

携程H5性能优化实战

从携程H5框架-Lizard分析
H5优化

携程旅行网 - 魏晓军

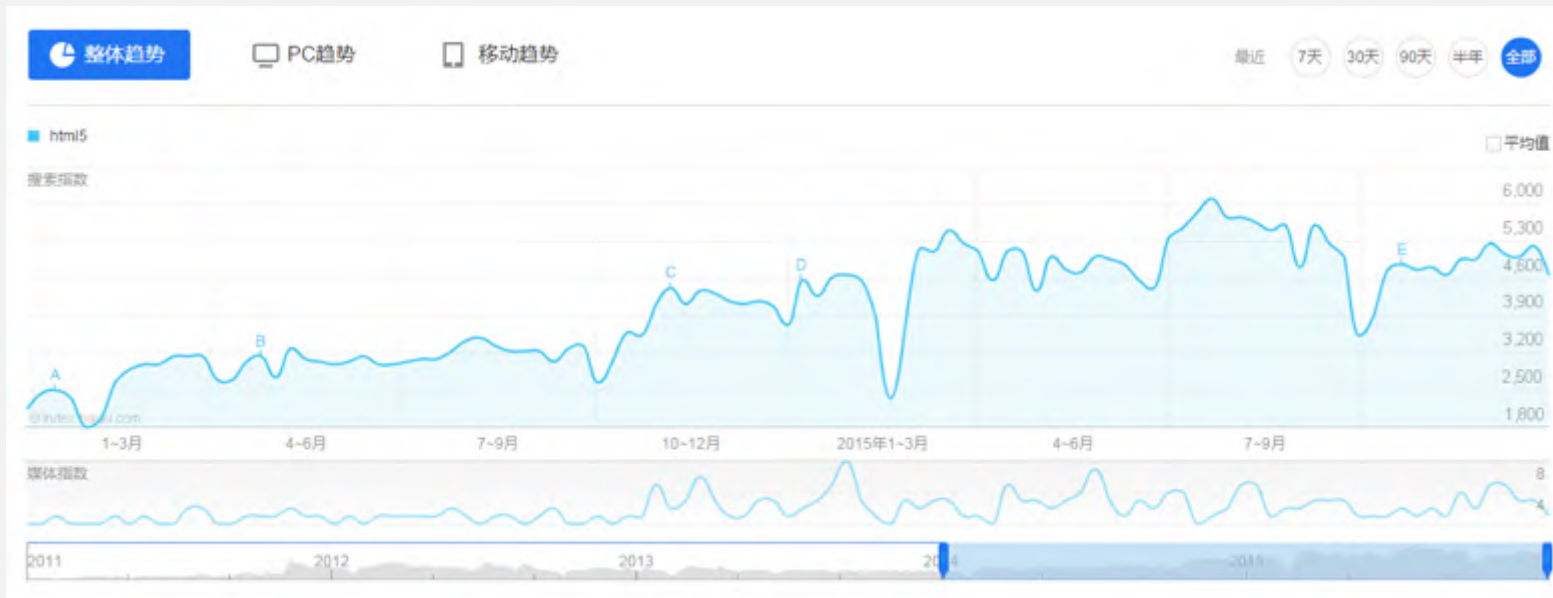


- 1、什么是H5、H5的发展趋势、H5的宿主载体
- 2、携程H5框架简介
- 3、优化知识点



HTML5是HTML最新的修订版本，2014年10月由万维网联盟（W3C）完成标准制定。目标是替换1999年所制定的HTML 4.01和XHTML 1.0标准，以期能在互联网应用迅速发展的时候，使网络标准达到匹配当代的网络需求。广义论及HTML5时，实际指的是包括HTML、CSS和JavaScript在内的一套技术组合。

这里我们把HTML5简称做H5



旅游行业、游戏领域

导流、广告

微信

跨平台，一套代码能跑多个环境、快速迭代，无需审核 开发成本低，入门容易
维护成本低，更新容易、有完善的开发生态链

浏览器

微信

TBV

第三方App

结论：H5的优化结合载体的优化才会更彻底



.....

