

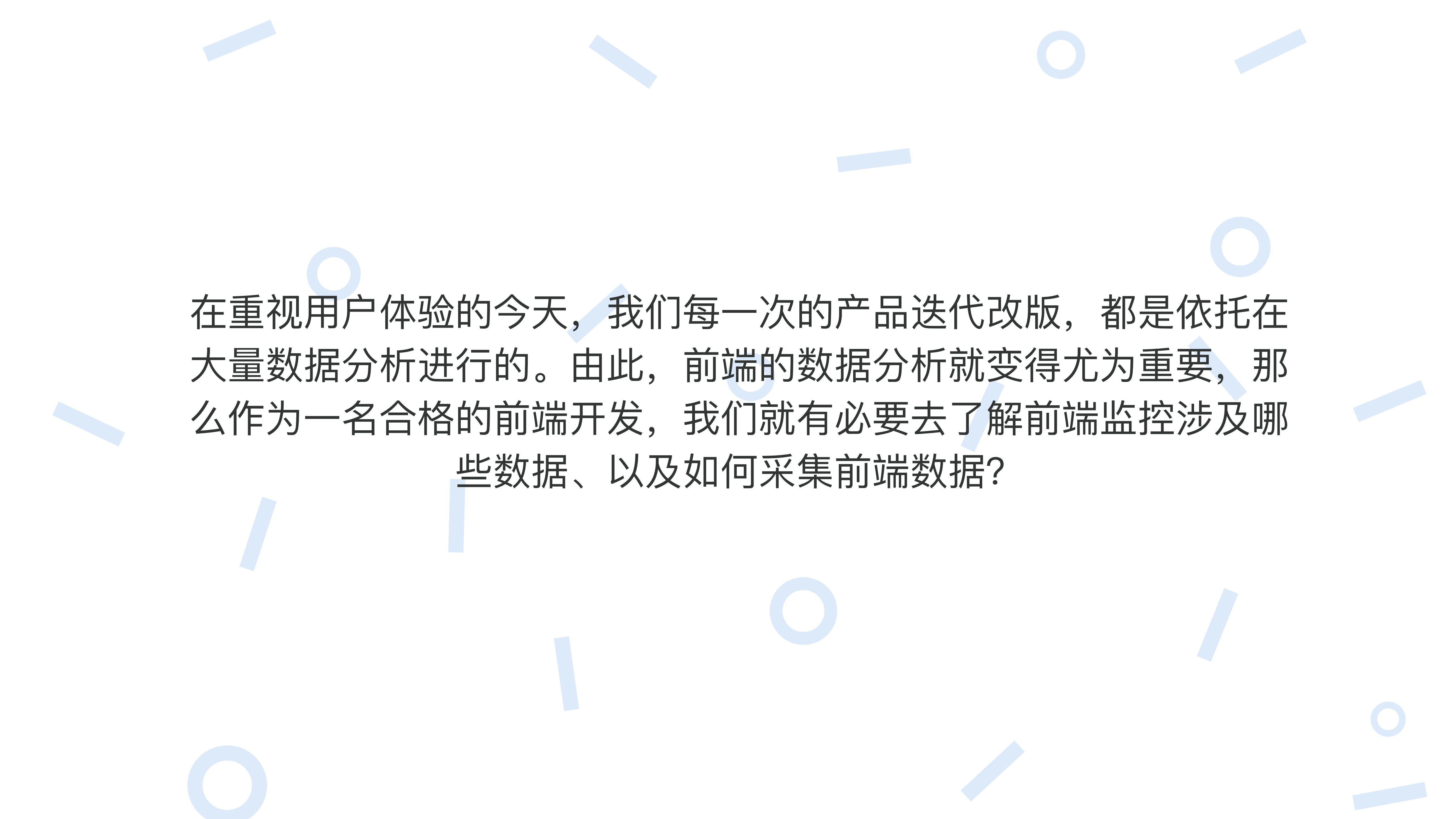
02

OPEN ORIENTED

凹凸实验室

不可不知的前端数据监控

高露



在重视用户体验的今天，我们每一次的产品迭代改版，都是依托在大量数据分析进行的。由此，前端的数据分析就变得尤为重要，那么作为一名合格的前端开发，我们就有必要去了解前端监控涉及哪些数据、以及如何采集前端数据？

