



链滴

Android 使用 URLConnection 下载音频文件

作者: [RustFisher](#)

原文链接: <https://ld246.com/article/1569332636769>

来源网站: [链滴](#)

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

本文链接: [Android 使用URLConnection下载音频文件](#)

使用MediaPlayer播放在线音频, 请参考[Android MediaPlayer 播放音频](#)

有时候我们会需要下载音频文件。这里提供一种思路, 将在线音频文件通过流写到本地文件中。使用URLConnection来建立连接, 获取到的数据写到文件中。

URLConnection建立连接后, 可以获取到数据长度。由此我们可以计算出下载进度。

```
public class DownloadStreamThread extends Thread {
    String urlStr;
    final String targetFileAbsPath;

    public DownloadStreamThread(String urlStr, String targetFileAbsPath) {
        this.urlStr = urlStr;
        this.targetFileAbsPath = targetFileAbsPath;
    }

    @Override
    public void run() {
        super.run();
        int count;
        File targetFile = new File(targetFileAbsPath);
        try {
            boolean n = targetFile.createNewFile();
            Log.d(TAG, "Create new file: " + n + ", " + targetFile);
        } catch (IOException e) {
            Log.e(TAG, "run: ", e);
        }
        try {
            URL url = new URL(urlStr);
            URLConnection connection = url.openConnection();
            connection.connect();
            int contentLength = connection.getContentLength();
            InputStream input = new BufferedInputStream(url.openStream());
            OutputStream output = new FileOutputStream(targetFileAbsPath);

            byte[] buffer = new byte[1024];
            long total = 0;
            while ((count = input.read(buffer)) != -1) {
                total += count;
                Log.d(TAG, String.format(Locale.CHINA, "Download progress: %.2f%%", 100 * (total / (double) contentLength)));
                output.write(buffer, 0, count);
            }
            output.flush();
            output.close();
            input.close();
        } catch (Exception e) {
            Log.e(TAG, "run: ", e);
        }
    }
}
```

启动下载，即启动线程。

```
new DownloadStreamThread(urlStr, targetFileAbsPath).start();
```

值得注意的是，如果本地已经有了文件，需要做一些逻辑判断。例如是否删掉旧文件，重新下载。或判断出已有文件，中止此次下载任务。

例如可以用`connection.getContentLength()`与当前文件长度来比较，如果不一致，则删掉本地文件重新下载。

实际上，`URLConnection`能处理很多流媒体。在这里是用来下载音频文件。可以实现下载功能和类“边下边播”的功能。

代码可以参考示例工程：<https://github.com/RustFisher/android-MediaPlayer>