

Спикер



Константин Шкурко

Разработчик мобильных приложений с 10+ годами опыта. Разрабатываю под iOS и Android (нативно и кроссплатформенно). Умею в веб-разработку и бэк на Python и Golang. Имею черный пояс по пальцевому кунг-фу в области мануального и не только тестирования.

Шью сумки и рюкзаки из кожи и заставляю всех это носить. Мастер спорта по силовому троеборью и многоборец с тремя своими детьми.

PWA вместо App Store

Опыт замены нативного iOS-приложения и технические ограничения

Константин Шкурко
Россельхозбанк

Что же делать?



Мы перебрали все варианты

✗ TestFlight

- Нужен активный аккаунт Apple
- Сборки живут 30 дней

✗ Enterprise Distribution

- Только для внутреннего использования (установка по шнурку)
- И снова без аккаунта Apple никуда

PWA ≠ mobile web

Установка

Иконка

Offline

Push

RWA – выход найден + ПЛЮШКИ

Деплоим когда хотим
Никакого ревью от Apple

Выглядит как нативное
приложение

Аккаунт разработчика не нужен

Никаких ограничений по
гайдлайнам

**Пока всё норм, но где-то внутри сидит маленький
дед с тревогой и бурчит.**

Капелька истории, позже пригодится

- 1999** Microsoft .hta (HTML App)
- 2007** Стив Джобс Web Clip (Ajax)
- 2015** Google PWA (дизайнер Фрэнсис Бэрримэн и инженер Алекс Расселл)
- 2018** Для iOS разработчиков PWA появляется только сейчас (iOS 11.3 поддерживает Service Worker и Web App Manifest)

manifest



HTTPS

service worker



2015

- ✓ Manifest
- ✓ Service worker
- ✓ Push-notifications



2018

- ✓ ✗ Manifest
- ✓ Service worker
- ✗ Push-notifications

manifest.json



Основной
документ PWA

Как браузер понимает что это PWA?

```
<link rel="manifest" href="/manifest.json">
```

```
{  
  "name": "My App",  
  "short_name": "My App",  
  "start_url": "/",  
  "display": "standalone",  
  "icons": [  
    {  
      "src": "/icons/icon-192.png",  
      "sizes": "192x192",  
      "type": "image/png"  
    },  
    {  
      "src": "/icons/icon-512.png",  
      "sizes": "512x512",  
      "type": "image/png"  
    }  
  ]  
}
```

Минимальная версия
манифеста





Обычный manifest.json + meta-тэги

```
<link rel="manifest" href="/manifest.json">  
<link rel="apple-touch-icon" href="/icons/apple-touch-icon.png">  
<meta name="apple-mobile-web-app-capable" content="yes">  
<meta name="apple-mobile-web-app-title" content="My App">
```

