Atitit pwa 技术总结

目录

[1.1.1. Step 1：开启HTTPS 1](#_Toc5783)

[2. PWA安装 1](#_Toc27434)

[3. 实现过程 2](#_Toc7516)

[4. Install active fetch事件 2](#_Toc11959)

[4.1. 更新你的service worker 2](#_Toc7104)

[5. Chrome setting 2](#_Toc26243)

[5.1. 多页应用 PWA 改造实践 3](#_Toc12562)

[6. Prblm 3](#_Toc28558)

[6.1. Dbg tol chrome,,, 3](#_Toc15760)

[6.2. An SSL certificate error occurred when fetching the script. 4](#_Toc19285)

[6.3. 未捕获（承诺）类型错误：无法在“addAll”上执行“缓存”：渐进式 Web 应用程序上的请求失败 4](#_Toc15897)

[6.4. Cant show install icon 4](#_Toc18248)

[6.5. PWA 专属问题 5](#_Toc9784)

### Step 1：开启HTTPS

显然，PWAs需要HTTPS连接。虽然成本会增加，但这些成本和代价是值得的，毕竟谷歌搜索会将安全的网站排名更高。

HTTPS对于上边的演示代码是非必需的，因为Chrome允许使用localhost或任何127.x.x.x地址进行测试。如果想在HTTP网站上测试PWA技术，在启动Chrome时可以使用下面的命令行标记：

·--user-data-dir

·--unsafety-treat-insecure-origin-as-secure

动提示用户添加到桌面 — 通过HTTPS提供（这是使用服务工作线程的一项要求）。 被访问至少两次(第一次访问执行安装和激活，第二次访问生效，提示安装到桌面)，这两 ...

浏览

# PWA安装

**安装PWA**

1. 在计算机上打开Chrome。
2. 转到您要**安装**的网站。
3. 在地址栏的右上角，点击“**安装**”图标 。
4. 按照屏幕上的说明**安装**该**PWA**。

地址栏会出现个安装图表

# **实现过程**

Chrome 将在您的应用符合以下条件时自动显示横幅（添加到桌面的提示）：

* 拥有一个网络应用清单文件（manifest.json），该文件****至少****具有：
  + 一个 short\_name（用于主屏幕）
  + 一个 name（用于横幅中）
  + 一个 192x192 png 图标（图标声明必须包含一个 mime 类型的 image/png）
  + 一个加载的 start\_url
* 拥有一个在您的网站上注册的****服务工作线程****（service worker）
* 通过HTTPS提供（这是使用服务工作线程的一项要求）。
* 被访问至少两次(第一次访问执行安装和激活，第二次访问生效，提示安装到桌面)，这两次访问至少间隔五分钟

# Install active fetch事件

1. install 事件会绑定在 Service Worker 文件中，在 Service Worker 安装成功后，install 事件被触发。**install**事件一般是被用来填充浏览器的离线缓存能力****。为了达成这个目的，使用了 cache API ，它使我们可以存储网络响应发来的资源，并且根据它们的请求来生成key。它会一直持久存在，直到你告诉它不再存储，你拥有全部的控制权。
2. 当 service worker 安装完成后，会接收到一个激活事件(activate event)。 **activate**主要用途是清理先前版本的service worker 脚本中使用的资源。****
3. fetch 事件在****您刷新网页的时候，会在您的缓存里面找，如果没有缓存才去做http请求。****

Service Worker注册成功之后就会触发install事件，在触发install事件后，我们就可以开始缓存一些静态资。waitUntil方法确保所有代码执行完毕后，Service Worker 才会完成Service Worker的安装。需要注意的是只有CACHE\_LIST中的资源全部安装成功后，才会完成安装，否则失败，进入redundant状态，所以这里的静态资源最好不要太多。如果 sw.js 文件的内容有改动，当访问网站页面时浏览器获取了新的文件，它会认为有更新，于是会安装新的文件并触发 install 事件。但是此时已经处于激活状态的旧的 Service Worker 还在运行，新的 Service Worker 完成安装后会进入 waiting 状态。直到所有已打开的页面都关闭，旧的 Service Worker 自动停止，新的 Service Worker 才会在接下来打开的页面里生效。为了能够让新的Service Worker及时生效，我们使用skipWaiting直接使Service Worker跳过等待时期，从而直接进入下一个阶段。

