



架构迎接未来变化
IAS2017 • NANJING

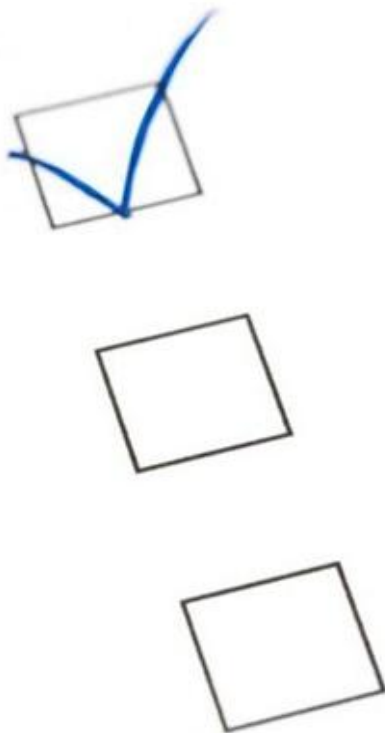


IT大咖说
知识分享平台



前后端分离架构的落地思考

苏宁云商 禹立彬

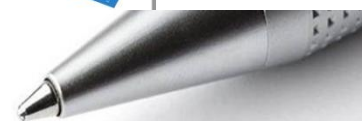


目录

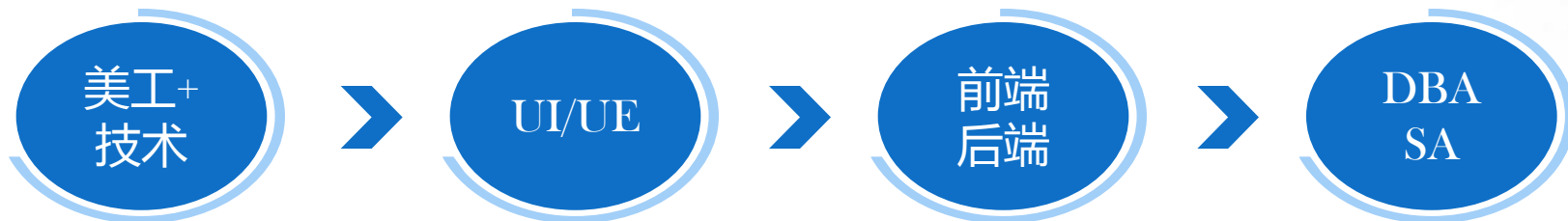
- 1 为什么要前后端分离
- 2 典型的业务场景
- 3 前后端常用技术利弊
- 4 其他思考

目录

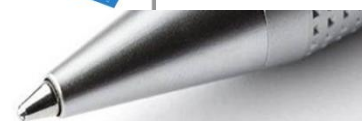
- 1 为什么要前后端分离
- 2 典型的业务场景
- 3 前后端常用技术利弊
- 4 其他思考



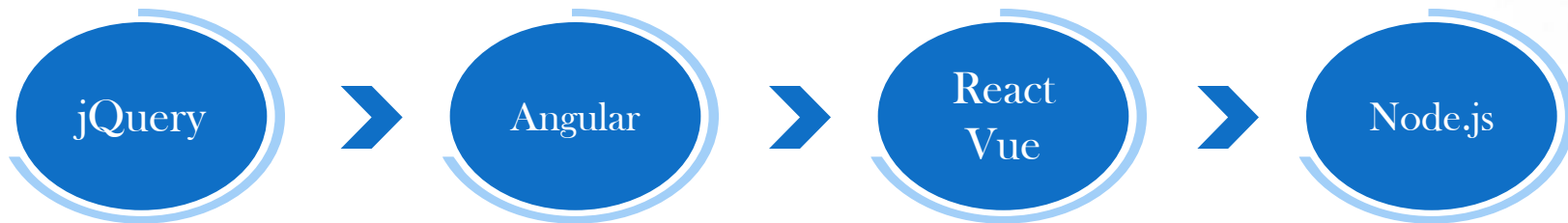
工作职责的细化



随着互联网的发展，工作细分是长久以来的趋势



前端门槛的增高



年抛的前端轮子们在把前端搞得筋疲力尽的同时，极大的增加了全栈的学习成本.

