

# TaitanWeex----基于 Weex 的电商促销 页搭建方案

## 大纲

泰坦（Taitan）系统，是苏宁易购内部 CMS(Content Management System 即内容管理系统)系统的别称。泰坦是希腊神话中的神族，他们是造物主，能够创造世界和世间万物。泰坦系统与泰坦的相似之处在于，苏宁易购消费者中心各类渠道（包括移动端、PC 端、IPAD 端等）面向用户的内容（商品展示）都依赖于此系统，它就像造物主一般“稳健”而又“强大”。

泰坦Weex（TaitanWeex）系统，是在开源框架Weex的基础上，进行深度融合和二次开发，对页面的UI进行了深度的二次封装，提供了稳定的API服务的同时，实现了模块的动态编译，减少了不必要的资源引用，提高了页面性能。目前苏宁易购的客户端的促销页都是使用了这套方案，支撑了日常和大促的用户体验和稳定性。

## 业务背景介绍

### 每个月都是大促

苏宁作为中国零售业的巨头，几乎每个月都在搞促销活动，除了几个大促，例如 618、818、十一国庆、双十一、双十二、元旦、过年。在这个频繁的促销活动的背景下，要求业务开发人员高效的完成促销活动的开发工作，对于业务开发人员将是一个非常大的挑战。



## 极致用户体验

在这非常频繁而又紧张的促销活动中，开发人员面临着超级多的业务要发布上线的同时还要保证页面加载性能的良好以及良好的用户体验。一旦用户体验降低，客户流量也随之降低，对苏宁的损失也是巨大的。因此，这就对开发人员的要求也上升了一个档次。

## 原生&H5

苏宁是一个零售行业，几乎所有用户能访问到的平台都要顾及到：PC端、Wap端、安卓端、IOS端、微信端等环境都必须支持，这面临着原生和H5版本要同时上线的问题，这种日益增加的需求导致了多端开发效率的矛盾。因为一旦任意一个端开发进度落后于其他端，会面临无法同时上线的问题，这将给公司的业务带来非常大的损伤。

## 频道页&大促会场

苏宁的业务主要分为频道页和促销会场。



频道页



促销会场

对于频道页面临的问题：

1. 模块定制化，复用性差。
2. 需求统一，可控性强。
3. 特性复杂的交互。
4. 数据量大。

对于促销会场面临的问题：

1. 通用业务模块，可动态搭建
2. 时效性。
3. 性能问题。
4. 轻交互，重展示。

## 痛点

为了解决上述的矛盾，需要逐步攻破这些问题。

1. 为了保证多端的同步开发，提高开发效率，我们采用统一的数据流。
2. 为了解决快速上线，保证开发效率的问题，我们开发业务模块库，极大的提高了业务的复用性。
3. 为了保证系统的稳定性，零事故。所以我们的系统必须得有降级方案。
4. 为降低运营成本，所有我们将业务模块可配置化，方便运营动态的配置业务模块。
5. 为了兼容 H5 和 WEEX 的同步开发，我们要将模块公共的抽象剥离，同时兼容 H5 和 WEEX，来提升开发效率。
6. 为了提升开发者的开发体验，所以我们开发了配套工具来提升开发效率和体验。



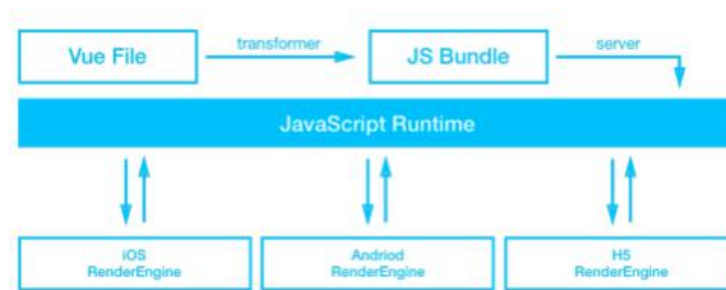
## 系统架构解析

### 如何使用 Weex

Weex 致力于使开发者能基于当代先进的 Web 开发技术，使用同一套代码来构建 Android、iOS 和 Web 应用。具体来讲，在集成了 WeexSDK 之后，你可以使用 JavaScript 和现代流行的前端框架来开发移动应用。

Weex 的结构是解耦的，渲染引擎与语法层是分开的，也不依赖任何特定的前端框架，目前主要支持 Vue.js 和 Rax 这两个前端框架。在这里苏宁采用的 Vue.js 框架作为开发框架。