一、采集前端的数据

访问类数据

PV/UV
操作系统
浏览器
分辨率
登录率
地域分布
网络类型
访问时段
停留时长
到达深度

点击类数据

页面总点击量
人均点击量
流出 url
点击时间
首次点击时间
点击热力图

性能类数据

白屏时间
首屏时间
用户可操作时间
页面总下载时间
自定义的时间点

异常类数据

异常的提示信息
JS 文件名
异常所在行
发生异常的浏览器
堆栈信息

访问数据类统计PV/UV

200

深圳 : 41



点击类数据

• 页面总点击量

• 人

```
window.addEventListener("click", function(e){
  var target, ptag;
  var elem = e.target || e.srcElement;
  var root = document.body;

  while(!elem == root){
    if(elem.getAttribute && elem.getAttribute('ptag')){
      target = el;
      ptag = elem.getAttribute('ptag');
      break;
    } else {
      elem = elem.parentNode || root;
    }
  }

  report(url, data)
}, false)
```

首次点击时间

• 点击热力图

性能类数据

指标： •白屏时间 •首屏时间 •流出 url •点击时间 •首次点击时间

方式： •前端埋点监控 •Performance API 监测页面性能

CSS加载完毕

```
<script>var cssLoadStart = +new Date</script>  
<link rel="stylesheet" href="xxx1.css" type="text/css" media="all">  
<script>var cssLoadTime = (+new Date) - cssLoadStart;var jsLoadStart = +new Date;</script>
```

JS加载完成

```
<script>var jsLoadStart = +new Date;</script>  
<script type="text/javascript" src="xx3.js"></script>  
<script>var jsLoadTime = (+new Date) - jsLoadStart;</script>
```

首屏图片加载完成

```
window.imageLoaded = (function() {  
    var i = 0  
    return function() {  
        var total = document.querySelectorAll('[onload="imageLoaded()"]').length  
        if (++i == total) _PFM_TIMING[5] = new Date()  
    } })();
```

主接口加载完成

HTML加载完成

```
window.addEventListener("load", function(){  
    var htmlLoadTime = new Date() - startTime  
})
```