目录

- ① 为什么要前后端分离
- ② 典型的业务场景
- ③ 前后端常用技术利弊
- ④ 其他思考



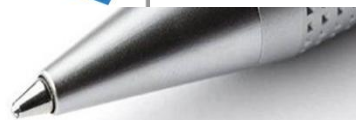
业务场景多





应用场景多





技术挑战

A

大流量下要求高可用

访问量大，每个问题都会被放大

B

浏览器兼容要求高

还有坚持使用低版本浏览器的用户

C

迭代速度要求快

促销活动页很多

D

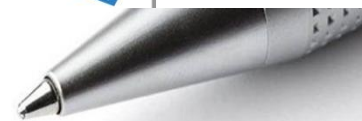
SEO支持

商品搜索

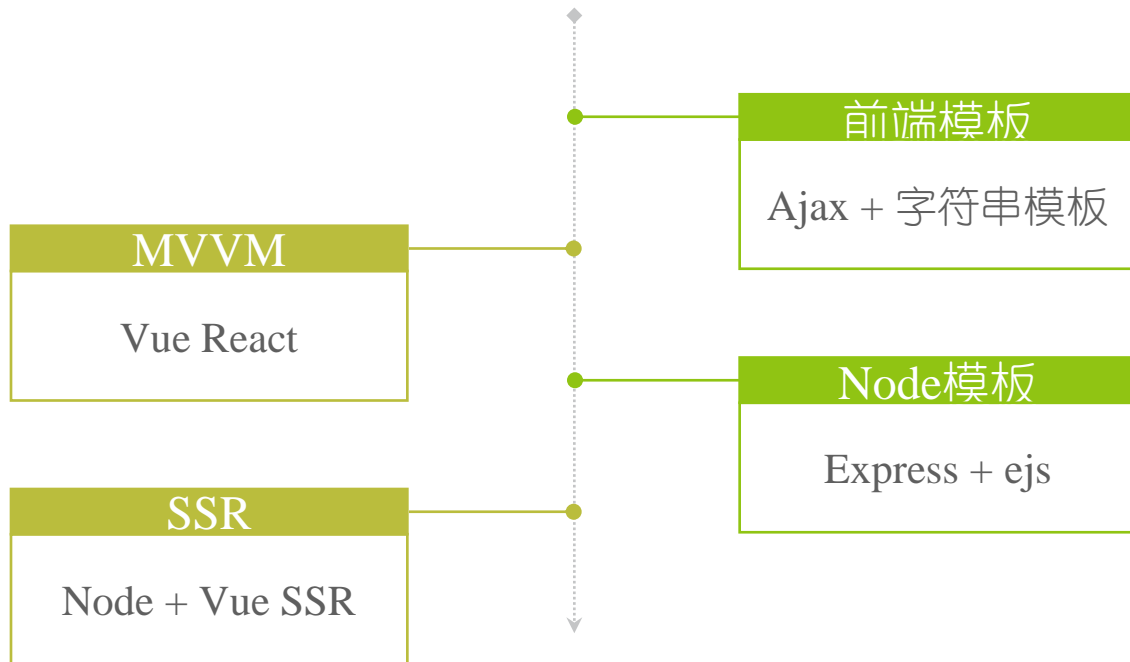
CONTENTS

目录

- 1 为什么要前后端分离
- 2 典型的业务场景
- 3 前后端常用技术利弊
- 4 其他思考

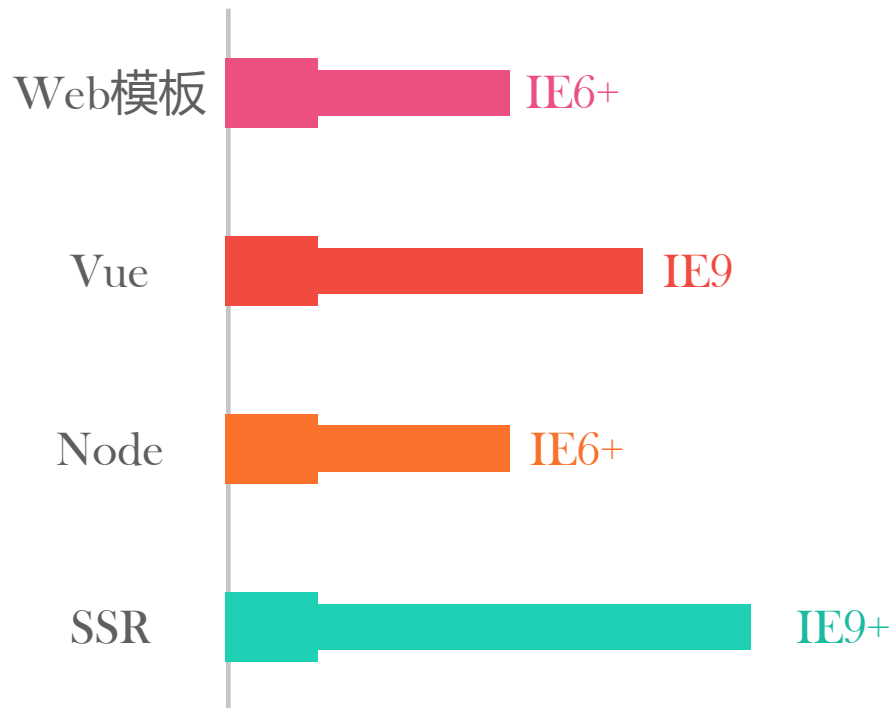


技术方案



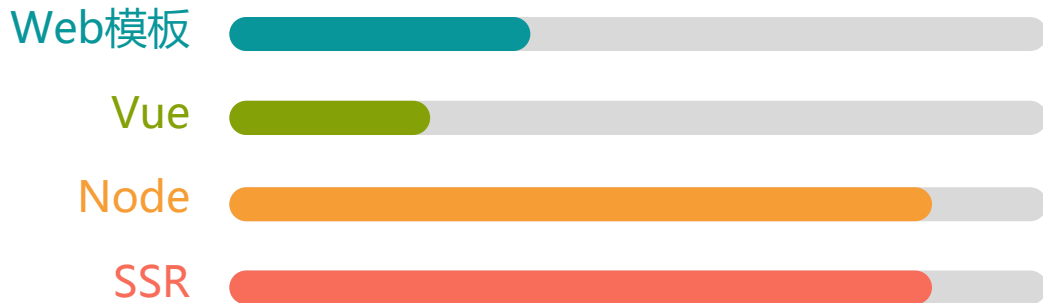


浏览器兼容



在浏览器兼容中，基于现代MVVM框架的技术方案，处于劣势，在浏览器兼容性要求较高的场合中，无法使用

SEO支持

