

打造微前端工程化体系

贾建容

- 使用场景
- 概念与架构
- 基座服务

- > 考虑如下业务场景:
 - 车贷
- ▶角色
 - 法务、财务
 - 反欺诈、风控
 - 产运研
 - 培训、市场、销售
 - 资金方(银行)、征信(央行)
 - 客户(车商、购车人)

- ▶贷前
 - 资金
 - 金融产品
 - 征信
 - 授信
 - 车商
 - 电销
 - 培训

- ▶贷中
 - 订单
 - 支付、退还款
 - 合同、签章
 - 用户
 - 放款
 - 放车
 - 风控、反欺诈

▶贷后

- 回款、对账
- 分润
- 权证、(解)抵押
- 催收
- 客服
- 销售绩效
- 数据报表

微前端的使用场景



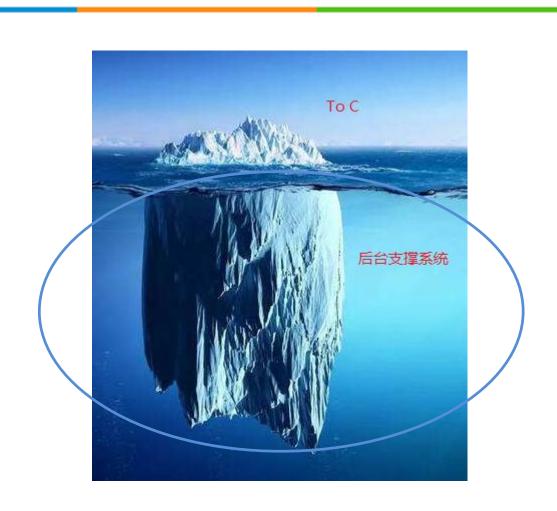


▶特点

- 业务流程复杂
- 系统多
- 系统生命周期长
- 参与角色多

▶常见于

- To B业务
- 中后台系统



- 使用场景
- 概念与架构
- 基座服务

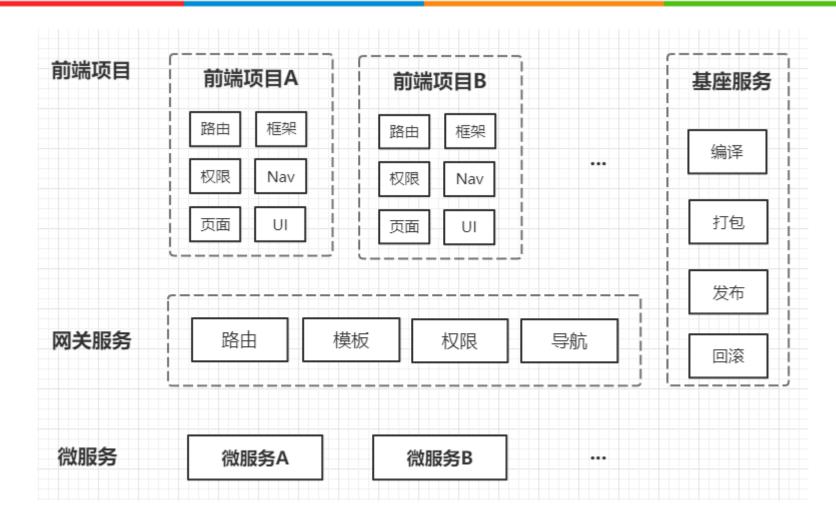


- >可控体系下的各项目协同(独立)开发及持续集成
 - 单个域名
 - 多个前端项目
 - · 每个前端项目拥有独立的Git仓库及CI流程, 互不影响



- > 基于Server端的微前端工程化体系
 - 基座服务
 - CI
 - 网关服务
 - 导航: UI + 数据
 - 权限: 登录 | 导航栏(页面) | 接口 | 数据
 - ・模板: velocity | ejs | html
 - 路由: 模板 + 接口

