基于SEAJS的前端框架实践

Seajs干什么？

* 以良好的形式组织前端资源：JS，CSS，tpl
* 配合SPM工具，压缩合并，轻松打包上线~
* 模块化的开发，简单，愉悦~

Seajs优点

* 简洁的API，五分钟上手，有文为证：http://seajs.org/docs/#quick-start
* 遵循CMD，如Nodejs一般的开发体验
* 方便的调试接口供君使用
* 适合Web Page，Web App的开发

以往一个小项目中页脚的那一堆<script>

```
</body>
<script src="/lib/jquery.js" type="text/javascript"></script>
<script src="/lib/jquery-ui.js" type="text/javascript"></script>
<script src="/lib/fancybox.js" type="text/javascript"></script>
<script src="/lib/underscore.js" type="text/javascript"></script>
<script src="/lib/backbone.js" type="text/javascript"></script>
<script src="/module/pic.js" type="text/javascript"></script>
<script src="/root/main.js" type="text/javascript"></script>
<script src="/root/slide.js" type="text/javascript"></script>
<script src="/root/dialog.js" type="text/javascript"></script>
<script src="/root/user.js" type="text/javascript"></script>
<script src="/root/index.js" type="text/javascript"></script>
</html>
```

使用Seajs以后~~

无需头疼<script>的顺序对JS执行造成的影响了

```
</body>
<script src="/Zonda/core/1.3.0-dev/sea.js" data-main="/Zonda/init"></script>
</html>
```

加载依赖的模块

do something~

comment.js (~/Desktop) – VIM

```
/**
+--  2 lines: * comment.js-------------------------------------------------------------------------------------
 */
define(function(require, exports, module){
+--  7 lines: *-----------------------------------------------------------------------------------------------

    var Util = require('util');
    var Backbone = require('backbone');
    var $ = require('jquery');

    var el = $(".ajax-pagination");

    // Model
    // -----
    // 分页模型
    var Model = require('./model');

    // 构造分页模型
    var pageModle = new Model( el.attr('page-url'), el.attr('data-value') );

+-- 22 lines: Controller-----------------------------------------------------------------------------------

    // 是否已经初始化过
    var INIT = 0;

    // function init
+-- 23 lines: function init ( type, page ) {-----------------------------------------------------------------

    // Hash Route
    var pageRoute = Backbone.Router.extend({

        routes : {
            '' : 'main',
            'comment=:type/page=:page' : 'main',
            'comment=:type' : 'main'
        },

        // Main Controller
        main : main

    }); // END pageRoute

    // 初始化
    new pageRoute();
});
```

框架是什么?

* 个人浅显的理解：一种可复用的设计
* 真正的框架？YUI3，KISS，JX，Google Closure...
* 那么jQuery是类库，Backbone，Seajs也是类库？（一个弱弱的声音：是的~）

自己折腾框架的意义

* 学习，理解，实践，积累~（对于吾等菜鸟来说确实如此）
* 可以使用自己或者团队喜欢的API（DIY？）
* 成就感（或者虚荣心？）~~（Nevermind，么么哒）

将会遇到的问题

* 自己折腾出来的是框架么？（废柴，畸形，蹩脚...）
* 重新发明轮子?
* .....

我们的框架要有哪些东西?

* 模块化，Seajs轻松胜任此项工作
* DOM操作，Ajax，jQuery说：干这一票，没有比我更适合的人选了！而且他还带来了一群战斗力超强的小弟~
* 事件，路由，MVC，Backbone靠你了
* UI组件，Bootstrap叼着雪茄说：没问题~
* 组织CSS，Less向前走了一步
* 谁来做坚实的后盾？Underscore嘴角荡漾着微笑 :)
* 最后一个问题：那你身为这个框架的老大，你干什么呢？
* 答：在小弟的帮助下，封装一些常用的方法，个性化API~（其实我的工作最轻松啦）

这就是框架?

* 远远不是吧？像YUI一样的框架？不要搞笑了
* 开发框架就这么简单？开玩笑？？
* 标题党?
* 这个应该算是工具集，它只是简单的规定了构建在它之上的应用程序能做什么，大概怎么做而已，它还有一些你用着比较习惯的API，所以.....作为菜鸟的起步，天真而勇敢的尝试还是值得鼓励的，哇哈哈（自己鼓励自己吧，骚年）~

