

# SSR、CSR、SSG

---

SSR、CSR、SSG 这些都是关于前端渲染方式的技术，它们决定了你的网页内容是在哪里生成的。

## SSR渲染

SSR 的全称是 Server-Side Rendering，即**服务器端渲染**。

SSR 想象成这样：当你访问一个网站时，**你的浏览器不是直接拿到一个空的 HTML 文件，而是向服务器发送请求。服务器在收到请求后，会先把页面所有需要的数据（比如文章列表、商品信息等）都准备好，然后把这些数据和 HTML 模板结合，生成一个完整的、包含了内容的 HTML 文件，最后把这个 HTML 文件发送给你的浏览器。**

### 流程：

1. **浏览器向服务器发送请求。**
2. **服务器查询数据，生成完整的 HTML。**
3. **浏览器收到完整的 HTML，直接显示页面内容。**
4. 等待 JavaScript **文件下载和执行后，页面变为可交互状态。**

### 优点：

- **对 SEO 友好：**搜索引擎爬虫可以直接抓取到完整的 HTML 内容，非常利于网站排名。
- **首屏加载速度快：**用户能更快地看到页面内容，因为浏览器收到的就是完整的页面。

### 缺点：

- **服务器压力大：**每次用户请求一个页面，**服务器都要进行渲染，在高并发场景下会消耗较多资源。**
- **用户体验可能不佳：**在 JavaScript **下载和执行完成之前，页面是静态的，用户无法进行点击等交互操作。**

# CSR渲染

## CSR (Client-Side Rendering) - 客户端渲染

CSR 是最常见的单页应用 (SPA) 渲染方式, 比如我们用 React、Vue 或 Angular 开发的应用。

你可以把 CSR 想象成这样: 你的浏览器访问网站时, 服务器只给你一个空的 HTML 文件, 里面只有一个根 `div` 元素和一个 `script` 标签。然后, 浏览器开始下载 JavaScript 文件, 当 JS 文件下载并执行后, 它会负责获取数据、动态生成所有 HTML 内容, 并把内容填充到那个空的 `div` 中。

### 流程:

1. 浏览器向服务器发送请求, 收到一个空的 HTML 文件。
2. 浏览器下载并执行 JavaScript 文件。
3. JavaScript 向服务器请求数据 (通常通过 API) 。
4. JavaScript 拿到数据, 生成 HTML 内容并渲染到页面上。

### 优点:

- 服务器压力小: 服务器只需提供静态文件, 不需要每次都渲染页面。
- 交互体验好: 一旦 JS 加载完成, 页面切换和交互会非常流畅, 因为不需要重新加载整个页面。

### 缺点:

- 对 SEO 不友好: 搜索引擎爬虫一开始看到的是一个几乎空白的 HTML 文件, 不利于内容抓取。
- 首屏加载慢: 用户需要等待 JS 和数据都加载完成后才能看到页面内容, 可能出现白屏。

# SSG渲染

## SSG (Static Site Generation) - 静态站点生成

SSG 是指在项目构建时就将所有页面预先渲染成静态 HTML 文件。

你可以把 SSG 想象成这样：你在部署项目之前，就通过工具（如 Next.js、Gatsby）把网站的所有页面都生成为 .html 文件，并把这些文件存放在 CDN 上。当用户访问网站时，浏览器直接从 CDN 上获取预先生成好的 HTML 文件。

#### 流程：

1. **项目构建时**，把所有页面都生成为**完整的 HTML 文件**。
2. **浏览器**向服务器（通常是 CDN）请求页面。
3. **服务器**直接返回预先生成的 HTML 文件。

#### 优点：

- **速度极快**：页面内容是预先生成好的，**用户访问时直接返回**，几乎没有延迟。
- **服务器压力极低**：服务器不需要任何渲染操作，**只负责发送静态文件**。
- **对 SEO 友好**：同样，搜索引擎可以轻松抓取到完整的 HTML 内容。

#### 缺点：

- **不适合频繁变动的内容**：如果网站内容需要频繁更新（比如股票行情、在线聊天），SSG 就不太合适，因为每次更新都需要重新构建和部署。
- **页面数量有限制**：对于页面数量非常庞大的网站（比如电商网站的所有商品页面），**全部静态生成会非常耗时**。

## 总结

SSR：服务器“实时”渲染，适合**内容变化较快**且对**SEO 要求高**的网站，比如博客、新闻网站。

CSR：浏览器“实时”渲染，适合**交互性强**且对**首屏速度和 SEO 要求不高**的应用，比如后台管理系统、个人工具类应用。

SSG：构建时“预先”渲染，适合**内容基本固定**且**追求极致性能和 SEO**的网站，比如**公司官网、技术文档、个人博客**。