

01Node基础

01Node基础

课程大纲

课前思考

什么是Node

官方介绍

通俗一点

Node的诞生

创始人

历程

Node的实现

Node在web中的用途

环境配置

安装包方式

如何更新node版本

问题

Nvm方式

步骤

环境变量

Windows常用命令行

常见问题

如何运行

案例

最大对象

案例

常用模块

fs（文件系统）

fs.readFile

fs.writeFile

案例

http

创建服务

监听链接

端口

服务器响应

get

post

案例

querystring（查询字符串）

字符串转对象

url

urlObject组成

字符串转url对象

课程大纲

- Node 是什么？带来了什么？能用来做什么？
- 如何搭建一个Node的开发和生产环境
- 快速上手使用Node，操作入门
- 常见的和辛没款和对象的使用
- 服务器端开发的基本概念以及如何通过Node实现
- 常用的快速开发Node应用程序框架
- 数据库的概念和如何通过Node操作数据库
- 综合项目实战

课前思考

- 什么是JavaScript？
- JavaScript的运行环境？
- 浏览器中的JavaScript可以做什么？
- 浏览器中的JavaScript不可以做什么？
- JavaScript只可以运行在浏览器中吗？
- 编程语言的能力取决于什么？（平台）

什么是Node

官方介绍



Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境。
Node.js 使用了一个事件驱动、非阻塞式 I/O 的模型，使其轻量又高效。
Node.js 的包管理器 npm，是全球最大的开源库生态系统。

通俗一点

- Node就是JavaScript语言在服务器端的 **运行环境**
- 运行环境有两层
 1. 意思首先，JavaScript 语言通过 Node 在服务器运行，在这个意义上，Node 有点像 JavaScript 虚拟机；
 2. 其次，Node 提供大量工具库，使得 JavaScript 语言与操作系统互动（比如读写文件、新建子进程），在这个意义上，Node 又是 JavaScript 的工具库。

