link: null
title: 珠峰架构师成长计划
description: 可以使用zlib模块进行压缩及解压缩处理,压缩文件以后可以减少体积，加快传输速度和节约带宽
代码
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=19 sentences=61, words=344

1. 压缩与解压缩处理

可以使用 `zlib`模块进行压缩及解压缩处理,压缩文件以后可以减少体积，加快传输速度和节约带宽 代码 (https://github.com/zhufengnodejs/static-server/tree/master/lesson/zlib)

2. 压缩对象

压缩和解压缩对象都是一个可读可写流

方法 说明 zlib.createGzip 返回Gzip流对象，使用Gzip算法对数据进行压缩处理 zlib.createGunzip 返回Gzip流对象，使用Gzip算法对压缩的数据进行解压缩处理 zlib.createDeflate 返回Deflate流对象，使用Deflate算法对数据进行压缩处理 zlib.createInflate 返回Deflate流对象，使用Deflate算法对数据进行解压缩处理

3. 压缩和解压缩

```
var zlib = require('zlib');
var fs = require('fs');

function zip(src) {
    var gzip = zlib.createGzip();
    var inputStream = fs.createReadStream(src);
    var outputStream = fs.createWriteStream(src+'.gz');
    inputStream.pipe(gzip).pipe(outputStream);
}
zip('source.txt');

function unzip(src){
    var gunzip = zlib.createGunzip();
    var inputStream = fs.createReadStream(src);
    var outputStream = fs.createWriteStream(src.slice(0,-3));
    inputStream.pipe(gunzip).pipe(outputStream);
}

gnzip('source.txt.gz');
```

4. 在http中的应用

```
var zlib = require('zlib');
var fs = require('fs');
var http = require('http');
http.createServer(function (request, response) {
    var raw = fs.createReadStream('.' + request.url);
    var acceptEncoding = request.headers['accept-encoding'];
    if (!acceptEncoding) {
        acceptEncoding = '';
    }
    if (acceptEncoding.match(/\bdeflate\b/)) {
        response.setHeader('Content-Encoding','deflate');
        raw.pipe(zlib.createDeflate()).pipe(response);
    } else if (acceptEncoding.match(/\bgzip\b/)) {
        response.setHeader('Content-Encoding','gzip');
        raw.pipe(zlib.createGzip()).pipe(response);
    } else {
        raw.pipe(response);
    }
}).listen(9090)
```

```
var zlib = require('zlib');
var fs = require('fs');
var http = require('http');

var request = http.get({
    host: 'localhost',
    path: '/index.html',
    port: 9090,
    headers: {
        'accept-encoding': 'gzip,deflate'
    }
})

request.on('response', function (response) {
    var output = fs.createWriteStream('test.txt');
    switch (response.headers['content-encoding']) {
        case 'gzip':
            response.pipe(zlib.createGunzip()).pipe(output);
            break;
        case 'deflate':
            response.pipe(zlib.createInflate()).pipe(output);
            break;
        default:
            response.pipe(output);
            break;
    }
});
request.end();
```

5. 方法调用

```javascript
var zlib = require('zlib');
var fs = require('fs');

var out = fs.createWriteStream('input.log');
var input = 'input';
zlib.gzip(input, function (err, buffer) {
    if (!err) {
        zlib.unzip(buffer, function (err, buffer) {
            if (!err) {
                console.log(buffer.toString());
                out.end(buffer);
            }
        })
    }
})
```