

Что такое PWA и как из вашего SPA сделать PWA с возможностью Offline доступа

Куценко Олег

JS Meetup, Харьков, 27 апреля 2020



Обо мне

1. Пишу на реакте 5+ лет с версии 13.0
2. Разработал три продакшен PWA проекта
3. Создаю сложные конструкторы сайтов
4. В свободное время увлекаюсь робототехникой



Сегодня мы рассмотрим

1. Как сделать кеширование файлов для доступа без интернета
2. Зачем надо service-worker и почему они полезны
3. Как подобрать свой стиль. Подробнее о manifest.json
4. Заключение - показ готового сайта с PWA



Приступая к работе

Король и Шут

Бал
лицемеров
Король и Шут





Как работают сайты без интернета?



No internet

Try:

- Checking the network cables, modem, and router
- Reconnecting to Wi-Fi
- [Running Windows Network Diagnostics](#)

ERR_INTERNET_DISCONNECTED



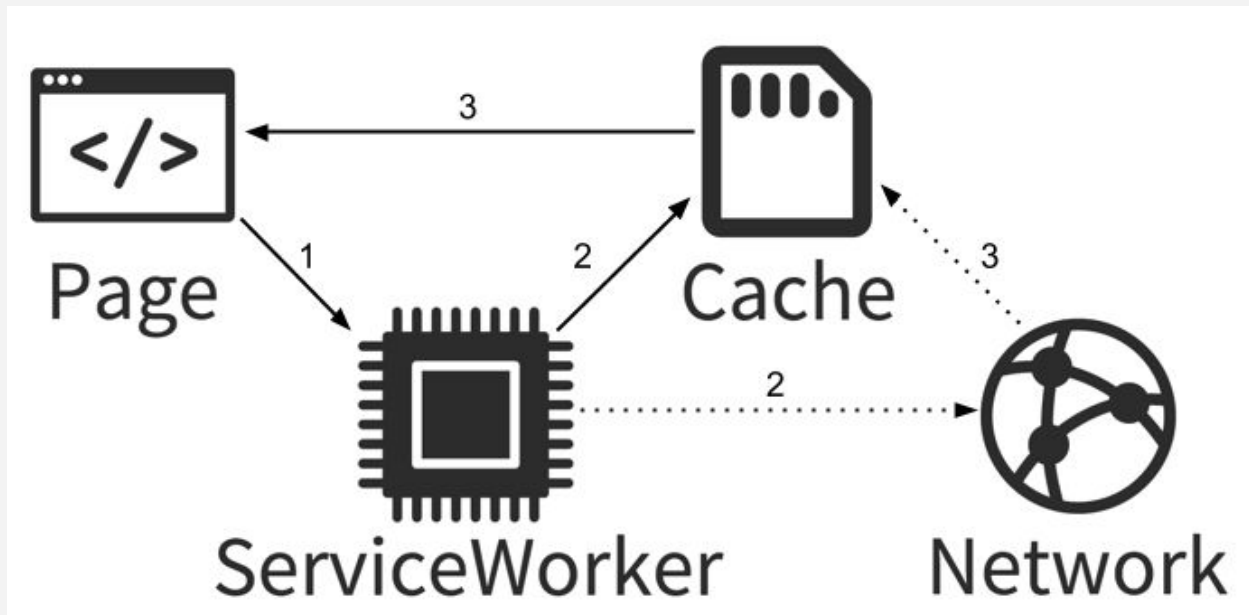
Что такое PWA

1. Доступ к приложению без интернета
2. Ускорение загрузки сайта из кеша
3. Свой цвет меню в мобильном приложении
4. Добавление приложения на рабочий стол
5. Push notifications



Service Worker

Service worker — это событийно-управляемый [worker](#), который регистрируется между web страницей и сетью.





