
Progresywne aplikacje webowe

Potęga aplikacji natywnych w przeglądarce

Tal Alter

przekład: Maria Chaniewska

APN Promise
Warszawa 2018

O'REILLY®

Spis treści

Przedmowa	xi
1. Wprowadzenie do progresywnych aplikacji webowych	1
Powrót sieci WWW	2
Obecny krajobraz aplikacji mobilnych	3
Zalety progresywnych aplikacji webowych	5
Karta, WWW i service worker	7
2. Pierwszy skrypt service worker	9
Konfigurowanie projektu przykładowego	9
Witamy w hotelu Gotham Imperial	10
Poznanie kodu	12
Obecne działanie w trybie offline	12
Tworzenie pierwszego skryptu service worker	14
Co to jest progresywne udoskonalanie?	18
Protokół HTTPS i skryptu service worker	19
Pobieranie zawartości z sieci WWW	19
Przechwytywanie żądań w trybie offline	21
Tworzenie odpowiedzi HTML	23
Pojęcie zasięgu skryptu service worker	25
Podsumowanie	26
3. Interfejs API CacheStorage	27
Co to jest CacheStorage, a co ważniejsze, czym nie jest	28
Decydowanie o czasie buforowania	29
Zapisywanie żądań w CacheStorage	30
Pobieranie żądań z CacheStorage	31
Buforowanie w przykładowej aplikacji	32
Dopasowywanie właściwych odpowiedzi do poszczególnych żądań	34
Buforowanie HTTP i nagłówki HTTP	36
Podsumowanie	37
4. Cykl życia skryptu service worker i zarządzanie buforowaniem	39
Cykl życia skryptu service worker	42

Cykl życia skryptu service worker i znaczenie metody waitUntil	45
Aktualizowanie skryptu service worker	46
Dlaczego potrzebne jest zarządzanie buforowaniem	48
Zarządzanie buforowaniem i czyszczenie starych pamięci cache	51
Ponowne używanie zbuforowanych odpowiedzi	56
Konfigurowanie serwera, aby udostępnił właściwe nagłówki buforowania	58
Narzędzia dla programistów	58
Konsola	58
Próbowałeś wyłączyć i włączyć to ponownie?	59
Inspekcja CacheStorage i IndexedDB	59
Dławienie sieci i symulowanie warunków offline	60
Lighthouse	60
Podsumowanie	61
5. Podejście najpierw tryb offline	63
Na czym polega podejście najpierw tryb offline?	64
Typowe wzorce buforowania	65
Mieszanie i łączenie: tworzenie nowych wzorców	69
Planowanie strategii buforowania	71
Implementacja strategii buforowania	74
Architektura powłoki aplikacji	85
Dołączanie treści do początkowego renderowania	87
Implementacja powłoki aplikacji	88
Nowe możliwości	90
Podsumowanie	92
6. Lokalne przechowywanie danych w IndexedDB	93
Co to jest IndexedDB	94
Używanie IndexedDB	97
Otwieranie połączenia z bazą danych	97
Wersjonowanie bazy danych/modyfikowanie magazynu obiektów	98
Dodawanie danych do magazynu obiektów	99
Odczytywanie danych z magazynu obiektów	100
Zarządzanie wersjami IndexedDB	102
Odczytywanie obiektów przy użyciu kursora	104
Tworzenie indeksów	105
Odczytywanie danych przy użyciu indeksu	107
Ograniczenia zakresu kursora	108
Ustawianie kierunku kursora	109
Aktualizowanie obiektów w magazynie obiektów	110
Usuwanie obiektów z magazynu obiektów	111
Usuwanie wszystkich obiektów z magazynu obiektów	112