Lizard `['lɪzəd]` 是一种两栖动物。

Lizard是为H5、Hybrid和SEO开发而产生的一套前端框架。

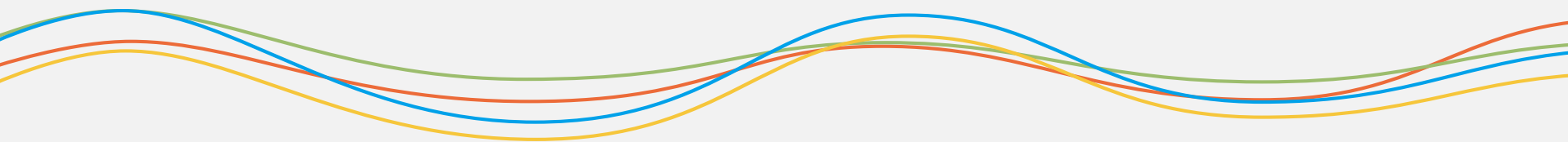
它遵循CommonJS标准，将Javascript做模块化封装，采用requirejs作为模块加载器。依赖zepto构建底层库，在其之上架构MVC框架，封装网络访问、本地数据库操作等基础操作。

通过对BridgeJS的封装，实现与native 的通信。

此外，它还提供携程特色风格的前端UI组件库和面向多种环境的Widget功能组件库。



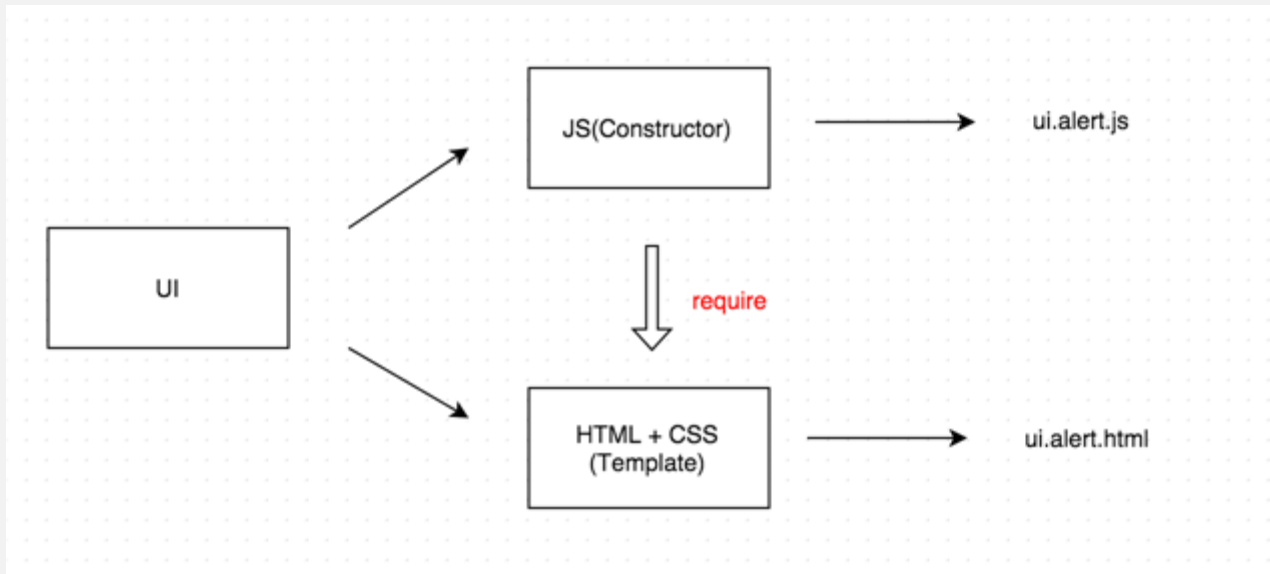
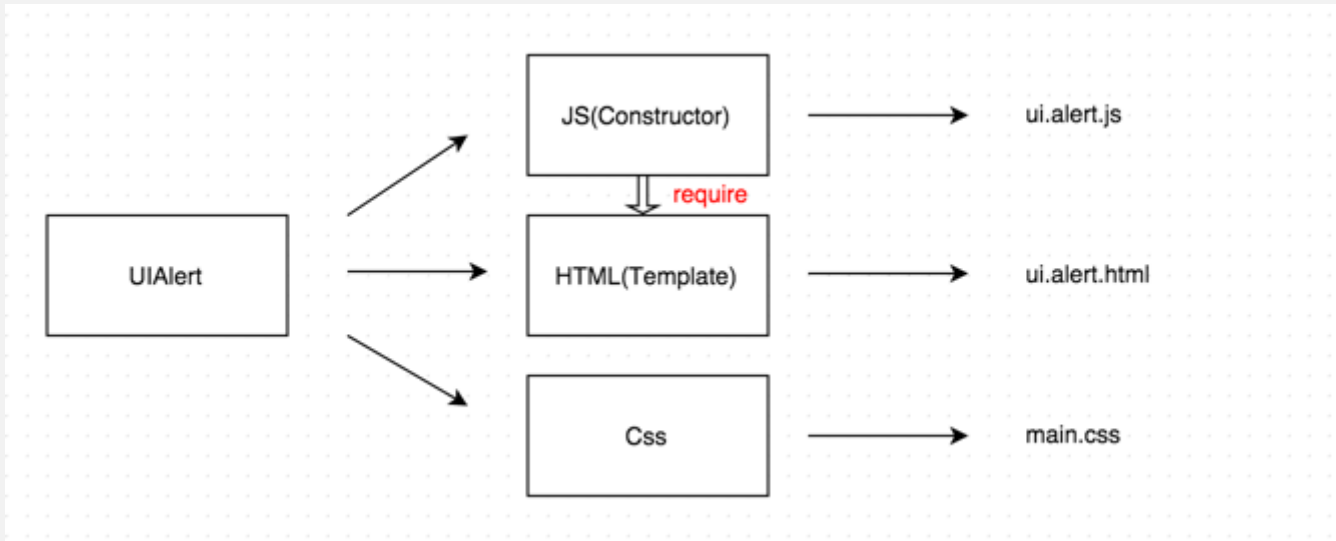
优化知识点

