02

OPEN ORIENTED

凹 凸 实 验 室

前端路由实现探索

张燕婷

2017.07.10

目录

02

背景

实现核心

实现方案

方案对比

背景

单页应用

- 用户访问一个网址，服务器返回的页面始终只有一个
- 浏览器地址栏内容改变或者页面内发生跳转，服务器不会重新返回新的页面，而是通过相应的js操作来实现页面的更改

背景

前端路由

- 根据不同的URL渲染不同的视图页面的实现手段

[作用]

1. 减轻服务器端的压力：为每一个不同的 url 执行一段处理逻辑，且可能存在高并发的情况

2. 提升用户体验：页面的切换不需要刷新整个页面，没有网络延迟，没有闪烁刷新

实现核心

02

实现前端路由，需要解决的问题有哪些?

在页面不刷新的前提下实现 url 变化

捕捉到 url 的变化，以便执行页面替换逻辑

支持浏览器（历史记录）前进与后退

实现方案

02

hash(#)

location.hash + hashchange 事件

history

history.pushState() + popstate 事件

实现方案

02

hash(#)

- # 本身以及它后面的字符称之为 hash
- 可通过 window.location.hash 属性读取

实现方案

hash(#)

02

[特点]

1. 不会被包括在 HTTP 请求中：用来指导浏览器动作的，对服务器端完全无用

在页面不刷新的前提下实现 url 变化

实现方案

hash(#)

02

[特点]

2. 可监听 hash 改变：window.addEventListener("hashchange", funcRef, false)

捕捉到 url 的变化，以便执行页面替换逻辑

实现方案

hash(#)

02

如何触发 hashchange?

- 在当前 url 的后面增加 hash：在 a 标签的 href 属性中设置，如
- 直接在 js 中对给 location.hash 赋值即可：如 location.hash = '#/index'

实现方案

hash(#)

02

[特点]

3. 记录访问历史：每一次改变 hash（window.location.hash），都会在浏览器的访问历史中增加一个记录

支持浏览器（历史记录）前进与后退

实现方案

hash(#)


```
var hash = window.location.hash
var path = hash.substring(1)

switch (path) {
  case '/':
    showHome()
    break
  case '/users':
    showUsersList()
    break
  default:
    show404NotFound()
}
```

实现方案

hash(#)


点击
相应链接

触发
hashchange
前进/后退
触发
hashchange
根据路由配置
更新dom
刷新
页面
服务端
只返回
入口页面

实现方案

02

history

- History 对象包含用户（在浏览器窗口中）访问过的 URL，即浏览器历史记录

- HTML5 新增的 history API： history.pushState() 和 history.replaceState()，可操作浏览器历史记录

实现方案

history

02

history.pushState 方法

在浏览记录（history对象）中添加一个新记录

- state：与指定网址相关的状态对象，popstate 事件触发时，该对象会传入回调函数
- title：新页面的标题，但是所有浏览器目前都忽略这个值
- url：新的网址，必须与当前页面处在同一个域，浏览器的地址栏将显示这个网址

实现方案

history

02

history.pushState 方法

在浏览记录（history对象）中添加一个新记录

[特点]

不会触发页面刷新，只是导致 history 对象发生变化，地址栏会有变化

在页面不刷新的前提下实现 url 变化

支持浏览器（历史记录）前进与后退

实现方案

history

02

history.popstate 事件

- 每当同一个文档的浏览历史（即 history 对象）出现变化时，就会触发 popstate 事件

[特点]

- 仅仅调用 pushState 方法或 replaceState 方法 ， 不会触发该事件
- 只有用户点击浏览器倒退按钮和前进按钮，或者使用JavaScript调用 back、forward、go 方法时才会触发

实现方案

history

02

history.popstate 事件

- 每当同一个文档的浏览历史（即 history 对象）出现变化时，就会触发 popstate 事件

[特点]

可以为 popstate 事件指定回调函数。这个回调函数的参数是一个 event 事件对象，它的 state 属性指向pushState 和 replaceState 方法为当前 URL 所提供的状态对象（即这两个方法的第一个参数）

