

УДК 004.4

**КРОССПЛАТФОРМЕННЫЕ ПРОГРЕССИВНЫЕ ВЕБ-ПРИЛОЖЕНИЯ,  
СРАВНЕНИЕ С НАТИВНЫМИ ПРИЛОЖЕНИЯМИ****Мишагин Дмитрий Владимирович**

магистрант

Национальный исследовательский университет ИТМО

Санкт-Петербург, Российская Федерация

+79195759358

[mi5ha6in@gmail.com](mailto:mi5ha6in@gmail.com)**Аннотация**

Статья посвящена исследованию применения прогрессивных веб-приложений (PWA), описанию их достоинств по сравнению с нативными приложениями. Прогрессивные веб-приложения позволяют разрабатывать веб-приложения, поддерживающие работу без доступа к сети интернет, позволяют загружать, устанавливать и использовать их в автономном режиме на различных системах, включая мобильные устройства и персональные компьютеры. Имеют доступ к API платформы, на которой запущены, что позволяет им приблизиться к функционалу нативных приложений. Статья отвечает на вопрос могут ли прогрессивные веб-приложения стать лучшей альтернативой традиционной разработке мобильных и десктопных приложений. Проведен сравнительный анализ и тестирование производительности каждого подхода. На основании проведенного сравнения и тестирования приведены рекомендации, которые позволяют определиться с выбором технологий в различных командах инженеров.

**Ключевые слова:** прогрессивные веб-приложения, кроссплатформенные приложения, веб-технологии, service worker, нативные приложения, push-нотификация.

**CROSS-PLATFORM PROGRESSIVE WEB APPLICATIONS, COMPARISON  
WITH NATIVE APPLICATIONS****Dmitry V. Mishagin**

undergraduate

National Research University ITMO

St. Petersburg, Russian Federation

+79195759358

[mi5ha6in@gmail.com](mailto:mi5ha6in@gmail.com)

---

**ABSTRACT**

---

The article is devoted to the study of the use of progressive web applications (PWA), a description of their advantages compared to native applications. Progressive web applications allow you to develop web applications that support work without access to the Internet, allow you to download, install and use them offline on various systems, including mobile devices and personal computers. They have access to the API of the platform on which they are running, which allows them to get closer to the functionality of native applications. The article answers the question of whether progressive web applications can be the best alternative to the traditional development of mobile and desktop applications. A comparative analysis and performance testing of each approach was carried out. Based on the comparison and testing, recommendations are given that allow you to decide on the choice of technology in various engineering teams.

---

**Keywords:** progressive web applications, cross-platform applications, web technologies, service worker, native applications, push-notification.

---

Нативные приложения состоят из двоичных исполняемых файлов, которые загружаются непосредственно на устройство пользователя и хранятся локально, распространяются через специализированные магазины приложений, такие как Google Play Store для приложений Android и магазин приложений App Store для iOS.

Приложения разрабатываются непосредственно на основе сервисов, предоставляемых базовой мобильной платформой. Эти сервисы предоставляются через специальный интерфейс прикладного программирования (API) с методами, связанными со связью и обменом сообщениями, графикой, местоположением, безопасностью и т.д. Языки программирования и инструменты для собственных мобильных приложений зависят от платформы; например, приложения Android создаются на Java или Kotlin с помощью Android SDK, тогда как приложения Apple iOS разрабатываются с использованием Objective-C или Swift с помощью инструмента XCode.

Благодаря API и инструментам, специфичным для платформы, разработчики могут создавать собственные мобильные приложения с богатым пользовательским интерфейсом, мощной продвинутой графикой и высокой производительностью. Однако использование технологий, специфичных для платформы, приводит к общеизвестной проблеме фрагментации мобильной платформы, поскольку код, написанный для одной мобильной платформы (например, код Java приложения Android), не может использоваться на другой (например, Objective-C-код приложения Apple iOS). Фрагментация делает разработку и обслуживание нативных приложений для нескольких платформ одной из основных технических проблем, затрагивающих сообщество разработчиков.

Стандартные веб-технологии, такие как HTML, CSS, JavaScript, способны помочь в создании приложений через общий согласованный технологический стек, таким образом, уменьшая проблему фрагментации. Прогрессивные веб-приложения стали связующим звеном между разработкой нативных приложений и веб-приложениями [1].

Progressive Web Applications (PWA) разработаны в 2015 году, с 2018 года приложения этого класса поддерживаются всеми основными браузерами [2; 3]. Внешне похожи на нативные приложения. Браузер используется как виртуальная машина, хранящая и запускающая в себе приложение. Как Андроид является виртуальной

машиной для андроид-приложений, так и браузер становится виртуальной машиной для PWA. Как нативное приложение обращается через файловую систему к своим ресурсам, так же и PWA обращается к своим ресурсам по HTTPS хранящимся локально.

Progressive Web Application содержит в себе большой перечень современных технологий, таких как: Service Worker, HTTPS, Application Shell, Web App manifest, Push Notifications.

