

# Node.js 与猪八戒前端基础设施介绍

万孝东

Node.js

由 Ryan Dahl 于 2009 年创造

# 基于 Chrome V8 引擎的 JavaScript 执行环境

开源、跨平台

非阻塞的异步 IO

单线程 (主线程)

事件驱动

# Node.js 和浏览器端的 JavaScript 区别

## Node.js

- 没有 dom
- 没有浏览器相关的 api 接口
- 可以访问文件系统
- 可以和网络层进行交互 (tcp / udp)
- 可以基于 N-API 写原生扩展

# Node.js 适合的场景

- web 开发
- 命令行应用和工具
- 桌面应用 (electron)
- 嵌入式 / IoT

# Node.js 不适合的地方

- 如果你不喜欢异步模型
- 如果你想对内存做精细化的控制
- 如果你的应用是 CPU / GPU 密集型的 (worker 线程? )

# Node.js 异步处理

- callback
- promise
- async / await

# Node.js 与 NPM

Node.js JavaScript 生态系统的包管理工具

包安装 / 版本 管理 / 依赖管理

随 Node.js 一起发布

# Node.js 在猪八戒网



# Node.js 在猪八戒网

## 第一阶段

FE

静态资源

back end

PHP

smarty

# Node.js 在猪八戒网

## 第二阶段

FE

静态资源

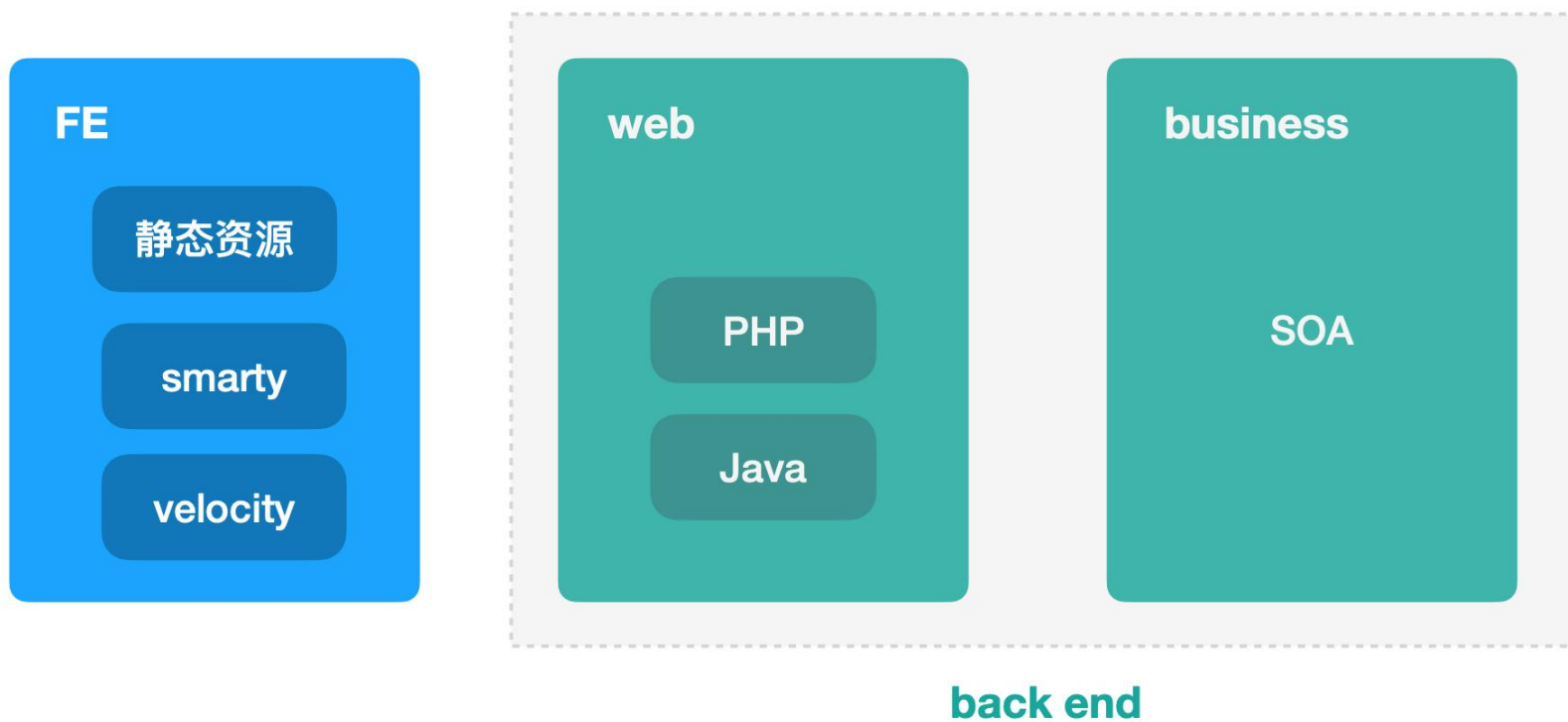
smarty

back end

PHP

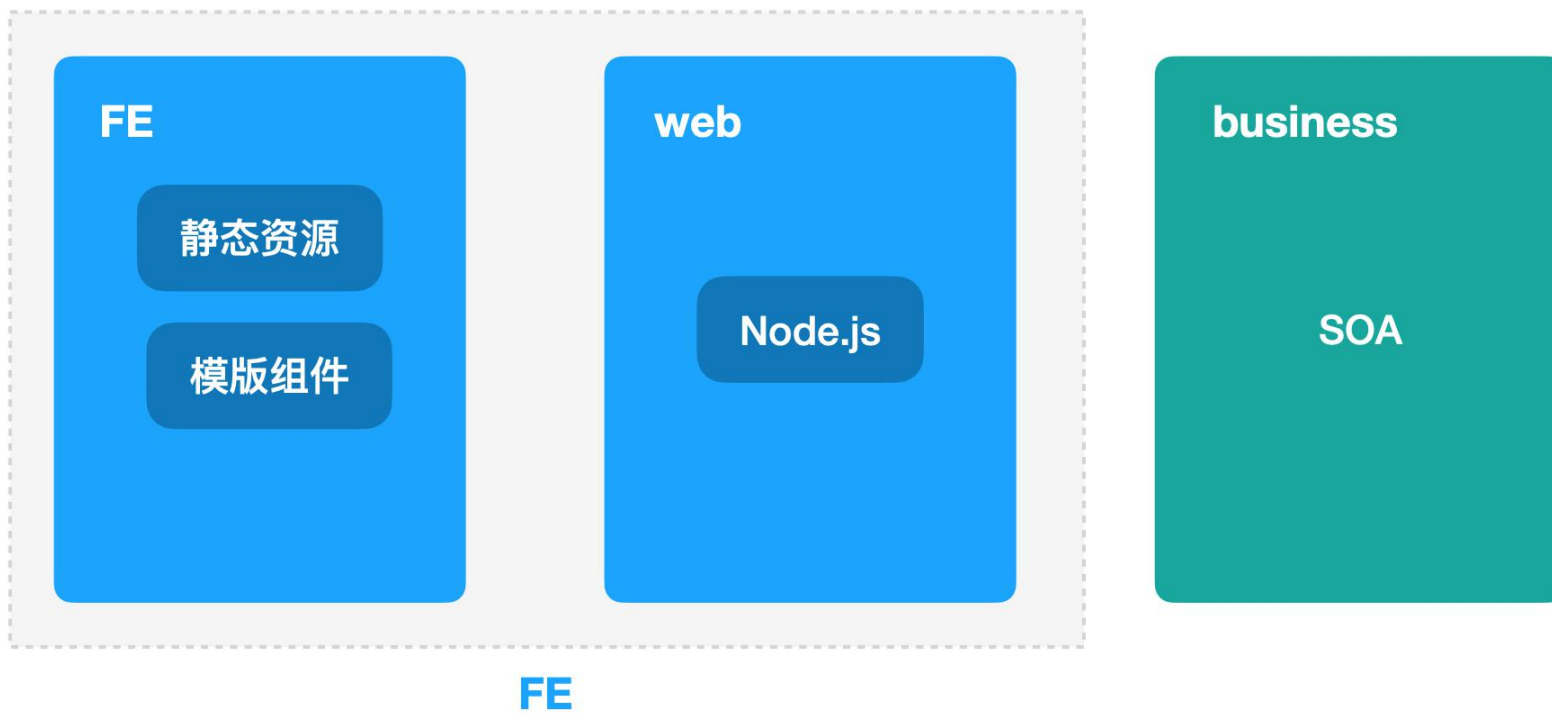
