

性能数据收集工具分析

- chrome dev Tools

我们经常使用 Chrome Dev Tools 来开发调试，还可以利用它来分析页面性能

- Lighthouse: 一个开源的自动化工具，可以安装为 Chrome 的扩展插件，也可以命令行直接运行，它将针对目标页面运行一连串测试，然后输出一个有关页面性能的评分报告。

优点：

1. 评分报告全面且具有一定的权威性
2. 提供解决方案
3. 发现大的性能问题

缺点：

1. 测试环境较单一，用户群体的环境各有不同，不可以一概而论
2. 存在「波动」，不同时刻的访客群体存在差异，数据只能反应当前时刻的「效果」

- 第三方收费方案

- NewRelic
- 阿里云 ARMS

一个字：贵！

- 自研性能监控系统：在目标页面注入脚本，在约定的时机收集性能指标数据，统一上报数据中心，数据中心集中整合生成报表，再根据报表分析性能。

优点：

1. 数据全面，可采集到所有用户各个环境下的性能，生成直观的分布图
2. 数据真实，来源于真实用户
3. 反馈及时，优化后效果可及时地在报表上反馈出来

缺点：

1. 依赖较多，数据中心和脚本都需要自主开发，相对成本较高

所谓监控，实际上就是性能「真实跟踪」，虽然依赖较多，但对性能指标的反馈最为真实有效。以下我们将围绕性能监控进行展开。

LightHouse

Lighthouse 是一个开源的自动化工具，用于改进网络应用的质量。只要为 Lighthouse 提供一个需要审查的网址，它将针对此页面运行一连串测试，然后生成一个有关页面性能的报告。

并对网站进行分析，将分析结果以可视化的方式展示给开发者，甚至还指出了可以优化的地方。

使用方式

- Chrome 开发者工具 (DevTools)
- Chrome 扩展
- **Node CLI 推荐:** <https://github.com/GoogleChrome/lighthouse#using-the-node-cli>
- Node Module

```
//安装
npm install -g lighthouse
//生成报告 并在浏览器中打开
lighthouse http://kaikeba.com --view

//报告格式是json，保存在当前目录
lighthouse http://kaikeba.com --output=json --output-path=./report.json

//设置浏览器的窗口尺寸
//lighthouse http://kaikeba.com --chrome-flags="--window-size=1190,660"

//模拟器设为桌面
//lighthouse http://kaikeba.com --emulated-form-factor=desktop
```

性能报告

- Performance 性能分析
- Accessibility 可访问性
- Best Practices 最佳实践
- SEO 搜索引擎优化
- Progressive Web App 是否集成pwa

准确衡量网站的性能

根据 Google 在 web.dev 上公布的数据，他们认为以用户为中心的性能指标，应该能回答以下四个问题 [3](#):

web.dev 是 Google Developer 提供的开发者社区，里面主要提到了一下列出的诸多类型的数据指标。

1. **是否发生？** 导航是否成功启动？服务器是否有响应？
2. **是否有用？** 是否已渲染可以与用户互动的足够内容？
3. **是否可用？** 用户可以与页面交互，还是页面仍在忙于加载？
4. **是否令人愉快？** 交互是否顺畅而自然，没有滞后和卡顿

是否发生

当用户访问一个网站的时候，关心的第一个问题永远是“是否发生”——浏览器是否成功地把我的请求发送出去，而服务器是否已经知道并开始处理我的请求。

TTFB (Time to First Byte)

首字节到达的时间点。

FP (First Paint)

首次绘制，标记浏览器渲染任何在视觉上不同于导航前屏幕内容的时间点。

FCP

First Contentful Paint

首次内容绘制，标记浏览器渲染来自 DOM 第一位内容的时间点，内容可能是文本、图像等元素。

测试加载速度的体验，页面最主要的内容何时呈现。

是否有用

当用户确定自己的请求发生了后，就会开始关心第二个问题：“是否有用？”

例如，用户在使用天气应用，在确定页面有反应了后，就开始关心，什么时候能展现有用的内容，从而得知今天的天气。

LCP

Largest Contentful Paint

最大内容绘制时间，计算从页面开始加载到用户与页面发生交互（点击，滚动）这段时间内，最大元素绘制的时间，该时间会随着页面渲染变化而变化，因为页面中的最大元素在渲染过程中可能会发生改变。

SI

Speed Index

速度指标，填充页面内容的速度，取开始加载到最后完成渲染，每一时刻页面未完成度的积分。

是否可用

在用户得到了有用的信息后，用户就会基于得到的信息作出反应，这就是页面“是否可用？”例如看到了新闻后，想要评论。TTI、FID、TBT 就是回答这些问题的指标。

TTI

Time to Interactive

可交互时间，用于标记页面已进行视觉渲染并能可靠响应用户输入的时间点。页面可能会因为多种原因而无法响应用户输入，例如页面组件运行所需的 Javascript 尚未加载，或者耗时较长的任务阻塞主线程。TTI 指标可识别页面初始 JavaScript 已加载且主线程处于空闲状态（没有耗时较长的任务）的时间点。

TBT

Total Blocking Time

总共阻塞时间，计算的是从 FCP 到 TTI 之间，主线程阻塞的总时间。阻塞时间是指单次任务占用主线程超过 50 ms 的部分。

FID (First Input Delay)

首次输入延迟，指用户首次输入到页面响应的时间。我们都知道第一印象的重要性，网站亦是如此。首次输入延迟会成为用户对网站很重要的第一印象，决定用户有可能成为忠实用户或者弃之而去。值得注意的是，FID 仅关注用户离散的操作，如点击，轻击，按键等，其他交互如滚动和缩放，并不是 FID 关注的，因为通常浏览器会用一个单独的线程来处理它们。

是否令人愉快

使用是否流畅，有没有发生预料外的视觉偏移。

CLS

Cumulative Layout Shift

累计布局偏移。测量在页面的整个生命周期中发生的每个意外的样式移动所造成的布局偏移分数的总和。

测试视觉稳定性上的体验，有多少内容发生了意外的偏移。