Node的诞生

创始人

Ryan Dahl

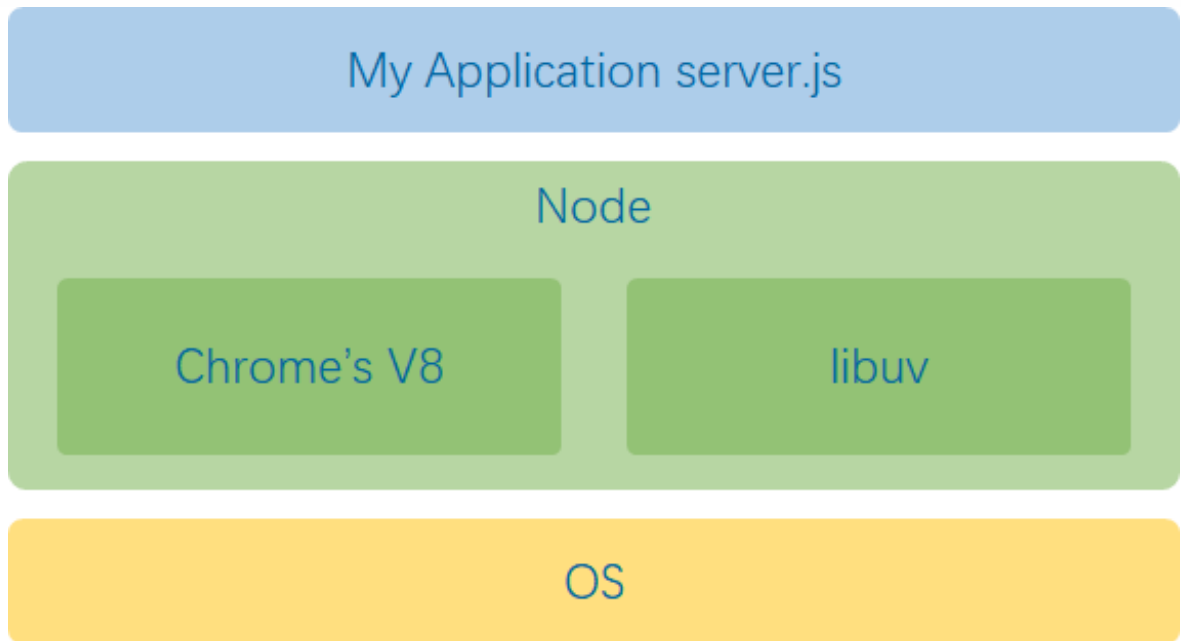


历程

- 2008年左右，随着 AJAX 的逐渐普及，Web 开发逐渐走向复杂化，系统化；
- 2009年2月，Ryan Dahl 想要创建一个轻量级，适应现代 Web 开发的平台；
- 2009年5月，Ryan Dahl 在 GitHub 中开源了最初版本，同年11月的 JSConf 就安排了 Node 讲座；
- 2010年底，Joyent 公司资助，Ryan Dahl 也加入了该公司，专门负责 Node 的开发；
- 2011年7月，在微软的支持下登陆 Windows 平台；

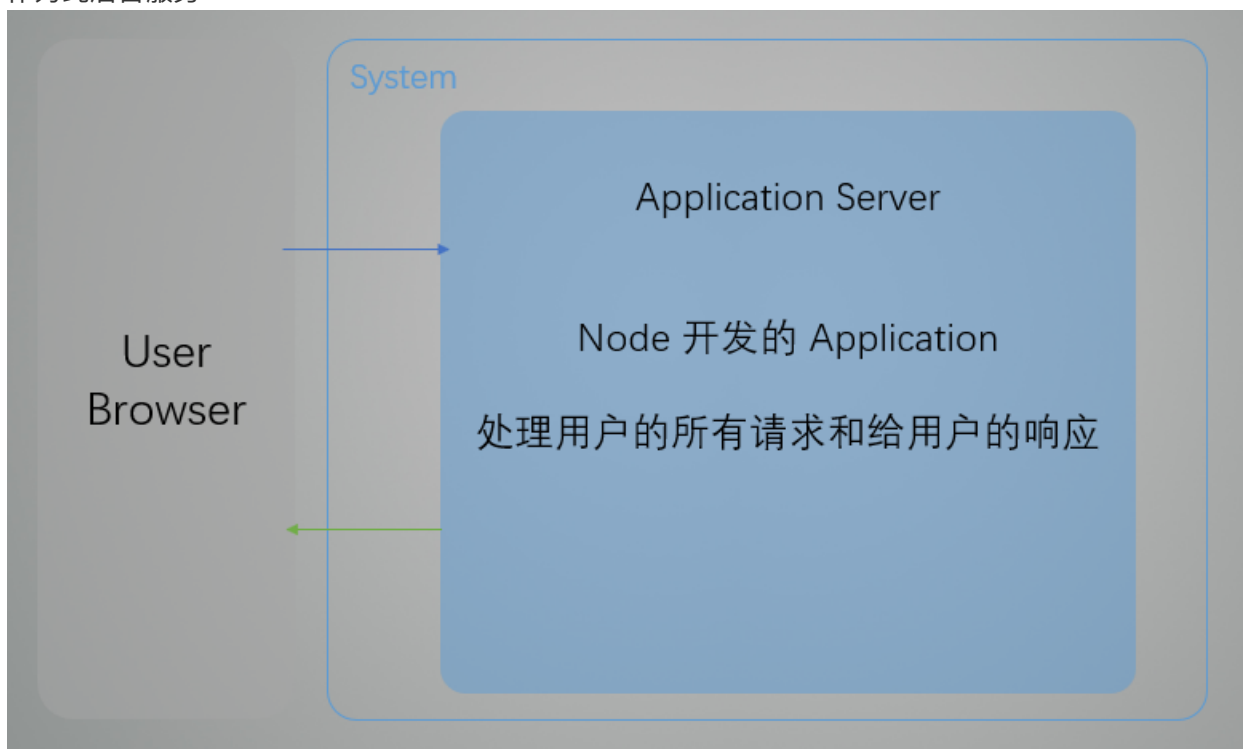
Node的实现

- Node 内部采用 Google Chrome 的 V8 引擎，作为 JavaScript 语言解释器；
- 通过自行开发的 libuv 库，调用操作系统资源。