name и short_name



short_name

Экран
установки
PWA

name

Иконки приложения

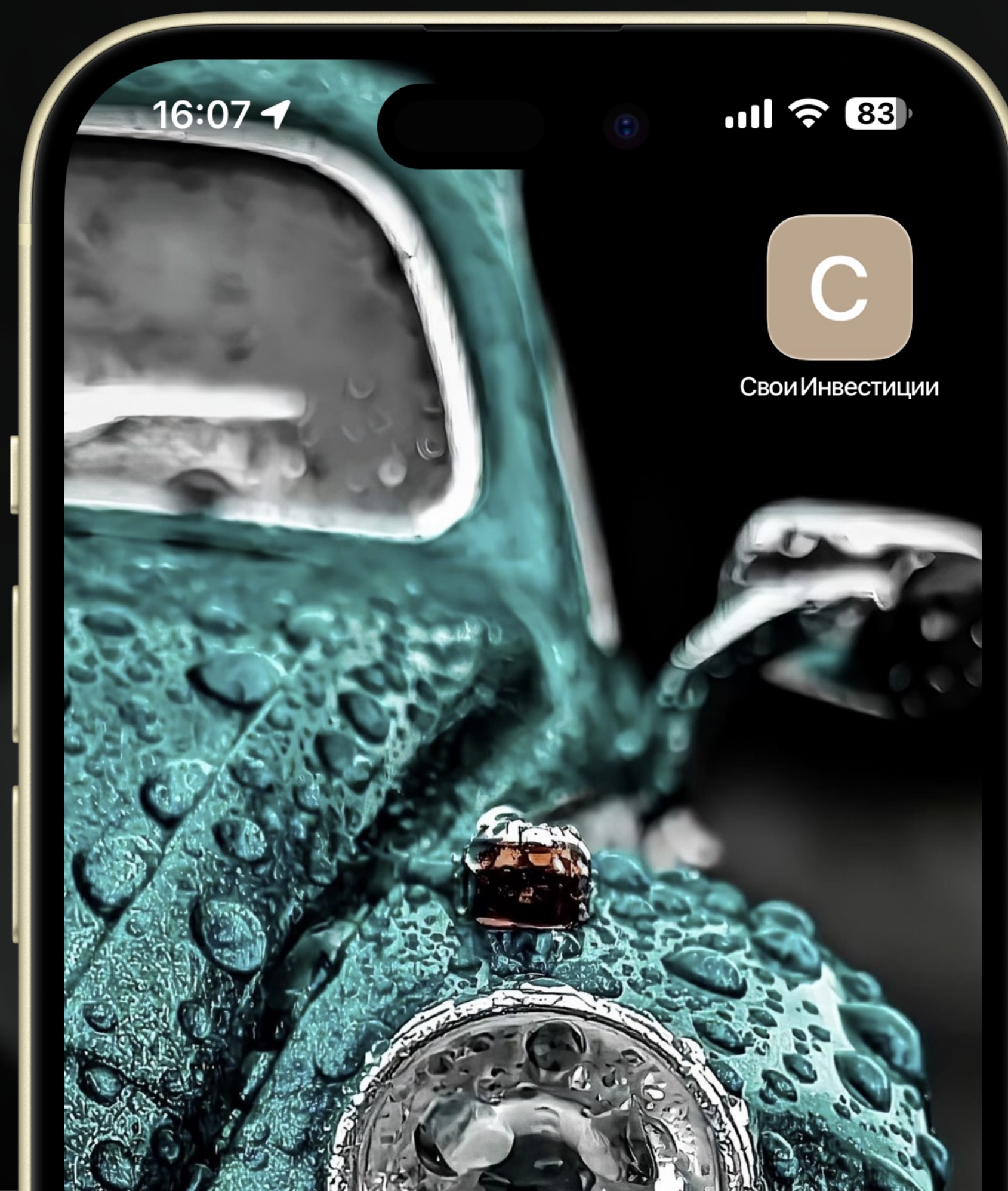
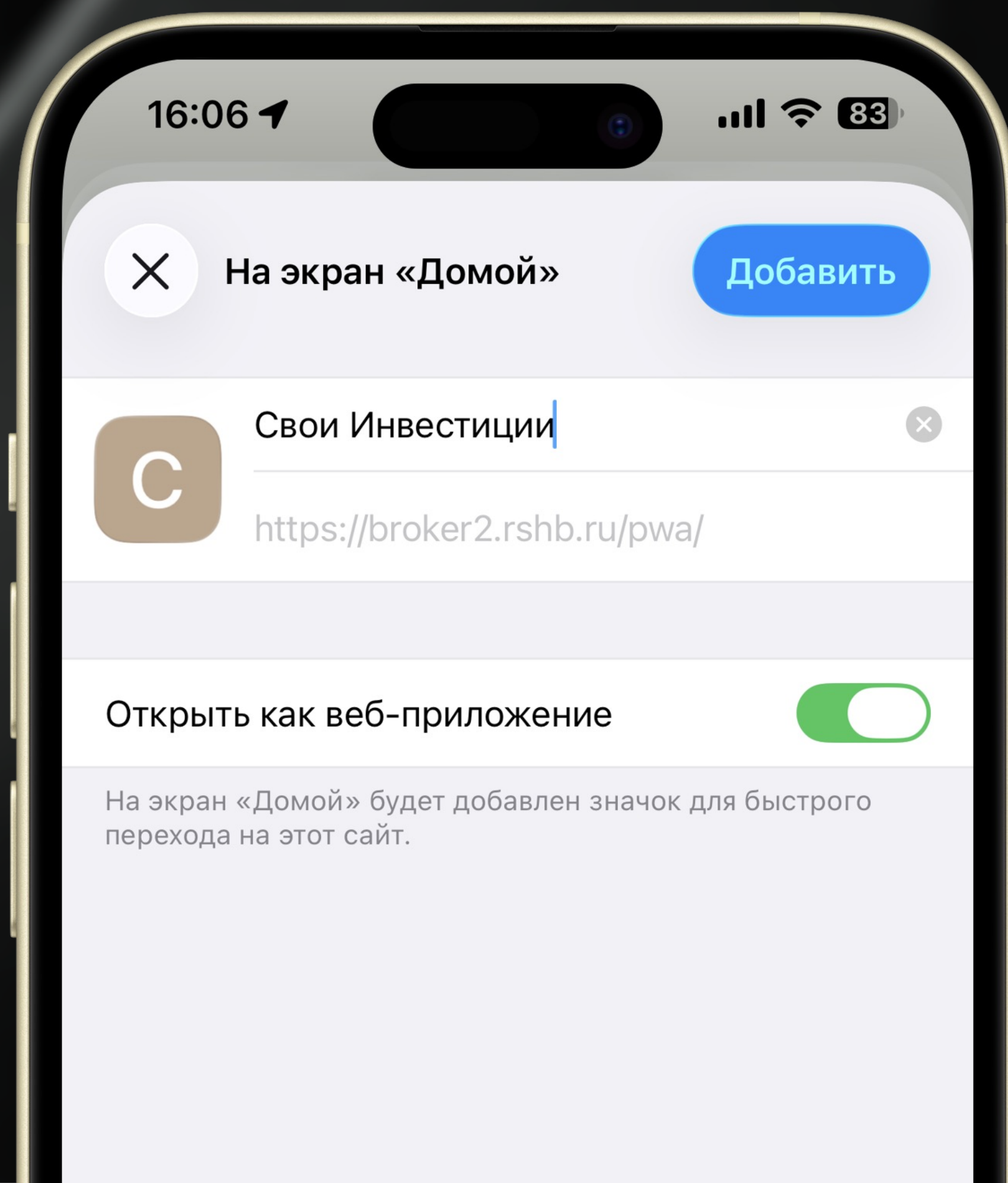
48
x
48

