**q.js学习文档**

# 相关网址

<https://github.com/itorr/q.js>

# 简介

Q.js 是一个炒鸡轻量的前端单页路由框架。

<http://mouto.org/#!54092>

轻量、快速、极简

　为了更好的利用缓存以及更少的后端支援，Q.js放弃了 HTML5 State，通过#!格式的 url hash 重现了 url 路由功能。

1. 无 JavaScript 库依托，可随意搭配使用
2. 源代码不及百行压缩后 834byte
3. 支持 IE6+ Chrome Safari FF
4. 未做情况判定，使用 Q.js 必然会注册 window.Q

# 万物之始

我们先来一段简单的 Hallo World

<!DOCTYPE html>

<meta charset="UTF-8">

<div id="m"></div>

<script src="q.js"></script>

<script>

Q.reg('home',function(){

document.getElementById('m').innerHTML='Hallo World';

});

Q.init({

index:'home'/\* 首页地址 \*/

});

</script>

打开例子后，浏览器会从 http://simple.com/ 跳转到 http://simple.com/#!home ，并且在页面显示 Hallo World。

如此，您的第一个通过 Q.js 实现的 URL 路由就工作了~ 接着我们再详细说明下，如何让 Q.js 工作的。

# 注册URL

在 Q.js 中，提供 关键字 和 正则表达式 两种注册 URL 方式。 关键字注册 URL 在 Q.js 中效率最高、但需要按照关键字模式规划 URL，在前者不能满足需求情况下可以使用正则注册法~

使用正则表达式注册URL可能会不同程度的增加运行时间（极不明显），在允许的情况下请优先使用关键字注册 URL~

接下来我们通过规划一个 blog 的URL，来学习如何使用 Q.js 注册URL。

## 首页

**我们先从首页开始~**

在注册之前我们分析一下注册首页 URL 都有哪些需求：

1. 首先首页有一个固定格式
2. 在大部分情况下不需要在 URL 中体现参数传递
3. 是打开域名时的默认展示页

**首先我们通过注册一个固定页面 home ，回调不期待传值**

Q.reg('home',function(){

console.log('打开了首页');

/\* JavaScript 代码 \*/

});

通过以上代码 访问 #!home 时，就会触发后面的回调。

在回调中我们修改页面 DOM 实现页面无刷新变换内容。 （这部分不在本篇文档所述范畴）

按照这样的方法，同理 我们也可以实现 类似 关于 #!about、友情链接 #!friend 等等页面的注册。

**实现了首页的 URL 注册之后，我们要在打开网页时默认跳转到 home 页。**

Q.js 的启动，由 Q.init 函数实现。

Q.init({

index:'home'/\* 首页地址 如果访问到不能访问页面也会跳回此页 \*/

});

在 init 的启动参数里面附带 index 参数，这样我们访问页面时就会默认打开 #!home

如果觉得 #!home 中的 ! 不符合预期，这…也是可以改的~

Q.init({

key:'^\_^', /\* url里#和url名之间的分割符号 默认为感叹号 \*/

index:'home' /\* 首页地址 不可访问路径也会跳回此地址 \*/

});

可以通过任意符合 URI 规范的字符串分割，甚至也可以设置为空。

设置为空时，Google等现代搜索引擎可能无法按照路由地址实现索引。请谨慎设置

## 文章页和翻页-带参数

文章页和首页不太一样、只有URL的一部分是相同的，需要通过URL中的不同部分区分不同文章页，不是默认打开页。

1. 文章页有一个固定的格式
2. 需要在URL中体现文章唯一ID

Q.js 关键字模式时，参数之间通过 / 分割，请根据顺序进行 URL 规划

Q.reg('article',function(aid){

if(!aid)

return alert('传入参数不正确，请确认您访问的地址。');

console.log('打开了文章，文章唯一ID为:'+aid);

/\* JavaScript 代码 \*/

});

注册之后访问页面 #!article/123 即可触发 article 页面回调，并传入 参数1 "123"

Q.js 不判断传入数据类型，请务必在回调中确认数据格式

翻页可文章页传值方式基本一致，在这里不重述

## 分类页-带多个参数

分类页其实和文章页很相似、只是参数一变成了分类名称，内容较多时需要传递下当前页码

1. 分类页有一个固定的格式
2. 需要在URL中体现分类唯一名称
3. 可能会有页码信息

依据上面的要求，我们依旧使用关键字注册模式，注册一个名为 category 的 url 预设两个参数 分类唯一名称 cstr ，页码 page

Q.reg('category',function(cstr,page){

if(!cid)

return alert('传入参数不正确，请确认您访问的地址。');

if(!page)/\* 如果 \*/

page=1;

console.log('打开了分类页面，分类名称是: '+cstr);

console.log('当前页面：'+page);