Service Worker for everyone

The screenshot displays the Chrome DevTools Application tab, which is used for managing and testing Service Workers. The left sidebar shows the 'Service Workers' section under the 'Application' tab, with options to view the 'Manifest', 'Service Workers', and 'Clear storage'. The main panel lists several registered Service Workers, each with its own 'Update' and 'Unregister' links. The first worker is for 'https://4j7m47v1m4.codesandbox.io/' with source 'sandbox-service-worker.js', received on 01/01/1970, 03:00:00, and status '#2966 activated and is stopped'. The second worker is for 'https://4j7m47v1m4.csb.app/' with source 'sandbox-service-worker.js', received on 30/07/2019, 22:38:33, and status '#3835 activated and is stopped'. The third worker is for 'https://airhorner.com/' with source 'sw.js', received on 18/08/2019, 21:56:28, and status '#4029 activated and is stopped'. The fourth worker is for 'https://alexjoverm.github.io/' with source 'service-worker.js', received on 01/01/1970, 03:00:00. Each worker entry includes a 'Push' button with the text 'Test push message from DevTools.' and a 'Sync' button with the text 'test-tag-from-devtools'.

URL	Source	Received	Status	Push	Sync
https://4j7m47v1m4.codesandbox.io/	sandbox-service-worker.js	Received 01/01/1970, 03:00:00	#2966 activated and is stopped start	Test push message from DevTools. Push	test-tag-from-devtools Sync
https://4j7m47v1m4.csb.app/	sandbox-service-worker.js	Received 30/07/2019, 22:38:33	#3835 activated and is stopped start	Test push message from DevTools. Push	test-tag-from-devtools Sync
https://airhorner.com/	sw.js	Received 18/08/2019, 21:56:28	#4029 activated and is stopped start	Test push message from DevTools. Push	test-tag-from-devtools Sync
https://alexjoverm.github.io/	service-worker.js	Received 01/01/1970, 03:00:00			



Кэширование файлов (CacheStorage)

Что позволяет делать:

- Хранить большие объемы данных (включая аудио и видео)
- Получать данные без доступа к интернету
- Реализовать стратегию кэширования (Cache only, Network only, Cache First, Network First)

Недостатки

- Доступно только из ServiceWorker



Стратегии кэширование - это просто

```
75 self.addEventListener("fetch", function(event) {
76   event.respondWith(
77     caches
78       .match(event.request)
79       .then(function(response) {
80         // Cache hit - return response
81         if (response) {
82           console.log("Found ", event.request.url, " in cache");
83           return response;
84         }
85         console.log("Network request for ", event.request.url);
86         return fetch(event.request).then(response => {
87           if (CACHE_WHEN_FETCH) {
88             // Check if we received a valid response
89             if (!response || response.status !== 200 || response.type !== "basic") {
90               return response;
91             }
92           |
93             caches.open(CACHE_NAME).then(function(cache) {
94               cache.put(event.request, response);
95             });
96
97           return response;
98         } else {
99           return response;
100         }
101       });
102     });
103   });
104 }
```



В Chrome кеширование можно найти тут

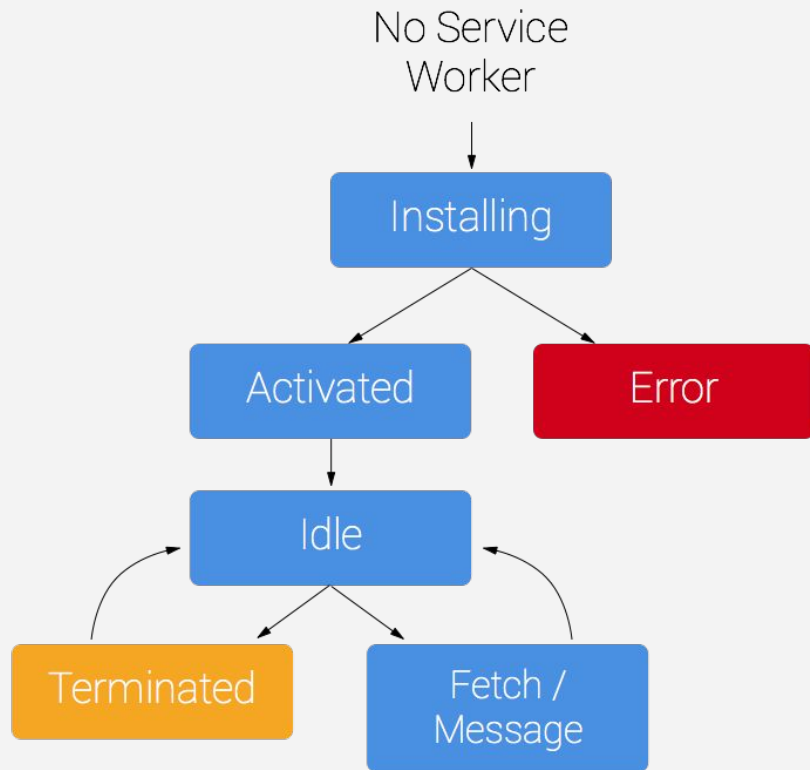
The screenshot shows the Chrome DevTools interface with the 'Application' tab selected. The left sidebar is divided into several sections: 'Application' (Manifest, Service Workers, Clear storage), 'Storage' (Local Storage, Session Storage, IndexedDB, Web SQL, Cookies), 'Cache' (Cache Storage, devsite.pwa_RUNTIME_v2, Application Cache), 'Background Services' (Background Fetch, Background Sync, Notifications, Payment Handler, Push Messaging), and 'Frames'. The 'Cache' section is expanded, and 'Cache Storage' is selected. The right pane displays a table of cached resources.

