/RHGL/wlh/wlh_srv/wlh3in1server/tomcat7/conf/tomcat.keystorekeytool -genkey -v -alias keystorekey -keyalg RSA
-validity 3650 -keystore /opt/RHGL/wlh/wlh_srv/wlh3in1server/tomcat7/conf/tomcat.keystor
输入密钥库口令： 123456 这里输入的是：123456 长度最少6位
再次输入新口令：
您的名字与姓氏是什么？ 名字与姓氏“是键
名，jdk安装目录下
[Unknown]: 118.145.3.104
您的组织单位名称是什么？
[Unknown]:
您的组织名称是什么？
[Unknown]:
您所在的城市或区域名称是什么？ 非必填
[Unknown]:
您所在的省/市/自治区名称是什么？
[Unknown]:
该单位的双字母国家/地区代码是什么？
[Unknown]:
CN=118.145.3.104, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown 是否正确？
[否]: y
正在为以下对象生成 2,048 位 RSA 密钥对和自签名证书 (SHA256withRSA) (有效期为 3,650 天):
CN=118.145.3.104, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
输入 <keystorekey> 的密钥口令
(如果和密钥库口令相同，按回车): 123456
再次输入新口令：
[正在存储/opt/RHGL/wlh/wlh_srv/wlh3in1server/tomcat7/conf/tomcat.keystor]
Warning:
JKS 密钥库使用专用格式。建议使用 "keytool -importkeystore -srckeystore /opt/RHGL/wlh/wlh_srv/wlh3in1server/tomcat7/conf/tomcat
.keystor -destkeystore /opt/RHGL/wlh/wlh_srv/wlh3in1server/tomcat7/conf/tomcat.keystor -deststoretype pkcs12" 迁移到行业标准格
式 PKCS12.
[root@localhost java-1.8.0-openjdk-1.8.0.242.b08-0.nfs.x86_64]#
```



注意：域名以tomcat地址为例

在核对信息的时候，如果直接Enter，或者输入其它，会重新开始录入信息

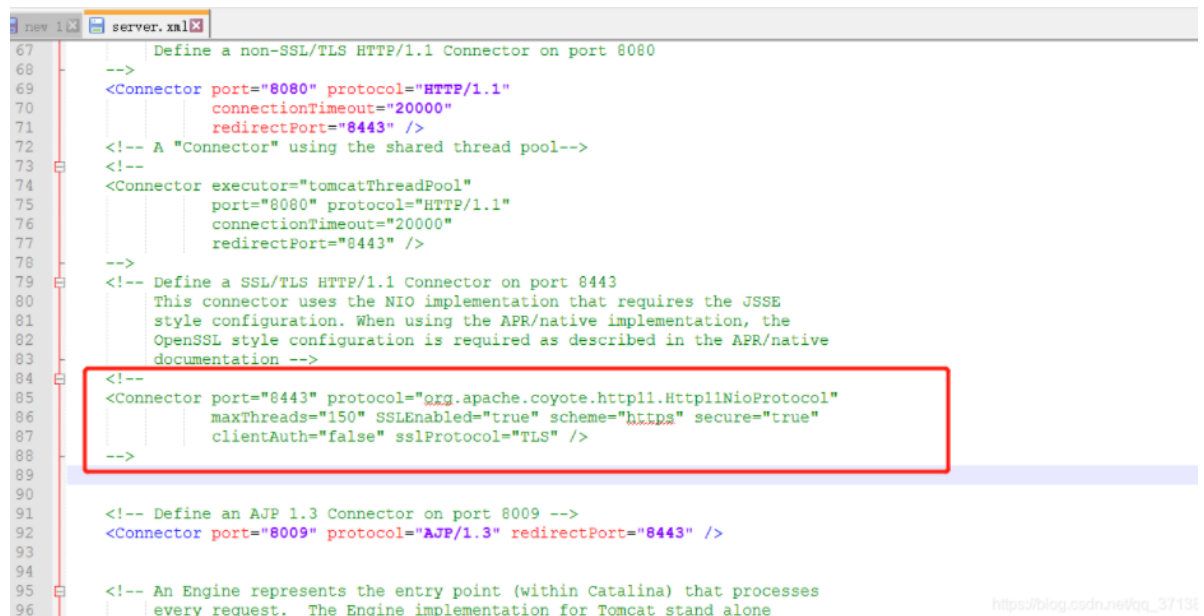
keystore指定证书存放目录必须存在，否则报系统找不到错误，如下：

```
[正在存储E:\conf\tomcat.keystore]
keytool 错误: java.io.FileNotFoundException: E:\conf\tomcat.keystore (系统找不到指定的路径。)
java.io.FileNotFoundException: E:\conf\tomcat.keystore (系统找不到指定的路径。)
    at java.io.FileOutputStream.open0(Native Method)
    at java.io.FileOutputStream.open(FileOutputStream.java:270)
    at java.io.FileOutputStream.<init>(FileOutputStream.java:213)
    at java.io.FileOutputStream.<init>(FileOutputStream.java:101)
    at sun.security.tools.keytool.Main.doCommands(Main.java:1194)
    at sun.security.tools.keytool.Main.run(Main.java:366)
    at sun.security.tools.keytool.Main.main(Main.java:359)
https://blog.csdn.net/qq_37138756
```

进入tomcat目录下，发现证书已经存在

```
[root@localhost tomcat7]# cd conf/
[root@localhost conf]# ls
Catalina      catalina.properties  logback-access.xml  server.keystore  tomcat.keystore  tomcat-users.xsd
catalina.policy context.xml          logback.xml         server.xml       tomcat-users.xml  web.xml
[root@localhost conf]# pwd
/opt/RHGL/wlh/wlh_srv/wlh3in1server/tomcat7/conf
```

6.1.3修改tomcat配置信息，打开tomcat目录下conf/server.xml文件，找到以下代码段



```
67      <!-- Define a non-SSL/TLS HTTP/1.1 Connector on port 8080 -->
68      <!--
69      <Connector port="8080" protocol="HTTP/1.1"
70              connectionTimeout="20000"
71              redirectPort="8443" />
72      <!-- A "Connector" using the shared thread pool-->
73      <!--
74      <Connector executor="tomcatThreadPool"
75              port="8080" protocol="HTTP/1.1"
76              connectionTimeout="20000"
77              redirectPort="8443" />
78      <!--
79      <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
80      This connector uses the NIO implementation that requires the JSSE
81      style configuration. When using the APR/native implementation, the
82      OpenSSL style configuration is required as described in the APR/native
83      documentation -->
84      <!--
85      <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
86              maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
87              clientAuth="false" sslProtocol="TLS" />
88      <!--
89
90
91      <!-- Define an AJP 1.3 Connector on port 8009 -->
92      <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
93
94
95      <!-- An Engine represents the entry point (within Catalina) that processes
96      every request. The Engine implementation for Tomcat stand alone
```

将注释去掉后修改代码如下：这里Https端口设置的是8443(https 访问默认是443端口，HTTP 访问默认是80)

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
           maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS"

           keystoreFile="/opt/RHGL/wlh/wlh_srv/wlh3in1server/tomcat7/conf/tomcat.keystore"
           keystorePass="123456" />
```

增加了keystoreFile 和 keystorePass 两个参数

keystoreFile：表示证书文件的放置位置

keystorePass：证书密钥库设置的密码

注意：redirectPort 端口号和port端口号可以自定义，端口号必须相同



```

-->
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
    port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
-->
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
    This connector uses the NIO implementation that requires the JSSE
    style configuration. When using the APR/native implementation, the
    OpenSSL style configuration is required as described in the APR/native
    documentation -->

<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
    maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="E:\tomcat\apache-tomcat-8.0.53\conf\tomcat.keystore" keystorePass="123456" />

<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />