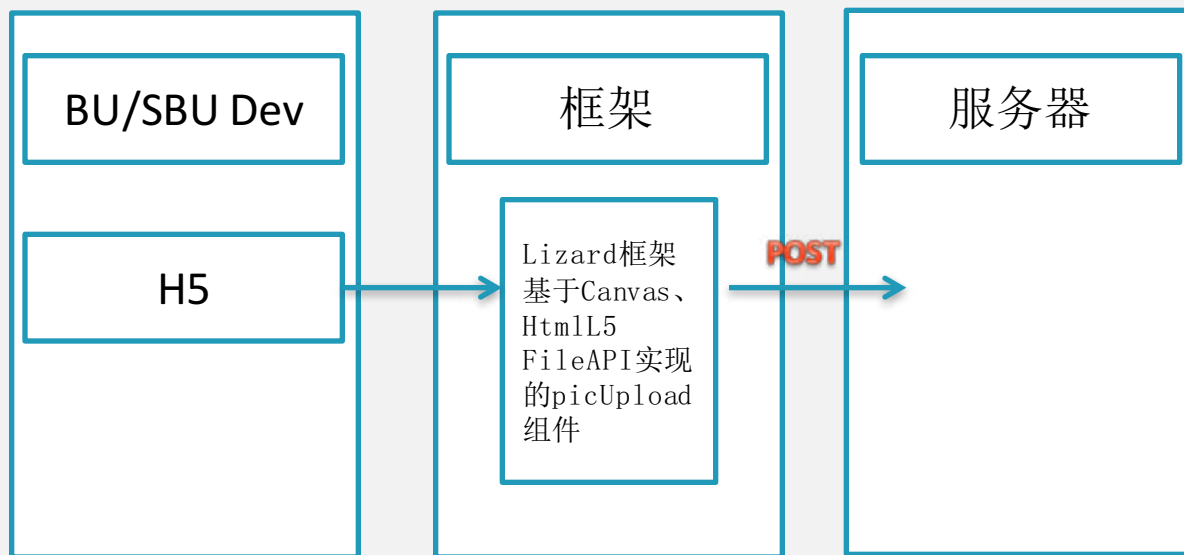


SPA (single page application) ，即单页webapp ，它具有以下优点：

用户体验，对于内容的改动不需要加载整个页面。这样不会出现白页情况，页面与页面无缝切换，甚至带有一定动画效果。

请求量少，请求内容无需服务器解析，对服务器压力较小，消耗更少的带宽。

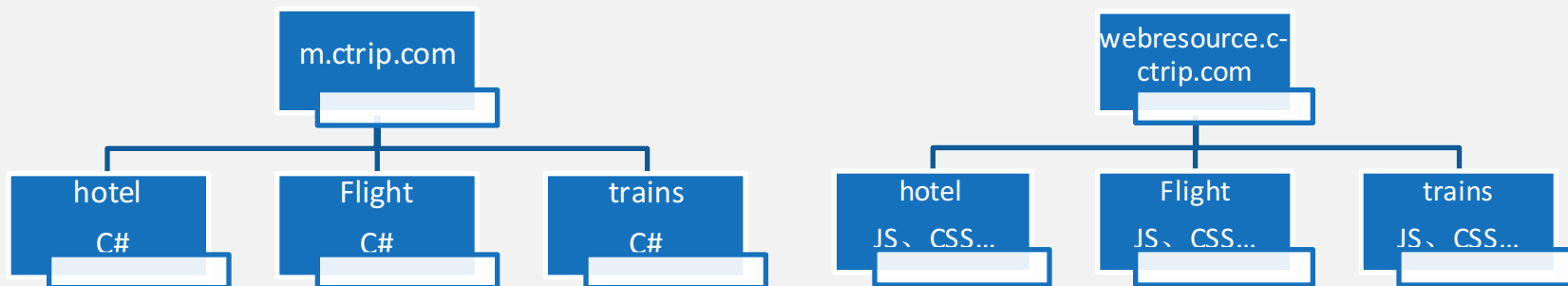




H5图片压缩上传

图片压缩效果	
体积	压缩率
>2M	90%
2M~1M	70%
<1M	10%

图片压缩效果



H5站点资源部署

```
<body>
.....
<script type="text/lizard-config">
{
  "url_schema": "{site}/hotel/{view}/{city}/{startdate}/{enddate}",
  "model":{
    "apis":{{
      url:"http://m.ctrip.com/restapi/hotelapi/product/productsearch",
      postdata:{"city":Lizard.P("city"),"startdate":Lizard.P("startdate"),"enddate":Lizard.P("enddate"),"isSale":0,"
        isPromo":0}
    }},
    "filter":function(datas){
      return datas;
    }
  },
  "model":{
    "apis":{{
      url:"http://m.ctrip.com/restapi/hotelapi/product/productsearch",
      postdata:Lizard.T("my_postdata")
    }},
    "filter":function(datas){
      return datas;
    }
  },
  "view":{
    "header":Lizard.T("my_header_template"),
    "viewport":Lizard.T("my_body_template")
