[部门现有框架介绍及demo实践](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579474)

[转至元数据结尾](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579474#page-metadata-end)

* 创建： [莫鹏程](http://10.11.112.49:8090/display/~MOPENGCHENG524)，最新修改于： [十二月 08, 2016](http://10.11.112.49:8090/pages/diffpagesbyversion.action?pageId=2579474&selectedPageVersions=20&selectedPageVersions=21)

[转至元数据起始](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579474#page-metadata-start)

**部门现有框架**[**下载地址**](http://git-ma.paic.com.cn/papuhui/pah5_base_frame)**(git)**

**现有项目中包含的所有引用包(node\_modules)**[**下载地址**](http://git-ma.paic.com.cn/papuhui/node_modules)**(git)**

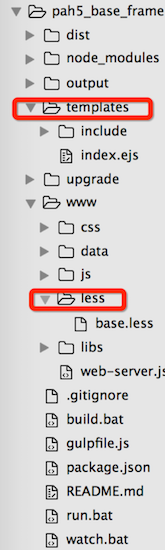
1. **框架主要技术**

Underscore + Backbone (View) + RequireJS + Zepto  
  
Underscore：JavaScript 工具库，负责框架的数据处理  
Backbone：分离其中View进行使用，负责框架的视图层处理，依赖于Underscore  
RequireJS：用于框架中模块化开发管理，需配置require-config.js  
Zepto：轻量级jQuery

1. **框架目录结构**

点击展开

 pah5\_base\_frame

1. **简单Demo实践**
   1. 前往内网git-ma下载基础框架（pah5\_base\_frame）和框架中引用的node\_modules，建议将node\_modules放在Mac当前用户根目录下（即“~”目录）
   2. 将重构提供的html(ejs)、css(less)文件分别放入如图所示的templates、www/less目录下  
      
   3. 根据项目需要，将静态页面布局中部分内容抽离成模板
   4. 抽离模板前：
   5. <body>
   6. <button id="js-add">添加一项</button>
   7. <ul class="list js-list">
   8. <li class="js-item" data-index="0">
   9. <div>2016/12/7 下午4:12:30</div>
   10. <div>Hello</div>
   11. </li>
   12. <li class="js-item" data-index="1" style="background:#fefbe5;">
   13. <div>2016/12/7 下午4:12:30</div>
   14. <div>Hello</div>
   15. </li>
   16. <li class="js-item" data-index="2">
   17. <div>2016/12/7 下午4:12:30</div>
   18. <div>Hello</div>
   19. </li>
   20. <li class="js-item" data-index="3" style="background:#fefbe5;">
   21. <div>2016/12/7 下午4:12:31</div>
   22. <div>Hello</div>
   23. </li>
   24. </ul>
   25. </body>
   26. 抽离模板后：
   27. <body>
   28. <button id="js-add">添加一项</button>
   29. <ul class="list js-list"></ul>
   30. </body>
   31. <script type="text/template" id="listTpl">
   32. <% $.each(data, function(index,item){ %>
   33. <li class="js-item" data-index="<%= index %>"
   34. <% if(index %2 != 0){ %>
   35. style="background:#fefbe5;"
   36. <% } %>
   37. >
   38. <div><%= item.time %></div>
   39. <div><%= item.name %></div>
   40. </li>
   41. <% }) %>
   42. </script>
   43. 将header、footer替换为公共的header、footer，其中header通过传入title设置页面标题，footer通过传入main指定该页面对应的js脚本文件
   44. <@- include('include/\_header.ejs', {data:{title:"演示"}}); @>
   45. <@- include('include/\_footer.ejs',{data:{main:"js/index"}}); @>
   46. 进行js脚本编写
       1. 按AMD规范书写，同步加载需要引用的模块
       2. 以Backbone.View为基础进行开发，将页面事件以extend()方法注入View中
       3. 几个主要方法/对象说明：
          * initialize()：页面初始化方法，会自动调用，且仅在页面生成或刷新后调用一次
          * render()：页面渲染方法，需手动调用
          * events:{}：页面交互事件对象，定义对象触发事件
   47. 接口相关
       1. 后端接口请求
          * 通过ajax请求，ajax基本配置在www/js/common/init.js中
          * 新增接口在www/js/common/api.js中增加（注意区分测试环境）
          * var api = env == "DEVELOPMENT" ? {
          * //DEVELOPMENT环境
          * GET\_ITEM : "data/getItem.json",
          * } : {
          * //非DEVELOPMENT环境
          * GET\_ITEM : "getItem.do",
          * };
          * 本地模拟接口数据json文件通常放置于www/data目录下
          * 调用ajax时按以下形式重写即可
          * $.ajax({
          * url: C.Api('GET\_ITEM'),
          * type:'get',
          * success: function (res) {
          * //成功回调
          * },
          * error: function (res) {
          * //失败回调
          * }，
          * complete: function (res) {
          * //完成回调
          * }
          * });
       2. Native接口请求
          * 通过C.Native方法调用，即www/js/common/native.js
          * 新增接口需先在www/js/common/adapter.js的methodList中增加，并在www/js/common/native.js定义接口传参默认值
          * 调用接口时按定义传参即可，如 C.Native.tip("提示内容");

**Demo下载：**[**pah5\_demo.zip**](http://10.11.112.49:8090/download/attachments/2579474/pah5_demo.zip?version=1&modificationDate=1481164615000&api=v2)

1. **调试方式**

**a. 浏览器调试**

* 1. *step 1*  
     Windows:在项目根目录下运行run.bat  
     Mac：在项目根目录下运行如下命令
  2. cd www
  3. node web-server.js
  4. -p 8877 设置端口号，默认8877
  5. -e DEVELOPMENT 设置测试环境，可选DEVELOPMENT、TEST、PRODUCTION
  6. *step 2*  
     浏览器打开  
     <http://localhost:8877/index.html?ssEnv=development>，其中index.html为需要调试页面，ssEnv=development为调试环境设置

**b. 在线资源调试(Hybird APP)**

* 1. *step 1*  
     打H5全量资源包

Windows:在项目根目录下运行build.bat  
Mac：在项目根目录下运行如下命令

gulp build

-env DEVELOPMENT 设置打包环境

* 1. *step 2*

前往Jenkins打包平台打Native在线调试包

在线资源地址配置（按项目有所不同）  


在线资源开关开启  
在线开关

* 1. *step 3*  
     将 *step 1* 打的H5全量资源包上传到 *step 2* 配置的在线资源服务器上
  2. *step 4*  
     运行 *step 2* 打的安装包即可看到代码结果
  3. *step 5*
     1. Android手机连接电脑后，在Chrome浏览器前往chrome://inspect可调试APP页面（每种型号手机首次使用该功能需科学上网）
     2. iPhone手机连接电脑后，在Safari菜单栏中选择“开发”（需先在Safari高级配置中设置）→对应调试手机名→需要调试的页面，可调试APP页面（需是可调试的ipa安装包）

**c. 映射调试(Hybird APP)**

* 1. *step 1*  
     如**方法 b *step 2***前往Jenkins打包平台打Native在线调试包
  2. *step 2*  
     参照[Charles映射请求](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579702)配置代理映射
  3. *step 3*  
     运行 *step 1* 打的安装包即可看到代码结果
  4. *step 4*  
     如**方法 b *step 5***通过浏览器调试