```

#### 6.1.4验证https:

重启tomcat,之后再浏览器输入<https://118.145.3.104:8443/webdav/>,点击高级,继续前往webdav

## 6.2 只能向https协议下的webdav上传与下载文件

如果只能向https协议下的webdav上传与下载文件,不支持向http上传与下载文件,可能是以下问题:

```

<!-- listings setting. -->
<!--
<!-- If you define welcome files in your own application's web.xml -->
<!-- deployment descriptor, that list *replaces* the list configured -->
<!-- here, so be sure to include any of the default values that you wish -->
<!-- to use within your application. -->

<welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
</welcome-file-list>

<!--强制使用https, http请求会自动转为https -->
<login-config>
    <auth-method>CLIENT-CERT</auth-method>
    <realm-name>Client Cert Users-only Area</realm-name>
</login-config>
<!--配置网站支持https, /* 表示全部请求都走https, transport-guarantee 标签设置为 CONFIDENTIAL以便应,
<security-constraint>
    <web-resource-collection>
        <web-resource-name>SSL</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
</web-app>

```

将图中方框里面代码注释掉即可。文件路径为tomcat安装路径的conf文件夹下。

## 6.3 文件上传时输错https端口号:

```

D:\webdavupload>curl --user admin:admin -T ceshiwebdav.rar https://118.145.3.104:8088/webdav/ceshiwebdav.rar
curl: (35) schannel: next InitializeSecurityContext failed: SEC_E_INVALID_TOKEN (0x80090308) - 给函数提供的标志无效

```

## 6.4 curl下指定协议无法实现webdav文件的上传与下载

## 6.4.1 查询本机jdk支持的版本协议如下图：

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19044.1526]
(c) Microsoft Corporation。保留所有权利。

C:\Users\wlh\Documents\WeChat Files\wxid_suafoxjwyjmi22\FileStorage\File\2022-03>java Test
Supported Protocols: 5
SSLv2Hello
SSLv3
TLSv1
TLSv1.1
TLSv1.2
Enabled Protocols: 4
SSLv3
TLSv1
TLSv1.1
TLSv1.2

C:\Users\wlh\Documents\WeChat Files\wxid_suafoxjwyjmi22\FileStorage\File\2022-03>
```

以下是curl命令：

--tlsv1	使用TLSV1（SSL）
--sslv2	使用SSLV2的（SSL）
--sslv3	使用的SSLV3（SSL）

在https协议下，执行如下文件上传命令：

```
curl --tlsv1 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
curl --tlsv1.1 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
curl --tlsv1.2 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
curl --sslv3 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
```

报错：

```
D:\webdavupload>curl --tlsv1 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

D:\webdavupload>curl --tlsv1.1 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

D:\webdavupload>curl --tlsv1.2 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

D:\webdavupload>curl --sslv3 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
Warning: Ignores instruction to use SSLv3
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

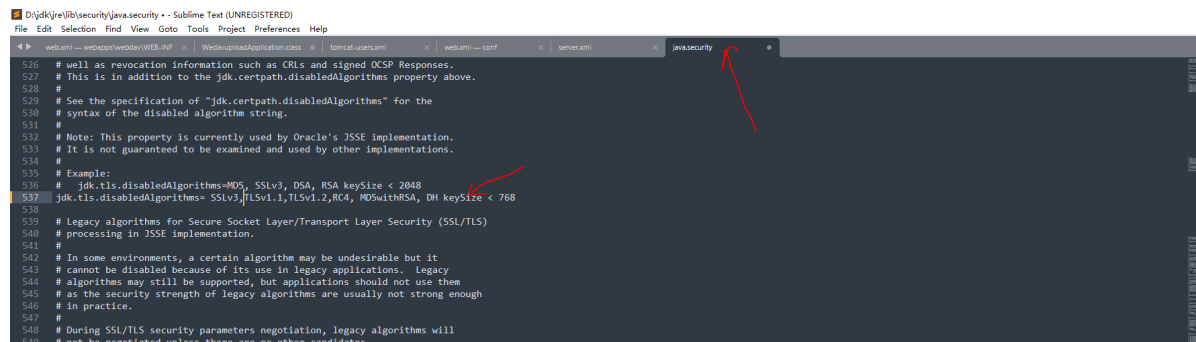
D:\webdavupload>
```

提示证书链不安全，关闭连接，且webdav页面并没有这个文件；

## 6.4.2 jdk禁用只剩一个协议

进入jdk目录: D:\jdk\jre\lib\security\

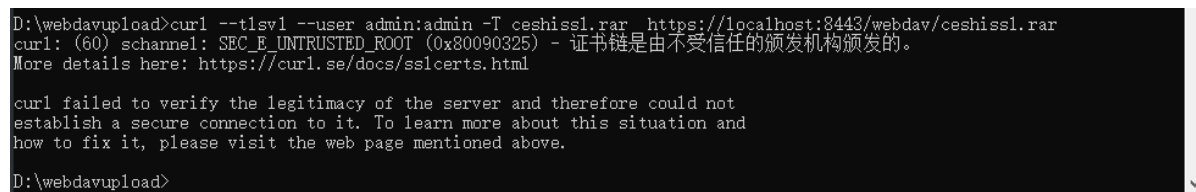
修改java.security文件



```
D:\jdk\jre\lib\security\java.security - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

526 # well as revocation information such as CRLs and signed OCSP Responses.
527 # This is in addition to the jdk.certpath.disabledAlgorithms property above.
528 #
529 # See the specification of "jdk.certpath.disabledAlgorithms" for the
530 # syntax of the disabled algorithm string.
531 #
532 # Note: This property is currently used by Oracle's JSSE implementation.
533 # It is not guaranteed to be examined and used by other implementations.
534 #
535 # Example:
536 # jdk.tls.disabledAlgorithms=MD5, SSLv3, DSA, RSA keySize < 2048
537 jdk.tls.disabledAlgorithms= SSLv3, TLSv1, TLSv1.2, RC4, MD5withRSA, DH keySize < 768
538 #
539 # Legacy algorithms for Secure Socket Layer/Transport Layer Security (SSL/TLS)
540 # processing in JSSE implementation.
541 #
542 # In some environments, a certain algorithm may be undesirable but it
543 # cannot be disabled because of its use in legacy applications. Legacy
544 # algorithms may still be supported, but applications should not use them
545 # as the security strength of legacy algorithms are usually not strong enough
546 # in practice.
547 #
548 # During SSL/TLS security parameters negotiation, legacy algorithms will
549 # not be negotiated unless there are no other candidates.
```

只剩下TLSv1协议, 重启tomcat;



```
D:\webdavupload>curl --tlsv1 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