192
x
192

512
x
512

```
<link rel="apple-touch-icon" href="%PUBLIC_URL%/icon-test-180.png" />  
<link rel="apple-touch-icon" sizes="152x152" href="%PUBLIC_URL%/icon-test-152.png" />  
<link rel="apple-touch-icon" sizes="180x180" href="%PUBLIC_URL%/icon-test-180.png" />  
<link rel="apple-touch-icon" sizes="192x192" href="%PUBLIC_URL%/icon-test-192.png" />
```

Иконки приложения



start_url и scope

Возврат из браузера
обратно в scope
не работает в iOS

/

- /app ← **start_url**
 - /app/payments
 - /app/profile
- /feature/logout ← **вне /app/* браузер**

display

- ✓ browser
- ✓ minimal-ui
- ✓ standalone
- ✓ fullscreen



- ✓ browser
- ✗ minimal-ui
- ✓ standalone
- ✗ fullscreen



theme_color и background_color



работают не стабильно и
перекрываются meta-тэгами

```
<meta name="theme-color" content="#fff" />  
<meta name="apple-mobile-web-app-status-bar-style" content="default" />
```

Все поля `manifest.json` и их поддержка



**Почему же Apple использует
архаичные meta-тэги и игнорит
ключи в manifest.json?**

**А помните я вам говорил, что знание истории даст нам
ответы на вопросы?**



Service Worker

сеть

кэш

offline

пуши



Жизненный цикл Service Worker

```
if ('serviceWorker' in navigator) {  
  window.addEventListener('load', async () => {  
    try {  
      const registration = await navigator.serviceWorker.register('/sw.js', {  
        scope: '/',  
      });  
  
      console.log('SW registered:', registration.scope);  
    } catch (error) {  
      console.error('SW registration failed:', error);  
    }  
  });  
}
```

**register > parsed
Service Worker**

Жизненный цикл Service Worker

Install - идеальное место для первого оффлайн запуска

 Из кэша

ИКОНКИ

js

css

html



ЧИСТКА КЭША

миграции indexedDB

бизнес логика

Жизненный цикл Service Worker

waiting - это ещё не активный воркер



показать UI
об обновлении, если
нужно обновить SW



тихое обновление
SW (моргание экрана
для пользователя)

Жизненный цикл Service Worker

activating - идеальное место для чистки и миграций



удаление старого
кэша
миграции данных



держат длинные
операции

Жизненный цикл Service Worker

activated - воркер становится текущим «контролёром» для страниц в своём scope и начинает получать функциональные события: `fetch`, `push`, `sync`, `message`



определение стратегии кэширования

`cache-first`, `network-first`, `stale-while-revalidate`, `offline fallback`



кэшировать все подряд без стратегии

Service Worker

Cache-first - сначала берём ответ из кэша и только если его нет — идём в сеть

Network-first - сначала пробуем сеть и кладём успешный ответ в кэш, сеть недоступна - отдаём последний кэш

Stale-while-revalidate - сразу отдаём кэш, параллельно делаем сетевой запрос и обновляем данные

Offline fallback - сначала network-first или cache-first, а если сеть и кэш не дали ответа, подсовываем заранее подготовленную заглушку

Предназначение *service worker*

Скорость

Устойчивость

Offline

Уведомления

Проблемы *service worker*

Залипание
кэша

Битые
offline-сценарии

Сложность
дебага

Безопасность

HTTPS – комбинация двух механизмов

- **Шифрование TLS (Transport Layer Security)**

шифрование трафика между клиентом и сервером

- **Аутентификация (SSL-сертификат)**

сервер доказывает свою подлинность клиенту

HTTPS

Да не, я на своем iPhone и без всяких HTTPS прилу установил!

