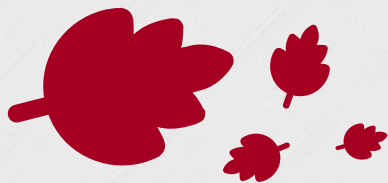




前端框架

Seajs+Handlebars模块化开发

作者：何清梅



目录



一、使用背景：为什么使用？



二、模块化开发的优缺点



三、Seajs简介



四、Handlebars简介



五、适合什么样的项目



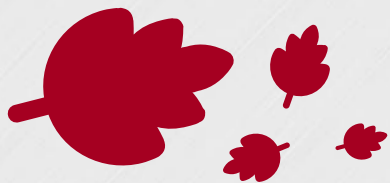
六：注意事项



七、目前使用的项目



八、我们的约定



传统开发模式的问题

由于代码的组织结构是非模块化的，所以代码无法复用，进而导致代码重复，这就为维护埋下了隐患 – 需求变更或功能添加将导致代码多处更改，随着应用规模的增大，代码将迅速进入难以维护的状态，常常是牵一发而动全身。

由于所有代码都混在一起，无法测试，我们也就无法获得保证代码质量的有效手段。

问题一

问题一

由于代码粒度太大，页面可能会加载大量根本用不到的代码，即便忽略网络传输的问题，过多无用代码，也会导致页面解析缓慢。

问题二

问题一

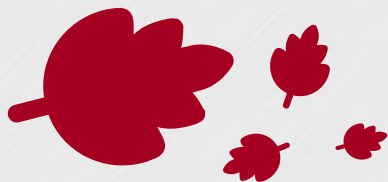
开发和维护成本大。比如1开发了个弹框叫dialog，然后他去做另外的项目。来了个2，他不知道原来之前开发过弹框，于是他又开发了个弹框叫popup，这样就容易造成代码冗余和维护困难。



模块化开发

针对前面这些弊端，我们做了一些思考，如何把一个大且复杂的场景分解成几个小部分，这些小的部分彼此之间互不干扰，可以单独开发，单独维护，而且他们之间可以自由组合（也就是按需加载），这就提出了模块化的概念。

所谓模块化就是把系统代码划分为一系列职责单一，且可替换的模块。模块化开发是指如何开发新的模块，和复用已有的模块来开发目前的功能。



模块化的优点

可维护性

代码的可维护性的一种理解是，新功能的添加无需修改已有代码，旧有功能的变更无需修改多处代码。

对于初期需求不明确，需要采用不断迭代方式开发的项目，代码可维护性就显的尤为重要。

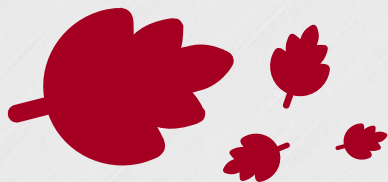
代码的复用

代码复用不仅仅是为了节省开发时间，同时也是保证代码质量的有效手段，代码的复用程度越高，其质量就越容易得到保证。

异步加载，如果一个JS出了问题，其他JS还可以继续加载。

多人协作的需求

大型应用无法通过一人之力完成，多人协作是不可避免的。在多人协作的环境下，经常要面临修改或使用别人写的代码的问题。只有那些功能单一，接口明确，模块化代码我们才敢放心大胆的修改或使用。



模块化的缺点

对代码的质量要求高

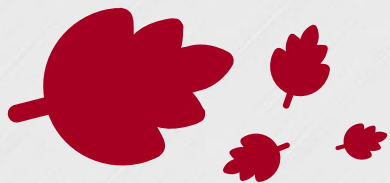
起步可能比较难。我们习惯了直接引用和复制粘贴代码，但是模块化有严格的要求，每个js必须都是以 `define(function(require,exports,module){})` 开始的，并且有时候要修改库文件才能使用。对代码的质量要求比较高，我们写的每个函数最好设定回调函数。

引用的文件个数多

因为把大功能拆分成了好多小功能，所以引入的js文件比之前写在一个文件的多。不过一般体积小，相对于它的好处，这个不足可以忽略不计。

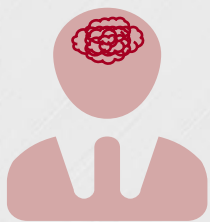
Handlebars在加载时被预编译

预编译是需要时间的，相对于浏览器直接加载string字符，加载handlebars是需要更多一些时间的。



Seajs介绍 |

是阿里的工程师叫玉伯开发的。

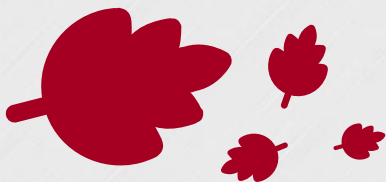


使用方法具体例子介绍。需要注意的是：默认路径是seajs所在的路径。

访谈：<http://www.iteye.com/magazines/119>

Seajs参数配置：<http://www.cnblogs.com/ada-zheng/p/3284478.html>





Handlebars介绍 |

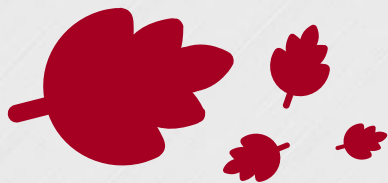
Handlebars 是 JavaScript 一个语义模板库，通过对view和data的分离来快速构建Web模板。它采用"Logic-less template"（无逻辑模版）的思路，在加载时被预编译，而不是到了客户端执行到代码时再去编译，这样可以保证模板加载和运行的速度。

官网：<http://handlebarsjs.com/>

使用教程：

<http://www.cnblogs.com/iyangyuan/archive/2013/12/12/3471227.html>

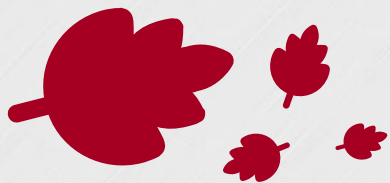




适合什么样的项目 |

大型的，功能复杂的，共用模块多的，后续有不定期新增功能的项目。比如后台开发、流+的企业平台、阅读。对于一些简单的活动页面，功能单一，甚至是静态页面，赶时间的项目不建议使用。





我们的约定



开发规范条例



所谓无规矩不成方圆，我们是一个集体大家庭，所以需要大家来一起遵守以下一些约定：

THANKS

前端开发：何清梅