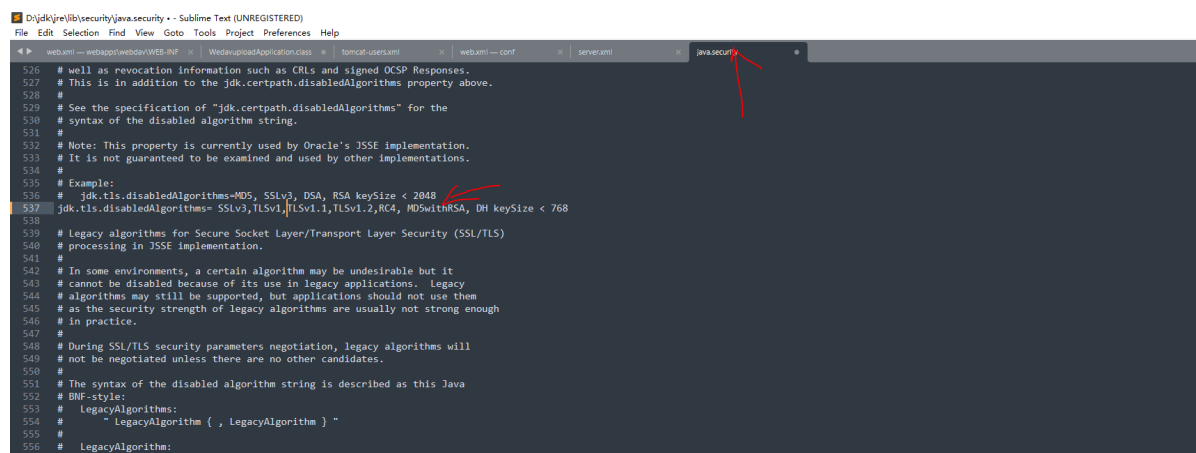
curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

D:\webdavupload>
```

报同样的错误;

## 6.4.3 禁用全部协议

参照6.4.2这次禁用全部协议, 如下图



```
D:\jdk\jre\lib\security\java.security - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

526 # well as revocation information such as CRLs and signed OCSP Responses.
527 # This is in addition to the jdk.certpath.disabledAlgorithms property above.
528 #
529 # See the specification of "jdk.certpath.disabledAlgorithms" for the
530 # syntax of the disabled algorithm string.
531 #
532 # Note: This property is currently used by Oracle's JSSE implementation.
533 # It is not guaranteed to be examined and used by other implementations.
534 #
535 # Example:
536 # jdk.tls.disabledAlgorithms=MD5, SSLv3, DSA, RSA keySize < 2048
537 jdk.tls.disabledAlgorithms= SSLv3, TLSv1, TLSv1.2, RC4, MD5withRSA, DH keySize < 768
538 #
539 # Legacy algorithms for Secure Socket Layer/Transport Layer Security (SSL/TLS)
540 # processing in JSSE implementation.
541 #
542 # In some environments, a certain algorithm may be undesirable but it
543 # cannot be disabled because of its use in legacy applications. Legacy
544 # algorithms may still be supported, but applications should not use them
545 # as the security strength of legacy algorithms are usually not strong enough
546 # in practice.
547 #
548 # During SSL/TLS security parameters negotiation, legacy algorithms will
549 # not be negotiated unless there are no other candidates.
550 #
551 # The syntax of the disabled algorithm string is described as this Java
552 # BNF-style:
553 # LegacyAlgorithms:
554 #   "LegacyAlgorithm { , LegacyAlgorithm } "
555 #
556 # LegacyAlgorithm:
```



```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19044.1526]
(c) Microsoft Corporation。保留所有权利。

C:\Users\wlh\Desktop\工作\202231能打出来支持协议的版本>java Test
Supported Protocols: 5
SSLv2Hello
SSLv3
TLSv1
TLSv1.1
TLSv1.2
Enabled Protocols: 0
C:\Users\wlh\Desktop\工作\202231能打出来支持协议的版本>
```

重启tomcat

### 6.4.3.1禁用全部协议的前提下，使用jar包测试webdav文件的上传与下载

#### 6.4.3.1.1 http协议

```
上传: java -jar wedavupload.jar 28.txt D:\wedavupload\  
http://localhost:8080/webdav/ null null null  
下载: java -jar wedavupload.jar null null null 15.txt D:\wedavupload\  
http://localhost:8080/webdav/
```

具体参数意义请参照5.1

执行结果如下：

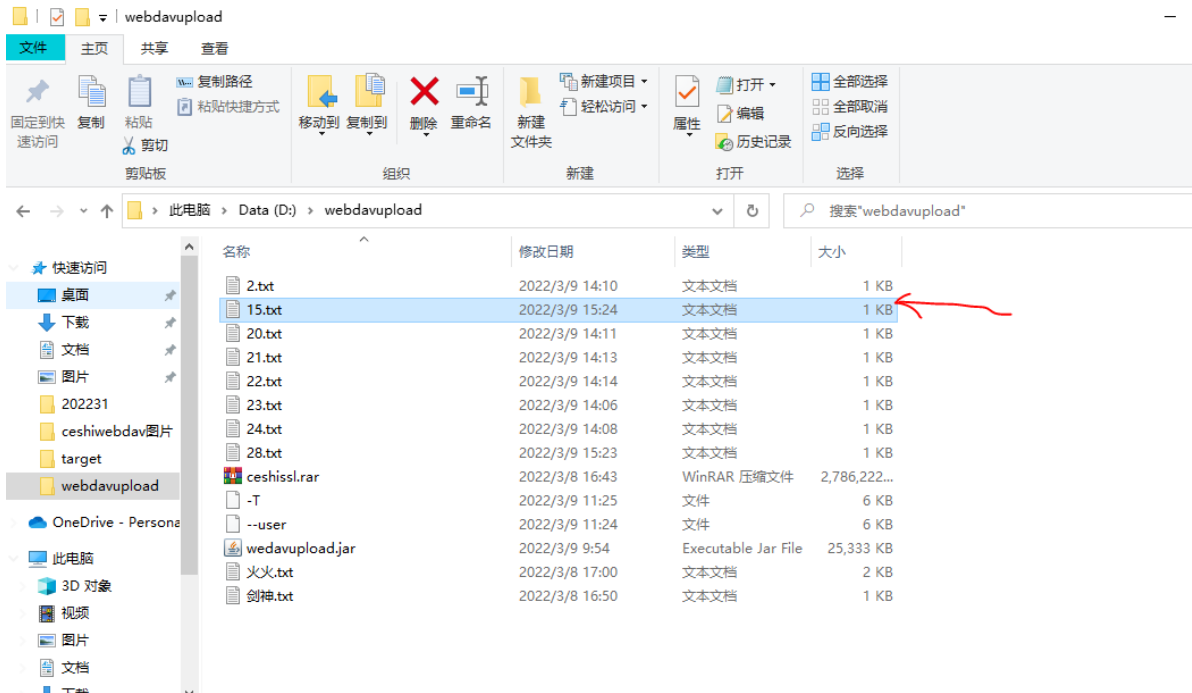
```
D:\wedavupload>java -jar wedavupload.jar 28.txt D:\wedavupload\ http://localhost:8080/webdav/ null null null  
-----文件上传开始-----  
getContentType, File ContentType is : text/plain  
-----文件上传结束-----  
-----文件下载开始-----  
Exception in thread "main" java.lang.IllegalArgumentException: Expected URL scheme 'http' or 'https' but no colon was fo  
und  
    at okhttp3.HttpUrl$Builder.parse$okhttp(HttpUrl.kt:1260)  
    at okhttp3.HttpUrl$Companion.get(HttpUrl.kt:1633)  
    at okhttp3.Request$Builder.url(Request.kt:184)  
    at com.superred.wedavupload.utils.FileDownloadUtil.downloadToServer(FileDownloadUtil.java:39)  
    at com.superred.wedavupload.utils.UploadAndDownloadMain.main(UploadAndDownloadMain.java:32)  
D:\wedavupload>java -jar wedavupload.jar null null null 15.txt D:\wedavupload\ http://localhost:8080/webdav/  
-----文件上传开始-----  
getContentType, File ContentType is : application/octet-stream  
15:24:50.582 [main] ERROR com.superred.wedavupload.utils.FileUploadUtil - 上传文件失败:  
java.lang.IllegalArgumentException: Expected URL scheme 'http' or 'https' but no colon was found  
    at okhttp3.HttpUrl$Builder.parse$okhttp(HttpUrl.kt:1260)  
    at okhttp3.HttpUrl$Companion.get(HttpUrl.kt:1633)  
    at okhttp3.Request$Builder.url(Request.kt:184)  
    at com.superred.wedavupload.utils.FileUploadUtil.uploadFileByHttpClient(FileUploadUtil.java:46)  
    at com.superred.wedavupload.utils.UploadAndDownloadMain.main(UploadAndDownloadMain.java:27)  
-----文件上传结束-----  
-----文件下载开始-----  
15:24:50.657 [main] INFO com.superred.wedavupload.utils.FileDownloadUtil - 文件下载成功,文件路径[D:\wedavupload\15.txt  
]  
-----文件下载结束-----  
D:\wedavupload>
```

从上图可以看到分别执行上传与下载命令，执行效果如下：

上传成功

127.0.0.1:8080/webdav		
15.txt	0.1 kb	Mon, 07 Mar 2022
16.txt	0.1 kb	Mon, 07 Mar 2022
17.rar	2786221.2 kb	Mon, 07 Mar 2022
19.rar	2786221.2 kb	Mon, 07 Mar 2022
2.txt	0.1 kb	Wed, 09 Mar 2022
20.rar	2786221.2 kb	Mon, 07 Mar 2022
21.txt	0.1 kb	Tue, 08 Mar 2022
22.txt	0.1 kb	Tue, 08 Mar 2022
23.txt	0.1 kb	Wed, 09 Mar 2022
24.txt	0.1 kb	Wed, 09 Mar 2022
25.txt	0.1 kb	Wed, 09 Mar 2022
26.txt	12231.5 kb	Wed, 09 Mar 2022
28.txt	0.1 kb	Wed, 09 Mar 2022
3.txt	0.1 kb	Wed, 09 Mar 2022
4.txt	0.1 kb	Sun, 06 Mar 2022
5.txt	0.2 kb	Sun, 06 Mar 2022
6.txt	0.2 kb	Sun, 06 Mar 2022
7.txt	0.2 kb	Mon, 07 Mar 2022

下载成功：



可见，当禁用全部协议时，http协议下是可以使用jar包实现webdav的上传与下载

#### 6.4.3.1.2 https协议

