PWA开发实践

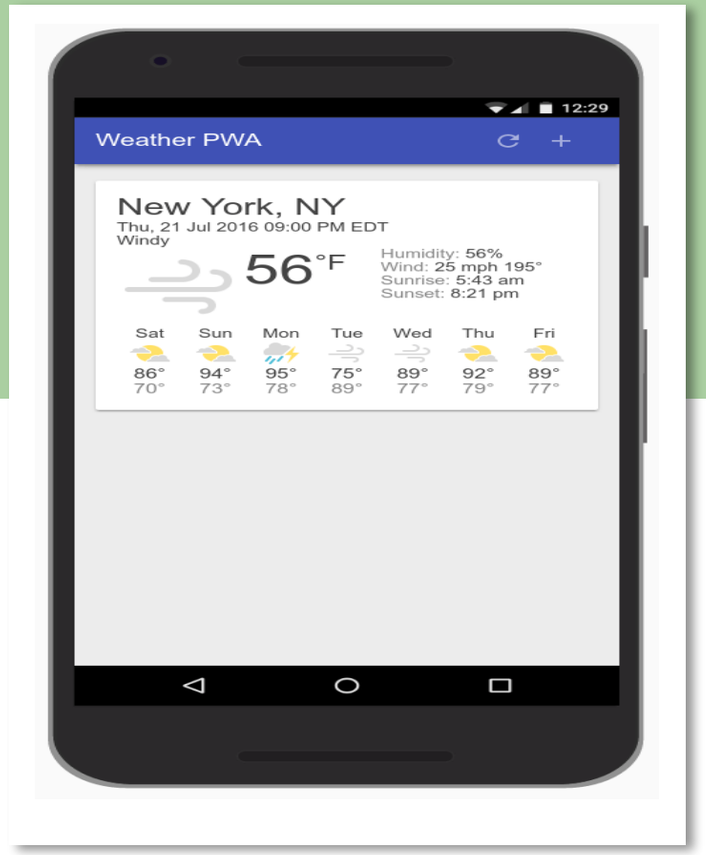
yuanzhijia@yidengxuetang.com

目录页

PAGE OF CONTENT

01

走进PWA的世界



Progressive Web Apps 是 Google 提出的用前沿的 Web 技术为网页提供 App 般使用体验的一系列方案。但是作为一个 web 应用，它可以 断网使用、推送消息、发送通知、从桌面启动，当然还包括 Web 应用的优势：免安装、快速开发、依赖浏览器跨平台(支持包括Edge在内的各种主流PC/手机浏览器)。

01 什么是 Progressive Web App?