Obsługa propagacji błędów w IndexedDB	112
IndexedDB dla mistrzów SQL	113
IndexedDB w praktyce	114
Obiecana baza danych	123
Utrzymanie IndexedDB	130
Używanie IndexedDB w skrypcie service worker	131
Ekosystem IndexedDB	132
PouchDB	133
localForage	133
Dexie.js	134
IndexedDB Promised	134
Podsumowanie	135
7. Zapewnianie funkcjonalności offline dzięki synchronizacji w tle	137
Działanie synchronizacji w tle	139
SyncManager	141
Dostęp od interfejsu SyncManager	141
Rejestrowanie zdarzeń	142
Zdarzenia synchronizacji	142
Tagi zdarzeń	143
Pobieranie listy zarejestrowanych zdarzeń synchronizacji	143
Ostatnie szanse	144
Przekazywanie danych do zdarzenia synchronizacji	145
Utrzymywanie kolejki akcji w IndexedDB	145
Utrzymywanie kolejki żądań w IndexedDB	148
Przekazywanie danych w tagu zdarzenia sync	150
Dodawanie synchronizacji w tle do przykładowej aplikacji	150
Podsumowanie	159
8. Komunikacja między skrypcem service worker a stroną przy użyciu publikowania wiadomości	161
Przesyłanie wiadomości z okna do skryptu service worker	162
Przesyłanie wiadomości ze skryptu service worker do wszystkich otwartych okien	164
Przesyłanie wiadomości ze skryptu service worker do konkretnego okna	166
Utrzymywanie otwartego łącza komunikacyjnego za pomocą MessageChannel	167
Komunikacja między oknami	171
Publikowanie wiadomości ze zdarzenia sync do strony	174
Podsumowanie	176

9. Zdobywanie miejsca na ekranie startowym dzięki instalacji aplikacji webowych	177
Możliwość instalacji aplikacji webowych	178
W jaki sposób przeglądarki decydują, kiedy pokazać baner instalacji aplikacji	179
Anatomia manifestu aplikacji webowej	180
Wady, skutki uboczne i przyszła zgodność	185
Podsumowanie	186
10. W zasięgu powiadomień z serwera	187
Działanie powiadomień z serwera	187
Interfejs API Notification	188
Interfejs API Push	188
Push + Notification	190
Tworzenie powiadomień	191
Żądanie pozwolenia na powiadomienia	191
Pokazywanie powiadomień	194
Dodawanie obsługi powiadamiania do aplikacji Gotham Imperial Hotel	199
Subskrypcja zdarzeń push przez użytkownika	202
Generowanie kluczy VAPID: publicznego i prywatnego	203
Generowanie klucza GCM	204
Tworzenie nowej subskrypcji	206
Subskrypcja wiadomości push dla użytkowników Gotham Imperial Hotel	208
Wysyłanie zdarzeń push z serwera	211
Nasłuchiwanie zdarzeń push i pokazywanie powiadomień	214
Przesłuchiwanie powiadomień	220
Podsumowanie	221
11. Środowisko użytkownika progresywnej aplikacji webowej	223
Wdzięk i zaufanie	223
Komunikowanie stanu ze skryptu service worker	225
Komunikacja przy użyciu biblioteki Progressive UI KIT'T	227
Typowe komunikaty w progresywnych aplikacjach webowych	230
Ukończenie buforowania	230
Zbuforowano stronę	230
Akcja się nie powiodła, ale zostanie ukończona po odzyskaniu łączności	231
Włączono powiadomienia	231
Dobieraj właściwe słowa	232
Zawsze finalizuj	232
Projekt progresywnej aplikacji webowej	235

Projekt powinien odzwierciedlać zmienne warunki	236
Projekt powinien pasować do środowiska	237
Projekt powinien adaptować się do konkretnego medium	237
Projekt powinien wzbudzać pewność i informować użytkownika	237
Projekt powinien pomóc osiągać cele użytkowników i firmy	238
Podjęcie odpowiedzialności za monit o instalacji	238
Mierzenie i określanie docelowej wydajności przy użyciu RAIL	239
Podsumowanie	242
12. Następne rozwiązania dla aplikacji PWA	243
Akceptowanie płatności przy użyciu interfejsu API Payment Request	243
Zarządzanie użytkownikami za pomocą interfejsu API Credential Management	245
Grafika w czasie rzeczywistym dzięki WebGL	246
Futurystyczne interfejsy API z rozpoznawaniem mowy	248
Rzeczywistość wirtualna w przeglądarce dzięki WebVR	249
Łatwe udostępnienie do aplikacji i z aplikacji	249
Udane interfejsy użytkownika odtwarzające multimedia	251
Nowa wspaniała era	252
Dodatek A. Skrypty service worker: wspaniała możliwość wykorzystania standardu ES2015	255
Literały szablonowe	256
Funkcje strzałkowe	256
Destrukturyzacja obiektu	257
Więcej ES2015	258
Dodatek B. Pełnoekranowe reklamy wewnętrzne: jak zniecierpliwienie trzaśnięcia drzwiami	259
Dodatek C. CORS kontra NO-CORS	261
Indeks	263
O autorze	277
Kolofon	278