

Java C# GZip 字符串压缩解压

作者: [88250](#)

原文链接: <https://ld246.com/article/1480569906672>

来源网站: [链滴](#)

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

压缩：将指定的原字符串用 gzip 算法压缩，然后以 BASE64 编码

解压：将指定的 BASE64 编码的字符串用 gzip 解压，返回原字符串

原字符串为 UTF-8 编码。

Java 版本

导入包

基本都是 JDK 内置的包，BASE64 部分可能需要替换一下（JDK8 已经自带 BASE64）。

```
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.zip.GZIPInputStream;
import java.util.zip.GZIPOutputStream;
import org.apache.commons.codec.binary.Base64;
```

实现部分

```
/**
 * 使用 gzip 进行压缩.
 *
 * @param str 压缩前的文本
 * @return 压缩后的文本（BASE64 编码）
 * @throws IOException 如果解压异常
 */
public static String gzip(final String str) throws IOException {
    if (str == null || "".equals(str)) {
        return str;
    }

    String ret = null;

    byte[] compressed;
    ByteArrayOutputStream out = null;
    GZIPOutputStream zout = null;
    try {
        out = new ByteArrayOutputStream();
        zout = new GZIPOutputStream(out);
        zout.write(str.getBytes());
        zout.close();
        compressed = out.toByteArray();
        ret = new String(Base64.encodeBase64(compressed), "UTF-8");
    } finally {
        if (out != null) {
            try {
                out.close();
            } catch (final IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

    }
}

return ret;
}

/**
 * 使用 gzip 进行解压缩.
 *
 * @param compressedStr 压缩后的文本 (BASE64 编码)
 * @return 解压后的文本
 * @throws IOException 如果解压异常
 */
public static String ungzip(final String compressedStr) throws IOException {
    if (null == compressedStr || "".equals(compressedStr)) {
        return compressedStr;
    }

    String ret = null;

    ByteArrayOutputStream out = null;
    ByteArrayInputStream in = null;
    GZIPInputStream zin = null;
    try {
        final byte[] compressed = Base64.decodeBase64(compressedStr);
        out = new ByteArrayOutputStream();
        in = new ByteArrayInputStream(compressed);
        zin = new GZIPInputStream(in);
        final byte[] buffer = new byte[1024];
        int offset = -1;
        while ((offset = zin.read(buffer)) != -1) {
            out.write(buffer, 0, offset);
        }

        ret = out.toString("UTF-8");
    } finally {
        if (zin != null) {
            try {
                zin.close();
            } catch (final IOException e) {
                e.printStackTrace();
            }
        }
        if (in != null) {
            try {
                in.close();
            } catch (final IOException e) {
                e.printStackTrace();
            }
        }
        if (out != null) {
            try {
                out.close();
            } catch (final IOException e) {

```

```
        e.printStackTrace();
    }
}

return ret;
}
```

C# 版本

命名空间

```
using System;
using System.Text;
using System.IO;
using System.IO.Compression;
```

实现部分

```
/// <summary>
/// 使用 gzip 进行压缩
/// </summary>
/// <param name="str">压缩前的文本</param>
/// <returns>压缩后的文本 (BASE64 编码) </returns>
public static string gzip(string str)
{
    if (null == str || "".Equals(str))
    {
        return str;
    }

    byte[] buffer = Encoding.UTF8.GetBytes(str);
    MemoryStream ms = new MemoryStream();
    using (GZipStream zip = new GZipStream(ms, CompressionMode.Compress, true))
    {
        zip.Write(buffer, 0, buffer.Length);
    }

    return Convert.ToBase64String(ms.ToArray());
}

/// <summary>
/// 使用 gzip 进行解压缩
/// </summary>
/// <param name="compressedStr">压缩后的文本 (BASE64 编码) </param>
/// <returns>解压后的文本</returns>
public static string ungzip(string compressedStr)
{
    if (null == compressedStr || "".Equals(compressedStr))
    {
        return compressedStr;
    }
}
```

```
using (var compressedStream = new MemoryStream(Convert.FromBase64String(compressedStr)))
using (var zipStream = new GZipStream(compressedStream, CompressionMode.Decompress))
using (var resultStream = new MemoryStream())
{
    var buffer = new byte[4096 * 2];
    int read;

    while ((read = zipStream.Read(buffer, 0, buffer.Length)) > 0)
    {
        resultStream.Write(buffer, 0, read);
    }

    return Encoding.UTF8.GetString(resultStream.ToArray());
}
}
```

Unity

如果 C# 版要用在 Unity 上，则需要安装[这个](#)免费插件，再将引入的命名空间从 `System.IO.Compression` 改为 `Unity.IO.Compression` 即可。

姊妹篇：<https://hacpai.com/article/1481701879422>