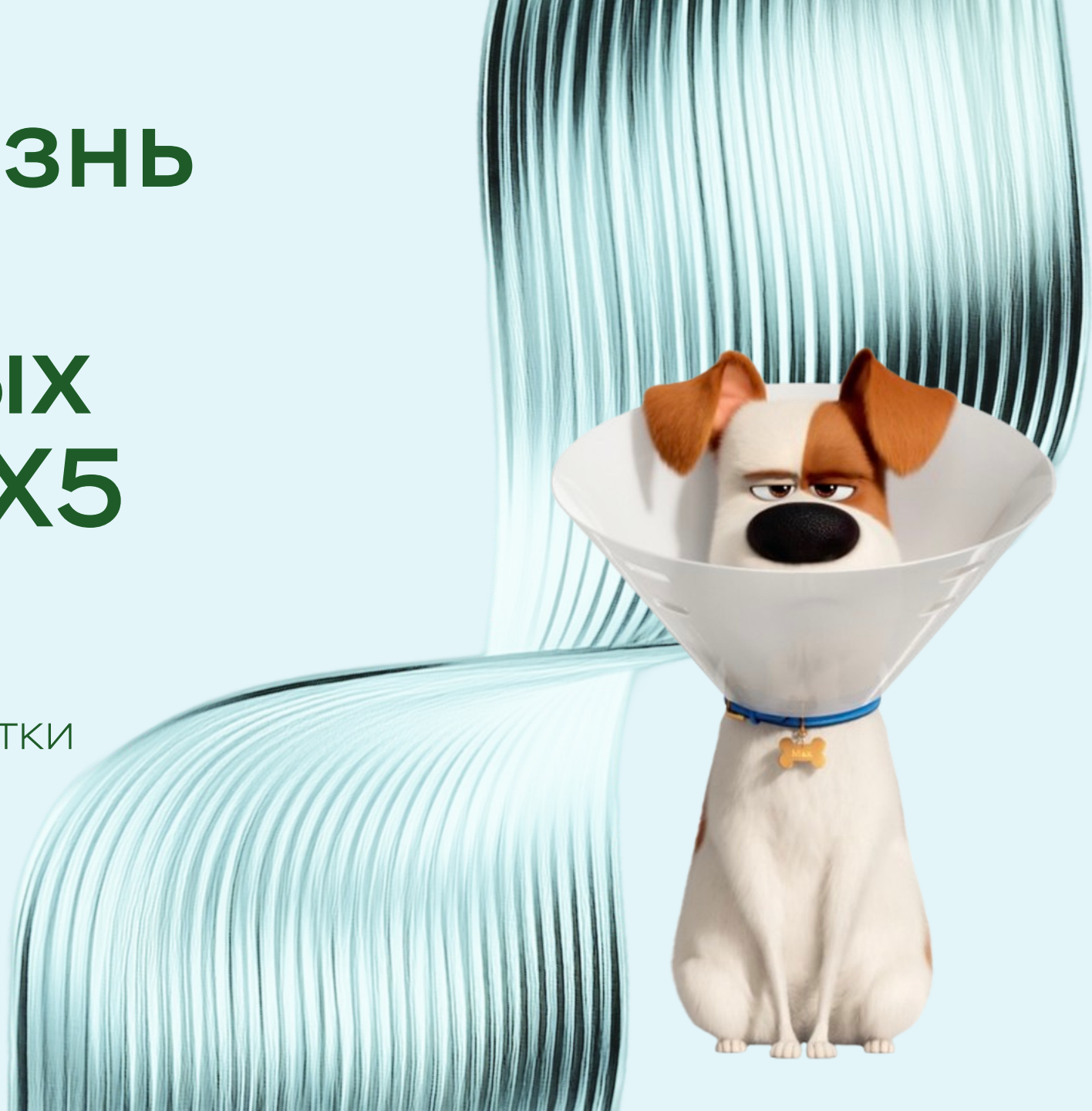


Секретная жизнь фотографий в Клубе Тайных Покупателей X5

Артур Басак

Руководитель команды разработки



Обо мне

- ✧ В Web-разработке 15 + лет
- ✧ Начинал как Java-программист
- ✧ В 2015 к Java прилип Script
- ✧ jQuery → ExtJS → EmberJS → AngularJS → React
- ✧ Не дизайнер, но рисую и занимаюсь иллюстрацией
- ✧ В X5 работаю над Клубом Тайных Покупателей



