

link: null
title: 珠峰架构师成长计划
description: ## 1. 静态资源服务器命令行工具
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=33 sentences=67, words=479

1. 静态资源服务器命令行工具

- 该选项指定下载字节的范围，常应用于分块下载文件
- range 的表示方式有多种，如100-500，则指定从100开始的400个字节数据；-500表示最后的500个字节；5000-表示从第5000个字节开始的所有字节
- 另外还可以同时指定多个字节块，中间用","分开
- 服务器告诉客户端可以使用range response.setHeader('Accept-Ranges', 'bytes')
- Server通过请求头中的 Range: bytes=0-xxx来判断是否是做Range请求，如果这个值存在而且有效，则只发回请求的那部分文件内容，响应的状态码变成206,如果无效，则返回416状态码，表明Request Range Not Satisfiable

```
curl -r 0-1024 -o music.mp3
```

2. 多语言切换

可以通过Accept-Language检测浏览器的语言

- 请求头格式 Accept-Language: Accept-Language:zh-CN,zh;q=0.9
- 响应头格式 Content-Language:zh-CN

```
let http = require('http');
let pack = {
  en: {
    title: 'hello'
  },
  cn: {
    title: '欢迎'
  }
}

function request(req, res) {
  let acceptLangulage = req.headers['accept-language'];
  let lan = 'en';
  if (acceptLangulage) {
    lan = acceptLangulage.split(',').map(item => {
      let values = item.split(';');
      return {
        name: values[0].split('-')[0],
        q: isNaN(values[1]) ? 1 : parseInt(values[1])
      }
    }).sort((lan1, lan2) => lan1.q - lan2.q).shift().name;
  }
  res.end(pack[lan] ? pack[lan].title : pack['en'].title);
}

let server = http.createServer();
server.on('request', request);
server.listen(8080);
```

3. 图片防盗链

- 从一个网站跳转，或者网页引用到某个资源文件时，HTTP请求中带有Referer表示来源网页的URL
- 通过检查请求头中的Referer来判断来源网页的域名
- 如果来源域名不在白名单内，则返回错误提示
- 用浏览器直接访问图片地址是没有referer的

```

let http = require('http');
let url = require('url');
let path = require('path');
let fs = require('fs');
let root = path.join(__dirname, 'public');

function removePort(host) {
  return host.split(':')[0]
}

function getHostName(urlAddr) {
  let {
    host
  } = url.parse(urlAddr);
  return removePort(host);
}

function request(req, res) {
  let refer = req.headers['referer'] || req.headers['referrer'];
  if (refer) {
    let referHost = getHostName(refer);
    let host = removePort(req.headers['host']);
    if (referHost !== host) {
      sendForbidden(req, res);
    } else {
      serve(req, res);
    }
  } else {
    serve(req, res);
  }
}

function serve(req, res) {
  let {
    pathname
  } = url.parse(req.url);
  let filepath = path.join(root, pathname);
  console.log(req.url, filepath);

  fs.stat(filepath, (err, stat) => {
    if (err) {
      res.end('Not Found');
    } else {
      fs.createReadStream(filepath).pipe(res);
    }
  });
}

function sendForbidden(req, res) {
  res.end('防盗链');
}

let server = http.createServer();
server.on('request', request);
server.listen(8080);

```

4. 代理服务器

代理（英语：Proxy），也称网络代理，是一种特殊的网络服务，允许一个网络终端（一般为客户端）通过这个服务与另一个网络终端（一般为服务器）进行非直接连接。一些网关、路由器等网络设备具备网络代理功能。一般认为代理服务有利于保障网络终端的隐私或安全，防止攻击。

```
npm install http-proxy --save
```

- web 代理普通的HTTP请求
- listen port
- close 关闭内置的服务

```

let httpProxy = require('http-proxy');
let http = require('http');
let proxy = httpProxy.createProxyServer();

http.createServer(function (req, res) {
  proxy.web(req, res, {
    target: 'http://localhost:8000'
  });
  proxy.on('error', function (err) {
    console.log('出错了');
    res.end(err.toString());
  });
}).listen(8080);

```

5. 虚拟主机

通过Host实现多个网站共用一个端口,多个网站共用一个服务器

```

let http = require('http');
let httpProxy = require('http-proxy');
let proxy = httpProxy.createProxyServer();

let hosts = {
  'a.zfpx.cn': 'http://localhost:8000',
  'b.zfpx.cn': 'http://localhost:9000'
}

let server = http.createServer(function (req, res) {
  let host = req.headers['host'];
  host = host.split(':')[0];
  let target = hosts[host];
  proxy.web(req, res, {
    target
  });
}).listen(80);

```

6. User-Agent

User-Agent中文名为用户代理，简称 UA，它是一个特殊字符串头，使得服务器能够识别客户使用的操作系统及版本、CPU 类型、浏览器及版本、浏览器渲染引擎、浏览器语言、浏览器插件等。

- 请求头 `User-Agent:Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36`
- `user-agent-parser`