**Lizard1.1升级2.0快速手册**

框架研发部 前端框架组

2014-04-24

目录

[Lizard2.0的新特性 3](#_Toc386109230)

[SEO的架构设计 3](#_Toc386109231)

[用户/SEM 3](#_Toc386109232)

[SEO 4](#_Toc386109233)

[SEO的部署 5](#_Toc386109234)

[服务器部署 5](#_Toc386109235)

[分服务器部署 5](#_Toc386109236)

[如何升级到2.0 5](#_Toc386109237)

[框架引用修改 5](#_Toc386109238)

[从html搬迁到.net页面 6](#_Toc386109239)

[代码整合 6](#_Toc386109240)

[View文件更新 7](#_Toc386109241)

[URL\_Schema 7](#_Toc386109242)

[Model 8](#_Toc386109243)

[View 11](#_Toc386109244)

[Controller 12](#_Toc386109245)

# Lizard2.0的新特性

## SEO的架构设计

### 用户/SEM



### SEO



## SEO的部署

### 服务器部署

用户/SEM页面和SEO页面部署在两台服务器上。

### 分服务器部署

更多详细内容请参看 <http://conf.ctripcorp.com/display/FRAM/H5+SEO+Solution>

# 如何升级到2.0

## 框架引用修改

框架引用只需要一个文件

[//webresource.c-ctrip.com/code/lizard/2.0/web/lizard.js](http://webresource.c-ctrip.com/code/lizard/1.1/web/common.js)

## 从html搬迁到.net页面

开发人员需要将原本的view html页面更新为.net页面。这样有以下几个优势

1. 一份代码同时兼容用户访问、SEM投放和SEO爬虫
2. 支持页面间POST数据
3. 实现服务器端路由控制
4. View与Controller路径解耦
5. 便于以后接入其他服务端框架，例如ABTesting

## 代码整合

在开发页面时，只需要关注以下4个要素

1. URL-Schema：用户访问的URL中QueryString参数定义
2. Model：包括请求数据的URL、POST Data和数据处理Filter函数
3. View：页面的HTML内容，包含head和body的template
4. Controller：页面逻辑代码处理

代码示例



### View文件更新

将原本的view更新为一份独立的静态页面，包括头部和尾部。

提示：更新为.net页面，可以使用master page来引入公共的头尾。

### URL\_Schema

将页面关键参数从LocalStorage中搬迁到QueryString。根据业务提取出关键参数。

例如某用户访问的页面是

<http://m.ctrip.com/webapp/hotel/index/shanghai/20140501/20140502>

我们需要定义这个URL-schema为

/{site}/hotel/{view}/{city}/{startdate}/{enddate}

* site : webapp/html5(user/seo)
* view : 此页面对应的view的html文件名称
* city/startdate/enddate : 关键参数定义

Lizard会根据schema来解析对应的参数。

提示：开发人员可以使用Lizard.P(“city”)来获取参数值

#### Lizard.P(param\_name)

Usage: 获取QueryString对应的参数值

Input: QueryString的参数名

Output: 对应参数的值

Example: Lizard.P(“city“) 返回为shanghai

代码示例：

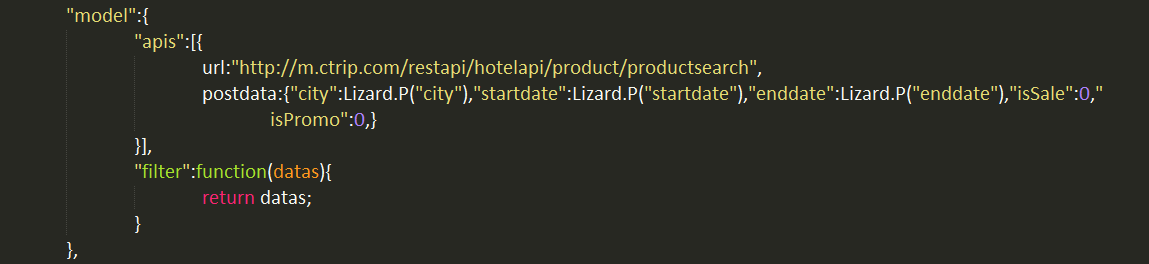


"url\_schema" : "/{site}/hotel/{view}/{city}/{startdate}/{enddate}"

可以参考SEO的规则来定义具体页面的schema.

### Model

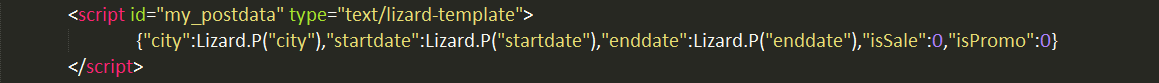
Model的声明包含两部分，apis和filter处理函数。



* apis是一个数组，每一项元素为一个数据请求，包含URL和POST Data.

1. post data中如果需要用到QueryString的数据，使用Lizard.P，如Lizard.P(“city”)声明
2. post data的内容也可以声明在页面中type为text/lizard-template的script标签内。

配置项中的postdata设置为Lizard.T(“my\_postdata”);即可



#### Lizard.T(id)

Usage: 根据ID获取页面特殊script元素

Input: id

Output: 此元素内容经过解析后的值

Example:

Lizard.T(“my\_postdata“)

返回为

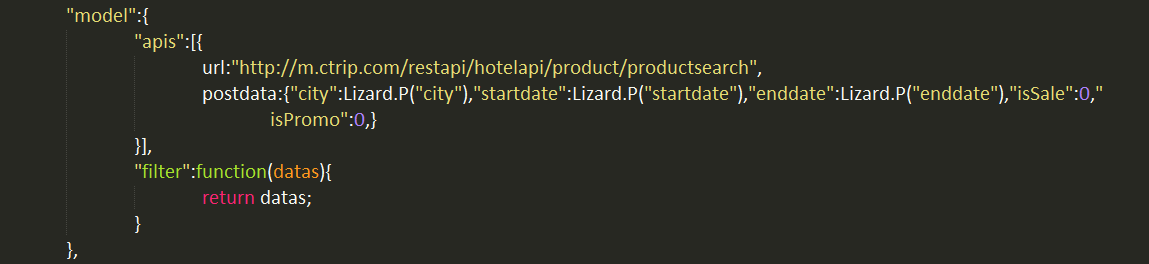
{"city":”shanghai”,"startdate":”20140501”,"enddate":”20140502”,"isSale":0,"isPromo":0}

* Filter是一个函数，函数对数据进行过滤和解析。

输入为一个数组，即apis中的每一项数据请求返回。

输出默认为原始的数据（即输入）。开发人员可根据业务逻辑修改Data，返回需要的应用给template数据.

代码示例：



"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;

}

},

### View

View包含header和viewport的两份Html模板。

header模板放置在header中，type为text/lizard-template的script标签内;

viewport模板放置在main-viewport中，type为text/lizard-template的script标签内

只需要将原本的html模板复制到对应的位置即可



代码示例：



"view":{

"header":Lizard.T("my\_header\_template"),

"viewport":Lizard.T("my\_body\_template")

},

Lizard.T参照上述说明

### Controller

只需要将原本的JS声明放到配置项中

代码示例：



"controller":"./res/views/vacationslist.js"

# Lizard2.0 Demo

请点击<http://conf.ctripcorp.com/display/FRAM/Lizard+2.0+Demo> 查看DEMO