01 响应式: 用户界面可以兼容多种设备，比如：桌面，移动端，平板。

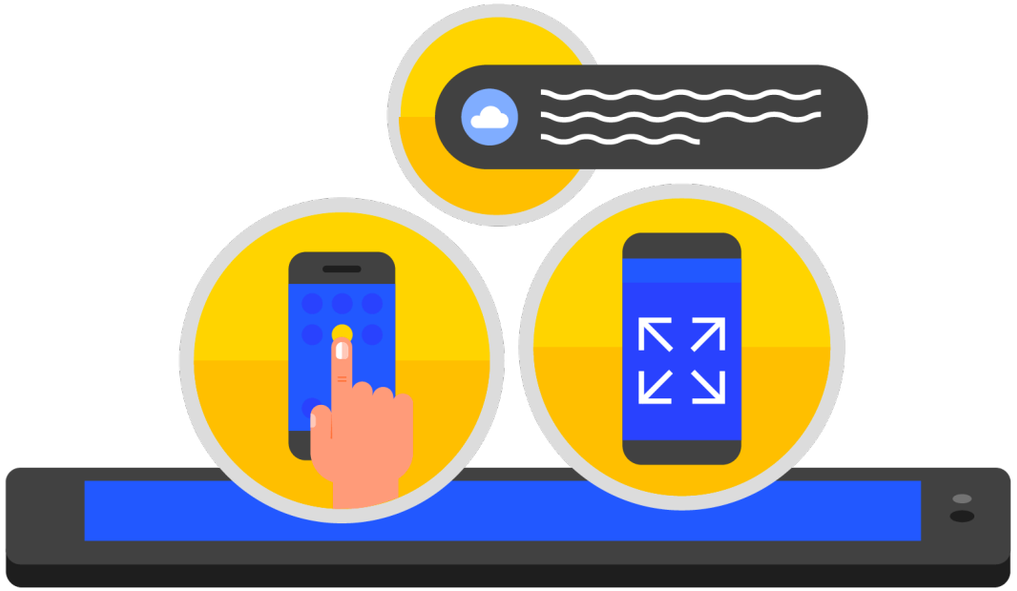
02 应用化: 交互体验接近native应用。

03 网络低依赖: 无网络或者低速网络下依然可以使用。

04 延续性: 借助推送功能，能够维持用户粘性。

05 可安装: 可以被安装到主屏，这样用户随时可以从主屏启动应用。

02 什么是 Progressive Web App?



06 可被发现: 被视为一个独立应用，而且可以被搜索.

07 可更新: 当用户重新联网时，可以更新内容.