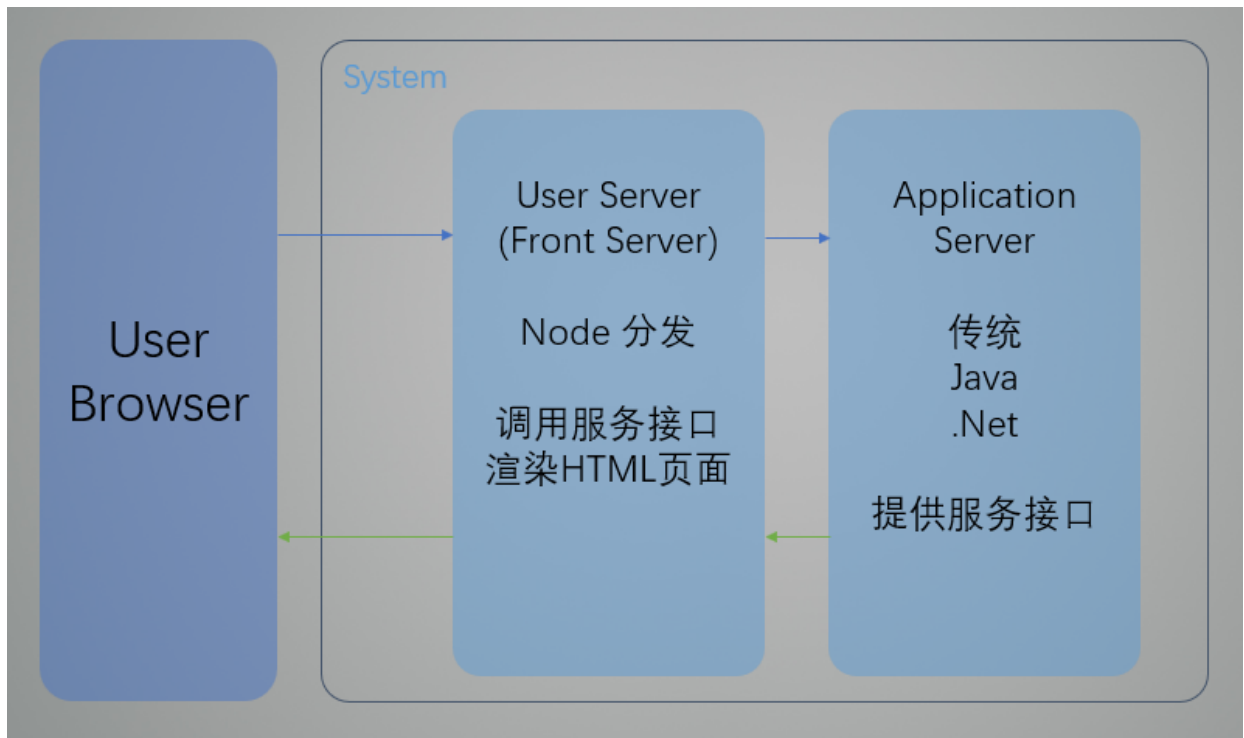


Node在web中的用途

作为纯后台服务



作为中间服务



环境配置

安装包方式

如何更新node版本

1. 重新下载新的安装包
2. 覆盖安装

问题

1. 以前版本安装的很多全局的工具包需要重新安装
2. 无法回滚到之前的版本
3. 无法在多个版本之间切换（很多时候我们要使用特定版本）

Nvm方式

步骤

1. nvm 下载

nvm 的下载地址：<https://github.com/coreybutler/nvm-windows/releases>。选择第一个 nvm-noinstall.zip，然后解压在系统盘（一般开发相关的文件我都放C盘，但是放别的盘也是可以的）。我放的目录路径是C:\dev\nvm。解压出来的文件有：