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

## **更新你的service worker**

# Chrome setting

关闭安全验证。。安全浏览》无防护

网站票据导入cert.pem.crt证书到受信任目录

Url左边，不安全，网站设置，不安全的内容，允许

## **多页应用 PWA 改造实践**

准备工作做完之后，就可以放心地开始写代码了。如果你的网站是单页架构，那么恭喜你，你只要：

* 用几个 SW 的库，如 [sw-precache-webpack-plugin](https://link.zhihu.com/?target=https://github.com/goldhand/sw-precache-webpack-plugin" \t "https://zhuanlan.zhihu.com/p/_blank)；
* 找个 [manifest.json](https://link.zhihu.com/?target=https://developer.mozilla.org/en-US/docs/Web/Manifest" \t "https://zhuanlan.zhihu.com/p/_blank) 抄一下，比如[我们的](https://link.zhihu.com/?target=https://h5.ele.me/manifest.json" \t "https://zhuanlan.zhihu.com/p/_blank)；（或者用这个超棒的 [App Manifest 生成器](https://link.zhihu.com/?target=https://app-manifest.firebaseapp.com/" \t "https://zhuanlan.zhihu.com/p/_blank)）
* 用 [lighthouse](https://link.zhihu.com/?target=https://developers.google.com/web/tools/lighthouse/" \t "https://zhuanlan.zhihu.com/p/_blank) 跑一下，按照提示改进改进；
* 对比一下谷歌的 [PWA Checklist](https://link.zhihu.com/?target=https://developers.google.com/web/progressive-web-apps/checklist" \t "https://zhuanlan.zhihu.com/p/_blank)，按照提示改进改进；
* Debug 一下微信/QQ 浏览器；
* Debug 一下 UC 浏览器；
* Debug 一下百度浏览器；
* Debug 一下 360 浏览器；
* Debug 一下猎豹浏览器；
* 找 10 台安卓机 Debug 一下自带的浏览器。

## 必须https或者测试用http local也可

****Make sure your site is served using HTTPS!**** Service worker functionality is only available on pages that are accessed via HTTPS. (http://localhost will also work, to facilitate testing.) The rationale for this restriction is outlined in the ["Prefer Secure Origins For Powerful New Features" document](http://www.chromium.org/Home/chromium-security/prefer-secure-origins-for-powerful-new-features).

## 批量生成缓存文件。。

sw-precache-webpack-plugin,使用 sw-precache 生成 service worker 的 Webpack 插件,缓存 webpack 的 bundles 发出的资产,下载sw-precache-webpack-plugin的源码\_GitHub\_酷徒 您可以选择通过此插件将 sw-precache 配置选项传递给 webpack。

# Prblm

## Dbg tol chrome,,,

## An SSL certificate error occurred when fetching the script.

index.html:12 servcie worker 注册失败

增加chorme qboot param,ingore slsl err..

## [未捕获（承诺）类型错误：无法在“addAll”上执行“缓存”：渐进式 Web 应用程序上的请求失败](https://stackoverflow.com/questions/66529102/uncaught-in-promise-typeerror-failed-to-execute-cache-on-addall-request)

对于第一个例外：-

未捕获（承诺）类型错误：无法在“addAll”上执行“缓存”：请求失败

当您在缓存列表中提到的任何文件返回 404 响应时，您会收到此异常。因此，请确保所有资源都提供 200。

对于第二个错误：-

当离线 start\_url 确实响应时，start\_url 不响应 200。

在您的情况下，由于文件没有被缓存（由于第一个异常），您会收到此错误，并确保在缓存列表中缓存根索引文件。

## Cant show install icon

Must paichu cache file and fetch event code...need...

## PWA 专属问题

建议使用 LightHouse 给你的 PWA 评分，报告中还会带有修改建议，十分实用。