08 安全: 可以使用HTTPS来防止内容伪造和中间人攻击。

09 渐进性: 所有用户都可以使用，浏览器无关。

10 兼容url: 可以通过url传播。

01

Progressive Web App

构建

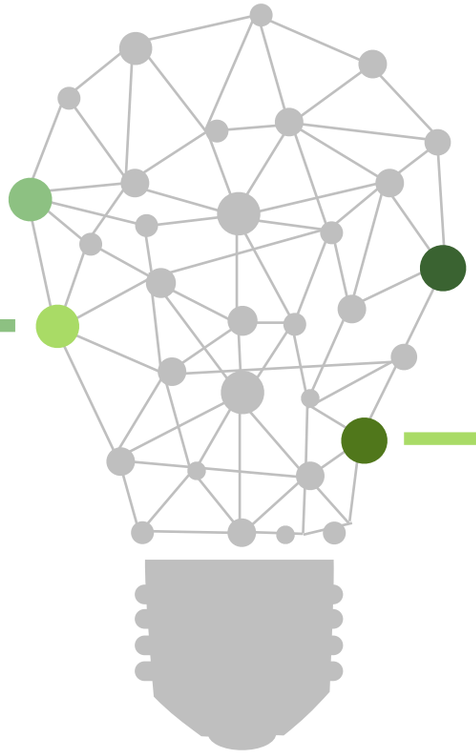
app shell

输入文字，点击添加相关标题文字，修改文字内容，也可以直接复制你的内容到此。

存储

输入文字

输入文字，点击添加相关标题文字，修改文字内容，也可以直接复制你的内容到此。



离线

输入文字

输入文字，点击添加相关标题文字，修改文字内容，也可以直接复制你的内容到此。

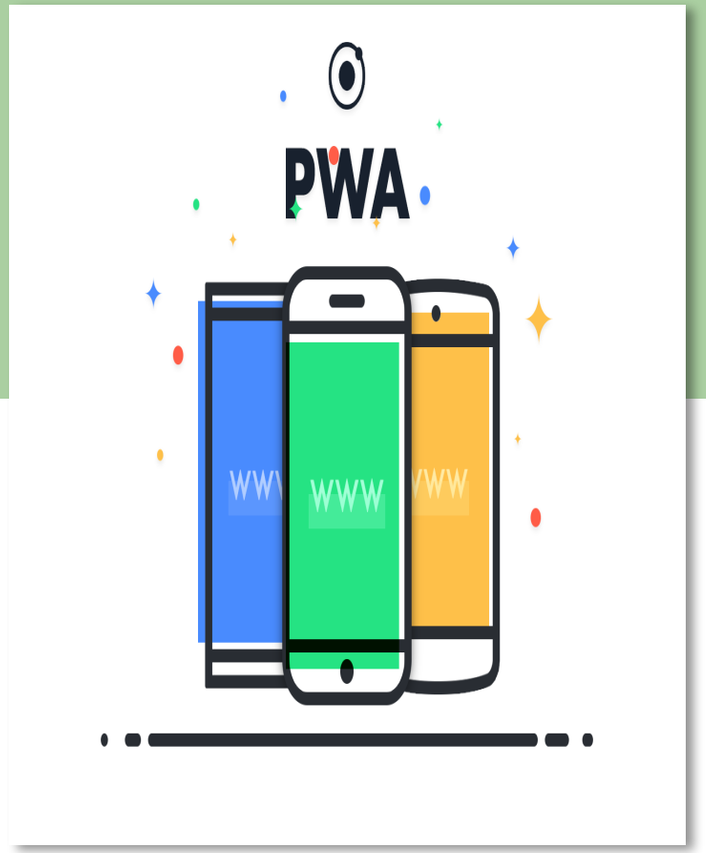
环境

输入文字

输入文字，点击添加相关标题文字，修改文字内容，也可以直接复制你的内容到此。

02

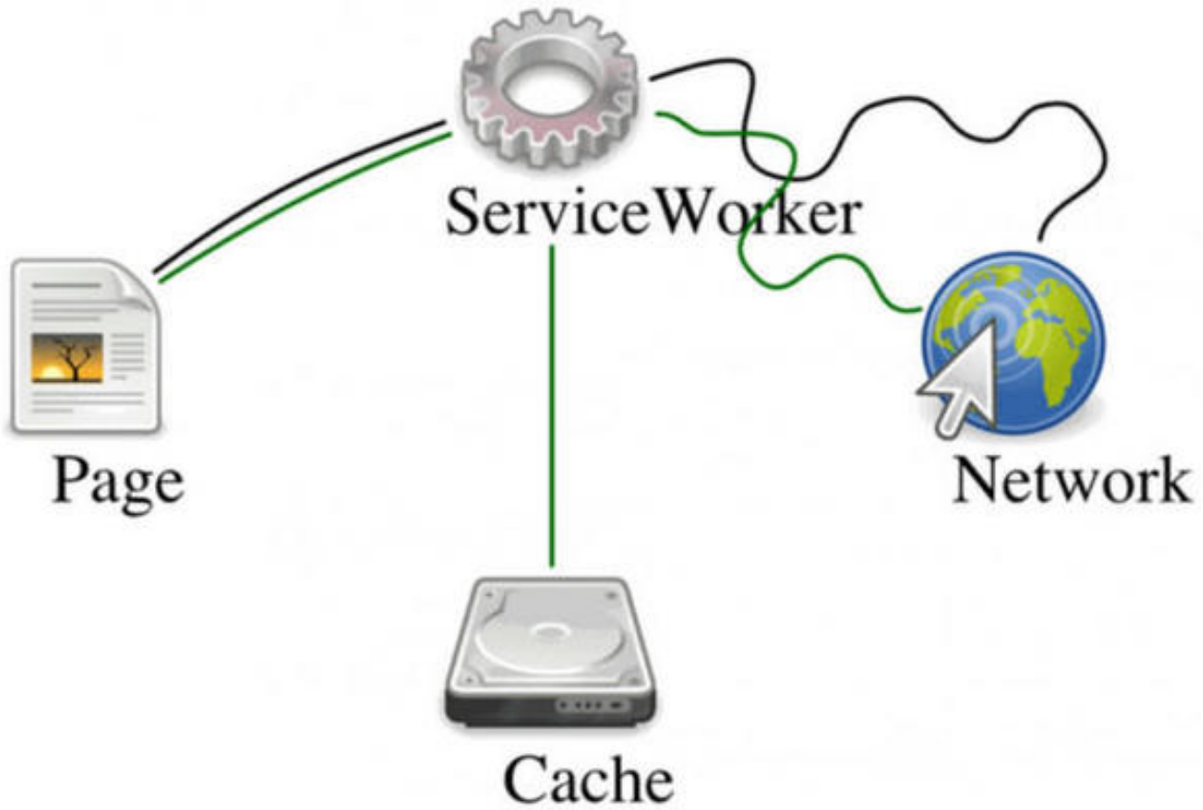
PWA核心技术



利用“App Shell”方法设计和构建应用、应用能够离线工作、存储供稍后离线使用的数据、代理的https服务器、Chrome 52 或更高版本

01

PWA技术体系



Intercept and reverse proxy

Instant Loading: Building offline-first PWA →

01 PWA核心技术

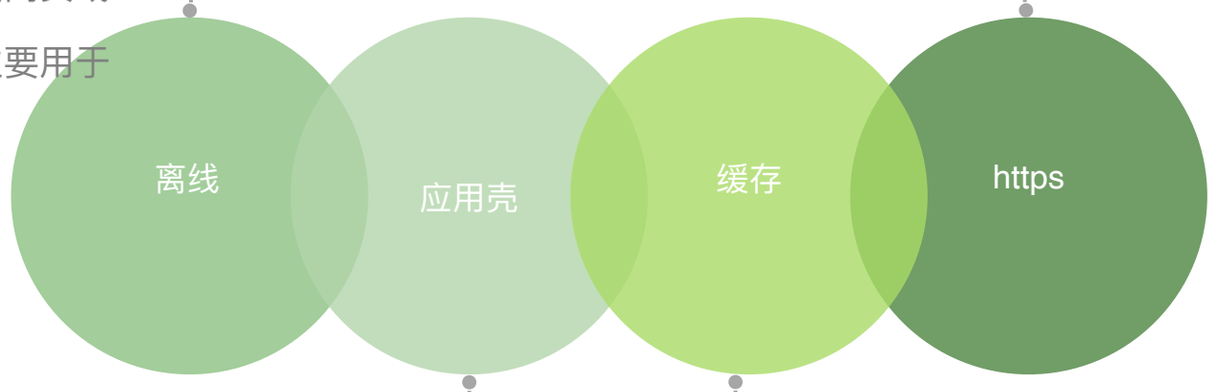
Service worker

它是一个在浏览器后台运行的脚本，与网页不相干，专注于那些不需要网页或用户互动就能完成的功能。它主要用于操作离线缓存。

