PWA

PWA (Progressive Web Application, Прогрессивное web-приложение) - это веб-приложение с характеристиками мобильного приложения. Приложения также запускается в браузере, но браузер пустой, без тулбаров и разных менюшек.

С помощью pwa можно:

- создать иконку приложения на рабочем столе устройства
- достучаться до аппаратуры
- отправлять push уведомления
- работать в оффлайн

Подробнее про pwa можно почитать здесь

Создание PWA

Для этого базово нужно 2 шага:

- иметь manifest.json
- и зарегистрированный service worker, который умеет кешировать запросы, то есть приложение может работать в оффлайн

manifest.json

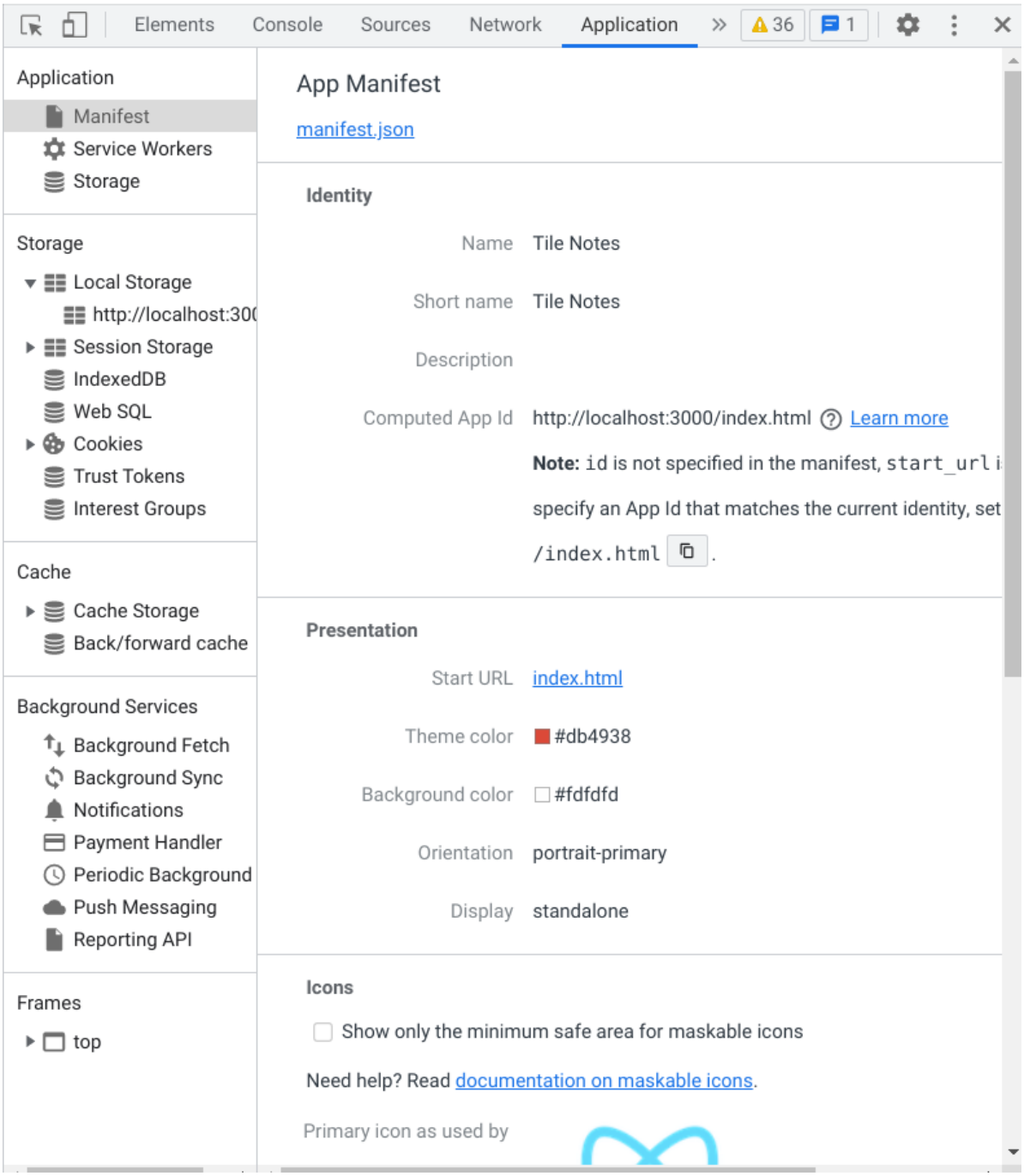
Начнем с manifest.json:

```
{
  "name": "Tile Notes",
  "short_name": "Tile Notes",
  "start_url": "/",
  "display": "standalone",
  "background_color": "#fdfdfd",
  "theme_color": "#db4938",
  "orientation": "portrait-primary",
  "icons": [
    {
      "src": "/logo192.png",
      "type": "image/png", "sizes": "192x192"
    },
    {
      "src": "/logo512.png",
      "type": "image/png", "sizes": "512x512"
    }
  ]
}
```