捕捉到 url 的变化，以便执行页面替换逻辑

实现方案

history

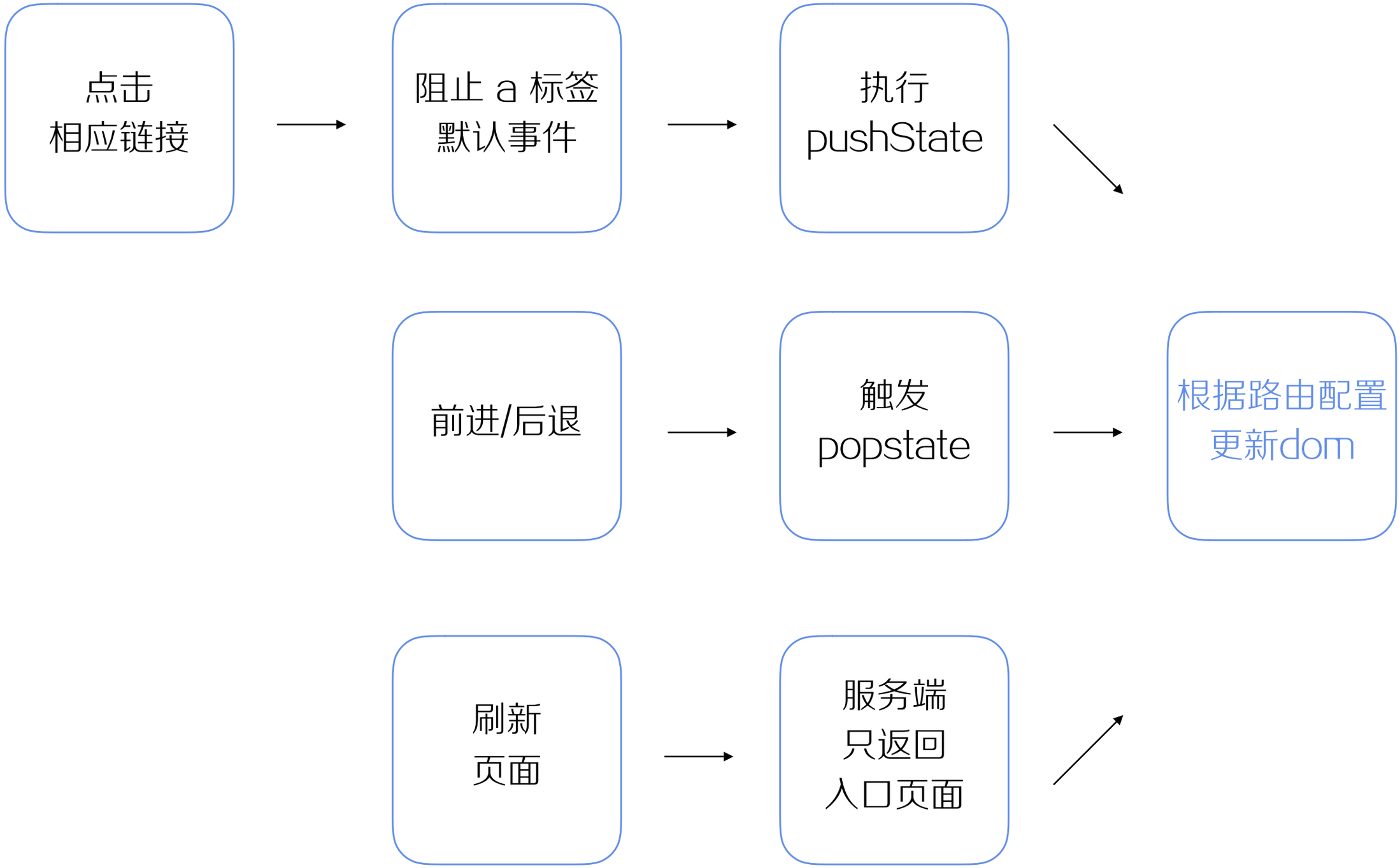

```
var path = window.location.pathname

switch (path) {
  case '/':
    showHome()
    break
  case '/users':
    showUsersList()
    break
  default:
    show404NotFound()
}
```

实现方案

history

02



方案对比

02

hash(#)

- 兼容性好
- 需使用#符号
- 只可修改#后面的部分，故只可设置与当前同文档的URL
- 服务器端无需进行处理

history

- ie 需要额外的兼容
- 更加直观和正式
- pushState设置的新URL可以是与当前URL同源的任意URL
- 纯相对路径需要在服务器端进行处理

参考文档

《【源码拾遗】从vue-router看前端路由的两种实现》

《理解Web路由》

《JS原生一步步实现前端路由和单页面应用》

《前端路由的实现方式》

《前端路由一探》

《前端路由实现 - history篇》

《javascript 标准参考教程（alpha）- history对象》