实践一把~

Zonda

拿自家的改装车Zonda来说事儿咯~

基本上Zonda的思路就是上一页那种工具集的思路（而非严格的框架），它的特点如下：

目录结构简单，清晰；

提供有三个项目状态：prod（生产），dev（开发），test（测试），“一键切换”；

一些实用模块：多文件上传（拖拽），幻灯片，对话框，表单验证，右键菜单，状态机等等（拿得出手的就这些啦~）

Zonda目录结构

app	4 days ago	删除无用的debug文件 [smallsmallwolf]
core	18 days ago	close #21 只需将seajs的插件合并放到sea.js的尾部即可 [smallsmallwolf]
css	a day ago	增加响应式布局支持 [smallsmallwolf]
demo	6 days ago	更新util至0.1.1 [smallsmallwolf]
dist	4 days ago	无法用正常模式使用spm，hack之 [smallsmallwolf]
images	a month ago	close #17 连接数至5个 [smallsmallwolf]
lib	a day ago	继承QUit测试环境 [smallsmallwolf]
test	21 hours ago	更新QUit至1.10.1版本，并调整测试方式 [smallsmallwolf]
tool	a day ago	继承QUit测试环境 [smallsmallwolf]
util	a day ago	继承QUit测试环境 [smallsmallwolf]
.gitignore	2 months ago	重新加入.gitignore [smallsmallwolf]
LICENSE.md	3 months ago	更改目录结构，比较大的修改 [smallsmallwolf]
README.md	a day ago	Update README.md [smallsmallwolf]
init.js	21 hours ago	更新QUit至1.10.1版本，并调整测试方式 [smallsmallwolf]
package.json	a day ago	继承QUit测试环境 [smallsmallwolf]

代码组织

CSS / Less

build/: 编译好的CSS

common/: 可复用的

lib/: Bootstrap, Typo,

unit/: 项目中不可复用的

ie/: 兼容IE

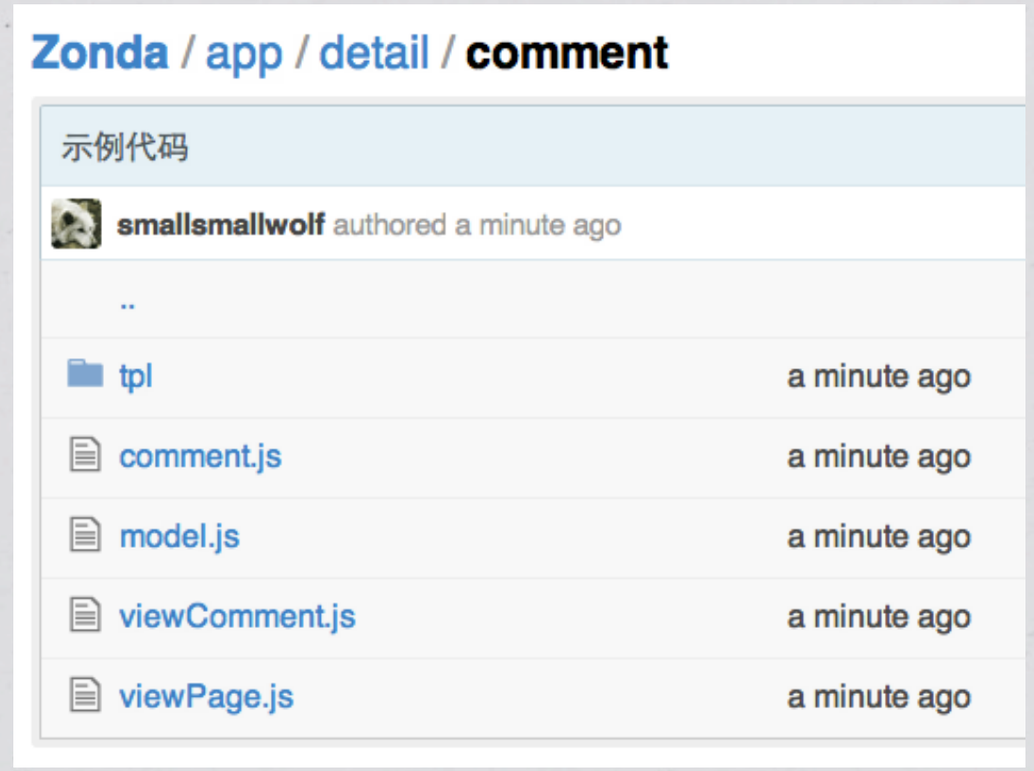
build	6 days ago	更新util至0.1.1 [smallsmallwolf]
common	a day ago	增加响应式布局支持 [smallsmallwolf]
ie	3 months ago	提供bootstrap栅格系统IE6支持 [smallsmallwolf]
lib	4 days ago	更新Bootstrap到2.2.1 [smallsmallwolf]
unit	23 days ago	改进less编译脚本 [smallsmallwolf]
init.less	a month ago	close #20 [smallsmallwolf]

代码组织

Javascript

这里只是个Web App的例子，按照Backbone的方式对app/目录下的某个子模块进行的划分。

对于Web Page式的项目，可以不必如此。



路由

在Web App中常根据Url Hash做路由，用Backbone就ok了~但是实际的应用场景中——我们做过的很多项目还是Web Page的，那么怎样按需加载执行Javascript呢？