#	Name	Respo...	Conten...	Conten...	Time C...
0	/_pwa/developers/icons/icon-144x144....	basic	image/...	3,513	2/11/2...
1	/_pwa/developers/manifest.json	basic	applica...	304	2/11/2...
2	/_static/images/lockup-developers.svg	cors	image/...	2,315	2/11/2...
3	/_static/images/video-placeholder.svg	cors	image/...	548	2/11/2...
4	/_static/offline?partial=1	basic	applica...	6,781	2/11/2...
5	/site-assets/logo-github.svg	basic	image/...	654	2/11/2...
6	/site-assets/logo-twitter.svg	basic	image/...	783	2/11/2...
7	/sw.js	basic	applica...	11,824	2/11/2...
8	/web/fundamentals/primers/service-wo...	basic	text/ht...	21,103	2/11/2...
9	/web/fundamentals/primers/service-wo...	basic	applica...	18,739	2/11/2...
10	/web/fundamentals/primers/service-wo...	basic	image/...	14,070	2/11/2...
11	/web/fundamentals/primers/service-wo...	basic	image/...	13,100	2/11/2...



Как с этим работать

1. Установка
2. Активация
3. Обработка ошибки
4. Ожидание запроса
5. Получение данных с сервера





Немного кода

```
var CACHE_NAME = 'my-site-cache-v1';
var urlsToCache = [
  '/',
  '/styles/main.css',
  '/script/main.js'
];

self.addEventListener('install', function(event) {
  // Perform install steps
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(function(cache) {
        console.log('Opened cache');
        return cache.addAll(urlsToCache);
      })
  );
});
```



Как добавить на сайт?

```
if ('serviceWorker' in navigator) {  
  window.addEventListener('load', function() {  
    navigator.serviceWorker.register('/sw.js').then(function(registration) {  
      // Registration was successful  
      console.log('ServiceWorker registration successful with scope: ', registration.scope);  
    }, function(err) {  
      // registration failed :(  
      console.log('ServiceWorker registration failed: ', err);  
    });  
  });  
}
```



Workbox - готовый service-worker



<https://developers.google.com/web/tools/workbox/guides/get-started>



Workbox - готовый service-worker

Преимущество:

1. Легко подключать
2. Стратегии кеширования уже заданы

Недостатки:

1. Мы не можем выйти за рамки апи
- 2.



Легко подключать

```
importScripts('https://storage.googleapis.com/workbox-cdn/releases/5.0.0/workbox-sw.js');
```

```
if (workbox) {  
  console.log(`Yay! Workbox is loaded 🎉`);  
} else {  
  console.log(`Boo! Workbox didn't load 😬`);  
}
```



Стратегии кэширования уже заданы

```
import {registerRoute} from 'workbox-routing';  
import {NetworkFirst} from 'workbox-strategies';
```

```
registerRoute(  
  /\.js$/,  
  new NetworkFirst()  
);
```