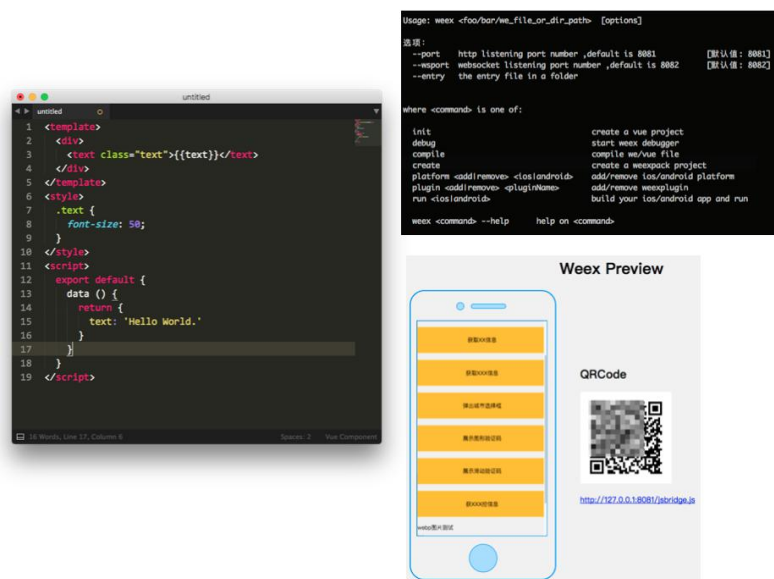
Weex 的另一个主要目标是跟进当代先进的 Web 开发和原生开发的技术，使生产力和性能共存。在开发 Weex 页面就像开发普通网页一样；在渲染 Weex 页面时和渲染原生页面一样。



开发人员通过编写 vue 文件，通过转换工具转换成各个平台的 JS Bundle，然后可以分别运行在对应的 IOS、安卓、H5 上面。

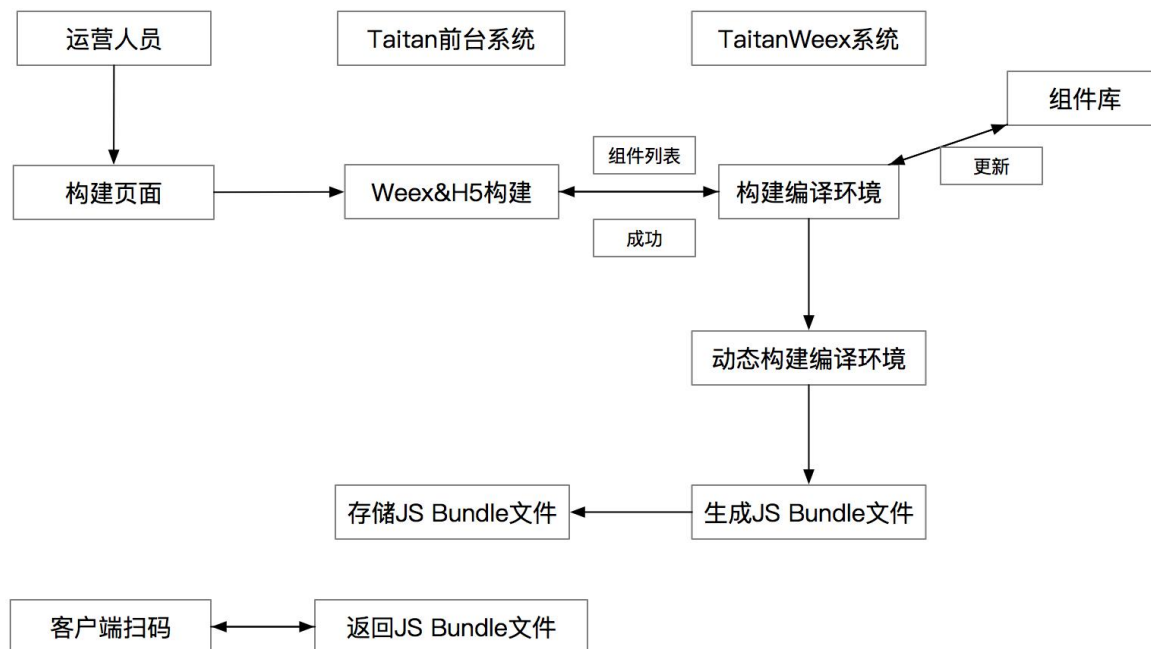
## weex&H5 组件

开发人员通过本地开发 WEEX&H5 组件，通过本地调试组件的基本功能，查看是否满足需求开发。通过测试之后，更新到组件库里面，更新最新的版本文件。



## 动态构建

利用 nodejs 我们搭建了一个动态编译 weex 组件的平台，运营人员通过在后台系统的图形界面构建出所要的大促页面，然后点击构建，后台系统就会统一页面所用到的组件，在内网通过 http 请求发送到 node 编译平台，然后平台根据所需的组件，在线更新组件文件，在线构建 WEEX 所需的编译环境，编译 JS Bundle 文件，异步发送给后台平台。