```
上传: java -jar wedavupload.jar 29.txt D:\webdavupload\  
https://localhost:8443/webdav/ null null null  
下载: java -jar wedavupload.jar null null null 16.txt D:\webdavupload\  
https://localhost:8443/webdav/
```

具体参数意义请参照5.2

执行结果如下：

文件上传失败：

```
C:\Windows\System32\cmd.exe
D:\webdavupload>java -jar wedavupload.jar 29.txt D:\webdavupload\ https://localhost:8443/webdav/ null null null
-----文件上传开始-----
getContentType, File ContentType is : text/plain
15:30:36.165 [main] ERROR com.superred.wedavupload.utils.FileUploadUtil - 上传文件失败:
java.net.UnknownServiceException: Unable to find acceptable protocols. isFallback=false, modes=[ConnectionSpec(cipherSui
tes=[TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA2
56, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA38
4, TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256, TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256, TLS_ECDHE_RSA_WITH_AES_12
8_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_256_GCM_SHA384, TLS_RSA
_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA], tlsVersions=[TLS_1_3, TLS_1_2], sup
portsTlsExtensions=true), ConnectionSpec()], supported protocols=[
at okhttp3.internal.connection.ConnectionSpecSelector.configureSecureSocket(ConnectionSpecSelector.kt:63)
at okhttp3.internal.connection.RealConnection.connectTls(RealConnection.kt:373)
at okhttp3.internal.connection.RealConnection.establishProtocol(RealConnection.kt:337)
at okhttp3.internal.connection.RealConnection.connect(RealConnection.kt:209)
at okhttp3.internal.connection.ExchangeFinder.findConnection(ExchangeFinder.kt:226)
at okhttp3.internal.connection.ExchangeFinder.findHealthyConnection(ExchangeFinder.kt:106)
at okhttp3.internal.connection.ExchangeFinder.find(ExchangeFinder.kt:74)
at okhttp3.internal.connection.RealCall.initExchange$okhttp(RealCall.kt:255)
at okhttp3.internal.connection.ConnectInterceptor.intercept(ConnectInterceptor.kt:32)
at okhttp3.internal.http.RealInterceptorChain.proceed(RealInterceptorChain.kt:109)
at okhttp3.internal.cache.CacheInterceptor.intercept(CacheInterceptor.kt:95)
at okhttp3.internal.http.RealInterceptorChain.proceed(RealInterceptorChain.kt:109)
at okhttp3.internal.http.BridgeInterceptor.intercept(BridgeInterceptor.kt:83)
at okhttp3.internal.http.RealInterceptorChain.proceed(RealInterceptorChain.kt:109)
at okhttp3.internal.http.RetryAndFollowUpInterceptor.intercept(RetryAndFollowUpInterceptor.kt:76)
at okhttp3.internal.http.RealInterceptorChain.proceed(RealInterceptorChain.kt:109)
at okhttp3.internal.connection.RealCall.getResponseWithInterceptorChain$okhttp(RealCall.kt:201)
at okhttp3.internal.connection.RealCall.execute(RealCall.kt:154)
at com.superred.wedavupload.utils.FileUploadUtil.UploadFileByHttpClient(FileUploadUtil.java:51)
at com.superred.wedavupload.utils.UploadAndDownloadMain.main(UploadAndDownloadMain.java:27)
Suppressed: java.net.ConnectException: Failed to connect to localhost/127.0.0.1:8443
at okhttp3.internal.connection.RealConnection.connectSocket(RealConnection.kt:297)
at okhttp3.internal.connection.RealConnection.connect(RealConnection.kt:207)
... 16 common frames omitted
Caused by: java.net.ConnectException: Connection refused: connect
at java.net.DualStackPlainSocketImpl.waitForConnect(Native Method)
at java.net.DualStackPlainSocketImpl.socketConnect(DualStackPlainSocketImpl.java:85)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:172)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
at java.net.Socket.connect(Socket.java:589)
at okhttp3.internal.platform.Platform.connectSocket(Platform.kt:120)
at okhttp3.internal.connection.RealConnection.connectSocket(RealConnection.kt:295)
... 17 common frames omitted
-----文件上传结束-----
-----文件下载开始-----
Exception in thread "main" java.lang.IllegalArgumentException: Expected URL scheme 'http' or 'https' but no colon was fo
und
at okhttp3.HttpUrl$Builder.parse$okhttp(HttpUrl.kt:1260)
at okhttp3.HttpUrl$Companion.get(HttpUrl.kt:1633)
at okhttp3.Request$Builder.url(Request.kt:184)
at com.superred.wedavupload.utils.FileDownloadUtil.downloadToServer(FileDownloadUtil.java:39)
at com.superred.wedavupload.utils.UploadAndDownloadMain.main(UploadAndDownloadMain.java:32)