Вот так он выглядит. Зачем он нужен? Он нужен, чтобы сказать браузеру, что наше приложение pwa, и задать некоторые настройки: название приложения - оно будет появляться на рабочем столе телефона, start_url - какую страницу браузеру запустить при старте, иконки, фоновый цвет и еще парочка опций.

Этот файлик должен быть доступен по пути /manifest.json относительно корня, то есть url примерно такой http://localhost:3000/manifest.json. Если мы используем react, то кладем данный файлик в public директорию, которая находится в корне проекта. Туда же кладем иконки.

Проверить то, что файлик корректно подтянулся можно так:



В DevTools (инструменты разработчика, Ctrl+Shift+I, браузер Chrome). Заходим во вкладочку Application, там должно быть что-то похожее как на скрине, без ворнингов и ошибок.

Service Worker

Что это такое? Это скрипт, который выполняется в фоне, в отдельном потоке, то есть отдельно от страницы. Он умеет разные штуки, в частности, можно перехватывать все сетевые запросы и кешировать их.

Регистрируем service worker, делаем это в файле `index.js` после рендера корневого компонента:

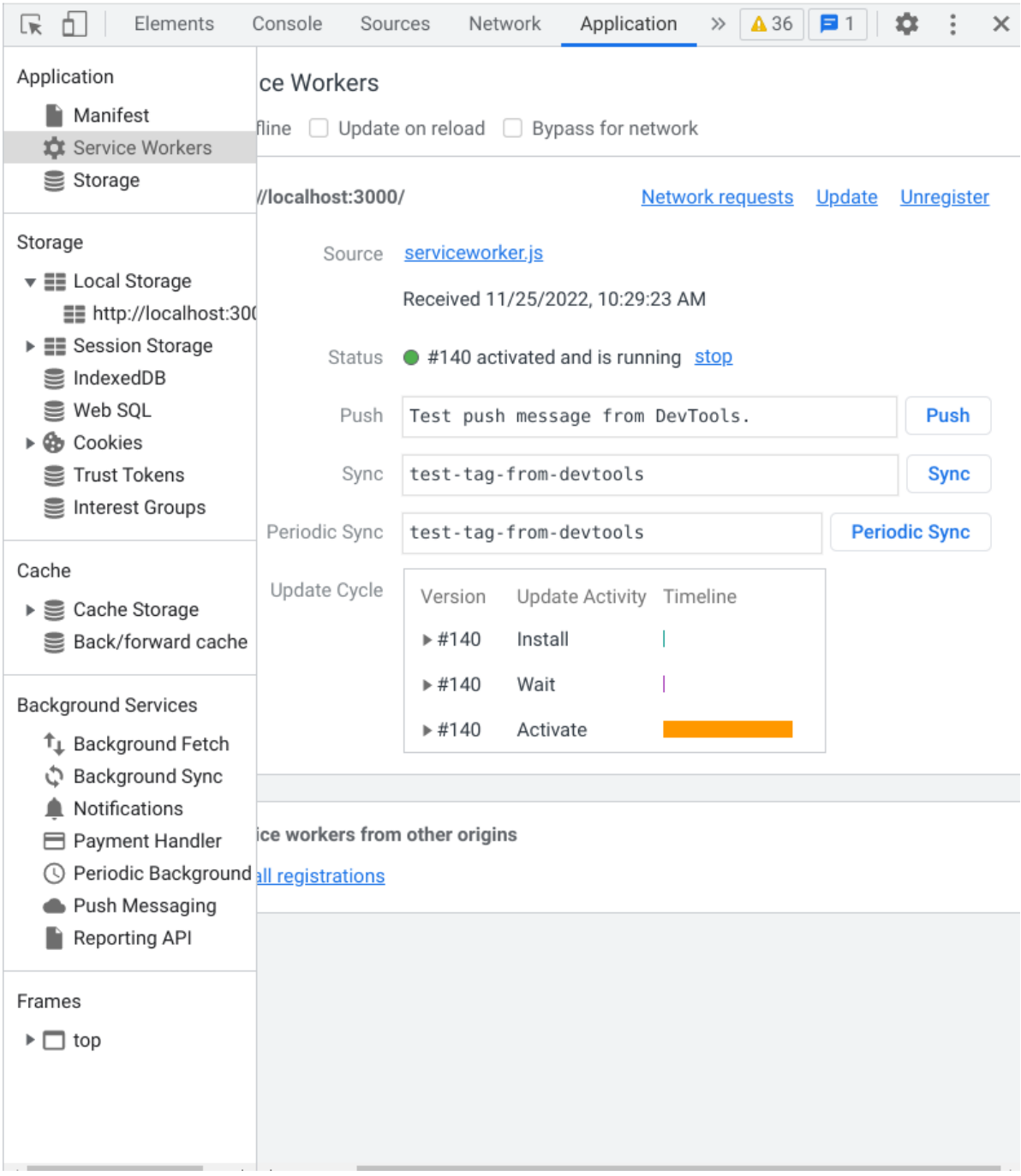
```
if ("serviceWorker" in navigator) {
  window.addEventListener("load", function() {
    navigator.serviceWorker
      .register("/serviceWorker.js")
      .then(res => console.log("service worker registered"))
      .catch(err => console.log("service worker not registered", err))
  })
}
```