测试环境&开发环境

http-server -c-1 # 注意设置关闭缓存

ngrok http 8080



App Shell

App Shell，顾名思义，就是“壳”的意思，也可以理解为“骨架屏”，说白了就是在内容尚未加载完全的时候，优先展示页面的结构、占位图、主题和背景颜色等，它们都是一些被强缓存的 html，css和javascript。

前端缓存介质

利用orm操作缓存的库?

offline.js实现数据的互通

02 Service Worker

作用

通过拦截网络请求，使得网站运行得更快，或者在离线情况下，依然可以执行。也作为其他后台功能的基础，比如消息推送和背景同步。

01 属于JavaScript Worker，不能直接接触DOM，通过postMessage接口与页面通信。

02 不用的时候会终止执行，需要的时候又重新执行，即它是事件驱动的。

03 只在HTTPs协议下可用，这是因为它能拦截网络请求，所以必须保证请求是安全的。

04 有一个精心定义的升级策略,内部大量使用Promise。

03

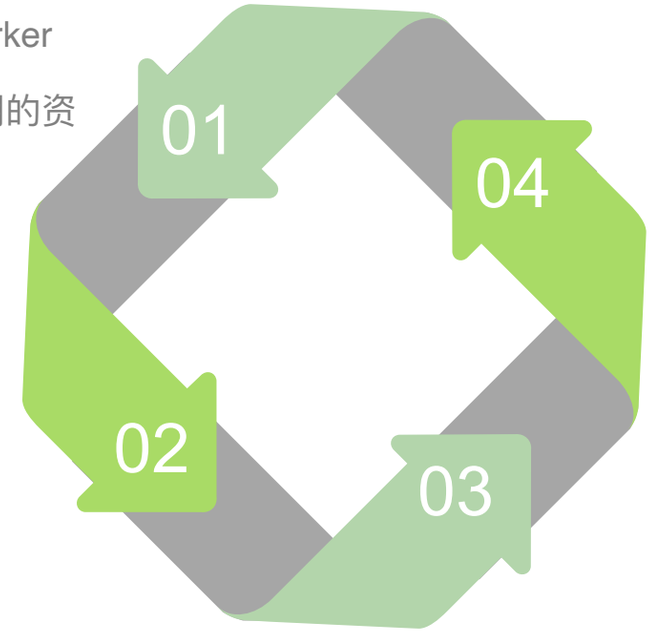
Service Worker 生命周期

INSTALLING 正在安装阶段

此阶段处于Service Worker被注册的时候。在这个阶段，Service Worker 的install() 方法将会被执行，声明的资源即将被加入缓存。

REDUNDANT 废弃阶段

当安装失败，激活失败或者当前 Service Worker 被其他 Service Worker 替换时，就会进入这个阶段，此时 Service Worker 将失去对页面的控制。