Внедрение на сайт

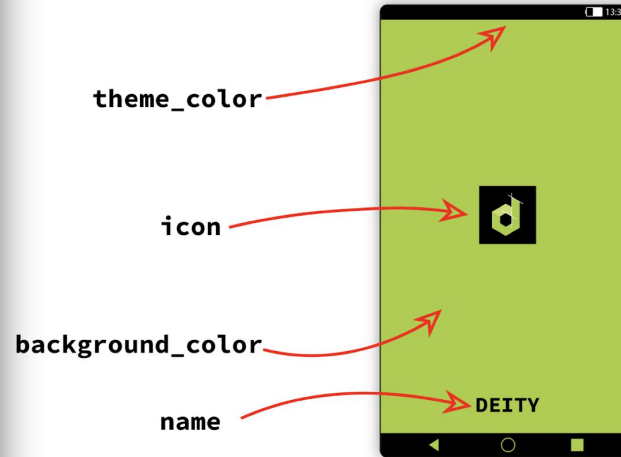
```
<script>
// Check that service workers are supported
if ('serviceWorker' in navigator) {
  // Use the window load event to keep the page load performant
  window.addEventListener('load', () => {
    navigator.serviceWorker.register('/service-worker.js');
  });
}
</script>
```



manifest.json

```
{
  "name": "Deity",
  "short_name": "Deity",
  "description": "Deity - Modular Commerce",
  "start_url": "/",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#ffffff",
  "icons": [
    {
      "src": "/static/media/deity.377581b7.svg",
      "sizes": "48x48",
      "type": "image/svg"
    },
    {
      "src": "/static/media/deity.377581b7.svg",
      "sizes": "72x72",
      "type": "image/svg"
    },
    {
      "src": "/static/media/deity.377581b7.svg",
      "sizes": "96x96",
      "type": "image/svg"
    },
    {
      "src": "/static/media/deity.377581b7.svg",
      "sizes": "144x144",
      "type": "image/svg"
    },
    {
      "src": "/static/media/deity.377581b7.svg",
      "sizes": "168x168",
      "type": "image/svg"
    },
    {
      "src": "/static/media/deity.377581b7.svg",
      "sizes": "192x192",
      "type": "image/svg"
    },
    {
      "src": "/static/media/deity.377581b7.svg",
      "type": "image/svg"
    }
  ]
}
```

WEB APP MANIFEST





Lighthouse Chrome



Progressive Web App

These checks validate the aspects of a Progressive Web App. [Learn more.](#)



Fast and reliable

- Page load is fast enough on mobile networks
- Current page responds with a 200 when offline
- start_url responds with a 200 when offline



Installable

- Uses HTTPS
- Registers a service worker that controls page and start_url
- Web app manifest meets the installability requirements



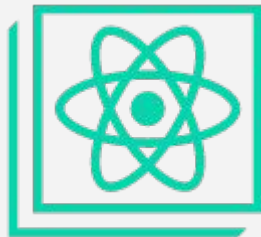
PWA Optimized

- Redirects HTTP traffic to HTTPS
- Configured for a custom splash screen
- Sets a theme color for the address bar.
- Content is sized correctly for the viewport
- Has a <meta name="viewport"> tag with width or initial-scale
- Contains some content when JavaScript is not available
- Provides a valid apple-touch-icon



Create-react-app

1. Доступно из коробки
2. Настроен через `workbox-webpack-plugin`
3. Необходимо удалить 2 буквы
(`serviceWorker.register()`)
4. `import` статика по умолчанию через `precache-manifest`





Рабочее PWA приложение

Король и Шут

Бал
лицемеров
Король и Шут





Недостатки PWA

1. Для работы offline требуется cache-first стратегия (приложение получит обновление при следующей загрузке)
2. Данные нужно кешировать сторонними либами (cache-persist, firestore)
3. В закрытых сетях больше вредит, т.к. требуется предавторизация
4. Пользователю надо хоть что то показать - нужно заботится о доступности без соединения





Выводы

1. Идеальное решения для портфолио
2. Плохо работает в закрытой среде (под авторизацией)
3. Клиент увидит новый контент только после последующего обновления
4. Для кеширования данных лучше использовать стороннее решение
5. Можно использовать Workbox





Полезные ресурсы

1. <https://codelabs.developers.google.com/dev-pwa-training/> - официальная тренинг документация от Google
2. <https://developers.google.com/web/tools/workbox> Workbox - утилита для быстрого подключения pwa к приложению
3. <https://developers.google.com/web/ilt/pwa/caching-files-with-service-worker> - описание стратегий кэширования
4. <https://github.com/inthedark122/music-portfolio/blob/v3/public/sw-v1.js> - Service worker используемы в демо приложении



Спасибо, вопросы?

Куценко Олег

Telegram: @inthedark122

o.kutsenko@andersenlab.com

www.andersenlab.com