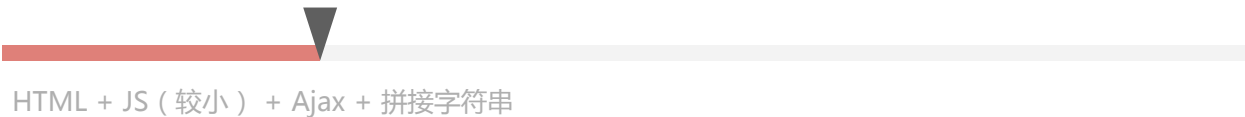


对于SEO有要求的网页来说，使用Web模板和Vue方案，不大适合



首屏渲染耗时

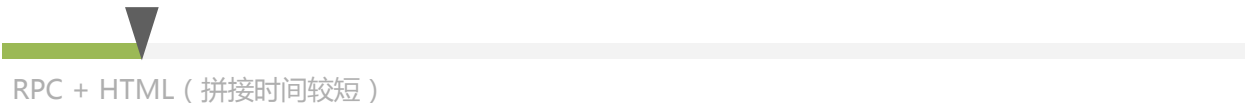
Web模板



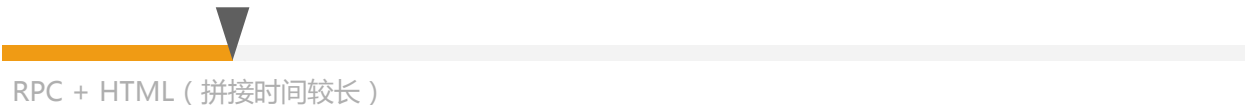
Vue



Node

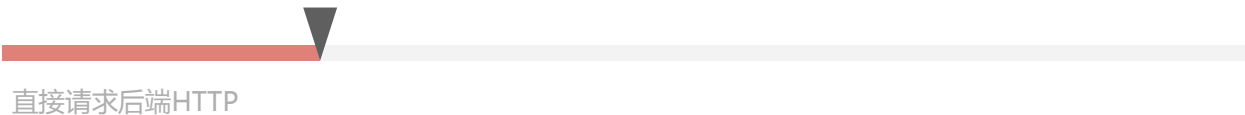


SSR

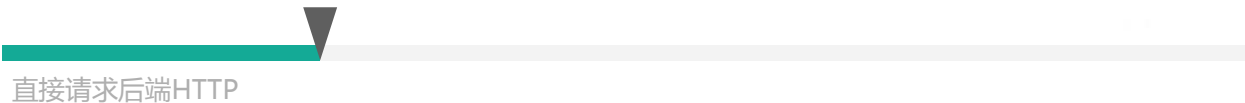


异步接口速度

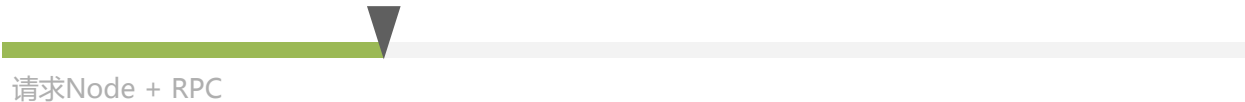
Web模板



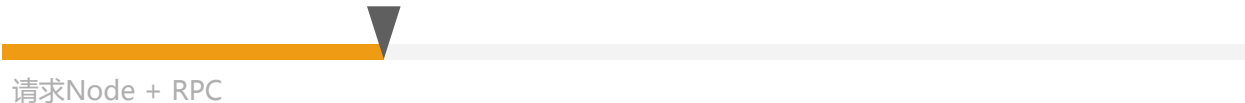
Vue



Node



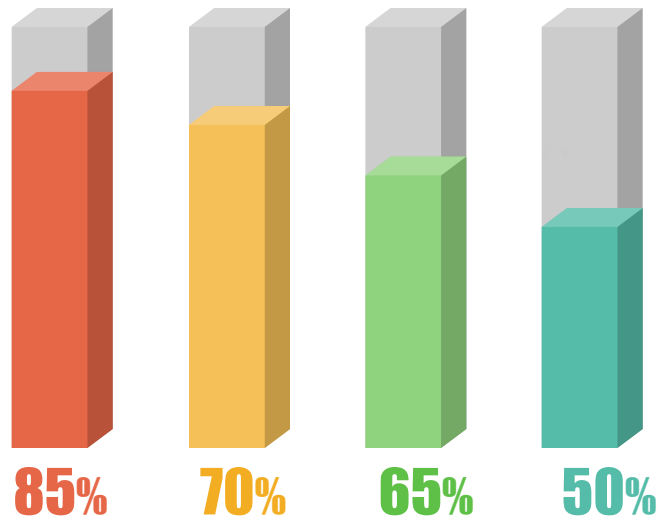
SSR





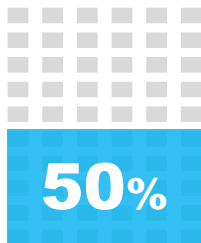
高可用

- Web模板
- Vue
- Node
- SSR