INSTALLED 安装完成阶段

当 Service Worker 完成其初始化安装之后就会进入到这个阶段。在这个阶段，它是一个“有效但尚未激活的 worker”，等待着客户端对其进行激活使用。我们可以在这个阶段告知用户，PWA 已经可以进行升级了。

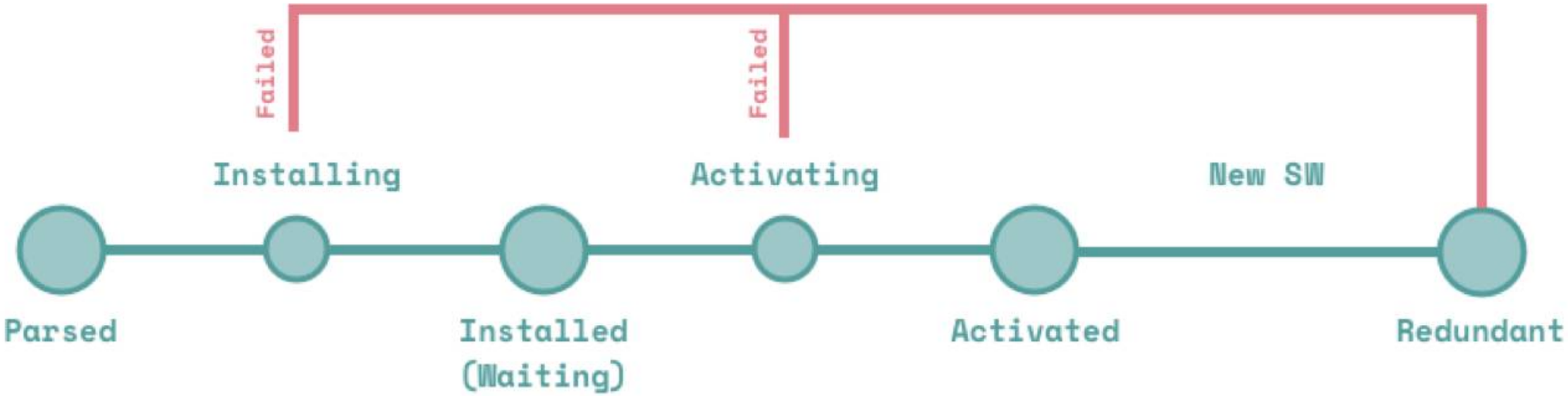
ACTIVETED 激活完成阶段

此时 Service Worker 已经被激活并生效，可以开始控制网站的资源请求了。此时 Service Worker 内的fetch 和message 事件已经可以被监听。

ACTIVATING 正在激活阶段

当客户端已无其它激活状态的 Service Worker、或者脚本中的skipWaiting() 方法被调用、或者用户关闭了该 Service Worker 作用域下的所有页面时，就会触发 ACTIVATING 阶段。在这个阶段中，Service Worker脚本中的activate() 事件会被执行，此时可以清除掉过期的资源，并将进入下一个“激活完成阶段”。