- elevate.cmd
- elevate.vbs
- install.cmd
- LICENSE
- nvm.exe

2. nvm 安装 双击 install.cmd，是以控制台形式显示的，第一下直接按回车，然后会在C盘根目录产生settings.txt，把这个文件放进刚刚解压的那个目录，然后修改settings.txt内容，改成下面那样：

```
root: C:\dev\nvm
path: C:\dev\nodejs
arch: 64
proxy: none
node_mirror: http://npm.taobao.org/mirrors/node/
npm_mirror: https://npm.taobao.org/mirrors/npm/
```

但是有些人很不幸，这个方法行不通，因为打开 install.cmd按下回车后，显示拒绝访问注册表路径，并弹出一个settings.txt。这时候，你只要淡定地叉掉那个文本以及控制台，然后在刚刚的目录里新建一个文件settings.txt，最后把上面的内容复制进去就可以了。

root： nvm的存放地址 path： 存放指向node版本的快捷方式，使用nvm的过程中会自动生成。一般写的时候与nvm同级。 arch： 电脑系统是64位就写64,32位就写32 proxy： 代理

3. nvm 配置 以控制台方法执行成功的，在环境变量里会自动配置了 NVM_HOME 和 NVM_SYMLINK，这时候只要修改相应的路径就行了。直接创建settings文件的可以在环境变量里创建 NVM_HOME 和 NVM_SYMLINK，并添加路径

```
NVM_HOME: C:\dev\nvm
NVM_SYMLINK: C:\dev\nodejs
```

在PATH里加上;%NVM_HOME%;%NVM_SYMLINK%;。一键控制台install的还要检查 环境变量 PATH 上的路径有没有添加C:\dev\nvm以及C:\dev\nodejs,有的话就删掉。

4. 检测安装结果 打开控制台，输入： nvm -v,若是出现版本信息，则安装。若报错，那就重新把步骤再捋一遍。 检查环境变量是否配置成功：可以在控制台输入： set [环境变量名]，查看路径是否填写错误

5. 使用node 控制台下载 => 输入： nvm install [版本号]，下载最新版的可以直接输nvm install latest 下载完成后，在控制台输入： nvm use [版本号]。即使用这个版本号的node了。在use后，上面所说的nodejs文件夹就自动生成了。

环境变量

- 环境变量就是操作系统提供的系统级别用于存储变量的地方
- 系统变量和用户变量
- 环境变量的变量名是不区分大小写的
- 特殊值：
PATH 变量:只要添加到 PATH 变量中的路径，都可以在任何目录下搜索

Windows常用命令行

- 切换当前目录（change directory）： cd
- 创建目录（make directory）： mkdir
- 查看当前目录列表（directory）： dir

- 别名：ls (list)
- 清空当前控制台：cls
- 别名：clear
- 删除文件：del
- 别名：rm
- 注意：所有别名必须在新版本的 PowerShell 中使用
- pwd：获取当前路径

常见问题

如何运行

node js文件路径 例如：node hello.js 一般后缀名可以省略 node hello

案例

1. 01hello.js

最大对象

在node中都是ECMAScript代码和node lib uv模块代码 没有浏览器中的BOM、DOM 在node中最大对象是global

案例

1. 02最大对象.html

常用模块

fs（文件系统）

浏览器中的 JavaScript 是没有文件操作的能力的 但是 Node 中的 JavaScript 具有文件操作的能力 fs 是 file-system 的简写，就是文件系统的样子 在 Node 中如果想要进行文件操作，就必须引入 fs 这个核心模块 在 fs 这个核心模块中，就提供了所有的文件操作相关的 API

```
const fs = require('fs');
```

fs.readFile

异步地读取一个文件的全部内容。

```
fs.readFile('/etc/passwd', (err, data) => {  
  if (err) throw err;  
  console.log(data);  
});
```

fs.writeFile

异步地写入数据到文件，如果文件已经存在，则覆盖文件。data 可以是字符串或 buffer。

```
fs.writeFile('文件名.txt', 'Node.js 中文网', (err) => {  
  if (err) throw err;  
  console.log('文件已保存! ');  
});
```