Без HTTPS иконка на экране вашего телефона будет просто ссылкой на веб-страницу и не более, хоть и похоже на приложение.

Кстати, если хотите видеть `display: minimal-ui`, то это оно



HTTPS



- WebAuth
- Push
- Геолокация
- Bluetooth

с учетом ограничений на iOS

**Без HTTPS ваш кэш, куки, токены,
localStorage занимают очередь
для атаки злоумышленников**

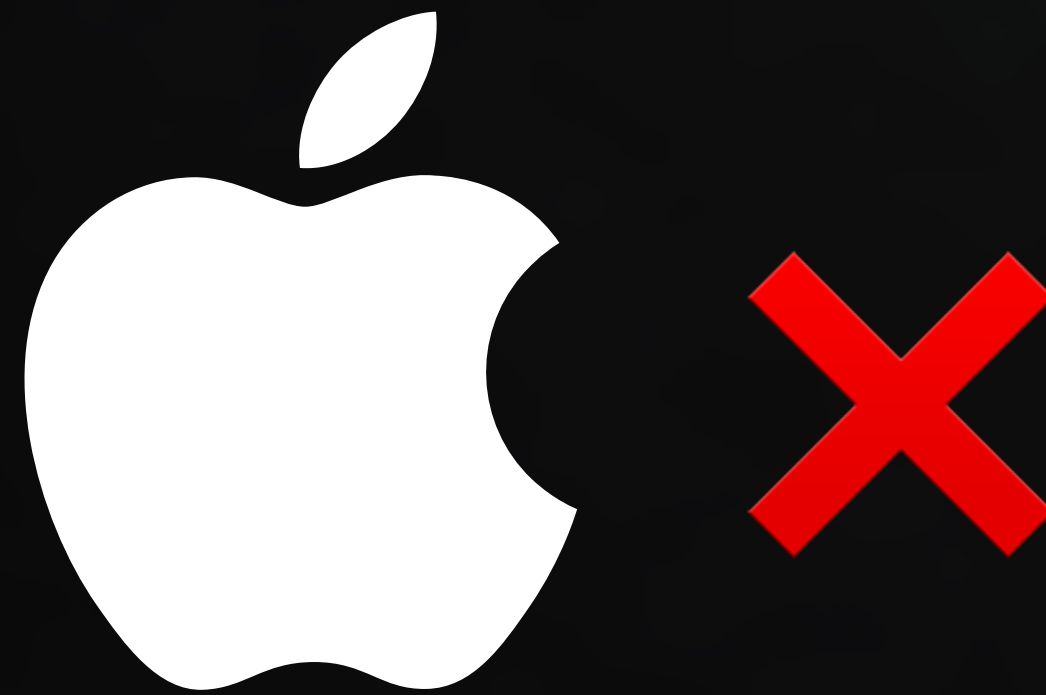
Secure context

localhost — поблажка для разработки

install-prompt



Manifest
HTTPS
Показ prompt без
действий
пользователя



Вобщe не отобразится,
пользователь должен сам
установить приложение
Без SW, manifest и HTTPS
иконка - просто закладка
браузера

Доступность API на платформах

	Android	iOS
Камера/микрофон	✓	✓
Геолокация	✓	✓ ✗
Оффлайн	✓	✓
Хранение	✓	✓ ✗
Web Push	✓	✓
Background Sync	✓	✗
Bluetooth/NFC	✓	✗
Deep link	✓	✓ ✗

Биометрия WebAuthn

Приложение
само
предложит



Только если
пользователь
совершит
действие



Хранение данных

Всё доступно,
можно
запрашивать
устойчивое
хранение



Local storage, indexedDB,
кэш, куки, но с
ограничения из-за
Intelligent Tracking
Prevention в Safari



Web Push

Отлично
работают
уже давно



Появились только с версии ios 16.4 и работают только с явного согласия пользователя. Ограничения по количеству

Нет возможности проверить есть ли разрешение пользователя. Re-subscribe на каждом старте. После долгого неиспользования пропадает разрешение



Deep links

Работают без
нареканий



Доступны только из пушей

В PWA не перейти по
ссылке из СМС или по
ссылке с сайта из-за
контейнеров



Web Share

Можно делиться
из PWA

Можно
зарегистрировать
PWA как target
share



Можно делиться
из PWA

Нельзя
зарегистрировать
PWA как target
share



manifest

HTTPS

SW

PWA

СПАСИБО ЗА ВНИМАНИЕ!

Контакты:

Константин Шкурко
Россельхозбанк