04 Service Worker 生命周期



Offline.js

Every app goes offline

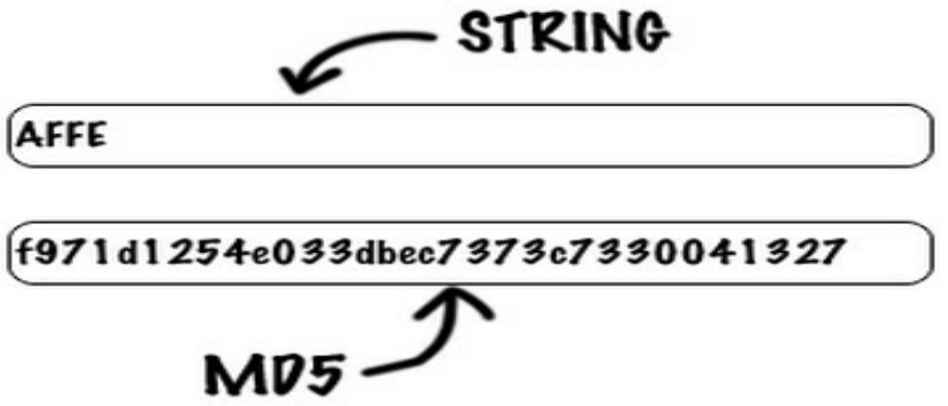
★ ON GITHUB

03

PWA工程化实践

PWA 的核心可谓是 Service Worker，任何一个PWA都有且只有一个service-worker.js 文件，用于为Service Worker添加资源列表，进行注册、激活等生命周期操作。但是在webpack构建的项目中，生成一个service-worker.js 可能会面临很多的问题。

01 Webpack&PWA

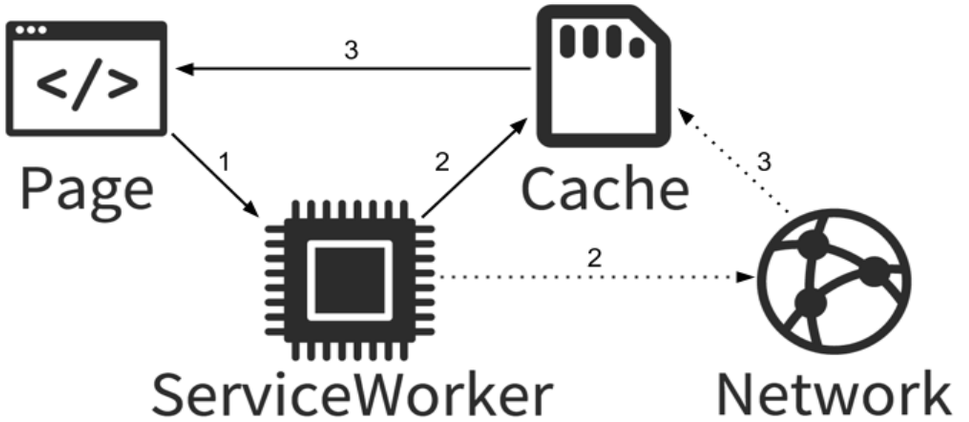


文件的版本戳

webpack生成的资源多会生成一串hash，Service Worker的资源列表里面需要同步更新这些带hash的资源；

sw.js文件的版本号

每次更新代码，都需要通过更新service-worker.js 文件版本号来通知客户端对所缓存的资源进行更新，（但使用明确的版本号会更加合适）。



02 offline-plugin PK Worker-precache-webpack-plugin

配置

更多的可选配置项，满足更加细致的配置要求；

文档

更为详细的文档和例子。

社区

更新频率相对更高，star数更多；

自动

自动处理生命周期，用户无需纠结生命周期的坑；

04

业界成熟方案

模板所有页面均有动画，修改方便，这里可以输入章节的小标题，内页图片可自由替换。

实时空气质量
60 良
更新于 14:56
9月28日 周四

23° 多云

今天： 19~24°C 多云 东南风1级

24小时 预报

昨天	今天	明天	周六	周日
28°	24°	24°	23°	29°
22°	19°	21°	21°	22°
西北风 3-4级	西北风 微风	东风 微风	东南风 微风	南风 3-4级

昨天

阴 22~28°C

风力 日出日

LAVAS

基于 Vue.js 的 PWA 解决方案

帮助开发者快速搭建 PWA 应用，解决接入 PWA 的各种问题

起步 GITHUB

01 什么是 Workbox?

01 workbox这个库和构建工具的集合使用了servicework 的 fetch event 和cache API，可以很容易地在用户的设备上本地存储您的网站文件。

02 让您的网站离线访问。

03 提高重复访问的负载性能，即使您不想完全离线，也可以使用 workbox 在本地存储和提供常用文件，而不是从网络中存储和提供。

04 迅速集成进workbox-webpack-plugin

02

帮助你的工具

https://realfavicongenerator.net/

tomitm.github.io/appmanifest

标题文字内容

标题文字内容

终 致谢