用DOM！嗯，Zonda确实这么做了，在项目中实践下来的感受是：基本够用，但是不灵活（水平有限，智商是硬伤，想不到其他更好的法子）.....

```
22 var Util = require('util');
23
24 Util.route({
25
26     '#main .sub-nav' : function () {
27         require('app/sub/navView');
28     },
29
30     '.sub-view' : function () {
31         require('app/sub/subView');
32     },
33
34     // ...
35 });
```

测试

集成QUit。

其实很容易的就把它拉进来了，就是每个地方要加个前缀有点麻烦，还在想办法解决~

```
15 +--  4 lines: *-------------------------------------
19 define(function(require, exports, module){
20     var Q = require('qunit');
21
22     Q.module( "group a" );
23
24     Q.test( "a basic test example", function() {
25
26         Q.ok( 1, "this test is fine" );
27
28     });
29
30     Q.test( "a basic test example 2", function() {
31
32         Q.ok( true, "this test is fine" );
33     });
34
35 });
```

Tool / build.sh

build.sh用来切换应用的状态：

./build.sh prod : 将app目录下的所有模块压缩合并为一个，放到Zonda/dist/下，最终上线版本的js连接数将会从开发版的35+降至3

./build.sh dev : 回到开发版本

./build.sh test : 启用测试

```
case $1 in
    #打包Zonda-Util模块
    util)
        cd ../util
        spm build -v
        cp dist/util.js ./
        rm -rf dist
        echo Util模块打包完成
    ;;
    # 打包项目
    prod)

        cd ../
        echo 开始打包Zonda

        #修改init.js中的线上版本为开发版本
        sed -i "s/app_version_type='\w*'/app_version_type='prod'/g" ./init.js

        #模拟spm标准目录
        mkdir src
        cp -r app/* src/

        #spm 打包
        spm build -v

        #删除打包临时源文件
        rm -rf src/

        #显示指明需要调用的模块ID
        echo 'seajs.use("#app/app");' >> dist/app.js

        echo Zonda-App打包完成
    ;;
    # 开发模式
    dev)
        cd ../
```

Tool / less2css.js

```
>>>>>>>>>> sub.less 编译错误！ <<<<<<<<<<

{ [Error: Command failed: NameError: .rating is undefined in /home/shiyang/www/Fruit84/Zonda/css/unit/sub.less:35:0
34
35 .rating ( @gray );
36

] killed: false, code: 2, signal: null }
```

./less2css.js 运行（需要执行权限），它跑在Node上，启动后会监听父级目录中 css/ 文件夹下的less文件的改变，保存less文件，即会编译为CSS文件，报错会显示在控制台中。

Zonda/css/下的less文件（包括Bootstrap，Typo）最终可以被Less编译成一个css文件，页面上加载的也是这个文件。

Zonda为什么而生?

* 每次做（中小型，周期短）项目时，都要逐个建文件夹，把jQuery什么的拖进来等等繁琐且重复的工作，于是就有了它的雏形，后来邂逅了Seajs，于是就基于Seajs对它做了很多的改进;

* Zonda只是个小“框架”（姑且称之为框架吧），不像YUI3那样的高富帅，而是给像我这样的屌丝用的，虽然它用意大利Pagani的超级跑车Zonda（风之子）做名字，但是它确实只是一辆改装过的公路摩托车而已（终于说出了事实）~

质疑

* “你是在拼拼凑凑的弄一个YUI出来么？很拙劣啊，切。”
* “这特么就不是框架，顶多就是个工具集嘛，太水了。”
* “里面都是别人的东西，你说个锤子哟.....”
* “只能在Linux下用，太鸡肋了吧？而且还要装Node，还要装SPM，Less什么的。”
* .....

回答

* YUI3是一辆McLaren（迈凯轮）车队的高科技F1赛车，高富帅也不一定玩得起，而Zonda只是辆经过改装和调教的公路摩托车，可以享受到骑乘的乐趣。但它还不是严肃的前端框架，所以没有可比性（而且我还没有水平模仿YUI3啊，努力啊努力）。

* Zonda确实更像工具集，目前还够用，自己写的代码量还不是很多，还要加油，努力掌握核心科技~~

* 至于只能在Linux下用的问题，因为我即使用了Mac也还是会打开终端ssh到Linux DEV服务器上工作的啊，这是个习惯问题，Zonda就是按照作者的习惯来的，就像有的人习惯手动挡，有的人习惯自动挡一样。

The END

非常感谢，欢迎批评指正~

About Me

Degas / 施扬

Github ID:smallsmallwolf

Email:smallsmallwolf@gmail.com