Service Worker (SW) – это JavaScript файл, содержащий хуки жизненного цикла для бизнес-логики и управления кешем, который запускается в браузере фоновым режимом, отдельно от страницы [4; 5]. SW перехватывает и обрабатывает сетевые запросы, включая механизмы управления кэшированием фоновой синхронизации [6]. Регистрируется при первом посещении пользователя (рис. 1).

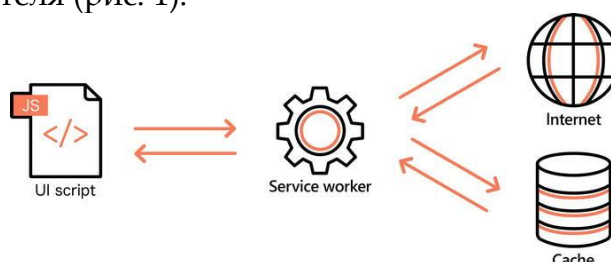


Рис. 1. Пример работы Service Worker [4]

Application Shell – минимальный набор HTML, CSS и JavaScript, который требуется для работы пользовательского интерфейса. Статические ресурсы кэшируются только при первом посещении, после чего они сразу же загружаются при повторном посещении приложения [7].

Технология Push Notifications – способ распространения информации в сети таким образом, что данные поступают от отправителя к получателю на основе установленных параметров [8]. Эту технологию еще называют серверными уведомлениями, поскольку здесь информация формируется на сервере и доставляется клиенту по инициативе сервера. В этом состоит ее отличие от технологии pull, где клиент инициирует запрос новой информации.

Push-уведомления могут появляться на экране любого устройства, если есть возможность вывода на экран данных, принятых из сети Интернет. При этом пользователю не обязательно поддерживать приложение постоянно работающим, чтобы получать уведомления.

HTTPS. Прогрессивные веб-приложения должны обслуживаться через HTTPS по соображениям безопасности [9].

В результате проведенного обзора технологий Progressive Web App и Native App, были выявлены их ключевые отличия, схожие черты и варианты использования. На основе этих закономерностей, для проведения полноценного сравнительного анализа, были определены следующие критерии сравнения:

- возможность работы без установки на платформу;
- возможность работы на любой платформе;
- доступ к аппаратным возможностям платформы;
- возможность работы без интернет-соединения;
- наличие уведомлений;
- фоновая синхронизация;
- тестируемость перед установкой.

*Таблица 1.*  
*Сравнительный анализ технологий Progressive Web App и Native App*

Критерий сравнения	Progressive Web App	Native App
Работа без установки на платформу	Есть	Нет
Кроссплатформенность	Есть	Нет
Доступ к аппаратным возможностям	Ограниченный	Есть
Работа без интернет-соединения	Есть	Есть
Наличие уведомлений	Есть	Есть
Фоновая синхронизация	Есть	Есть
Тестируемость перед установкой	Есть	Нет

Ключевым преимуществом Progressive Web App является кроссплатформенность таких приложений. Разработчику достаточно написать код один раз, и он будет работать на всех устройствах пользователя. У PWA отсутствуют проблемы с обновлением приложения. Для обновления достаточно передать новый HTML, CSS и JS через интернет-соединение в фоновом режиме, что будет не заметно для пользователя и не потребует с его стороны дополнительных действий.

Другим существенным плюсом является возможность работы приложения без его установки. Это повышает конверсию пользователей и удобство использования программного продукта.

Native App имеет такие преимущества как скорость работы и полная поддержка аппаратной платформы, так как приложение изначально создается с учетом особенностей устройства.

Тестирование производительности.

Прогрессивные веб-приложения были сравнены с помощью Performance API, позволяющий веб-страницам измерить производительность. Метки были установлены в коде, один для начала бенчмарка, и один на конец. После завершения тест был сделан расчет общее времени.

Тесты для нативных приложений Android были выполнены с помощью библиотеки Guava, созданной Google, в частности их секундомер учебный класс. Пример приложения, обращающегося к камере, запускается, когда пользователь нажал кнопку в приложении. Когда приложение получает доступ к камере, начинаются тесты. Когда камера успешно открылась, приложение сделает снимок и сохранит его. Когда картинки будут успешно сохранены тесты остановятся.

Для проведения полноценного сравнительного анализа, были определены следующие критерии сравнения:

- время отклика камеры в миллисекундах;
- время отклика геолокации в миллисекундах;
- размер установочного файла;
- число кадров в секунду;
- нагрузка на процессор в процентах;

- время отклика блютуз в миллисекундах;
- расход заряда аккумулятора;
- общая скорость работы.

*Таблица 2.*  
*Тестирование производительности Progressive Web App и Native App*

Критерий сравнения	Progressive Web App	Native App
Время отклика камеры, ms	6136	770
Время отклика геолокации, ms	507	708
Размер установочного файла	104KB	4.37MB
Число кадров в секунду	60	60
Нагрузка на процессор в процентах	62%	41%
Время отклика блютуз, ms	1255	528
Расход заряда аккумулятора	Средний	Низкий
Общая скорость работы	Средняя	Высокая

Результаты иллюстрируют наличие определенных компромиссов. Если минимальный размер приложения имеет высший приоритет, то выбор за PWA с результатом 104 КБ против 4 мегабайт у Native App.