D:\webdavupload>java -jar wedavupload.jar null null null 16.txt D:\webdavupload\ https://localhost:8443/webdav/
-----文件上传开始-----
getContentType, File ContentType is : application/octet-stream
15:31:34.357 [main] ERROR com.superred.wedavupload.utils.FileUploadUtil - 上传文件失败:
java.lang.IllegalArgumentException: Expected URL scheme 'http' or 'https' but no colon was found
at okhttp3.HttpUrl$Builder.parse$okhttp(HttpUrl.kt:1260)
at okhttp3.HttpUrl$Companion.get(HttpUrl.kt:1633)
```

文件下载失败:

```

D:\webdavupload>java -jar webdavupload.jar null null 16.txt D:\webdavupload\ https://localhost:8443/webdav/
-----文件上传开始-----
getContentType, File ContentType is : application/octet-stream
15:31:34.357 [main] ERROR com.superred.webdavupload.utils.FileUploadUtil - 上传文件失败:
java.lang.IllegalArgumentException: Expected URL scheme 'http' or 'https' but no colon was found
    at okhttp3.HttpUrl$Builder.parse$okhttp(HttpUrl.kt:1260)
    at okhttp3.HttpUrl$Companion.get(HttpUrl.kt:1633)
    at okhttp3.Request$Builder.url(Request.kt:184)
    at com.superred.webdavupload.utils.FileUploadUtil.UploadFileByHttpClient(FileUploadUtil.java:46)
    at com.superred.webdavupload.utils.UploadAndDownloadMain.main(UploadAndDownloadMain.java:27)
-----文件上传结束-----
-----文件下载开始-----
Exception in thread "main" java.net.UnknownServiceException: Unable to find acceptable protocols. isFallback=false, mo
s=[ConnectionSpec(cipherSuites=[TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256, TLS_ECDH
ECDH_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECD
RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256, TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA2
, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WIT
AES_256_GCM_SHA384, TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA], tlsVer
ons=[TLS_1_3, TLS_1_2], supportsTlsExtensions=true), ConnectionSpec(), supported protocols=[])
    at okhttp3.internal.connection.ConnectionSpecSelector.configureSecureSocket(ConnectionSpecSelector.kt:63)
    at okhttp3.internal.connection.RealConnection.connectTls(RealConnection.kt:373)
    at okhttp3.internal.connection.RealConnection.establishProtocol(RealConnection.kt:337)
    at okhttp3.internal.connection.RealConnection.connect(RealConnection.kt:209)
    at okhttp3.internal.connection.ExchangeFinder.findConnection(ExchangeFinder.kt:226)
    at okhttp3.internal.connection.ExchangeFinder.findHealthyConnection(ExchangeFinder.kt:106)
    at okhttp3.internal.connection.ExchangeFinder.find(ExchangeFinder.kt:74)
    at okhttp3.internal.connection.RealCall.initExchange$okhttp(RealCall.kt:255)
    at okhttp3.internal.connection.ConnectInterceptor.intercept(ConnectInterceptor.kt:32)
    at okhttp3.internal.http.RealInterceptorChain.proceed(RealInterceptorChain.kt:109)
    at okhttp3.internal.cache.CacheInterceptor.intercept(CacheInterceptor.kt:95)
    at okhttp3.internal.http.RealInterceptorChain.proceed(RealInterceptorChain.kt:109)
    at okhttp3.internal.http.BridgeInterceptor.intercept(BridgeInterceptor.kt:83)
    at okhttp3.internal.http.RealInterceptorChain.proceed(RealInterceptorChain.kt:109)
    at okhttp3.internal.http.RetryAndFollowUpInterceptor.intercept(RetryAndFollowUpInterceptor.kt:76)
    at okhttp3.internal.http.RealInterceptorChain.proceed(RealInterceptorChain.kt:109)
    at okhttp3.internal.connection.RealCall.getResponseWithInterceptorChain$okhttp(RealCall.kt:201)
    at okhttp3.internal.connection.RealCall.execute(RealCall.kt:154)
    at com.superred.webdavupload.utils.FileDownloadUtil.downloadToServer(FileDownloadUtil.java:43)
    at com.superred.webdavupload.utils.UploadAndDownloadMain.main(UploadAndDownloadMain.java:32)
Suppressed: java.net.ConnectException: Failed to connect to localhost/127.0.0.1:8443
    at okhttp3.internal.connection.RealConnection.connectSocket(RealConnection.kt:297)
    at okhttp3.internal.connection.RealConnection.connect(RealConnection.kt:207)
    ... 16 more
Caused by: java.net.ConnectException: Connection refused: connect
    at java.net.DualStackPlainSocketImpl.waitForConnect(Native Method)
    at java.net.DualStackPlainSocketImpl.socketConnect(DualStackPlainSocketImpl.java:85)
    at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
    at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
    at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
    at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:172)
    at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
    at java.net.Socket.connect(Socket.java:589)
    at okhttp3.internal.platform.Platform.connectSocket(Platform.kt:120)
    at okhttp3.internal.connection.RealConnection.connectSocket(RealConnection.kt:295)
    ... 17 more
D:\webdavupload>

```

由报错日志可以看出，支持的协议为空，所以得出结论：**禁用全部协议的前提下，https协议下不允许使用jar包测试webdav文件的上传与下载**

#### 6.4.3.2禁用全部协议的前提下，使用curl测试webdav文件的上传与下载

##### 6.4.3.2.1 http协议

```

上传: curl --user admin:admin -T 30.txt http://127.0.0.1:8080/webdav/30.txt
下载: curl --user admin:admin http://127.0.0.1:8080/webdav/17.rar>17.rar

```

如下图，执行完毕：