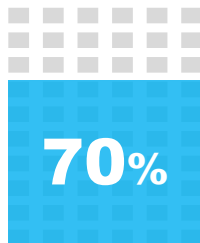




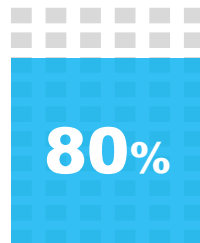
技术门槛



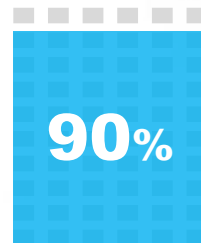
Web模板



Vue



Node

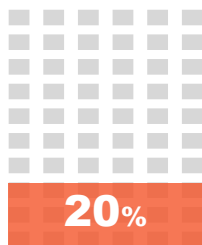


SSR

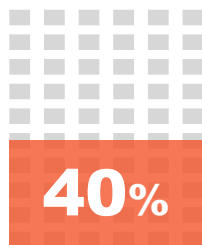
相对来说，Node相关的技术门槛较高。



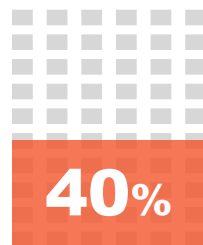
前端工作量



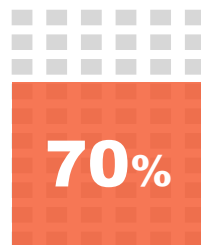
无分离



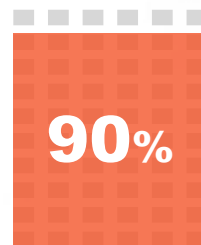
Web模板



Vue

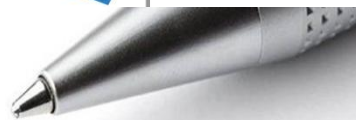


Node

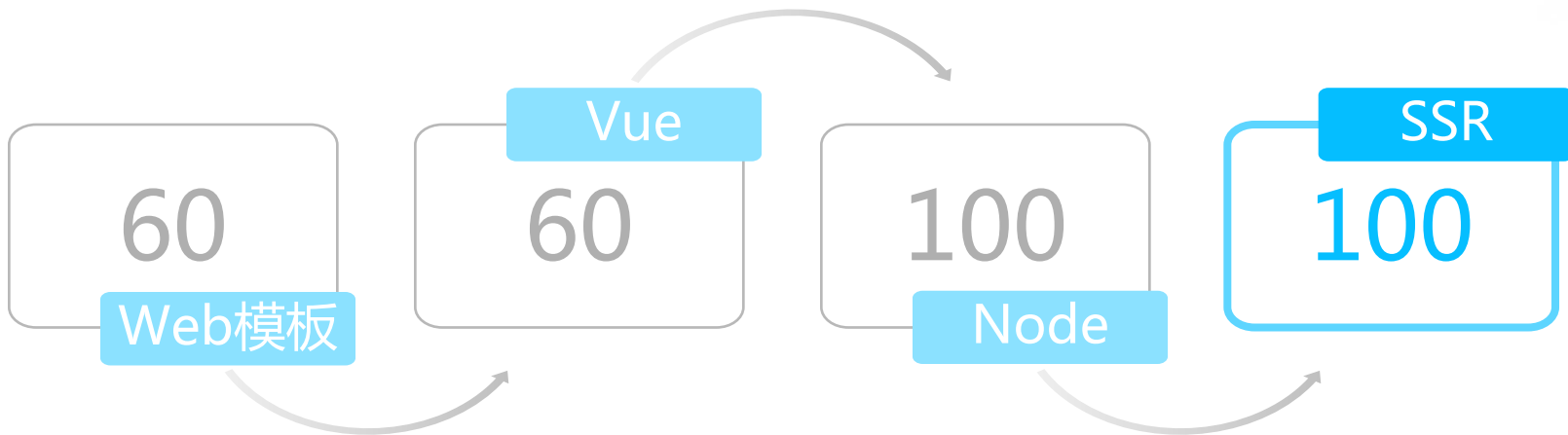


SSR

显然前后端分离会导致web前端工作量增加



