



链滴

java 实现 PDF 文件下载到服务器

作者: [Stone](#)

原文链接: <https://ld246.com/article/1481017749236>

来源网站: [链滴](#)

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

目前负责的项目遇到下载pdf文件到本地的需求，有http请求和https请求发个帖记录一下！

1.http请求pdf文件地址并下载到本地，别的不太会说，直接粘代码了

```
package cn.com.test;

import java.io.DataInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;

import javax.net.ssl.HttpURLConnection;

/**
 *
 * @ClassName: HttpDownloaderPDF
 * @Description: 下载pdf文件到本地
 * @author Stone
 * @date 2016年12月1日
 *
 */
public class HttpDownloaderPDF {

    String remoteFileUrl = ""; // pdf文件请求地址
    String localFilePath = ""; // 本地存放pdf文件路径

    public void httpDownloader(String remoteFileUrl, String localFilePath) {
        try {
            URL url = new URL(remoteFileUrl);

            HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection(
;
            httpURLConnection.setConnectTimeout(5 * 1000); // 5000 自定义连接超时毫秒数
            httpURLConnection.connect(); // 连接
            System.out.println("connect URL success!");

            int fileLength = httpURLConnection.getContentLength();
            System.out.println("file size: " + (fileLength / 1024.0) + " KB");

            System.out.println("start the download...");
            try (DataInputStream dis = new DataInputStream(
                httpURLConnection.getInputStream());
                FileOutputStream fos = new FileOutputStream(localFilePath)) {
                byte[] buf = new byte[10240]; // 根据实际情况自定义 buf 大小
                for (int readSize; (readSize = dis.read(buf)) > 0;) {
                    fos.write(buf, 0, readSize);
                }
                System.out.println("download is complete!");
            } catch (IOException ex) {
```

```

        System.out.println("download is error");
    }

    httpURLConnection.disconnect();
} catch (IOException ex) {
    ex.printStackTrace();
    System.out.println("URL Does not exist or connection timeout!");
}
}
}

class PDFTest {
    public static void main(String[] args) {
        HttpDownloaderPDF hdl = new HttpDownloaderPDF();
        hdl.httpDownloader("requestURL", "/home/Stone/Downloader/file.pdf");
    }
}

```

2.有http请求就应该考虑https请求，还是简单粗暴直接粘代码

```

package cn.com.test;

import java.io.DataInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.security.cert.X509Certificate;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;

/**
 *
 * @ClassName: HttpDownloaderPDF
 * @Description: 下载pdf文件到本地
 * @author Stone
 * @date 2016年12月1日
 *
 */
public class HttpDownloaderPDF {

    /**
     * 验证证书
     */

```

```

final static HostnameVerifier DO_NOT_VERIFY = new HostnameVerifier() {
    public boolean verify(String hostname, SSLSession session) {
        return true;
    }
};

private static void trustAllHosts() {
    // 创建信任管理器跳过证书校验
    TrustManager[] trustAllCerts = new TrustManager[] { new X509TrustManager() {
        public java.security.cert.X509Certificate[] getAcceptedIssuers() {
            return new java.security.cert.X509Certificate[] {};
        }
    }
    public void checkClientTrusted(X509Certificate[] chain, String authType) {

    }
    public void checkServerTrusted(X509Certificate[] chain, String authType) {

    }
};

// 安装 all-trusting 信任管理
try {
    SSLContext sc = SSLContext.getInstance("TLS");
    sc.init(null, trustAllCerts, new java.security.SecureRandom());
    HTTPSURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
} catch (Exception e) {
    e.printStackTrace();
}

}

public void httpDownloader(String remoteFileUrl, String localFilePath) {
    try {
        trustAllHosts();
        URL url = new URL(remoteFileUrl);

        HTTPSURLConnection https = (HTTPSURLConnection) url.openConnection();
        https.setHostnameVerifier(DO_NOT_VERIFY);

        HTTPURLConnection httpURLConnection = https;
        httpURLConnection.setConnectTimeout(5 * 1000); // 5000 自定义连接超时毫秒数
        httpURLConnection.connect(); // 连接
        System.out.println("connect URL success!");

        int fileLength = httpURLConnection.getContentLength();
        System.out.println("file size: " + (fileLength / 1024.0) + " KB");

        System.out.println("start the download...");
        try (DataInputStream dis = new DataInputStream(
            httpURLConnection.getInputStream());
            FileOutputStream fos = new FileOutputStream(localFilePath)) {
            byte[] buf = new byte[10240]; // 根据实际情况自定义 buf 大小
            for (int readSize; (readSize = dis.read(buf)) > 0;) {
                fos.write(buf, 0, readSize);
            }
        }
    }
}

```

```
        System.out.println("download is complete!");
    } catch (IOException ex) {
        System.out.println("download is error");
    }

    httpURLConnection.disconnect();
} catch (IOException ex) {
    ex.printStackTrace();
    System.out.println("URL Does not exist or connection timeout!");
}
}
}

class PDFTest {
    public static void main(String[] args) {
        HttpDownloaderPDF hdl = new HttpDownloaderPDF();
        hdl.httpDownloader("requestURL", "/home/Stone/Downloader/file.pdf");
    }
}
```

注意：记得在服务器自定义目录下（不要在需要权限的目录下创建，很麻烦还要授权）创建你存文件文件夹，不然的话程序顺利执行完，但是你要的文件不一定能下载成功！就好比告诉我，石头哥哥上我请你吃饭记得来我家哈！说的挺好听，你不告诉我你家地址和门牌号，我怎么知道去哪里找你，以啊写程序也得多点儿真诚少点儿套路0.0

版权声明：本文为本人原创文章，转载请注明出处[java实现PDF文件下载到服务器](#)谢谢！