



Being Young is an Attitude

Technology

web前后端开发实践和探索

融合OR分离

关于我



网名：Hbomb (H1bomb)

Github：<https://github.com/h1bomb>

WeChat&Weibo：h1bomb

热衷于web前端技术的

全端攻 (chen) 城(xu)狮(yuan)

现就职于YOHO！



一家年轻活力的公司（平均年龄26岁）
从事潮流文化领域（媒体,零售,展会）
目前已完成D轮融资（1亿美刀）有上市计划



SO

欢迎加入我们

qiqi.zhou@yoho.cn

微信: H1bomb

做技术界最潮的程序猿
做潮流界最牛的攻城狮

Agenda

- web开发的痛点
- 应对方法
- YOHO! Web前后端演进实践
- 分离OR融合?

web开发的痛点

1. 频繁变更的需求
2. 代码质量把控
3. 无处不在的业务逻辑
4. 各种兼容
5. 重复的部分
- ...

工程化

应对方法

1. 界定前后端切面（接口标准化）
2. 前后端职责分离（职能专业化）
3. 组件，服务，模块
4. 依赖管理，切面治理

YOHO! 的实践



前端页面制作

交付静态页面

前后端集成

YOHO! 的实践

第二种方式

页面部件化



```
graph TD; A[页面部件化] --> B[页面布局 and 组合]; B --> C[分离产物 js+css+html]; C --> D[界面集成 CDN+模板];
```

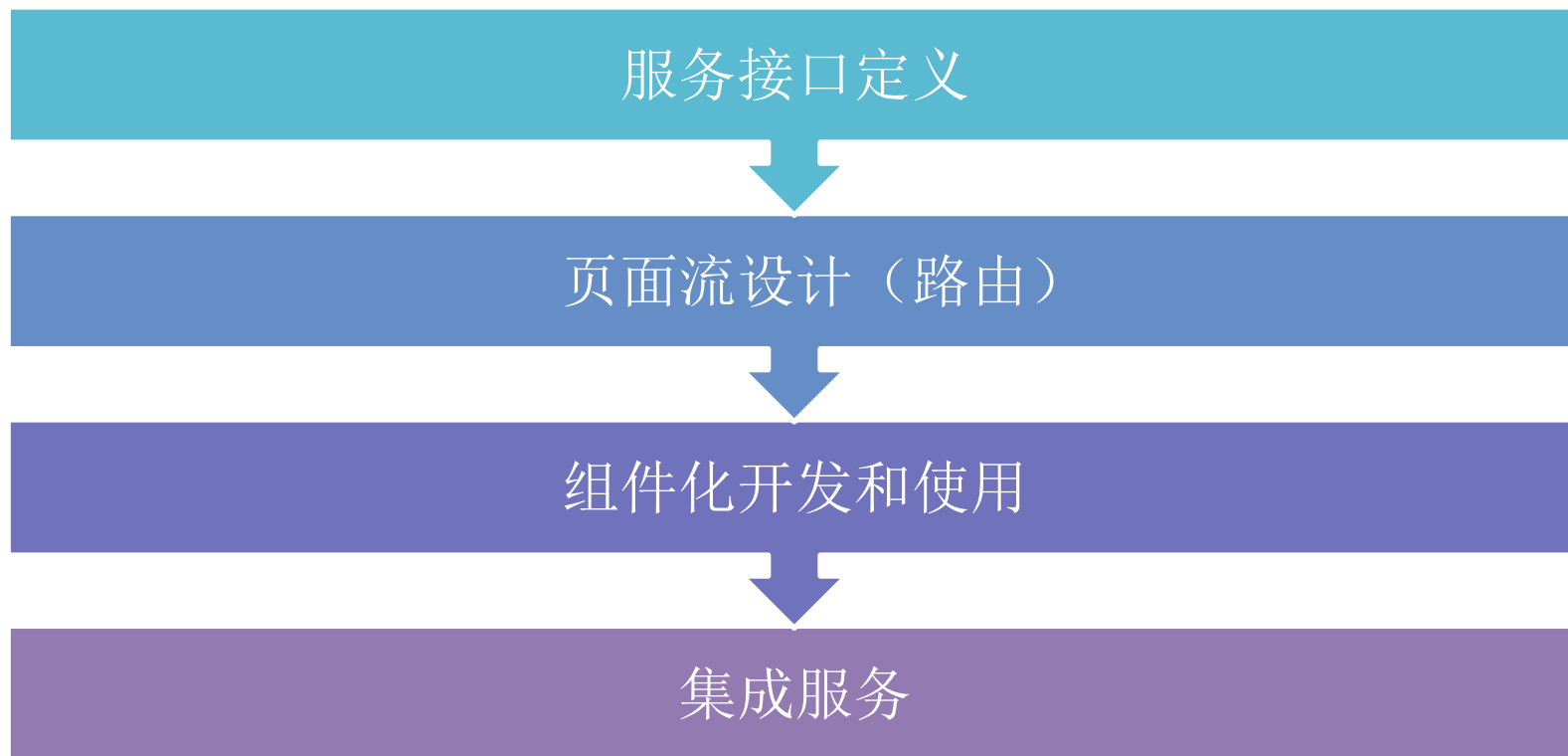
页面布局 and 组合

分离产物 (js+css+html)

界面集成 (CDN+模板)

YOHO! 的实践

第三种方式



前端都使用了哪些轮子？



Talk is cheap. Show me the code.

Demo

yo framework

<https://github.com/h1bomb/yo>

yo example

<https://github.com/h1bomb/yo-todos>

yo template

<https://github.com/h1bomb/slush-yo>

分离OR融合？

- 分离什么？
 - 前端能力封装
 - 标准化的组件
 - 通用开发模式
 - 开箱即用的流程化工具（ 桩服务生成 ）
 - 前端切面的分离（ 数据，结构，样式，行为 ）
 - 单元测试，端对端测试

分离OR融合？

- 分离什么？
 - 后端服务封装
 - 标准的数据源(基于资源，基于服务)
 - 标准的增删改从的行为封装(RESTful)
 - 流程化的客户端工具（服务调试，验证，服务文档生成）

已有的项目

- 组件化 (React, Polymer, WebCoomponents)
- 开发模式 (MVC, MVVM, FLUX...)
- 应用框架 (angularjs, vuejs, Ember.js...)
- 服务标准 (RESTFul, GraphQL, JSON API, json-schema...)
- 开箱即用工具 (Gulp, Yeoman, mocha, jsmine, webpack, swagger, jshint, babel, git...)

分离OR融合？

- 融合什么？
 - 对前后端标准的共识(服务，组件，切面)
 - 业务开发的融合（业务理解更透彻）
 - 工程化的融合（开发过程，开发工具，系统规范）

Qiqi.zhou@yoho.cn

微信:H1 bomb

Thanks!