前后端分离度



01

你的项目要SEO吗

Node.js就是不二选择了

需要Server渲染，不管是ejs还是SSR



Node.js风险

缺少企业级工具。

错误调试困难

资料少于主流语言

02

你的项目要兼容IE

过度使用框架

大量使用React Vue等框架



风险

使用这些框架可以显著的提高前端工程师的薪水

使得个人利益和项目利益产生冲突

03 有足够的前端工程师吗

前后端分离的工作量

前后端分离的越彻底，前端工作量越多



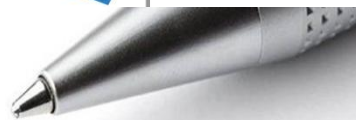
招聘风险

有经验的前端工程师不那么好招。

靠加班？

渐进式前后端分离

苏宁的经验



渐进提高前后端分离度



转向Node.js中间层



转向Node.js中间层
视场景使用SSR

渐进提高前后端分离度

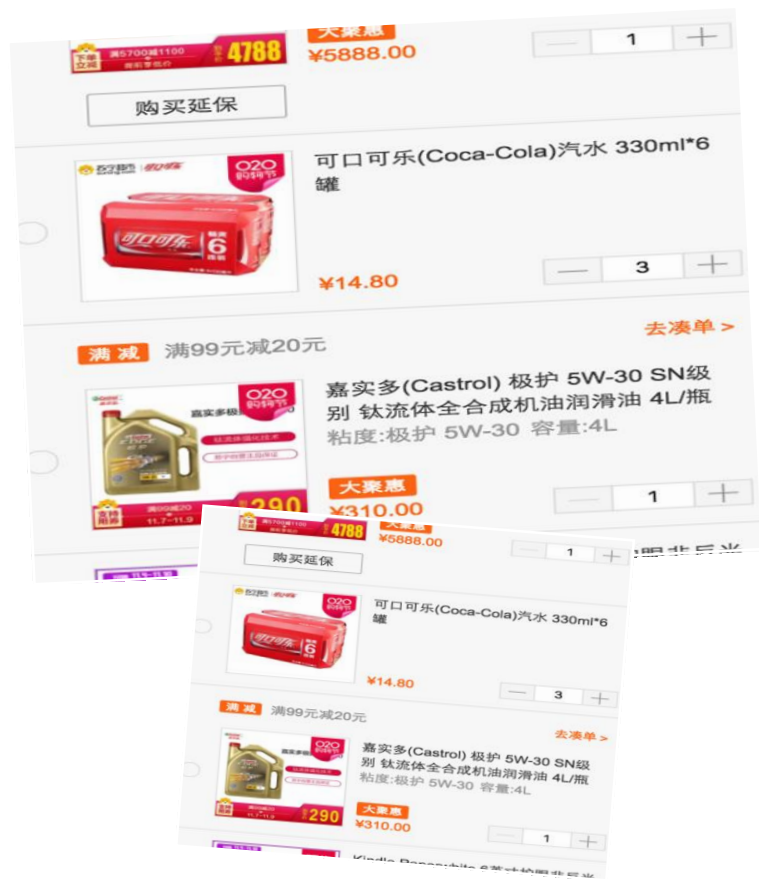


对于浏览器兼容性要求，可用性要求，
页面性能要求都极高的电商类页面
不使用前后端分离配合少量web模板。

渐进提高前后端分离度

浏览器兼容要求较高的活动展示页，逐渐从Web模板过渡为Node模板





渐进提高前后端分离度

核心应用型Web页，对可用性要求
占主导。

过渡为Node + Vue.js方案

渐进提高前后端分离度



渐进过渡到 S S R 方案

渐进的在团队中推进前后端分离



初期团队



成熟团队



技术实力



人力富余

团队推进中，根据团队实际的情况，也应该是渐进的。

渐进的在团队中推进前后端分离



技术风险



团队配比



进度要求

评估风险边界，保持业务稳定

渐进的在团队中推进前后端分离

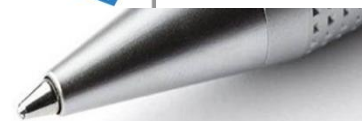


业务开发中，多选择新业务推进高级分离方案
对于老业务改造，应该循序渐进，选择新需求

CONTENTS

目录

- 1 为什么要前后端分离
- 2 典型的业务场景
- 3 前后端常用技术利弊
- 4 其他思考



全栈化思潮





前端开发职业前景



VR可能代替网页



AI代替重复性劳动



职业天花板



多语言能力要求



轮子学习成本

“标准总是过时的，这让它们成为了标准。”

-Alan Bennett



**THANK
YOU**