## 如何打通 java 系统和 node 系统之间的交互

在 nodejs 中，在系统启动之后，监听后台系统的发送过来的请求，如果有请求过来，就先校验参数的合法性，如果不合法直接返回 error，如果合法，则立马返回一个 success。然后放入到编译队列里面去。接着通知消费者去消费编译队列里面的任务。消费者这个时候会增加正在被消费的个数，如果当消费个数超过一个上限，就会停止操作。

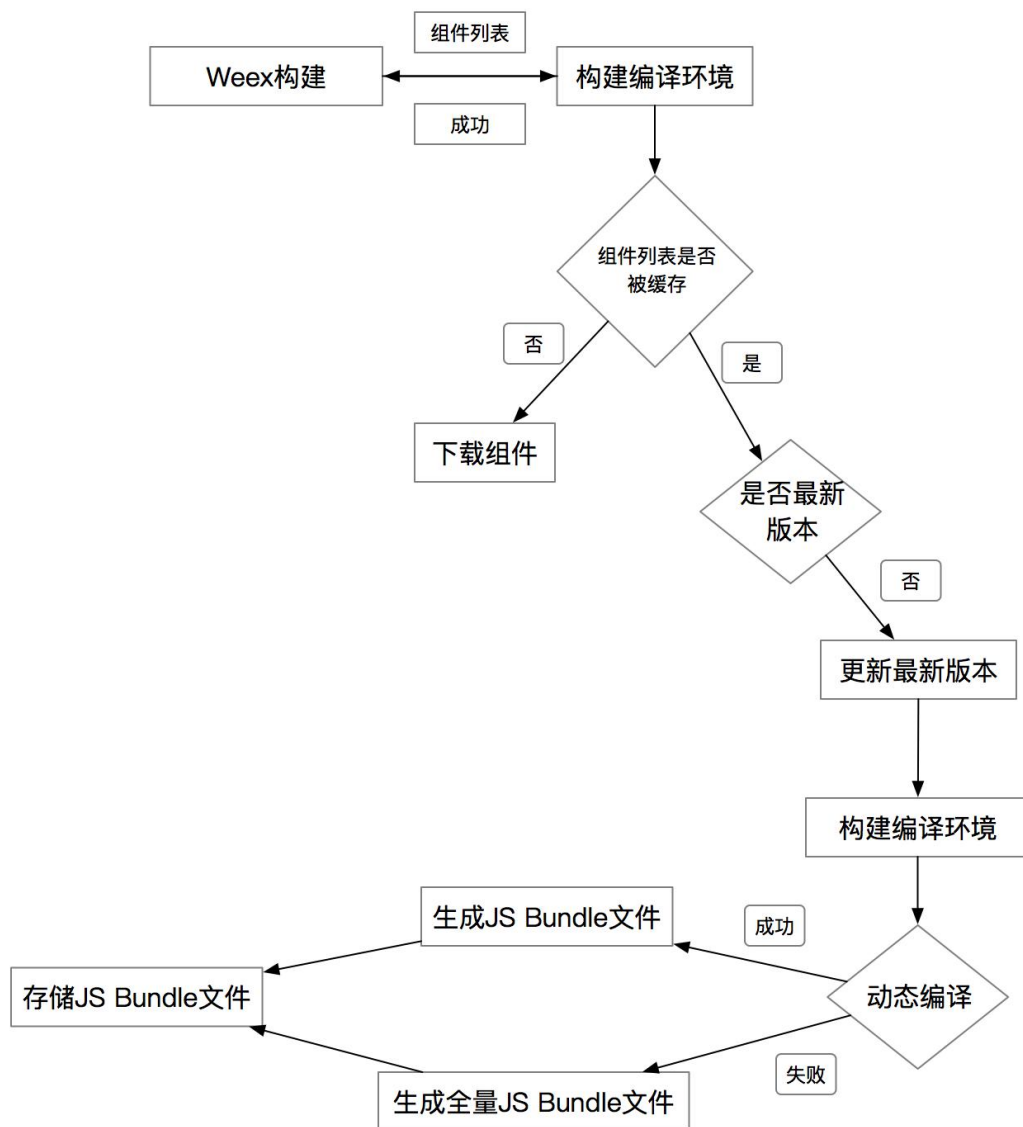
接下来就会根据请求参数中自带的组件列表，查看是否本地有缓存，如果有缓存，则请求 WEEX 组件库对应组件的最新版本号，如果与本地不同，则先更新组件。

