

HTTP 模块

Node.js 全解

版权声明

本内容版权属杭州饥人谷教育科技有限公司（简称饥人谷）所有。

任何媒体、网站或个人未经本网协议授权不得转载、链接、转贴，或以其他方式复制、发布和发表。

已获得饥人谷授权的媒体、网站或个人在使用时须注明「资料来源：饥人谷」。

对于违反者，饥人谷将依法追究法律责任。

联系方式

如果你想要购买本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

如果你发现有人盗用本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

HTTP 协议

理论上我们应该先学习这个协议再学习 http 模块

不管了，直接干！

在实践中学习

TypeScript

理论上应该先看一下 TS 文档

不管了，直接干！

干完了再看文档

一些有用的工具

- node-dev

- ✓ 当文件更新时自动重启的 node
- ✓ 避免每次改完代码都要重新运行的麻烦
- ✓ 不宜在生产环境使用

- ts-node

- ✓ 让 node 支持直接运行 TypeScript 代码
- ✓ 不宜在生产环境使用

- ts-node-dev

- ✓ 这个工具结合了上面两个工具
- ✓ 可以用 TypeScript 开发 Node.js 程序，且会自动重启
- ✓ 不宜在生产环境使用，但非常适合用来学习

一些有用的工具2

- WebStorm 自动格式化

- ✓ Reformat Code 设置为 Ctrl + L 或者其他键位
- ✓ 可以快速帮你格式化 TS / JS 代码

- WebStorm 自动写代码

- ✓ Complete Current Statement 设为 Ctrl + Enter
- ✓ Show Context Actions 设为 Alt + Enter
- ✓ 把冲突的键位改成其他快捷键
- ✓ 当然你可以设为其他键位，这只是我的习惯

- VSCode 配置

- ✓ 开启自动保存，开启保存时自动格式化（有些人不喜欢）
- ✓ 当然也可以 Shift + Alt + F 手动格式化
- ✓ Quick Fix 默认快捷键 Ctrl + 句号（功能稍弱）

一些有用的工具3

• WebStorm Git 配置

- ✓ 把 WebStorm 的 git 路径设置为 cmd 里的
- ✓ 示例：D:\Fang\Software\cmd\vendor\git-for-windows\bin\git.exe
- ✓ 好处是统一管理统一配置

• VSCode Git 配置

- ✓ 在 settings.json 里添加

```
"git.enabled": true,  
"git.path": "D:\\... \\git-for-windows\\cmd\\git.exe",  
"terminal.integrated.shell.windows": "D:\\... \\git-for-windows\\bin\\bash.exe",
```
- ✓ 目的是让 VSCode 使用 cmd 里的 git 和 bash
- ✓ 你如果没装 cmd 就自行修改为其他路径

一些有用的工具3

- curl

- ✓ GET 请求: `curl -v url`
- ✓ POST 请求: `curl -v -d "name=frank" url`
- ✓ 设置请求头: `-H 'Content-Type: application/json'`
- ✓ 设置动词: `-X PUT`
- ✓ JSON 请求: `curl -d '{"name":"bob"}' -H 'Content-Type: application/json' url`
- ✓ 后面会用到 curl 来构造请求

yarn global add ts-node-dev

或者 npm i -g ts-node-dev

创建项目

• 步骤

- ✓ yarn init -y
- ✓ 新建 index.ts
- ✓ 使用命令行或者WebStorm启动
- ✓ yarn add --dev @types/node 安装 node 声明文件
- ✓ 引入 http 模块 (WebStorm 自动导入)
- ✓ 用 http 创建 server (WebStorm 自动命名)
- ✓ 监听 server 的 request 事件 (可以简写)
- ✓ server.listen(8888) 开始监听 8888 端口
- ✓ 使用 curl -v <http://localhost:8888> 发请求

server 是个什么东西

- http.Server 类的实例

- ✓ 根据文档知道 [http.createServer](#) 的返回值的类型
- ✓ 所以 server 是 http.Server 的实例
- ✓ 因此 server 拥有[几个事件和方法](#)
- ✓ 其中也就 request 事件和 listen() 方法目前能用上

- 继承 net.Server 类

- ✓ 根据文档知道 [http.Server](#) 继承了 [net.Server](#)
- ✓ 隐私 server 有拥有了[几个事件和方法](#)
- ✓ 其中也就 error 事件和 address() 方法目前能用上

- 这就是看文档的技巧

用 Node.js 获取请求内容

- get 请求

- ✓ request.method 获取请求动词
- ✓ request.url 获取请求路径（含查询参数）
- ✓ request.header 获取请求头
- ✓ get 请求一般没有消息体/请求体

- post 请求

- ✓ curl -v -d "name=frank" <http://localhost:8888>
- ✓ request.on('data', fn) 获取消息体
- ✓ request.on('end', fn) 拼接消息体

(request, response) 是啥

- 找类

- ✓ 根据文档，request 是 `http.IncomingMessage` 的实例
- ✓ 根据文档，response 是 `http.ServerResponse` 的实例

- Request

- ✓ 拥有 `headers`、`method`、`url` 等属性
- ✓ 从 `stream.Readable` 类继承了 `data/end/error` 事件
- ✓ 为什么不能直接拿到请求的消息体呢？跟 TCP 有关

- Response

- ✓ 拥有 `getHeader/setHeader/end/write` 等方法
- ✓ 拥有 `statusCode` 属性，可读可写
- ✓ 继承了 `Stream`，目前用不上

根据 url 返回不同的文件

目标一

处理查询参数

目标2

提示：使用 `url.parse`

匹配任意文件

目标3

处理不存在的文件

目标4

提示：返回一个404页面

处理非 GET 请求

目标5

提示：405 Method Not Allowed

添加缓存选项

目标6

提示：[Cache-Control](#)

响应内容启用 gzip

目标7，可选

提示：[zlib 文档](#)

对比业界优秀案例

http-server / node-static

找到不足，超过它们

发布到 npm

不要浪费你的代码，持续维护三个月，可以换 WebStorm

发布代码

- 步骤

- ✓ 首先要把 TypeScript 变成 JS
- ✓ 使用 tsc 命令可以做到（全局安装 typescript）
- ✓ 然后把 js 作为 package.json 里的 main 字段
- ✓ 测试 JS 是否可用
- ✓ npm adduser; npm publish（或者用 yarn login）

免费正版 JetBrains 全家桶

• 申请条件

- ✓ 你是开源项目的核心贡献者
- ✓ 你的项目是开源的（如 MIT 许可证）
- ✓ 你的项目是不盈利的
- ✓ 你的项目至少活跃了 3 个月
- ✓ 你定期发布更新的版本
- ✓ 满足条件就[点击申请按钮](#)吧

• 填写材料

- ✓ 邮箱地址一定不要填错
- ✓ No. of required licenses 处填1
- ✓ 项目描述用 Google 翻译就行
- ✓ 正版 JetBrains 全家桶唾手可得，价值 \$649/年

总结

- 学习方法

- ✓ 先定一个小目标
- ✓ CRM 完事

- 本节课了解了

- ✓ http 模块
- ✓ request / response 两个对象
- ✓ url 模块
- ✓ zlib 模块
- ✓ HTTP 状态码
- ✓ TypeScript !

再见

下节课继续干