Создаем файл serviceWorker.js и кладем его в директорию public:

```
self.addEventListener('fetch',() => console.log("fetch"));
```

Здесь мы формально выполняем требования браузера: нужно повесить обработчик на событие fetch. Если хотите поиграться с разными стратегиями кеширования, то можно глянуть тут

Если по пути, что-то пошло не так, то возможно нужно будет перезагрузить service worker:

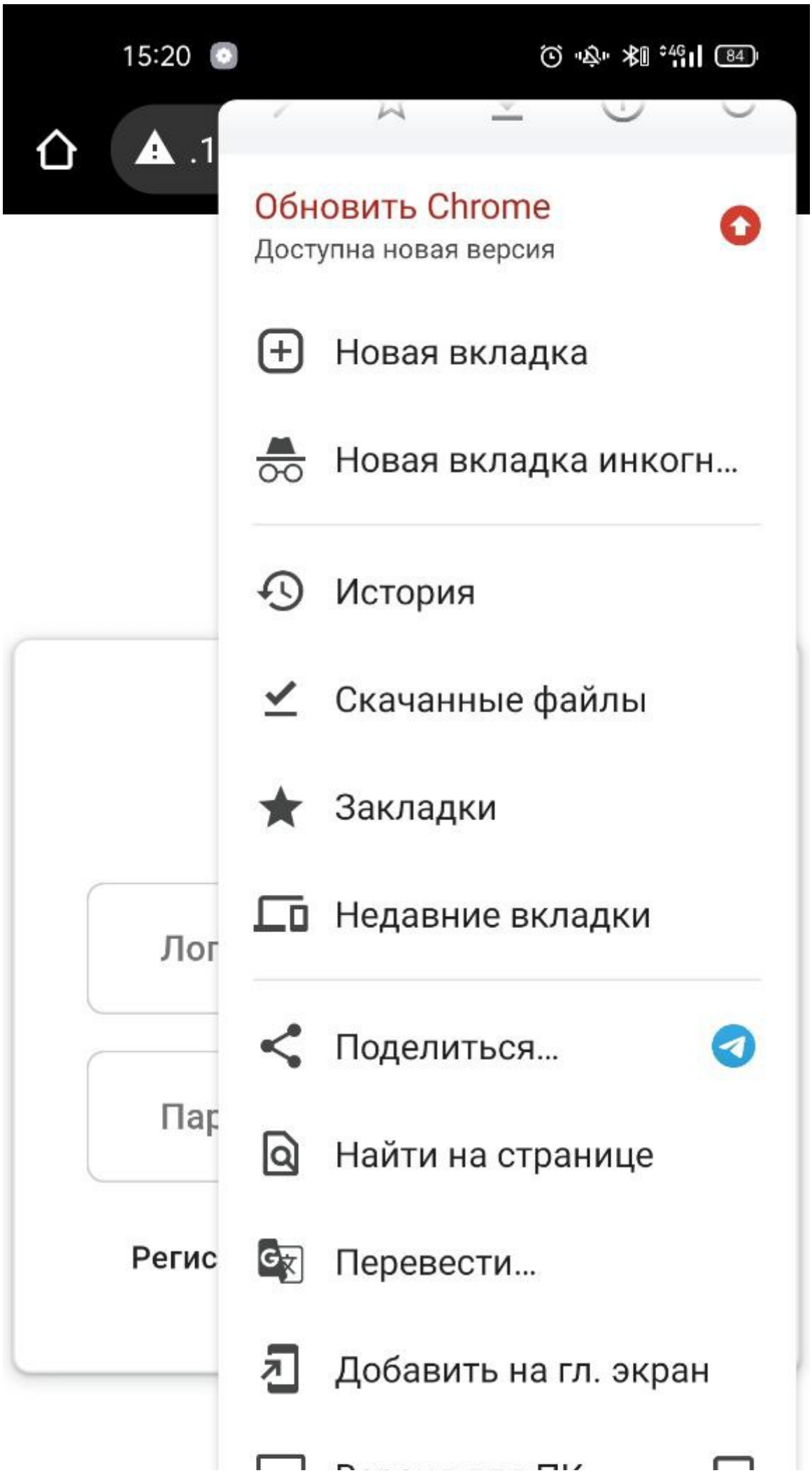


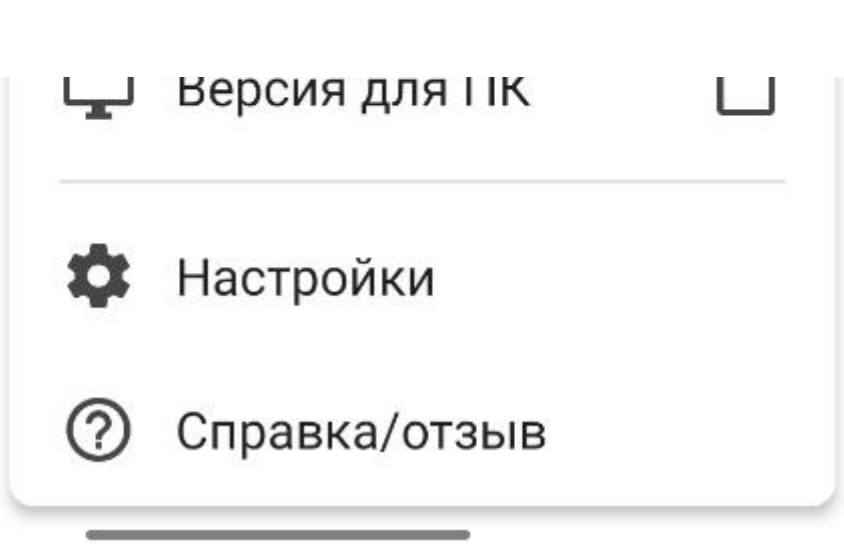
Делается это нажатием кнопочки `Unregister` и перезагрузкой страницы.

Теперь можно проверить, что все работает:

Должна появиться иконочка `Install Apllication` справа в десктопном браузере. Нажимаем на нее, должно открыться отдельное окно с приложением.

Делаем тоже самое на мобилке. Для этого нужно поместить комп и телефон в одну сеть (привет СТ), взять адрес компа (на Linux/Mac можно сделать через `ip addr`) Должно быть что-то похожее на 192.168.199.97. Открыть приложения в мобильном браузере по ссылке http://192.168.199.97:3000. И установить приложение через менюшку:


15:20
84
Обновить Chrome
Доступна новая версия
Новая вкладка
Новая вкладка инкогн...
История
Скачанные файлы
Закладки
Недавние вкладки
Поделиться...
Найти на странице
Перевести...
Добавить на гл. экран



Жмякаем `Добавить на главный экран`. Готово - иконка должна появиться на рабочем столе.

Что еще можно сделать? Можно добавить адаптивность, чтобы приложение было юзабельно на мобилке. Делается это через media queries в css. Здесь можно почитать про них.

Базово в css объявляем `@media (max-width: 480px) {}`. И внутри фигурных скобок пишем стили, они будут применяться, когда разрешение устройства меньше 480px. Обычно телефоны примерно таких размером, можно отдельно сделать для совсем малюшек 320px (iPhone SE). Или для плашнетов 768px.