

移动端前端框架比较及设计

传统页面结构

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <meta name="viewport" content="width=device-width,minimum-
scale=1.0,maximum-scale=1.0,user-scalable=no"/>
    <link rel="stylesheet" href="/css/master.css" media="screen"
title="no title" charset="utf-8"/>
  </head>
  <body>
    <div class="">
      <img src="" alt="" />
      <a href="#"></a>
    </div>
    <script type="text/javascript" src="/js/master.js"></script>
  </body>
</html>
```

框架任务一 组装页面

- mvc的抽取
mode: data、Rest API
view: html
controller: javascript
- 自定义组件实现

```
< script type="text/myFramework-config"></script>

<textarea class="myFramework-config"></textarea>

<myFramework-config></myFramework-config>
```

- 解析页面的能力

```
function initParser(html){
    var pageDocNode = $('<DIV>' + html + '</DIV>');
    var templates = pageDocNode.find('script[type="text/lizard-template"]');
    var configStr = pageDocNode.find('script[type="text/lizard-
config"]').text()
};
```

- 添加配置项

```
< script type="text/myFramework-config">
    //真实的URL为 http://m.ctrip.com/webapp/hotel/detail/110
    {
        url:'/webapp/hotel/detail/{id}'
        model:{
            apis:[
                {
                    url:"",
                    postdata:{},
                    name:'api1',
                    ajaxOnly:true
                },
                {
                    url:"",
                    postdata:{
                        name:myFramework.depend('api1').name,
                        id:myFramework.params("id")
                    },
                    name:'api2',
                    suspend:function(){}
                },
            ],
            filter:function(datas){},
            initdata:{},
            errorback:function(){}
        },
        views:{
            body:myFramework.template('#template1'),
            header:"<h1>header</h1>"
        },
        controller:'myController.js'
    }
</script>
```

- model的实现
 - post、get获取
 - 数据依赖的处理

数据缓存处理

数据校验处理 : 登陆信息、安全信息

- view的实现

模版的定义

```
< script type="text/myFramework-template" id="tpl1">
  <h3>
    <% if (typeof content === "string") { %>
      <%= content %>
    <% } %>
  </h3>
</script>
```

模版的解析

```
var compiled = _.template($('#tpl1').text());
compiled({epithet: "stooge"});
```

模版预编译

```
var compiled = function($data){
  var content = $data.content, $out = "";
  $out += "<h3>",
  if(typeof content === "string"){
    $out += content;
  }
  $out += "</h3>";
  return $out;
}
```

- controller的实现

```
new pageView({
  events:{
    "click .starting": "getDepart",
  },
  "getDepart":function(){
  }
})
```

框架任务二 页面生命周期管理

- 单页面结构

```
<section>
  <div class="" id="page1"></div>
  <div class="active" id="page2"></div>
  <div class="" id="page3"></div>
</section>
```

- 提供pageView类

```
new pageView({
  pagecreat:function(){

  },
  pageshow:function(){

  },
  pagehide:function(){

  }
})
```

框架任务三 路由实现

- 前往下一个浏览页面

```
forward(url,opts)
```

- 返回前一个浏览页面

```
back(url)
```

- 支持的类型

```
ctrip://    跳转至native页面
http://同域
http://跨域
{targetModel:4} 多webview打开
```

- 部分代码实现

```
$('#body').on('click', $.proxy(function (e)
    var el = $(e.target);
    if (el[0].nodeName == 'A') {
        var href = el.attr('href');
        var opts = JSON.parse(el.attr('data-opts')||'{}');
        var jumptype = el.attr('data-jumptype')
        if (jumptype == 'back') {
            back(href, opts);
        } else if ( jumptype == 'forward') {
            forward(href, opts);
        }
    }
})

$(window).bind('popstate', function (e) {
    var data = e.state || (e.originalEvent && e.originalEvent.state);
    history.replaceState({},title, url);
})

$(window).bind('hashchange',function (e) {
    var hash = decodeURIComponent(window.location.hash);
})
```

框架任务四 跨团队合作

- 事件订阅机制的实现

```

var MessageCenter = {
  messageQueue : {},
  publish: function(message, args){

  },
  subscribe: function(message, handler, scope){

  },
  unsubscribe: function(message, handler)

  },
}
MessageCenter.subscribe('viewReady', function (inView) {

})
MessageCenter.publish('viewReady', inView);

```

框架任务五 与native交互

- 简单代码实现

```

window.app.callback = function(options) {
  var tagname = options.tagname;
  var param = options.param;
  switch(tagname){
    case "web_view_finished_load":

    break;
  }
}
var fn = new HybridShell.Fn('web_view_finished_load');

```

- 包括功能

登陆

分享

图片浏览器

获取网络状态

...

框架任务六 多平台适配

```
var ua = window.navigator.userAgent;
```

- 支持的平台

微信 MicroMessenger

winphones IEMobile

攻略APP gs_wireless

主板APP Ctrip_CtripWireless

青春版 Youth_CtripWireless

周末游 we_wireless

TV AndroidTV_CtripWireless

H5

直连 `!!(navigator.userAgent.match(/ctripwireless/i) && (window.location.protocol !== "file:"))`

- 代码处理逻辑

```
if("微信"){  
    require(["WeixinJSBridge.js"])  
}else{  
    ....  
}
```

框架任务七 常用工具提供

- 模块加载器

requireJS

alameda

almond

seaJS

- dom选择器

zepto

jqmobi

- javascript能力增强

undersore

Lo-Dash

- 点击事件处理
- fastclick
- localStorage复写
- cookie读写
- 隐私模式替代方案
-

框架任务八 seo支持

- 支持服务器渲染首屏

框架任务九 Hybrid支持

- 构建build工具

lizard 原理