```

D:\webdavupload>curl --user admin:admin -T 30.txt http://127.0.0.1:8080/webdav/30.txt
D:\webdavupload>curl --user admin:admin http://127.0.0.1:8080/webdav/17.rar>17.rar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2720M  100 2720M    0     0  401M      0  0:00:06  0:00:06 --:--:-- 374M
D:\webdavupload>

```

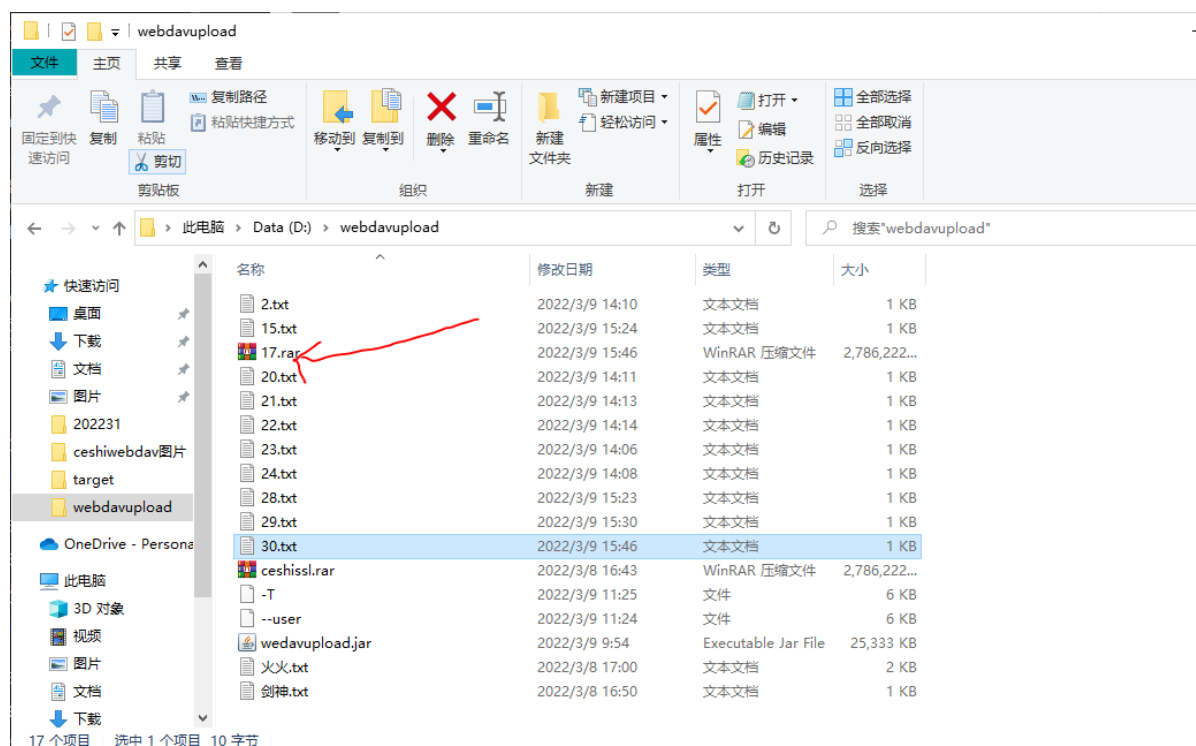
执行效果：

上传成功



19.rar	2786221.2 kb	Mon, 07 Mar 2022 09:48
2.txt	0.1 kb	Wed, 09 Mar 2022 03:00
20.rar	2786221.2 kb	Mon, 07 Mar 2022 10:08
21.txt	0.1 kb	Tue, 08 Mar 2022 01:56
22.txt	0.1 kb	Tue, 08 Mar 2022 01:57
23.txt	0.1 kb	Wed, 09 Mar 2022 03:16
24.txt	0.1 kb	Wed, 09 Mar 2022 03:18
25.txt	0.1 kb	Wed, 09 Mar 2022 03:28
26.txt	12231.5 kb	Wed, 09 Mar 2022 03:27
28.txt	0.1 kb	Wed, 09 Mar 2022 07:24
3.txt	0.1 kb	Wed, 09 Mar 2022 03:09
30.txt	0.1 kb	Wed, 09 Mar 2022 07:46
4.txt	0.1 kb	Sun, 06 Mar 2022 16:29
5.txt	0.2 kb	Sun, 06 Mar 2022 17:33
6.txt	0.2 kb	Sun, 06 Mar 2022 17:44
7.txt	0.2 kb	Mon, 07 Mar 2022 01:58
8.txt	0.2 kb	Mon, 07 Mar 2022 02:00
9.txt	0.2 kb	Mon, 07 Mar 2022 02:00
license.txt	0.1 kb	Tue, 01 Mar 2022 06:18

下载成功



可见：在禁用全部协议的情况下，在http协议下，可以使用curl实现webdav文件的上传与下载

#### 6.4.3.2.2 https协议

指定协议：

上传：

```
curl --tlsv1 --user admin:admin -T ceshissl.rar
https://localhost:8443/webdav/ceshissl.rar
curl --tlsv1.1 --user admin:admin -T ceshissl.rar
https://localhost:8443/webdav/ceshissl.rar
curl --tlsv1.2 --user admin:admin -T ceshissl.rar
https://localhost:8443/webdav/ceshissl.rar
curl --sslsv3 --user admin:admin -T ceshissl.rar
https://localhost:8443/webdav/ceshissl.rar
```



下载:

```
curl -tls1 --user admin:admin https://localhost:8443/webdav/9.txt>9.txt
curl -tls1.1 --user admin:admin https://localhost:8443/webdav/9.txt>9.txt
curl -tls1.2 --user admin:admin https://localhost:8443/webdav/9.txt>9.txt
curl --sslv3 --user admin:admin https://localhost:8443/webdav/9.txt>9.txt
```

上传命令执行完毕, 如下:

```
D:\webdavupload>curl --tls1 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

D:\webdavupload>curl --tls1.1 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

D:\webdavupload>curl --tls1.2 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

D:\webdavupload>curl --sslv3 --user admin:admin -T ceshissl.rar https://localhost:8443/webdav/ceshissl.rar
Warning: Ignores instruction to use SSLv3
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

D:\webdavupload>
```

下载命令执行如下:

```
D:\webdavupload>curl -tls1 --user admin:admin https://localhost:8443/webdav/9.txt>9.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
  0   0   0    0    0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

D:\webdavupload>curl -tls1.1 --user admin:admin https://localhost:8443/webdav/9.txt>9.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
  0   0   0    0    0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

D:\webdavupload>curl -tls1.2 --user admin:admin https://localhost:8443/webdav/9.txt>9.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
  0   0   0    0    0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
curl: (60) schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

D:\webdavupload>curl --sslv3 --user admin:admin https://localhost:8443/webdav/9.txt>9.txt
* Trying 127.0.0.1:8443...
* Trying ::1:8443...
* Connected to localhost (::1) port 8443 (#0)
* schannel: disabled automatic use of client certificate
* schannel: ALPN, offering http/1.1
* schannel: SEC_E_UNTRUSTED_ROOT (0x80090325) - 证书链是由不受信任的颁发机构颁发的。
* Closing connection 0

D:\webdavupload>
```

检查上传与下载目录，如下图并没有找到这两个文件，或者即使存在，文件大小为0kb：

The screenshot shows two windows. The top window is a web browser displaying a file list for a webdav directory. The bottom window is a Windows File Explorer showing the local file system for the 'webdavupload' directory.

**Web Browser File List:**

File Name	Size	Date/Time
25.txt	0.1 kb	Wed, 09 Mar 2022 03:28:31
26.txt	12231.5 kb	Wed, 09 Mar 2022 03:27:31
28.txt	0.1 kb	Wed, 09 Mar 2022 07:24:00
3.txt	0.1 kb	Wed, 09 Mar 2022 03:09:22
30.txt	0.1 kb	Wed, 09 Mar 2022 07:46:11
31.rar	2786221.2 kb	Wed, 09 Mar 2022 07:57:11
4.txt	0.1 kb	Sun, 06 Mar 2022 16:29:00
5.txt	0.2 kb	Sun, 06 Mar 2022 17:33:44
6.txt	0.2 kb	Sun, 06 Mar 2022 17:44:00
7.txt	0.2 kb	Mon, 07 Mar 2022 01:58:31
8.txt	0.2 kb	Mon, 07 Mar 2022 02:00:11
9.txt	0.2 kb	Mon, 07 Mar 2022 02:02:11
license1.txt	0.1 kb	Tue, 01 Mar 2022 06:18:31
test2.md	3.3 kb	Mon, 07 Mar 2022 10:06:11
webdavupload.rar	0.5 kb	Sun, 06 Mar 2022 17:45:00
剑.txt	0.0 kb	Tue, 08 Mar 2022 08:57:11
剑来.txt	0.1 kb	Tue, 08 Mar 2022 08:41:11
和.md	0.1 kb	Mon, 07 Mar 2022 03:25:11
火火.txt	0.1 kb	Tue, 08 Mar 2022 08:57:11

**Windows File Explorer File List:**

Name	Modified Date	Type	Size
2.txt	2022/3/9 14:10	Text Document	1 KB
9.txt	2022/3/9 16:05	Text Document	0 KB
15.txt	2022/3/9 15:24	Text Document	1 KB
20.txt	2022/3/9 15:57	Text Document	1 KB
21.txt	2022/3/9 14:13	Text Document	1 KB
22.txt	2022/3/9 14:14	Text Document	1 KB
23.txt	2022/3/9 14:06	Text Document	1 KB
24.txt	2022/3/9 14:08	Text Document	1 KB
28.txt	2022/3/9 15:23	Text Document	1 KB
29.txt	2022/3/9 15:30	Text Document	1 KB
30.txt	2022/3/9 15:46	Text Document	1 KB
31.rar	2022/3/9 15:46	WinRAR 压缩文件	2,786,222...
ceshissl.rar	2022/3/8 16:43	WinRAR 压缩文件	2,786,222...
-T	2022/3/9 11:25	File	6 KB
..user	2022/3/9 11:24	File	6 KB
wedavupload.jar	2022/3/9 9:54	Executable Jar File	25,333 KB
火火.txt	2022/3/8 17:00	Text Document	2 KB
剑神.txt	2022/3/8 16:50	Text Document	1 KB

所以，在禁用全部协议的前提下，https协议下curl无法使用指定协议实现webdav文件的上传与下载

-k命令参数：