  },
  "controller":"./res/views/vacationslist.js"
}
</script>
.....
</body>
```

Core、UI拆分

延迟加载功能

FakeData

ErrorData

模板预编译

ShowView功能

首屏优化效果			
频道	首页(s)	列表页(s)	详情页(s)
团购	339.4	202.2	-199.25
门票	-30.5	123.7	254.8
跟团游	229.2	1606.4	711.8
自由行	346.6	217.8	457.25
游轮	521.2	823.55	1932.6

优化效果图

AjaxReady

Domready时间 (ms)

TemplateRender

2.1

2.2

Request

Domready

内网

730

299

50K

19460

8030

Onload

250K

4510

2010

750K

1570

720

1M

1060

350

```
<div class="main-frame">
  <div class="main-viewport">
    <div id="client_id_viewport_1_1451203210113" page-url="/webapp/mkt/b2b-online/login_result?from=http://m.ctrip.com/webapp/mkt/b2b-online/" data-
      login_result" style="display: none;"></div>
    <div id="client_id_viewport_2_1451203211296" page-url="/webapp/mkt/b2b-online/" data-view-name="index" style="display: none;"></div>
    <div id="client_id_viewport_3_1451207759820" page-url="/webapp/mkt/b2b-online/Order_list" style data-view-name="Order_list"></div>
  </div>
</div class="main-state"></div>
```