参考: <http://wqadmin.jd.com/athena/fd/index.php/point/>

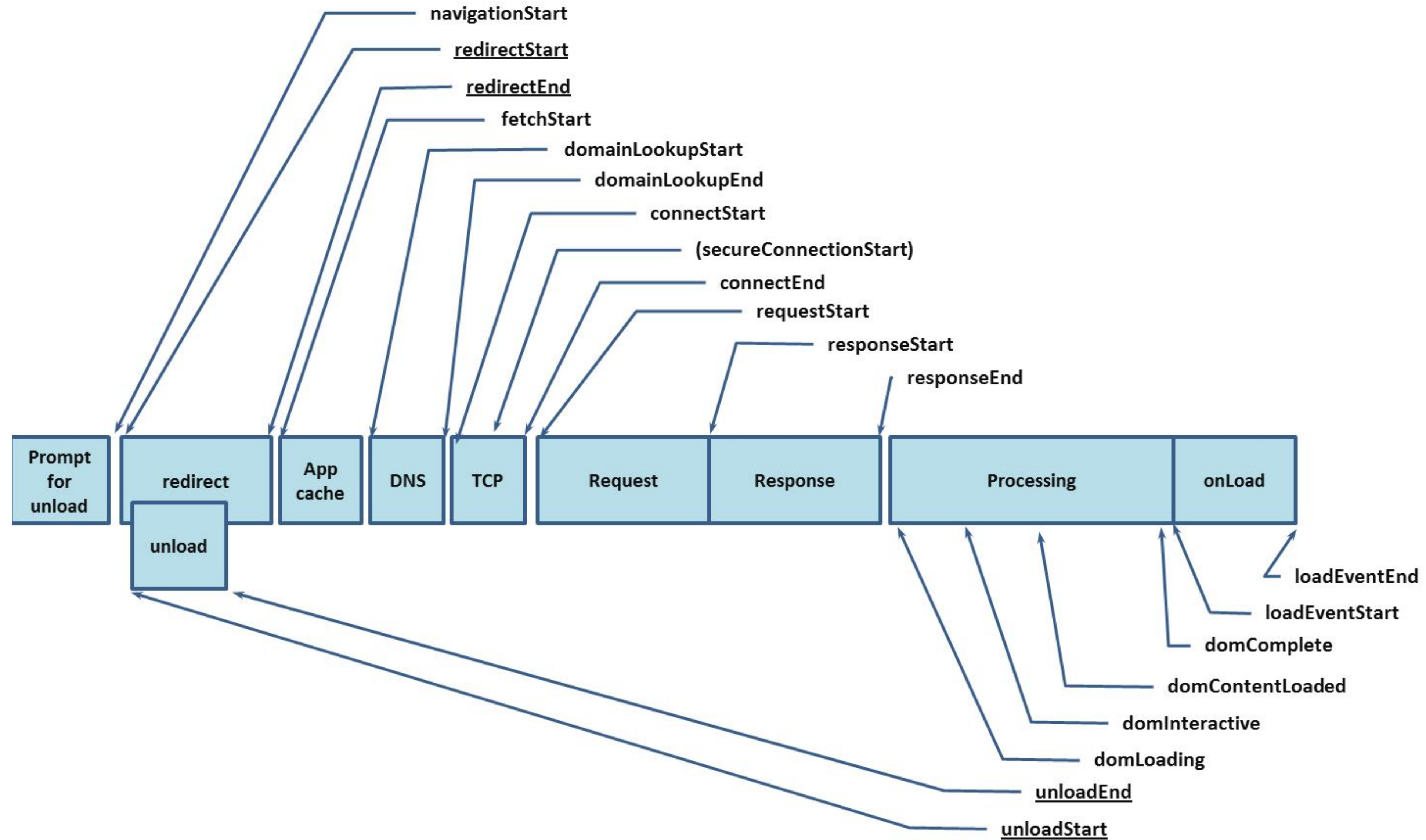
Navigation Timing API

主文档加载速度时间线

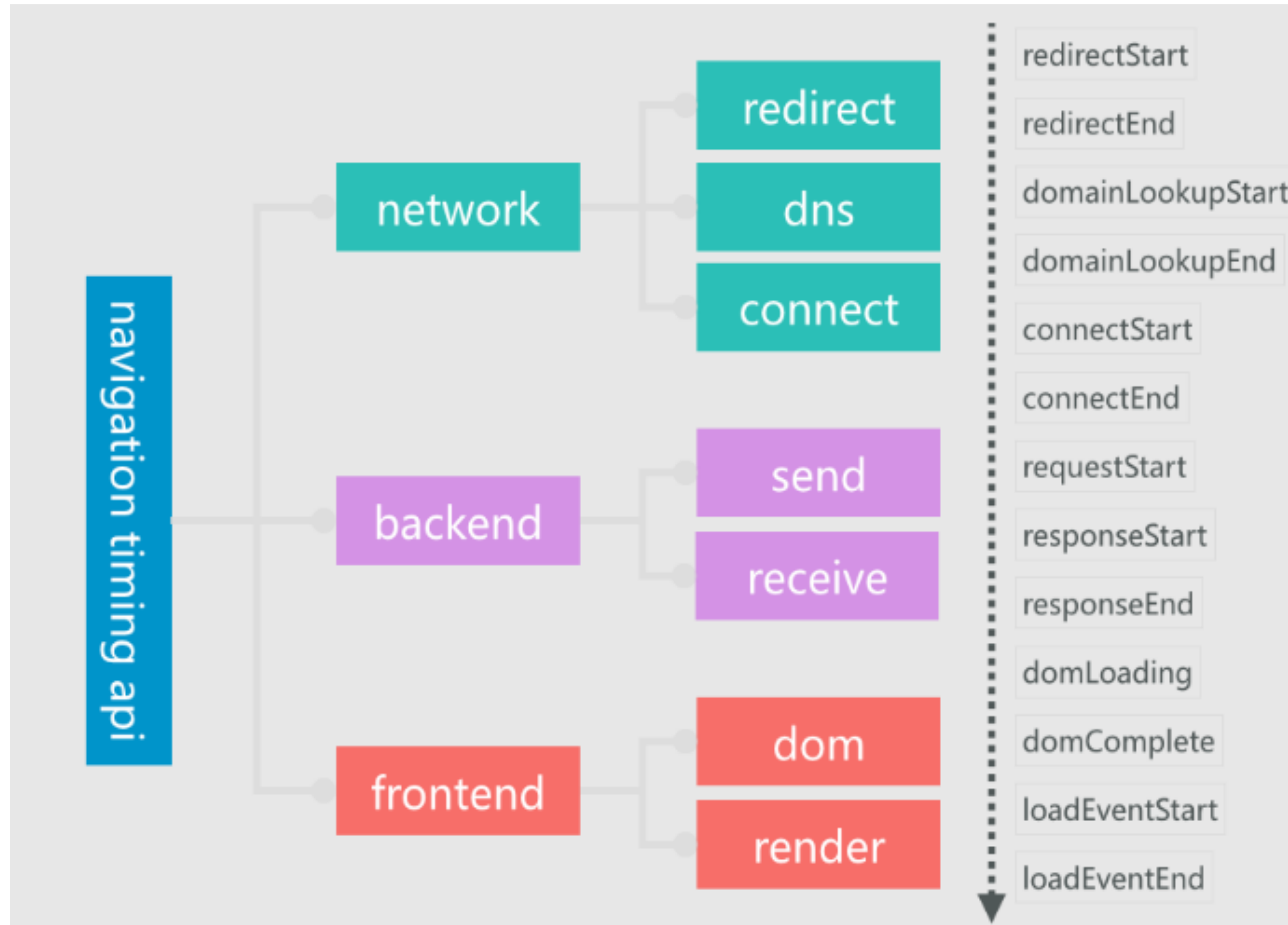
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser
		42	48	4	34	3.2											
		43	49	5	35	4.1											
		44	50	5.1	36	4.3		2.1									
		45	51	6	37	5.1		2.2									
		46	52	6.1	38	6.1		2.3									
		47	53	7	39	7.1		3									
6		48	54	7.1	40	8		4									
7		49	55	8	41	8.4		4.1									
8	12	50	56	9	42	9.2		4.3									
9	13	51	57	9.1	43	9.3		4.4		12							
10	14	52	58	10	44	10.2		4.4.4	7	12.1			10		4		
11	15	53	59	10.1	45	10.3	all	56	10	37	59	53	11	11.4	5	1.2	7.12

兼容性 Chrome 6.0及以上、Firefox 7.0及以上、InternetExplorer 9.0及以上、Opera 15.0及以上， Safari (WebKit) 8.0及以上

Navigation Timing API



Navigation Timing API



异常类数据

指标： •异常的提示信息 •JS 文件名 •异常所在行 •发生异常的浏览器 •堆栈信息

方式： •window.onerror捕获 •try..catch

window.onerror

When a **JavaScript runtime error** (including syntax errors and exceptions thrown within handlers) occurs, an **error** event using interface **ErrorEvent** is fired at **window** and **window.onerror()** is invoked (as well as handlers attached by **window.addEventListener** (not only capturing)).

```
window.onerror = function(messageOrEvent, source, lineno, colno, error) { ... }
```

监控数据： 错误信息 msg 出错文件url 行号row 列号col 错误详细信息 error

同源策略 & 'Script error.'