更新完了组件之后，开始创建一个编译空间，动态的创建编译的文件夹，以及移动所需的文件，主要包含公共组件，依赖组件，所需的业务组件等，然后动态的生成 index.js 和 index.vue 文件。

编译环境创建好了之后，就会调用 webpack 进行编译环境的编译，等待编译完成生成 JS Bundle 之后，获取到 JS Bundle 的内容，然后发送请求给后台系统，告知编译完成。然后减少一个被消费的个数。

然后继续查看编译队列里面有没有待编译的任务，如果有，则继续执行编译工作。

在每一个任务执行的时候，都会有一个降级方案，就是在任何一个编译的过程中，如果有任何报错的情况，就会返回一个之前缓存在本地的全量的 JS Bundle 文件，同时发出告警。当发生错误的时候，也会更新最新的全量 JS Bundle 文件。



## 一些问题

在开发 WEEX 组件库的过程中也遇到了一些问题，比如数据的扁平化处理，因为 WEEX 是基于数据驱动页面展示的，数据那块如果过于复杂化，也会引起性能的问题，所有在数据扁平化处理上面需要花一定的时间去优化和改善。

还有在开发过程中，由于业务模块达到了 100+，所以对于模块的规划分类，也是需要优化改进的。组件设计也不能过于复杂，从而引发性能问题，导致页面的崩溃，在这一块上面还需要加强治理。

## 泰坦系统所见即所得

苏宁易购的促销页面是通过泰坦系统创建的实现的。





泰坦后台系统

通过登录到泰坦的后台系统，可以进行页面的所见即所得的页面搭建，利用平台提供的各个组件搭建一个促销页面。



通用图片组件



组件的配置信息

通过对于组件的各个属性进行动态配置，可以给组件设置主键，通过可视化的操作，支持排序功能。



每个页面都支持保存草稿功能，支持 Weex 和 H5 的实时预览。



## 页面配置数据的存储

当页面搭建完毕之后，泰坦系统会把页面的各个组件的配置信息整合成一个 json 格式的数据，存储在 redis 集群里面。这套 json 数据同时支持 Weex 和 H5 页面的渲染。当页面配置发生变化的时候，更新 redis 数据。对外提供异步请求接口，方便 Weex 和 H5 页面请求数据。

## 带来的成果

### 运营配置

之前在没有泰坦系统的时候，只有 H5 端支持动态配置，运营人员之前只能配置 H5 端的页面效果，跑在原生 App 应用上面，有了泰坦系统之后，运营人员通过可视化配置，在线预览的方式来查看促销页是否满足他们的预期，客户端优先展示 Weex 页面，如果遇到什么问题降低到 H5 展示。

### 页面加载

在原本的 H5 页面的时候，页面的业务 js 基本都会比较大，尤其是促销页面，js 文件可能更大，达到 500K+ 的大小。这在用户初次请求页面的时候，页面的加载时间基本都要超过 2s 以上。

但是 Weex 页面的 bundle.js 文件相比较于 H5 的 js 大小小了很多，基本稳定在 90k+ 左右，这在页面加载的时候，页面的加载时间基本可以稳定在 500ms 以内。性能提升了 4 倍左右。