案例

1. 03读文件.js
2. 04写文件.js

http

HTTP 服务器与客户端

创建服务

`http.createServer`

监听链接

`server.listen` 开启HTTP服务器监听连接。

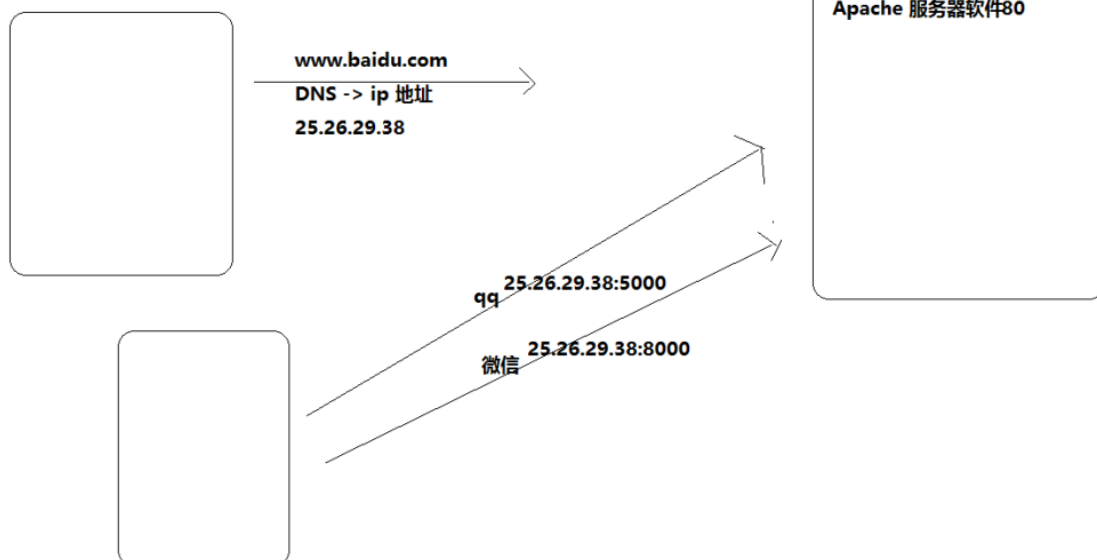
端口

所有联网的程序都需要进行网络通信

计算机中只有一个物理网卡，而且同一个局域网中，网卡的地址必须是唯一的。

网卡是通过唯一的 ip 地址来进行定位的。

IP 地址用来定位计算机
端口号用来定位具体的应用程序
(所有需要联网通信的软件都必须具有端口号)



服务器响应

request 客户端请求

response 服务器响应

get

用户传来的get数据在url里面，所以应该去url取数据

post

当用户上传的数据非常多时，需要分段传输，所以当数据传输完成之后才能获得所有数据

案例

1. 05创建服务
2. 06服务器响应
3. 07get
4. 08post
5. 09登录注册

querystring（查询字符串）

querystring 模块提供了一些实用函数，用于解析与格式化 URL 查询字符串。可以通过以下方式使用：

```
const querystring = require('querystring');
```

字符串转对象

[querystring.parse](#) 该方法会把一个 URL 查询字符串 str 解析成一个键值对的集合。

url

url 模块提供了一些实用函数，用于 URL 处理与解析。可以通过以下方式使用：

```
const url = require('url');
```

urlObject组成

href							
protocol	auth		host		path		hash
			hostname	port	pathname	search	
						query	
"	https:	//	user	:	pass	@ sub.host.com : 8080	/p/a/t/h ? query=string #hash "
			hostname	port			
protocol	username	password	host				
origin			origin		pathname	search	hash
href							

字符串转url对象

[url.parse](#) url.parse() 方法会解析一个 URL 字符串并返回一个 URL 对象。