```
上传: curl -k --user admin:admin -T 31.rar https://localhost:8443/webdav/31.rar
下载: curl -k --user admin:admin https://localhost:8443/webdav/20.txt>20.txt
```

如下图，执行完毕：

```
D:\webdavupload>curl -k --user admin:admin -T 31.rar https://localhost:8443/webdav/31.rar
D:\webdavupload>curl -k --user admin:admin https://localhost:8443/webdav/20.txt>20.txt
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 717 100 717 0 0 2757 0 --:--:-- --:--:-- --:--:-- 2757
D:\webdavupload>
```

执行效果如下：

上传成功

https://localhost:8443/webdav			
25.txt	0.1 kb	Wed,	
26.txt	12231.5 kb	Wed,	
28.txt	0.1 kb	Wed,	
3.txt	0.1 kb	Wed,	
30.txt	0.1 kb	Wed,	
31.rar	2786221.2 kb	Wed,	
4.txt	0.1 kb	Sun,	
5.txt	0.2 kb	Sun,	
6.txt	0.2 kb	Sun,	
7.txt	0.2 kb	Mon,	
8.txt	0.2 kb	Mon,	
9.txt	0.2 kb	Mon,	
license1.txt	0.1 kb	Tue,	
test2.md	3.3 kb	Mon,	
webdavupload.rar	0.5 kb	Sun,	

下载成功

webdavupload			
<div> <div>文件 主页 共享 查看</div> <div> <div> <div>固定到快速访问</div> <div>复制</div> <div>粘贴</div> <div>剪贴板</div> </div> <div> <div>复制到快速方式</div> <div>移动到</div> <div>复制到</div> <div>删除</div> <div>重命名</div> </div> <div> <div>新建项目</div> <div>轻松访问</div> <div>新建文件夹</div> </div> <div> <div>打开</div> <div>编辑</div> <div>历史记录</div> </div> <div> <div>全部选择</div> <div>全部取消</div> <div>反向选择</div> </div> </div> </div>			
此电脑 > Data (D:) > webdavupload			
名称	修改日期	类型	大小
2.txt	2022/3/9 14:10	文本文档	1 KB
15.txt	2022/3/9 15:24	文本文档	1 KB
20.txt	2022/3/9 15:57	文本文档	1 KB
21.txt	2022/3/9 14:13	文本文档	1 KB
22.txt	2022/3/9 14:14	文本文档	1 KB
23.txt	2022/3/9 14:06	文本文档	1 KB
24.txt	2022/3/9 14:08	文本文档	1 KB
28.txt	2022/3/9 15:23	文本文档	1 KB
29.txt	2022/3/9 15:30	文本文档	1 KB
30.txt	2022/3/9 15:46	文本文档	1 KB
31.rar	2022/3/9 15:46	WinRAR 压缩文件	2,786,222...
ceshissl.rar	2022/3/8 16:43	WinRAR 压缩文件	2,786,222...
-T	2022/3/9 11:25	文件	6 KB
--user	2022/3/9 11:24	文件	6 KB
webdavupload.jar	2022/3/9 9:54	Executable Jar File	25,333 KB
火火.txt	2022/3/8 17:00	文本文档	2 KB
剑神.txt	2022/3/8 16:50	文本文档	1 KB

可见，在禁用全部协议的前提下，curl可以通过命令参数-k实现webdav文件的上传与下载

### 6.4.4 测试结果表格

	可用协议：ssl3,tls1,tls1.1,tls1.2					禁用全部协议				
	jar包		curl测试			jar包		curl测试		
	http 协议	https 协议	http 协议	https 协议 下-k 命令	https 协议 下指定协 议	http 协议	https 协议	http 协议	https 协议 下-k 命令	https 协议 下指定协 议
上传	可行	可行	可行	可行	不可行	可行	不可行	可行	可行	不可行
下载	可行	可行	可行	可行	不可行	可行	不可行	可行	可行	不可行

## 6.5 下载文件为空

请检查下载命令中的webdav的ip是否正确

## 6.6 出现上传下载异常时输出异常日志

### 6.6.1 上传异常时输出异常日志

上传时对应http协议以及https协议,若要查看日志, 执行命令时加上以下参数中的一个即可

#### 6.6.1.1 -S命令参数

-S 参数指定只输出错误信息, 下面命令没有任何输出, 除非发生错误(注意是大写S)。

http协议对应的上传命令:

```
curl -S --user admin:admin -T 2.txt http://127.0.0.1:8080/webdav/2.txt
```

效果如下:

```
D:\webdavupload>curl -S --user admin:admin -T 2.txt http://127.0.0.1:8080/webdav/2.txt
D:\webdavupload>
```

https协议对应的上传命令

```
curl -S -k --user admin:admin -T 3.txt https://localhost:8443/webdav/3.txt
```

效果如下:

```
D:\webdavupload>curl -S -k --user admin:admin -T 3.txt https://localhost:8443/webdav/3.txt
D:\webdavupload>
```

### 6.6.1.2 -v命令参数

-v 参数输出通信的整个过程，用于调试。

http协议对应的上传命令：

```
curl -v --user admin:admin -T 23.txt http://127.0.0.1:8080/webdav/23.txt
```

效果如下：

```
D:\webdavupload>curl -v --user admin:admin -T 23.txt http://127.0.0.1:8080/webdav/23.txt
* Trying 127.0.0.1:8080...
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
* Server auth using Basic with user 'admin'
* PUT /webdav/23.txt HTTP/1.1
> Host: 127.0.0.1:8080
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.79.1
> Accept: */*
> Content-Length: 10
> Expect: 100-continue
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 100
* We are completely uploaded and fine
* Mark bundle as not supporting multiuse
< HTTP/1.1 201
< Content-Length: 0
< Date: Wed, 09 Mar 2022 03:16:28 GMT
<
* Connection #0 to host 127.0.0.1 left intact
D:\webdavupload>
```

上传成功

https协议对应的上传命令：

```
curl -v -k --user admin:admin -T 24.txt https://localhost:8443/webdav/24.txt
```

效果如下：