По двум критериям, а именно: время отклика геолокации и число кадров в секунду, результаты обоих подходов сопоставимы.

Если производительность и скорость доступа к аппаратной части имеет наивысший приоритет, то выбор стоит за Native App. Он показывает лучшие результаты в следующих критериях, отвечающих за быстродействие: время отклика камеры, нагрузка на процессор в процентах, время отклика блютуз, расход заряда аккумулятора, общая скорость работы.

Из ходя из вышеперечисленного можно сделать вывод, что Progressive Web App следуют использовать, когда есть необходимость в кроссплатформенности приложения и нет команды разработчиков под каждый тип устройств, а Native App – если критична скорость работы и полный доступ к аппаратным возможностям платформы.

### Список литературы

1. Золотарёв, С.П. Развитие web - технологий [Электронный ресурс] / С.П. Золотарёв – Вологдинские чтения – 2019. – № 73. – с. 16-18. – Электрон. текстовые дан. – Режим доступа: <https://elibrary.ru/item.asp?id=14869698> (дата обращения: 3.03.2020). – Загл. с экрана.
2. PWA – это просто [Электронный ресурс] – Электрон. дан. – Режим доступа: <https://habr.com/ru/post/418923/> (дата обращения: 3.03.2020).
3. Ильясов, Г.Э. Обзор технологии создания прогрессивного Веб-приложения [Электронный ресурс] / Г.Э. Ильясов – International scientific review of the

- problems and prospects of modern science and education - 2018. - с. 40-42. - Электрон. текстовые дан. - Режим доступа: <https://elibrary.ru/item.asp?id=35363747> (дата обращения: 8.03.2020). - Загл. с экрана.
4. Sheppard, D. Beginning Progressive Web App Development: Creating a Native App Experience on the Web [Текст] / D. Sheppard - Apress, 2017. - С. 266.
  5. Service Workers: an Introduction [Электронный ресурс] - Электрон. дан. - Режим доступа: <https://developers.google.com/web/fundamentals/primers/service-workers/> (дата обращения: 9.03.2020).
  6. Service Workers. Инструкция по применению [Электронный ресурс] - Электрон. дан. - Режим доступа: <https://habr.com/ru/company/2gis/blog/345552/> (дата обращения: 9.03.2020).
  7. The App Shell Model [Электронный ресурс] - Электрон. дан. - Режим доступа: <https://developers.google.com/web/fundamentals/architecture/app-shell> (дата обращения: 10.03.2020).
  8. Web Push Notifications: Timely, Relevant, and Precise [Электронный ресурс] - Электрон. дан. - Режим доступа: <https://developers.google.com/web/fundamentals/push-notifications/> (дата обращения: 15.03.2020).
  9. Why HTTPS Matters [Электронный ресурс] - Электрон. дан. - Режим доступа: <https://developers.google.com/web/fundamentals/security/encrypt-in-transit/why-https> (дата обращения: 3.04.2020).

### References

1. Zolotarev, S.P. Development of web - technologies [Electronic resource] / S.P. Zolotaryov - Vologda Readings - 2019. - No. 73. - p. 16-18. - The electron. text data - Access mode: <https://elibrary.ru/item.asp?id=14869698> (accessed date: 03/03/2020). - Zagl. from the screen [in Russian].
2. PWA is just [Electronic resource] - Electron. Dan. - Access mode: <https://habr.com/en/post/418923/> (accessed date: 03/03/2020) [in Russian].
3. Ilyasov, G.E. Overview of the technology for creating a progressive Web application [Electronic resource] / G.E. Ilyasov - International scientific review of the problems and prospects of modern science and education - 2018. -- p. 40-42. - The electron. text data - Access mode: <https://elibrary.ru/item.asp?id=35363747> (accessed date: 8.03.2020). - Zagl. from the screen [in Russian].
4. Sheppard, D. Beginning Progressive Web App Development: Creating a Native App Experience on the Web [Text] / D. Sheppard - Apress, 2017. - P. 266.
5. Service Workers: an Introduction [Electronic resource] - Electron. Dan. - Access mode: <https://developers.google.com/web/fundamentals/primers/service-workers/> (accessed date: 9.03.2020)
6. Service Workers. Instructions for use [Electronic resource] - Electron. Dan. - Access mode: <https://habr.com/en/company/2gis/blog/345552/> (accessed: 9.03.2020)
7. The App Shell Model [Electronic resource] - Electron. Dan. - Access mode: <https://developers.google.com/web/fundamentals/architecture/app-shell> (accessed March 10, 2020)

8. Web Push Notifications: Timely, Relevant, and Precise [Electronic resource] - Electron. Dan. - Access mode: <https://developers.google.com/web/fundamentals/push-notifications/> (accessed date: 03/15/2020)
9. Why HTTPS Matters [Electronic resource] - Electron. Dan. - Access mode: <https://developers.google.com/web/fundamentals/security/encrypt-in-transit/why-https> (accessed date: 3.04.2020)