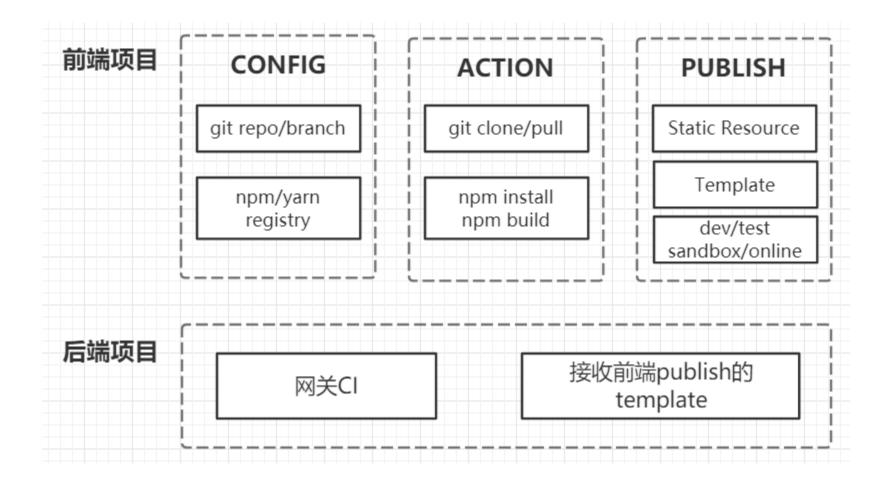




- 使用场景
- 概念与架构
- 基座服务

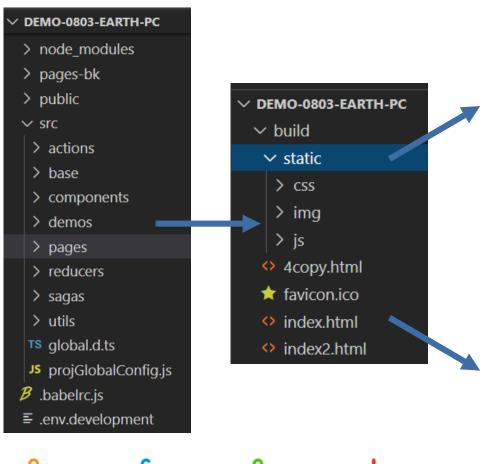
- 前端项目
 - 多个项目
 - 每个项目独立编译及打包
 - 每个打包后的项目独立上线及回滚
- 后端项目
 - 网关项目的上线及回滚







• 单个前端项目CI



```
"//j1.58cdn.com.cn/
                                      /redux.min.js"></script</pre>
"//j1.58cdn.com.cn
                                         act-redux.min.js">
"//j1.58cdn.com.cn,
                                      (redux-thunk.min.js">
"//j1.58cdn.com.cn/
                                       react-router-dom.min.j
"//j1.58cdn.com.cn
                                       (lodash.min.js"></scrip</pre>
"//j1.58cdn.com.cn/
                                       moment-with-zh-cn.js">
"//j1.58cdn.com.cn/
                                       antd v3.16.3/antd.min.
"<u>//j1.58cdn.com.cn/</u>
                                        intime.ef7e0b36.js">
"//j1.58cdn.com.cn
                                        endor.f913fa0e.js"></
"//j1.58cdn.com.cn/
                                       bms efficacyMgmt.e8e1d
```

```
[work(jiajianrong)@ static]$ pwd
/opt/web/finance_static_site/static
[work(jiajianrong)@ static]$ ls
anti_fraud.html
audit_index.html
auto-tel.html
bc_carBusinessSys.html
bc_carLoanMaterialConf.html
bc_cityConfig.html
bc_jrManageCenter.html
```

基座服务



· 聚合多个前端项目CI

项目信息						
ID	名称	Git地址				
项目A	项目A	地址A				
项目B	项目B	地址B				
项目C	项目C	地址C				

	编记			
项目ID	Git分支	打包版本	打包ID	
项目A	分支A1	v1.0	打包Av10	
项目A	分支A2	v1.1	打包Av11	
项目A	分支A3	v2.0	打包Av20	
项目A	分支A4	v2.1	打包Av20	
项目B	分支B1	v1.0	打包Bv10	
项目B	分支B2	v1.1	打包Bv11	
项目B	分支B3	v1.2	打包Bv12	
项目C	分支C1	v1	打包Cv1	
项目C	分支C2	v2	打包Cv2	
项目C	分支C3	v3	打包Cv3	

部署信息						
打包ID	部署环境					
打包Av10	dev					
打包Av10	test					
打包Av10	sandbox					
打包Av10	online					
打包Av11	dev					
打包Av20	test					
打包Bv10	dev					
打包Bv12	online					
打包Cv1	dev					
打包Cv2	dev					

基座服务



版本控制

打包版本		部署环境			
ID	version	dev	test	sandbox	online
项目A	v1.0	0	0	0	0
项目A	v1.1	0	0	0	0
项目A	v1.2				
项目A	v2.0	0	0	1	1
项目A	v2.1	0	1		
项目A	v2.2	1			
项目A	v3.0				
项目B	v1.0	0	0	0	0
项目B	v1.1	0	0	0	1
项目B	v2.0	0	0	1	
项目B	v2.1	1	1		
项目C	v1	0	0	0	0
项目C	v2	0	0	0	0
项目C	v3	0	0	1	1
项目C	v4	1	0		
项目C	v5	0	1		
项目C	v6	0			



- 编译细节优化
 - 存储与任务
 - 本地 vs DB及文件服务
 - Java/Jenkins与nodejs/pm2的区别
 - 并行/串行
 - 同一项目串行: 同步锁
 - 不同项目并行: child_process.spawn与spawnSync
 - Git clone/pull
 - SSH vs Https
 - 权限与日志
 - npm install
 - 性能
 - npm server
 - lock & --prefer-offline
 - 日志
 - child_process.exec不能顺序输出stdout, stderr, 且返回有限

- 发布细节优化
 - Publish时的环境
 - 基座服务与各环境(dev/test/online)网络互通
 - 各环境IP配置化
 - 推送协议
 - 回滚
 - publish之前的编译包
- 上线后体验优化
 - 日志查询: uuid
 - React -> Nginx -> nodejs -> Java
 - 多项目UI界面风格统一
 - 同一UI库、项目创建脚手架
- 安全
 - 执行的命令防止恶意注入
 - exec('npm install; rm -rf *' , function(...){})

Thanks!