# Node.js 在猪八戒网

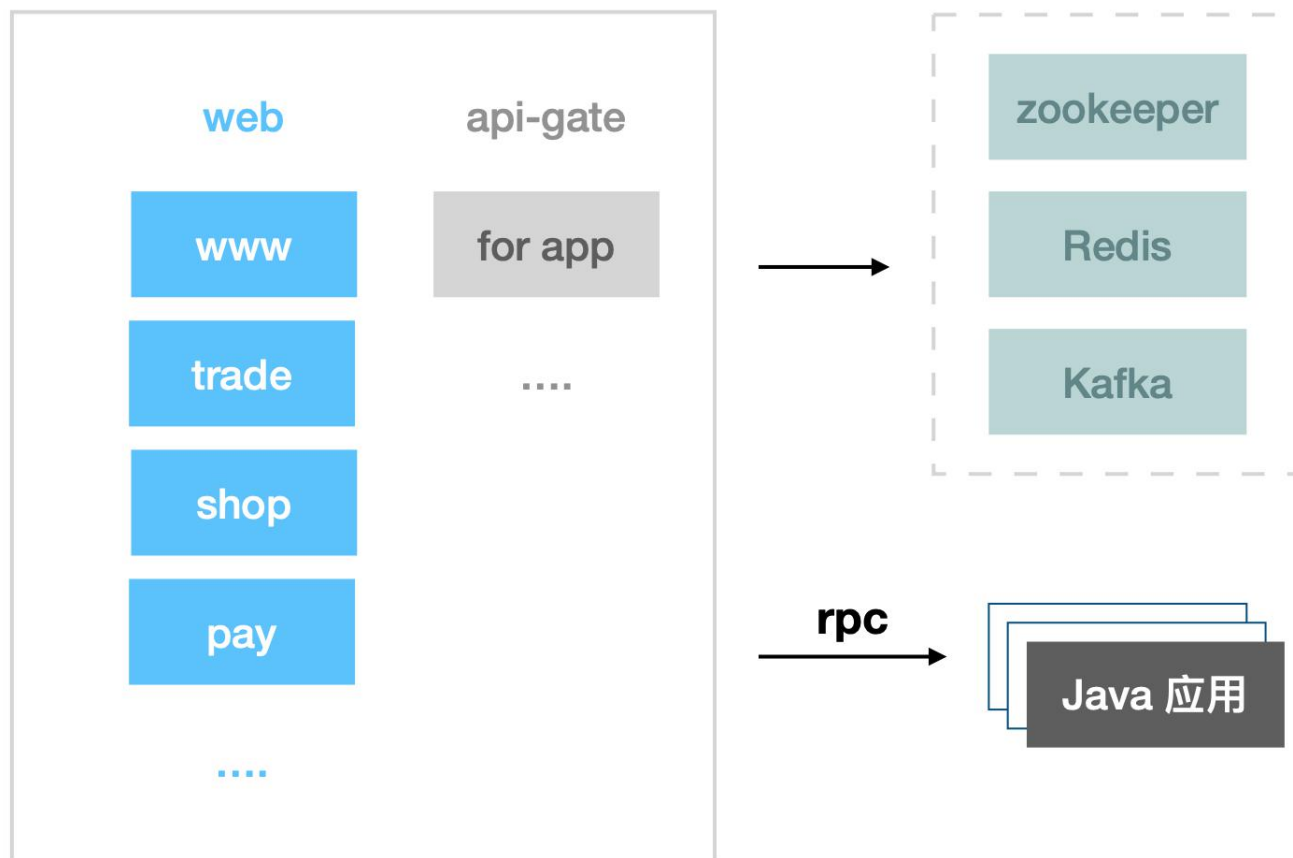
## 第三阶段



# Node.js 在猪八戒网

## 第四阶段





# Node.js 在猪八戒网



```
UA.onGet('/hello/world', (req, res, next) => {  
  const model = {  
    name: 'utopiajs',  
    version: versionDataApi.versionService.get()  
  }  
  console.info('[hello]')  
  return new UA.view(model, 'view/hello/world')  
})
```