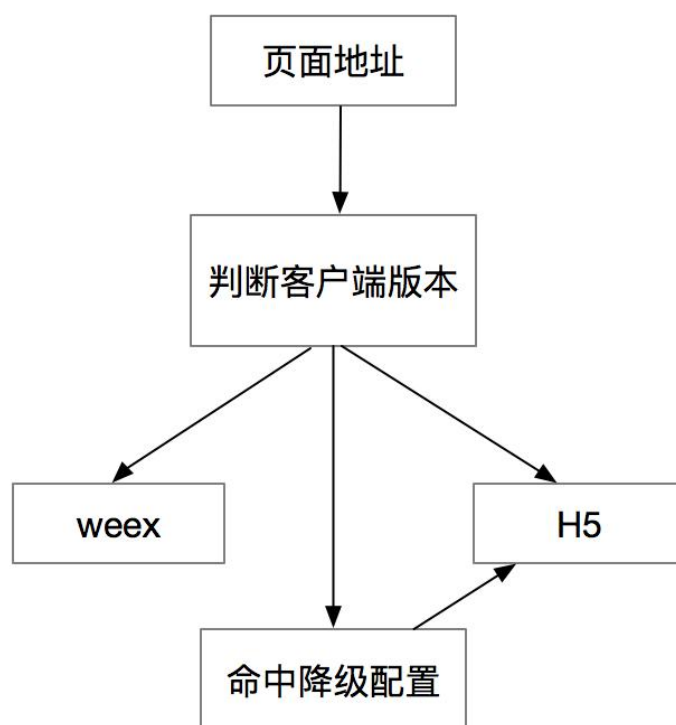
### 渲染性能

在原本的 H5 页面渲染的时候，基本稳定在 3s 左右，替换成 Weex 页面之后，页面的渲染时间基本稳定在 0.3s 左右，性能提升了 10 倍左右。极大的提升了页面的渲染能力，能在最短的时间内显示出促销内容。

## 稳定性保障

### 降级方案

为了保证促销活动的进行，不允许出现任何问题，所以必须得有降级方案来保障大促活动。

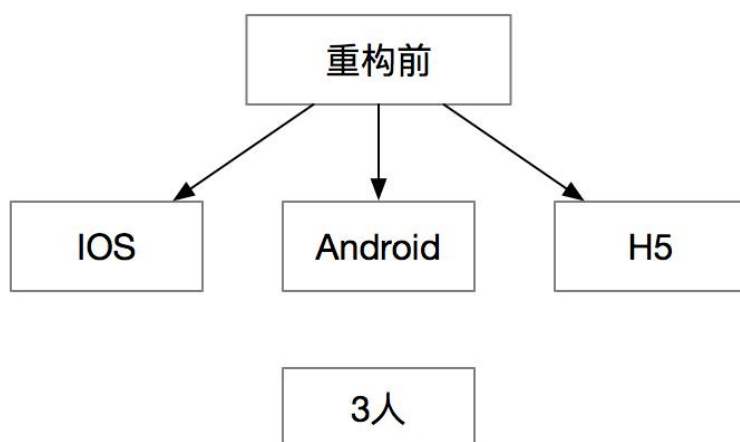


当请求一个页面的时候，通过判断客户端版本，来选择是否加载 WEEX 页面，还是加载 H5 页面，如果发现有需要降级处理，则直接选择渲染 H5 页面。

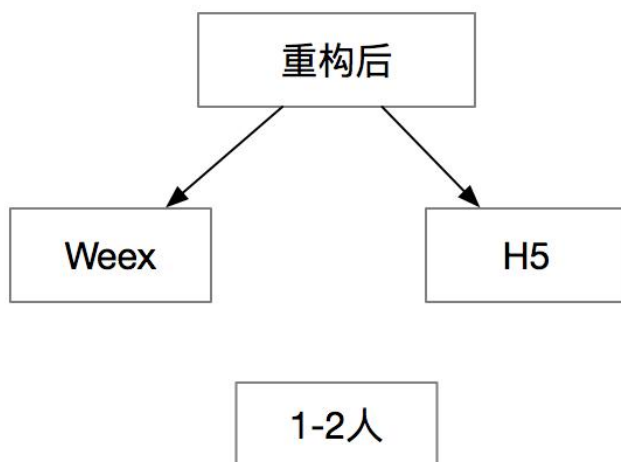
因为苏宁的任何一次促销活动，都不允许有事故发生，是事故零容忍的。在这种背景下，降级方案是任何一个系统都得有的，因为开发人员也不能保证系统在任意时刻都是可用的，不会出现问题的，所有必须的有降级方案来保证事故的不发生。

## 开发效率

原本采用 WEEX 组件动态编译，如果实现一个页面功能，需要一个 IOS 开发人员，一个安卓开发人员，一个 H5 开发人员。



重构之后，我们只需要一到两个人来编写 WEEX 代码通过代码同构编译出 H5 页面。



## 总结

在这套方案上线之后，已经服务过多次大促，现在线上页面已经达到了 10000+，可以支撑大促期间的页面搭建，同时也给用户带来了很好的用户体验，并且极大的提升运营的搭建效率。

当然后续我们还有很多工作要做，去完善优化我们的这套系统，比如业务代码的同构改造，以此来提高开发效率。还有 WEEX 端的 JsBridge 和 H5 端的同构方案。客户端 WEEX 组件的封装工作。对于系统的性能治理，崩溃治理上面还有很多工作要去完善和优化。