跨域问题。现在的站点，静态文件大多都是放在一个独立的域名下面。既可以减少浏览器并发的域名限制，又能通过 CDN 提高资源的访问速度。默认情况下，在本域名下捕获到一个跨域脚本的错误信息时，只能获取到一条信息 **Script error.**，没有文件信息，没有行列号数据，更没有详细的错误对象。

```
▼ error: Object  
  column: null  
  file: null  
  line: null  
  message: "Script error."  
  stack: null
```

设置请求header

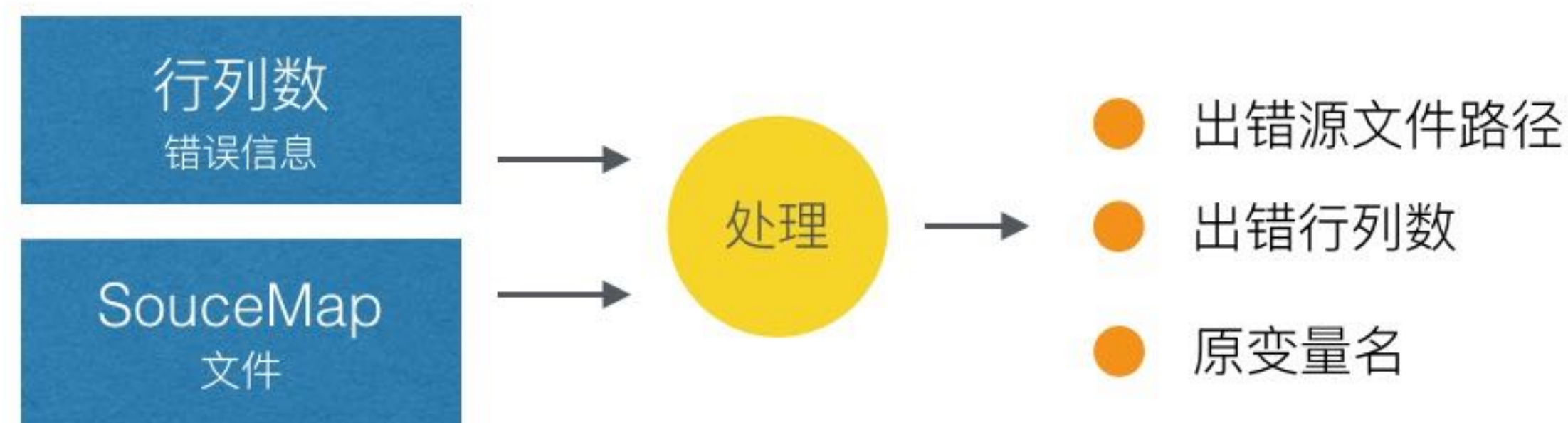
服务器端增加 Access-Control-Allow-Origin
客户端引用脚本增加 crossorigin="anonymous"

(uglifyjs + combo) vs sourcemap

目前大多数站点的静态脚本文件，上线时都要压缩混淆的。所以发生错误时，获取到的行号就是第 1 行，列号会是一个巨大无比的数。这时你只能依赖错误信息和文件路径来定位错误。

SourceMap 是一个信息文件，存储着源文件的信息及源文件与处理后文件的映射关系。

在定位压缩代码的报错时，可以通过错误信息的行列数与对应的 SourceMap 文件，处理后得到源文件的具体错误信息。



数据上报

方式： • query string •Ajax

优缺点： •img方式效果效率更高，可通过src跨域上报 •URL字符长度限制，则不适合大量数据上报

•Ajax适合大量数据上报，若存在跨域上报，在xhr2.0下需设置请求Header，允许跨域

Image 数据上报效率对比

new Image vs. createElement('img')

- [GlobalEventHandlers.onerror](#)
- [Performance API](#)
- [new Image vs. createElement\('img'\)](#)
- [前端数据之美](#)
- [域名劫持资源重加载方案](#)
- [CSS文件动态加载](#)
- [捕获页面中全局Javascript异常](#)