Прогрессивные веб-приложения

Прогрессивное веб-приложение (PWA) — это приложение, созданное с использованием технологий веб-платформы, но обеспечивающее взаимодействие с пользователем, аналогичное приложению для конкретной платформы.

Как и веб-сайт, PWA может работать на нескольких платформах и устройствах из одной кодовой базы. Как и приложение для конкретной платформы, его можно установить на устройство, работать в автономном режиме и в фоновом режиме, а также интегрировать с устройством и с другими установленными приложениями.

Гиды Эти руководства дают концептуальные объяснения различных аспектов PWA. Они предназначены для того, чтобы помочь вам понять, какие вещи возможны с PWA, и предоставить достаточно указателей, которые помогут вам понять, как их достичь.

Что такое прогрессивное веб-приложение? Введение в PWA, сравнение их с традиционными веб-сайтами и приложениями для конкретных платформ и описание их основных функций.

Возможность установки PWA Одним из определяющих аспектов PWA является то, что его можно установить на устройство, а затем представить пользователям как приложение для конкретной платформы, постоянную функцию их устройства, которую они могут запускать непосредственно из операционной системы, как и любое другое приложение. В этом руководстве мы рассмотрим, что означает «устанавливаемый», что должен предоставить PWA, чтобы его можно было установить, и как вы можете настроить процесс установки.

Установка и удаление веб-приложений В этом руководстве рассказывается, как пользователи могут устанавливать и удалять PWA на своих устройствах.

Офлайн и фоновая работа В этом руководстве мы представим набор технологий, которые позволяют PWA обеспечивать удобство работы пользователя, даже если устройство имеет прерывистое подключение к сети, и выполнять операции в фоновом режиме, даже когда основное приложение не запущено.

Кэширование Обзор API-интерфейсов, позволяющих PWA кэшировать ресурсы локально, и некоторых распространенных стратегий, используемых PWA для реализации автономных функций.

Лучшие практики для PWA PWA должны адаптироваться к различным браузерам и устройствам, быть доступными, иметь хорошую производительность и хорошо интегрироваться с операционной системой. В этом руководстве представлен список рекомендаций, которые помогут вам убедиться, что ваше PWA настолько хорошо, насколько это возможно.

Как Эти руководства содержат конкретные подробные инструкции о том, как реализовать определенные функции PWA.

Создать отдельное приложение Описывает, как указать, что PWA должен запускаться в отдельном выделенном окне при запуске, а не на вкладке браузера.

Определите значки вашего приложения Описывает, как определить собственный набор значков, которые будут использоваться при установке PWA на устройстве.

Настройте цвета вашего приложения Описывает, как установить цвета фона и темы для PWA.

Показать значки Описывает, как отобразить значок на значке PWA: например, чтобы пользователь знал, что он получил новые сообщения.

Предоставьте общие действия приложения в виде ярлыков Описывает, как предоставить общие действия для PWA, которые можно запустить из контекстного меню приложения операционной системы.

Делитесь данными между приложениями Описывает, как PWA могут обмениваться данными друг с другом с помощью механизма совместного использования приложений операционной системы.

Инициировать установку из PWA Описывает, как разработчики могут предоставить собственный пользовательский интерфейс, чтобы пригласить пользователей установить их PWA.

Свяжите файлы с вашим PWA Описывает, как вы можете создать связь между типами файлов и вашим PWA, чтобы, когда пользователь щелкает файл, запускалось ваше PWA для его обработки.

Учебники В этих руководствах вы создадите PWA с нуля. Учебники проходят этапы создания приложения от начала до конца, объясняя, как реализованы различные функции приложения.

Создание вашего первого PWA В этом учебном пособии для начинающих рассказывается о создании PWA для отслеживания менструальных циклов. Уроки включают в себя знакомство с HTML, CSS и JavaScript, необходимыми для создания полнофункционального веб-приложения, настройку среды тестирования и полные объяснения, помогающие учащемуся обновить веб-приложение до PWA; включая разработку и проверку манифеста, добавление работника службы и использование работника службы для удаления устаревших кешей.

Глубокое погружение в PWA В этом учебном пособии среднего уровня рассматривается создание PWA, в котором содержится информация об играх, представленных в категории A-Frame на конкурсе js13kGames 2017 . Это руководство включает в себя все основы создания PWA с дополнительными функциями, включая уведомления, push-уведомления и производительность приложения.

Ссылка Справочная документация по веб-технологиям, которые вы будете использовать для создания PWA.

Манифест веб-приложения Члены манифеста веб-приложения Разработчики могут использовать элементы манифеста веб-приложения для описания PWA, настройки его внешнего вида и более глубокой интеграции в операционную систему.

API сервис-воркеров Связь с приложением Следующие API-интерфейсы могут использоваться сервис-воркером для связи со связанным с ним клиентским PWA:

Client.postMessage() Позволяет сервисному работнику отправлять сообщение своему клиентскому PWA.

API широковещательного канала Позволяет сервисному работнику и его клиентскому PWA установить базовый двусторонний канал связи.

Автономная работа Сервисный работник может использовать следующие API, чтобы ваше приложение работало в автономном режиме:

Cache Механизм постоянного хранения для ответов HTTP, используемый для хранения ресурсов, которые можно повторно использовать, когда приложение находится в автономном режиме.

Clients Интерфейс, используемый для предоставления доступа к документам, которые контролируются сервис-воркером.

FetchEvent Событие, отправляемое в сервис-воркере с каждым HTTP-запросом, сделанным клиентским PWA. Событие можно использовать либо для отправки запроса на сервер в обычном режиме и сохранения ответа для использования в будущем, либо для перехвата запроса и немедленного ответа ранее кэшированным ответом.

Фоновая операция Следующие API-интерфейсы могут использоваться сервис-воркером для выполнения задач в фоновом режиме, даже если ваше приложение не запущено:

API фоновой синхронизации Способ отложить выполнение задач в сервис-воркере при стабильном сетевом соединении.

API периодической веб-синхронизации в фоновом режиме Способ регистрации задач, которые будут запускаться в сервис-воркере через определенные промежутки времени с сетевым подключением.

API фоновой выборки Метод, с помощью которого работник службы может управлять загрузками, которые могут занимать значительное время, например видео- или аудиофайлы.

Другие веб-API ИндекседБД API-интерфейс хранилища на стороне клиента для значительных объемов структурированных данных, включая файлы.

API бейджей Метод установки значка на значке приложения, обеспечивающий уведомление с низким уровнем отвлечения внимания.

API уведомлений Способ отправки уведомлений, которые отображаются на уровне операционной системы.

API веб-ресурсов Механизм обмена текстом, ссылками, файлами и другим содержимым с другими приложениями, выбранными пользователем на своем устройстве.

API наложения оконных элементов управления API для PWA, установленных в настольных операционных системах, который позволяет скрывать строку заголовка окна по умолчанию, позволяя отображать приложение по всей поверхности окна приложения.

From:
https://wwoss.ru/ - **worldwide open-source software**

Permanent link:
https://wwoss.ru/doku.php?id=software:development:web:docs:web:progressive_web_apps:progressive_web_apps

Last update: **2023/08/21 19:56**