缓存节点不超过5个

可自定义缓存节点

localStorage、cookie、Application Cache

缓存一些非实时数据，比如城市信息和常用联系人

数据会缓存30s-60s

1、只返回页面中需要的数据

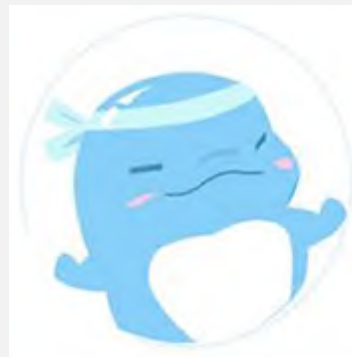
加载80kb的城市数据，包括国内和国际。建议默认加载国内的，当用户切换的时候再请求国际的数据，提升选择城市的数据

2、非首页只需返回模版和配置项数据

在当前生产环境中，由于我们采用的都是单页应用，除了进入页面是需要html文件的头尾之外，跳转到第二个页面时，其实需要的只是BU配置的lizard-config以及lizard-templete中的内容，其他的内容都是不需要的，但是在我们的ajax请求时，返回的仍然是一个完整的doom，这就增加了返回值的体积，



Base64图片



canvas

	image	css + html	js	total
原loading	7.5KB	8.85KB	1KB	9.95KB
改造后loading	0	713B	4.19KB	4.2KB

改造后的loading总体积减少58%

Resource	Method	Status	Type	Alias	Size	Time
lizard.seed.js	GET	200	script	cf1	117 KB	3.37 s
ui.loadFailed.js	GET	200	script	lizard.seed.js:7	19.0 KB	642 ms
ui.loading.js	GET	200	script	lizard.seed.js:7	8.3 KB	434 ms
config.js	GET	200	script	lizard.seed.js:7	15.8 KB	434 ms
_mubt.min.js?v=20159_26.js	GET	200	script	lizard.seed.js:1	12.0 KB	569 ms
new_guide.png	GET	200	png	Other	19.4 KB	1.01 s
index.js	GET	200	script	lizard.seed.js:7	2.0 KB	383 ms
data:image/png;base...	GET	200	png	lizard.seed.js:8	0 B	6 ms
c.page.factory.js	GET	200	script	lizard.seed.js:7	1.1 KB	216 ms
c.page.common.list.js	GET	200	script	lizard.seed.js:7	1.4 KB	242 ms
h5Redirect.js	GET	200	script	_mubt.min.js?v=2	4.0 KB	1.71 s

Resource	Method	Status	Type	Alias	Size	Time
car.css?v=20151023_11_VD	GET	200	stylesheet	car2	39.9 KB	1.03 s
lizard.seed.js	GET	200	script	car29	1.5 KB	293 ms
concat?f=/code/lizard/2.1/web/lizard.core.js/code/lizard/2.1/web/lizard.web.js/Res...	GET	200	script	lizard.seed.js:7	108 KB	2.08 s

1、图二的核心JS尺寸较之图一有了大幅的下降（从120K下降到80K—50%的减少）

2、利用CDNCombo技术，有效的减少了http请求的次数

不常用代码不要放入框架中、去除H5\Hybrid杂糅代码

拆分首屏必需框架JS和单页路由JS

做了功能拆分后，下载资源减少50%多，所以加载时间也缩短一半，可以认为从打开浏览器下载资源到页面渲染完成2.2 比2.1 提升约40~50%！

定位模块要包含国内，国外的功能（gzip有20多K）

情感化模块 UILoadingFailed, UIAlert, UIToast

非常用工具模块

以前的逻辑是，在bu首屏的数据取回来，然后doom节点展示完成之后，再去阻塞式的加载bu业务的controller，然后再绑定dom事件等等，这中间两个异步的请求事件是串行的，导致时间 \geq 两者之和；

优化后的逻辑是，两个异步的请求同时加载，然后在处理业务时相互等待，在不影响原来业务处理顺序的前提下，总时间为两个异步之中，请求时间最长的一个。



THANK YOU