```
D:\webdavupload>curl -v -k --user admin:admin -T 24.txt https://localhost:8443/webdav/24.txt
* Trying 127.0.0.1:8443...
* Trying ::1:8443...
* Connected to localhost (::1) port 8443 (#0)
* schannel: disabled automatic use of client certificate
* schannel: ALPN, offering http/1.1
* ALPN, server did not agree to a protocol
* Server auth using Basic with user 'admin'
* PUT /webdav/24.txt HTTP/1.1
> Host: localhost:8443
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.79.1
> Accept: */*
> Content-Length: 6
> Expect: 100-continue
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 100
* We are completely uploaded and fine
* Mark bundle as not supporting multiuse
< HTTP/1.1 201
< Content-Length: 0
< Date: Wed, 09 Mar 2022 03:18:58 GMT
<
* Connection #0 to host localhost left intact
D:\webdavupload>
```

### 6.6.1.3 -i命令参数

**-i** 参数可以显示http response的头信息，连同网页代码一起。

http协议对应的上传命令：

```
curl -i --user admin:admin -T 26.txt http://127.0.0.1:8080/webdav/26.txt
```

效果如下：

```
D:\webdavupload>curl -i --user admin:admin -T 26.txt http://127.0.0.1:8080/webdav/26.txt
HTTP/1.1 100
HTTP/1.1 204
Date: Wed, 09 Mar 2022 03:27:13 GMT

D:\webdavupload>
ip:admin -T 25.txt https://localhost:8443/webdav/25.txt
```

https协议对应的上传命令：

```
curl -i -k --user admin:admin -T 25.txt https://localhost:8443/webdav/25.txt
```

效果如下：

```
D:\webdavupload>curl -i -k --user admin:admin -T 25.txt https://localhost:8443/webdav/25.txt
HTTP/1.1 100
HTTP/1.1 201
Content-Length: 0
Date: Wed, 09 Mar 2022 03:28:52 GMT

D:\webdavupload>
```

## 6.6.2 下载异常时输出异常日志

下载时对应http协议以及https协议,若要查看日志，执行命令时加上以下参数中的一个即可

### 6.6.2.1 -S命令参数

**-S** 参数指定只输出错误信息，下面命令没有任何输出，除非发生错误(注意是大写S)。

http协议对应的下载命令：

```
curl -S --user admin:admin http://127.0.0.1:8080/webdav/23.txt>23.txt
```

效果如下：

```
D:\webdavupload>curl -S --user admin:admin http://127.0.0.1:8080/webdav/23.txt>23.txt
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100    10    100    10    0    0    457      0  --:--:-- --:--:-- --:--:--    500

D:\webdavupload>
```

https协议对应的下载命令

```
curl -S -k --user admin:admin https://localhost:8443/webdav/24.txt>24.txt
```

效果如下：

```
D:\webdavupload>curl -S -k --user admin:admin https://localhost:8443/webdav/24.txt>24.txt
% Total    % Received % Xferd  Average Speed   Time    Time     Time    Current
           Dload  Upload   Total     Spent    Left     Speed
100      6  100      6    0     0    25      0  --:--:-- --:--:-- --:--:--    25

D:\webdavupload>
```

### 6.6.1.2 -v命令参数

-v 参数输出通信的整个过程，用于调试。

http协议对应的下载命令：

```
curl -v --user admin:admin http://127.0.0.1:8080/webdav/2.txt>2.txt
```

效果如下：

```
D:\webdavupload>curl -v --user admin:admin http://127.0.0.1:8080/webdav/2.txt>2.txt
* Trying 127.0.0.1:8080...
% Total    % Received % Xferd  Average Speed   Time    Time     Time    Current
           Dload  Upload   Total     Spent    Left     Speed
0         0    0     0    0     0    0      0  --:--:-- --:--:-- --:--:--    0* Connected to 127.0.0.1 (127.0.0.1) port
8080 (#0)
* Server auth using Basic with user 'admin'
> GET /webdav/2.txt HTTP/1.1
> Host: 127.0.0.1:8080
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.79.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200
< Cache-Control: private
< Expires: Thu, 01 Jan 1970 00:00:00 GMT
< Accept-Ranges: bytes
< ETag: W/"25-1646794999275"
< Last-Modified: Wed, 09 Mar 2022 03:03:19 GMT
< Content-Type: text/plain
< Content-Length: 25
< Date: Wed, 09 Mar 2022 06:10:52 GMT
<
{ [25 bytes data]
100    25  100    25    0     0   629      0  --:--:-- --:--:-- --:--:--   694
* Connection #0 to host 127.0.0.1 left intact

D:\webdavupload>
```

https协议对应的下载命令：

```
curl -v -k --user admin:admin https://localhost:8443/webdav/20.txt>20.txt
```

效果如下：

```
D:\webdavupload>curl -v -k --user admin:admin https://localhost:8443/webdav/20.txt>20.txt
* Trying 127.0.0.1:8443...
% Total    % Received % Xferd  Average Speed   Time    Time     Time    Current
           Dload  Upload   Total     Spent    Left     Speed
0         0    0     0    0     0    0      0  --:--:-- --:--:-- --:--:--    0* Trying ::1:8443...
* Connected to localhost (::1) port 8443 (#0)
0         0    0     0    0     0    0      0  --:--:-- --:--:-- --:--:--    0* schannel: disabled automatic use of client
certificate
* schannel: ALPN, offering http/1.1
* ALPN, server did not agree to a protocol
* Server auth using Basic with user 'admin'
> GET /webdav/20.txt HTTP/1.1
> Host: localhost:8443
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.79.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 404
< Cache-Control: private
< Expires: Thu, 01 Jan 1970 00:00:00 GMT
< Content-Type: text/html; charset=utf-8
< Content-Language: zh-CN
< Content-Length: 717
< Date: Wed, 09 Mar 2022 06:11:22 GMT
<
{ [717 bytes data]
100   717  100   717    0     0  2826      0  --:--:-- --:--:-- --:--:--  2868
* Connection #0 to host localhost left intact

D:\webdavupload>
```

### 6.6.1.3 -i命令参数

-i 参数可以显示http response的头信息，连同网页代码一起。

http协议对应的下载命令：

```
curl -i --user admin:admin http://127.0.0.1:8080/webdav/21.txt>21.txt
```

效果如下：

```
D:\webdavupload>curl -i --user admin:admin http://127.0.0.1:8080/webdav/21.txt>21.txt
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100    12    100    12    0    0    515      0  --:--:-- --:--:-- --:--:--    545
```

https协议对应的下载命令：

```
curl -i -k --user admin:admin https://localhost:8443/webdav/22.txt>22.txt
```

效果如下：

```
D:\webdavupload>curl -i -k --user admin:admin https://localhost:8443/webdav/22.txt>22.txt
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100    31    100    31    0    0   121      0  --:--:-- --:--:-- --:--:--   121
D:\webdavupload>
```

## 七、如何定位现场问题？

### 7.1 是否终端有输出日志

无论是使用jar包还是用curl来实现文件的上传与下载，遇到问题，首先第一步看终端是否有日志，根据日志定位问题，如下图所示：

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19044.1526]
(c) Microsoft Corporation。保留所有权利。

D:\webdavupload>curl -k --user admin:admin -T 31.rar https://localhost:8443/webdav/31.rar
curl: (7) Failed to connect to localhost port 8443 after 2229 ms: Connection refused
D:\webdavupload>
```

### 7.2 查看命令行是否输入正确

如果第一步没有日志，那么需要参照四、使用curl测试webdav文件的上传与下载 五、使用jar包测试webdav文件的上传与下载检查输入的命令行中，文件名，webdav的路径url,ip和端口是否输入正确

### 7.3 输出日志

参照6.6出现上传下载异常时输出异常日志，逐个使用命令参数，看打出来的日志是否有异常

### 7.4 postman模拟命令的发出

如果使用命令参数仍然没有发现日志异常，那么可以使用postman来模拟命令行的输入；如果还不行可以联系技术人员排查问题