# Utopia.js

统一的 web 接入层框架

## 设计思路

声明式编程范式 屏蔽实现细节

定位和解决问题优先 路由注册的完整性 (从网址查问题)

API 设计覆盖大概率场景 简化的 model 层

# 智子组件平台 tomoko

代码 + demo + 文档 （跨异构系统的通用组件调用）

## 目标

本地化的开发、测试、构建、部署、跨异构系统引用

demo 即文档



# 智子组件平台 tomoko

跨项目组件调用会需要使用到这个系统

前端组件项目支持自定义构建脚本，如果项目有自定义构建脚本，devops 就不会调用 tomoko cli 来进行代码构建

你可以基于此来实现自己的构建打包输出。

## 猪八戒网前端基础设施

[git.zhubajie.la](https://git.zhubajie.la)

内部代码托管系统，使用 boss 账号即可登录

公司内部的项目代码均在此系统中托管

**⚠ 不要将公司内部的项目提交到外部的 git 仓库中**

## devops 系统

目前主要基于 devops 系统进行项目的创建和代码发布上线操作。

涉及到前端相关的项目创建主要包括 Utopia nodejs web 项目和前端静态资源项目

两者都可以自定义构建脚本进行个性化的自定义代码构建

新建项目需要通过 devops 进行操作，  
不然系统里没有项目信息无法进行上线操作

## 域名相关

猪八戒网有很多二级域名，基本上每个 web 工程会对应到一个公网可访问的二级域名。

二级域名最好和工程名字保持一致，方便定位项目  
如 ucenter.zbj.com 对应 nodejs-zbj-ucenter-web

线上环境的主域名是 zbj.com，测试环境的主域名是 test.zbjdev.com

如商家中心，u.zbj.com 和 u.test.zbjdev.com

## Utopia.js 和配置中心

配置中心主要解决各个不同环境的环境变量差异，让代码能够通用化

utopia.js 工程根目录下的 config.json 是本地的配置文件  
在运行时会同配置中心的配置进行合并，配置中心的配置优先级要高于本地配置，配置中心没有设置的，会使用 config.json 里的配置

配置中心的项目数据的读取是根据 utopia.js 工程根目录下的 package.json 里的 name 来读取的

## Utopia.js 和日志系统

utopia.js 使用 `console.info`、`console.warning` 和 `console.error` 来进行日志文件的写入。

`console.log` 是不会写入到日志文件系统中的，在进行重要操作的日志记录时，需要注意这一点。

日志查看地址：[elkn.zhubjie.la](http://elkn.zhubjie.la)  
选择 Node.js 日志

## Utopia.js 和 apm 系统

公司目前使用 apm 系统来进行应用性能的监控，Node.js 项目也已接入。

日志查看地址：[apmn.zhubjie.la](http://apmn.zhubjie.la)  
输入 Node.js 项目名称



# 常见问题

## 如何从网址定位到代码

### 1. 服务端代码

> 打开 chrome 开发者工具的网络面板，可以看到 http response 字段中有 Project-Name 的字段，此即为 Node 的项目名称，去 devops.zbj.com 系统搜寻该项目名，即可看到具体的 git 地址

> 如果没有看到该 header，可以直接去 knight.zhubajie.la 项目搜索中输入域名，按照域名搜索

> 也可以打开页面源代码，查看 script / css 的标签，Node.js 项目在构建时会将项目名构建在静态资源产出物中

> 找到 git 后，即可在代码中搜索网址的 path 相关信息



# 常见问题

## 如何从网址定位到代码

### 1. 前端代码

> 打开 chrome 开发者工具，按 esc 按键，即可看到全局搜索框，输入关键信息搜索即可，比如 dom 节点名称、class 名之类的。搜索完成后，既可看到相关的前端代码，前端代码所在的 ur 路径即显示了该脚本所属的项目，通常为前端组件项目，或者具体的 Node.js 项目，

> 从前端组件项目的 id 即可看出该代码所属的项目名

# 额外要了解的

Promise

Async await

CommonJS / AMD / UMD / ES modules

Express / Koa

谢谢