

# VUE-FPG框架介绍：What's it?

<https://github.com/ixjs/vue-fpg>

基于VUE框架制定的企业级多项目工程组框架

2018-11-29

[lance\\_amethyst@hotmail.com](mailto:lance_amethyst@hotmail.com)

wechat:lance\_amethyst

# 企业级Web前端开发的困境

- 在介绍FPG框架之前，我们先了解下常规企业级Web前端开发的困境：
  - 项目要求都是紧急快，其他环节复用程度且不论，前端目前以剪切、复制、修改来新建项目；
  - 同样的功能／组件每个项目都有差异，升级维护太受累，公司级的功能／组件库迟迟无法建立
  - 项目／人员流转非常受限，特别是用不同框架的前端项目
  - 前端人员成本高，流程规范不清晰，组内和组外沟通语言不统一
  - 前后端开发因为调试，容易陷入相互等待，很无奈的延期
  - 前端开发已经越来越专业化，工序和内容很多，很容易陷入混乱，失焦，接手困难等
  - 新接手项目总是出现各种问题，环境跑不起来，依赖关系又要重建，调试和开发环境切换频繁

# FPG框架能解决的问题

- 基于VUE，统一整合前端框架，把框架本身也作为项目来管理，自身可持续升级进化
- 统一开发环境，快速创建项目，快速切换项目，将前端开发规范直接包含在框架中，更聚焦，上手更快
- 功能／组件库直接融合在框架中，方便维护，可以自然而然的积累
- 对小图片集，代码检查，项目发布等提供了脚本化管理工具，方便集成和自动化管理，同样将前端开发规范和工程管理隐含在框架中
- 按照框架要求进行开发，前端完全独立于后端开发，自行开发模拟的数据接口就是与后端的接口需求；开发好的界面就是和产品设计沟通校正的原型，方便快速迭代；
- 发布部署时可以根据现场环境简单调整配置参数，配合其他工具，很容易就能自动化
- 现场发现前端问题一般不需要现场调测，只要按照要求提供产生问题的数据，可以自行模拟调测；即使要到现场，也能简单的修改接口指向和调试代码，不影响生产环境的正常运行；
- 还有很多前端性能优化的方法和常见问题的解决方法等直接融在框架中，降低对前端开发人员的技能和经验要求

# FPGA不关注的问题

- 不在前端工作范围内的问题
- 工作流程的执行，监督，工作量评估等
- 具体的技术问题

# FPG相关的工程管理思路

- 自我管理：管好自己的工作，不要干扰和受干扰，建立信誉
- 惯例优先：惯例更有利于建立默契，有助于减少沟通障碍，合作更具可预见性，有可预见性自然就易于管理，效率自然也将提升
- 边界定位：做每件事都要有边界，定义好边界才好拆解，解构，实现，重组，完善；有边界就能明确接口，才好考虑耦合和复用等
- 模式思维／组件开发：每个企业的模式和业务特征基本是一致的，这就导致模式化的功能和相似的组件很多，可复用的远比现实高

# FPG是什么

- FPG是一种工程思路：Frontend Projects Group；前端项目工程组
- FPG框架是用于管理多个类似前端项目的企业级框架
- FPG面向的是解决工程问题，而不是技术问题
- VUE-FPG是基于前端框架VUE构建的项目实施框架，其重心在于项目工程化管理，解决企业内部多个前端项目之间共享，迁移，积累和进化等工程化问题

# 为FPGA举个栗子 – 1. 公司现状

- A公司，人力资源是有限的，但每个项目有新建、开发、维护：
  - 有N个项目：如果只一个项目麻烦就少很多
  - 只有1个前端架构师：多了没必要了，太贵了，很难招到，实在不行，高级也是可以的，一般来说肯定是兼职的，毕竟成本还是太高
  - M个前端：  $M \ll N$ ，如果不是这样，要么就是项目太大，要么就是企业老有钱了
  - 管理规则：都不那么明确谁一定会在哪个项目上，一般是哪里着急哪里去

# 为FPG举个栗子 – 2.代码库和职责

- 前端代码库共有 $N + 1$ 个：
  - 框架库framework.git：
    - 架构师可以提交
    - 其他人员只读
  - 项目库p1.git, .... pn.git：
    - 只有该项目人员可提交
    - 架构师只读
    - 无关人员不可见
- 架构师的职责：
  - 不定期调研p1~pn代码库，抽取组件，模式，以及实用工具集等，对framework.git进行开发，升级以及维护
  - 定期对M个前端发布framework.git升级公告
  - 对其他前端人员进行指导和培训
- 前端人员的职责：
  - 克隆framework.git,；并在架构师的提示下升级；
  - 克隆常规项目pi~pj.git，在各项目经理的要求下开发维护前端项目
  - 在架构师的帮助下使用framework.git创建新项目



# FPGA适用的典型场景

- 创业公司；原因：人少，事多，一般都要经历多次试错
  - 项目型公司；原因：项目多，利润一般来自压缩人力成本和技术积累
  - 多业务系统（产品）公司；原因：业务系统多，每个业务系统要至少一个前端项目
- 共同特征：
    - 基于B/S结构的软件开发／集成需要
    - 多套前端项目并存
    - 有技术积累的意愿

# VUE-FPG优势分析

- 使用VUE框架作为基础框架，充分降低前端开发人员技能需求，也就直接降低了人力成本
- 使用了简单的命令行工具，只需要熟记几个命令就能完成项目生命周期的大部分管理工作
- 内嵌了页面路由，服务响应路由，可扩展的页面功能权限配置等引擎，高可定制化，高可扩展性
- 建立成体系的开发惯例，有利于团队配合和新成员融入
- 直接将模块／组件库融入进框架，有利于公司的技术积累和快速实现

# VUE-FPGA未完成的工作

- 为简化框架开发工作，将自动化测试的部分内容从框架中剥离出来，后续有可能会补充回来
- 多语言支持已经实现，但出于现实考虑，暂不列入框架内容
- 考虑了plugin的接入方式，但暂时没有列入开发计划
- 准备发布到npm上方便使用，但目前还没来得及完成
- 文档工作
- 还有很多。。。