/\* 下面这段伪代码依托iTorr.js 仅作参考 \*/

/\* 根据分类唯一名称和页码发起 AJAX 请求 \*/

$.x('/api/category/'+cstr+'/page/'+page,function(d){

console.log(d);

/\* 在这里修改页面DOM \*/

});

});

访问 #!category/photo 会传入 photo 到 category 的回调函数 访问 #!category/photo/2 会传入 photo 和 2 到 category的回调函数

参数二不存在时，需要在回调函数内进行处理~

## 导航条

有了这些 URL 之后，我们还需要一个导航条来提供页面之间的入口， 导航条需要在每次页面变更时修改导航条样式，指示当前打开的页面。

我们的 HTML 一般是这样的

<nav>

<a href="#!home" class="active">首页</a>

<a href="#!about">关于</a>

<a href="#!friend">友情链接</a>

</nav>

CSS 一般是这样的

nav{line-height:2em;}

nav a{display:inline-block;vertical-align:top;padding:0 .5em;}

nav a.active{background:#369;color:#FFF;}

这时候我们需要在每次 URL 变更时，通过 JavaScript 动态修改 nav 中的 active class 的位置，以修改样式。

/\* 这段伪代码依托iTorr.js 仅作参考 \*/

var navchange=function(L){/\* 每次有url变更时都会触发pop回调 \*/

/\* L 为当前回调函数名称（目前仅支持关键字回调情况） \*/

var a;//临时变量

if(a=$('nav a.active')) //如果存在这个DOM

a.className=''; //修改它的ClassName

if(a=$('nav a[href="#!'+L+'"]')) //如果存在这个DOM

a.className='active';//修改它的ClassName

};

Q.js 提供了 Q.pop 事件，可以通过注册 Q.pop 实现在每次 URL 变更时发生回调， Q.pop 事件和其他 URL 事件不发生冲突，并在其他 URL 回调函数之前运行，请注意触发时机。

注册和修改 Q.pop 有多种途径。可以放在启动函数中注册，比如：

Q.init({

/\* ...各种其他参数... \*/

pop:navchange //注册 pop 函数

});

也可以通过 关键字注册

Q.reg('pop',navchange);

甚至可以直接修改 Q.pop

Q.pop=navchange

请注意，使用 Q.js 请避免使用 Q.js 保留关键字为 关键字URL 地址 保留关键字包括：reg,pop,go,V

到这里，我们的简单的博客差不多就能用啦~

还剩下一个正则注册 URL 没有说，正则注册在关键字判定前进行判定，不论注册前后，请注意判定顺序。所以也因为这样，我们关键字判定中的关键字部分可以用正则表达式注册法替代。

但是如果我们依旧想用保留关键字，用正则注册法是这样实现的

Q.reg(/pop/,function(){

console.log('pop事件');

});

正则注册法需要在正则里指定可能传入的值

以上我们的博客URL规划基本就已经完成啦~

# 如何实现更加丰富的 url 格式

关键字注册法虽然方便，但仅仅支持 关键字/参数1/参数2/参数3 这种格式，想要实现更多不科学的 URL 格式，可以通过正则注册法~ 比如视频网站的链接，由v开头后面跟数字ID的情况，预期必然会有参数一 并且为数字

Q.reg(/v(\d+)/,function(vid){

console.log('pop事件 参数一：'+vid);

});

这样访问 http://simple.com/#!v1234567 就会传递值 1234567 给回调函数。

# 我想在 JavaScript 里知道当前在哪个页面如何实现？

访问 Q.lash 变量即可返回当前页面关键字

Q.lash 目前仅可返回关键字注册的 URL，预计会在下次版本更新时增加正则支持。

# 如何通过JavaScript跳转到特定链接

因为在 Q.js 中！关键字是可以被修改的，如果通过JavaScript实现URL跳转，可以调用 q.go 函数

Q.go('category/code');

当然，大部分情况我们推荐使用 HTML 原生的 a 标签 link 通过访客触发，进行跳转

<a href="#!category/code">代码分类</a>

以上。

# 更新历史

## 2015-3-29 增加了数组绑定模式

感谢@youye 提交更新!

Q.reg([

['地址1',function(){

//回调函数1

}],

['地址2',function(){

//回调函数2

}]

]);

Q.reg(['地址3','地址4']);

Q.reg(['猫娘1','猫娘2','猫娘3'],function(){

M.innerHTML='喵喵喵';

});