兰亭前端模块化开发小结

-- 董玉伟

回顾

- https://github.com/dongyuwei/litb-fe-prototype
- **兰亭前端模块化架构设计目标：**

---

1. **前后端分离**：前端源码独立控制，消除页面内嵌js，css。
2. **模块化**：js，css，html均模块化，提高代码复用度。清晰的依赖关系管理，使修改更安全，扩展更容易。
3. **模板复用**：前端提供模板，后端只提供渲染模板需要的数据及接口。模板和数据是严格分离的。
4. **并行开发**：前后端可并行开发，后端只关心提供数据及接口服务。多项目可并行开发，可快速分支开发。
5. **独立发布**：前端代码（js，css，img）可独立发布上线（快速发布，频繁发布），加快功能迭代和hotfix速度。
6. **提高性能**：消除大部分页面内嵌js和css，提高静态资源缓存利用率和浏览器解析渲染html的性能。

- 基本达成当初的设计蓝图

模块化工程目录结构

- lightsource/
- ├── _docs
- ├── img
- ├── js
- ├── less
- ├── page
- ├── pagelet
- ├── _test
- ├── _tools
- └── widget
-

有哪些新花样？

- Less--- 辅助 css 模块化开发
- Bootstrap 模块化的 less(css) 组件，新的 reset
- Mustache(nodejs,php)--- 无逻辑模板，强制分离 model 和 view. 子模板复用．前端也可以使用 Mustache.js

辅助开发工具 ria-packager

- Install nodejs
- [sudo] npm install -g ria-packager
- ria-packager -start # 默认使用当前目录为documentRoot
- 打包工程 ria-packager -from xxx -to xxx [-v]

三大组件级别

- Widget　最小模块单元
- Pagelet　中型粒度模块
- Page　完整页面

plugin

- jQuery js plugin 可复用的插件
- Less plugin(mixin, variables, module)

Reset && Grid

- 新的 reset
- 固定宽度 Grid
- 后续可能会采用流式布局 grid

组件交互 -- 自定义事件

- 采用自定义事件在不同组件间交换信息
- 组件级别
- 全局事件　$(window).on(“xyz”,function(e,data){});
- 事件传递的自定义数据为 Object 类型

按需加载子模块 json

- require('x/y/z/_test/main.json')
-

```
{
    "product_list" : [
        require("/widget/prod_item/_test/main.json"),
        require("/widget/prod_item/_test/main2.json"),
        require("/widget/prod_item/_test/main.json"),
        require("/widget/prod_item/_test/main2.json"),
        require("/widget/prod_item/_test/main.json"),

        require("/widget/prod_item/_test/main2.json"),
        require("/widget/prod_item/_test/main.json"),
        require("/widget/prod_item/_test/main2.json"),
        require("/widget/prod_item/_test/main.json"),
        require("/widget/prod_item/_test/main2.json"),

        require("/widget/prod_item/_test/main.json"),
        require("/widget/prod_item/_test/main2.json"),
        require("/widget/prod_item/_test/main2.json"),
        require("/widget/prod_item/_test/main2.json"),
        require("/widget/prod_item/_test/main.json")
    ]
}
```

前后端联调

- 先沟通确定接口
- Fe 开发机上代码每 ***2*** 分钟 自动更新
- Php 使用 mustache 引擎时远程调用 fe 服务器上的 mustache 模板 (.html)
- http://fe.tbox.me:8888/lightsource/pagelet/recommended_products/main.html**?dev=true**

代码管理

- Lightsource 使用独立 svn 仓库
- Php 使用 svn 外链引用 tags/lightsource
- tags/lightsource 是最新稳定可发布版 , 是打包压缩 , 合并文件后的版本 .

发布上线

- 静态资源发布到 cdn: lbox.me/lightsource
- Md5 版本号控制静态资源缓存
- ria-packager 打包模块化工程时自动生成每个 js 和 css 的 md5 hash
- php 从 **md5_mapping.json** 中读取每个 js,css 的 md5 hash, 附加到 url 后 . 如 main.js?v=123abc
- 模板随 php 代码一块部署

特殊目录文件处理

- 以 . 及 _ 下划线开头的目录及文件 , 打包时自动忽略 , 不处理 .
- _docs 是模块化组件文档
- _tools 有几个自动化工具脚本

Widget 和 pagelet 测试模板

- Widget 和 pagelet 中使用的独立 html 片段 , 没有头尾 .
- 开发测试时需要自动附加合适的头尾 .
- 头尾引入模板对应的 js 和 css.
- 默认使用 ria-packager 自带的 _layout.html
- _test 目录下 _layout.html 可重载模板

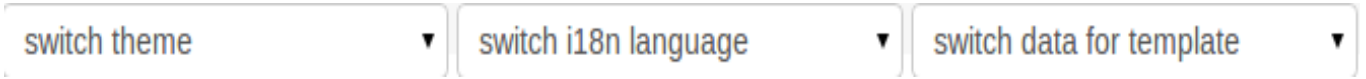
HTML5

- <!DOCTYPE html>
- <html dir="{{{dir}}}" lang="{{{lang}}}">
- 拯救 IE9:
- <meta http-equiv="X-UA-Compatible" content="IE=edge" />
- 拯救 IE6~8:

```
$.each(["abbr","aside","details","footer","header","section","hgroup","nav","summary"],
    function(i,el){
    document.createElement(el);
});
```

切换模板数据，皮肤，语言

- 一份模板，多份 (测试) 数据
- 一份模板 ,2 份皮肤
- 切换语言 : 希伯来语样式 !!!!



Global config

- JSON.stringfy && $.data()
-
```
{
    "google_analytics" : "<script type='text/javascript'>/* php should
-
    "globalConfig" : JSON.stringify({
        "baseURL": "http://www.lightinthebox.com/",
        "resourceURL": "http://cloud.lbox.me/resource_90989/dev_v2/",
        "siteWater": "",
        "imageURL": "http://cloud.lbox.me/",
```

```
<script type="text/javascript" id='_GlobalConfig_' data-config='{{globalConfig}}'>
    !window.jQuery && document.write('<script src="' + litb.lightsourceURL + 'js/base/jquery.
        js?v={{{jquery_md5}}}"><\/script>');

    var litb = $('#_GlobalConfig_').data('config') || {};
```

工具篇

- ria-packager
- html prettify 格式化 html(js,css) 代码 sublime 插件
- jsonLint 精确诊断 json 数据结构错误 .
- Fiddler2 调试线上 bug 或者测试环境 bug 时 , 可通过 fiddle 转发特定资源到本地服务器上 .

工具篇

- Css Sprite Online Builder
- export all language js 在线导出 litb 和 mobile 工程的语言包文件 (en.js,...,he.js) lightsource 工程使用 litb 相同语言包 (纯语言包 , 不包含其他 js 逻辑代码).

有待改进点

- Html 压缩
- 模块错误隔离 : try catche 模块级错误 , 避免影响整体代码执行 . 隐藏炸弹 <---> 性能考量
- js 模块使用严格的 namespace 来管理 .jQuery 插件都绑定到 $.fn 或者 $ 上 , 容易冲突
- button 和 icon, 新系统计划使用模板和翻译文本来做新 ui 设计 .
- 第三方代码加载方式优化 , 统一管理 .

展望

- 按计划推进模板模块化迁移
- 继续完善基础架构和辅助工具
- 自动发布，独立发布，频繁发布。

end

- 3ks!