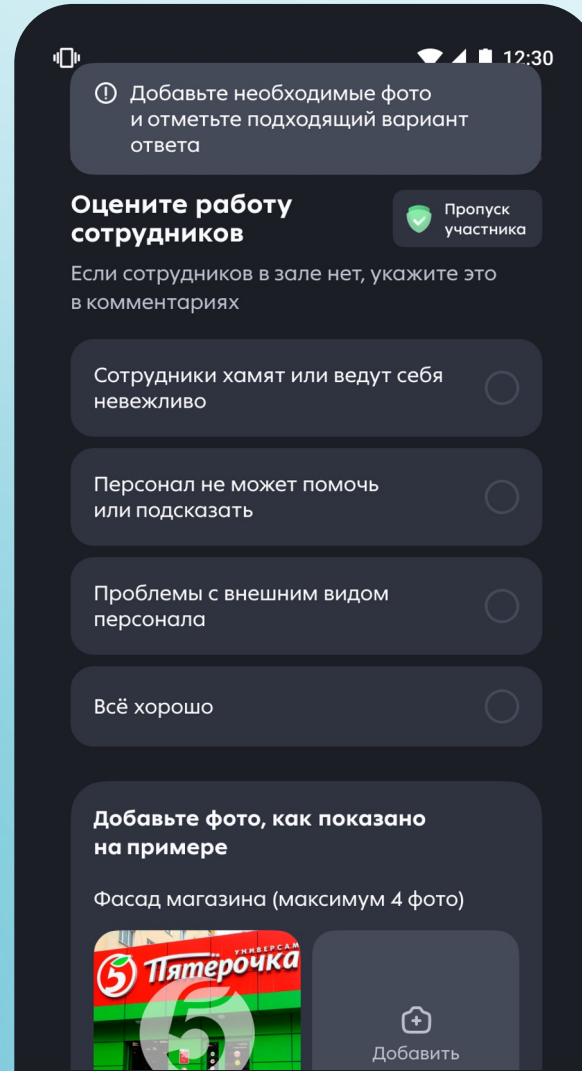
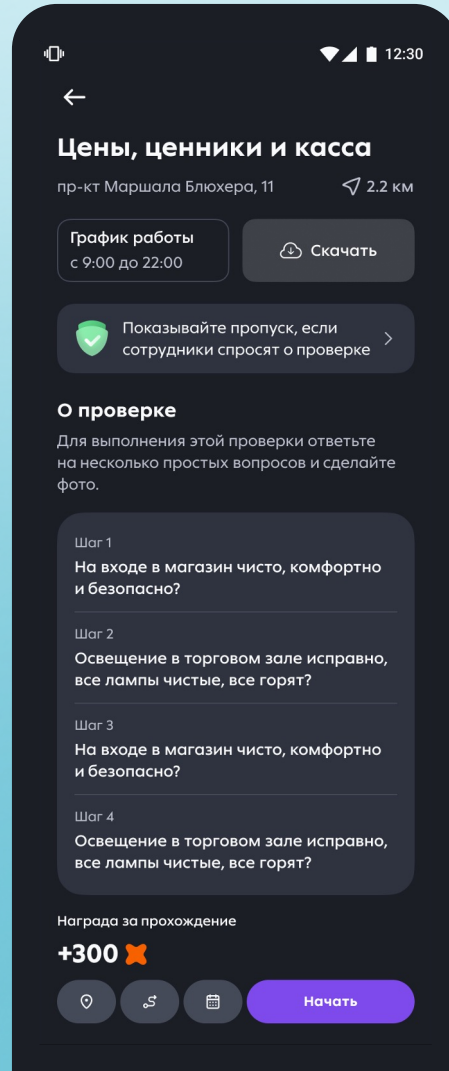
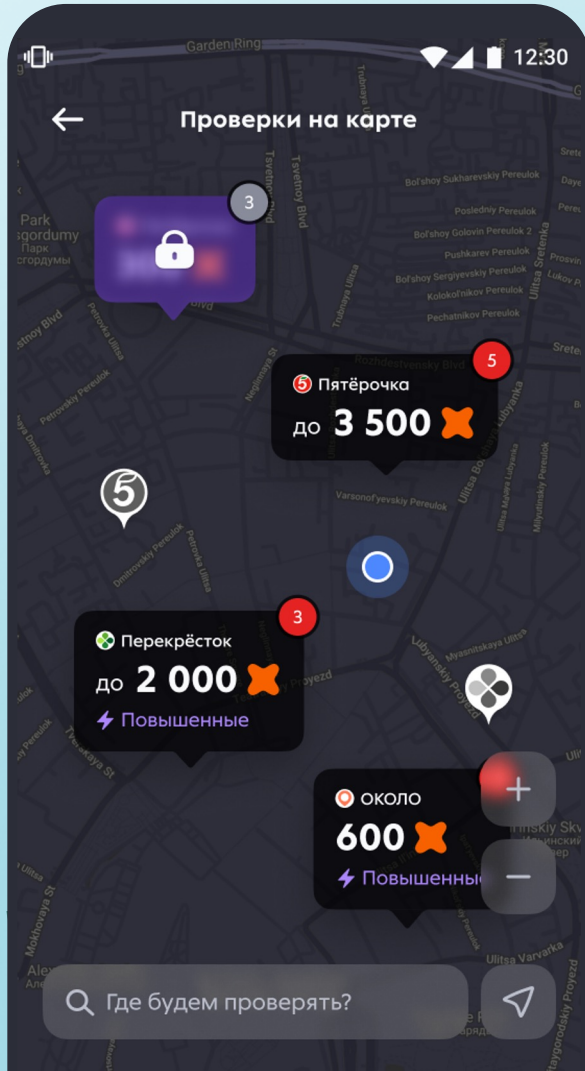
Басак Артур

X5 Tech · Руководитель команды разработки

artu.basak@x5.ru

📍 arturbasak

Клуб тайных покупателей (КТП) в МП Х5



Подход Web-First

Гибрид (Web + iOS/Android)

WebView

+

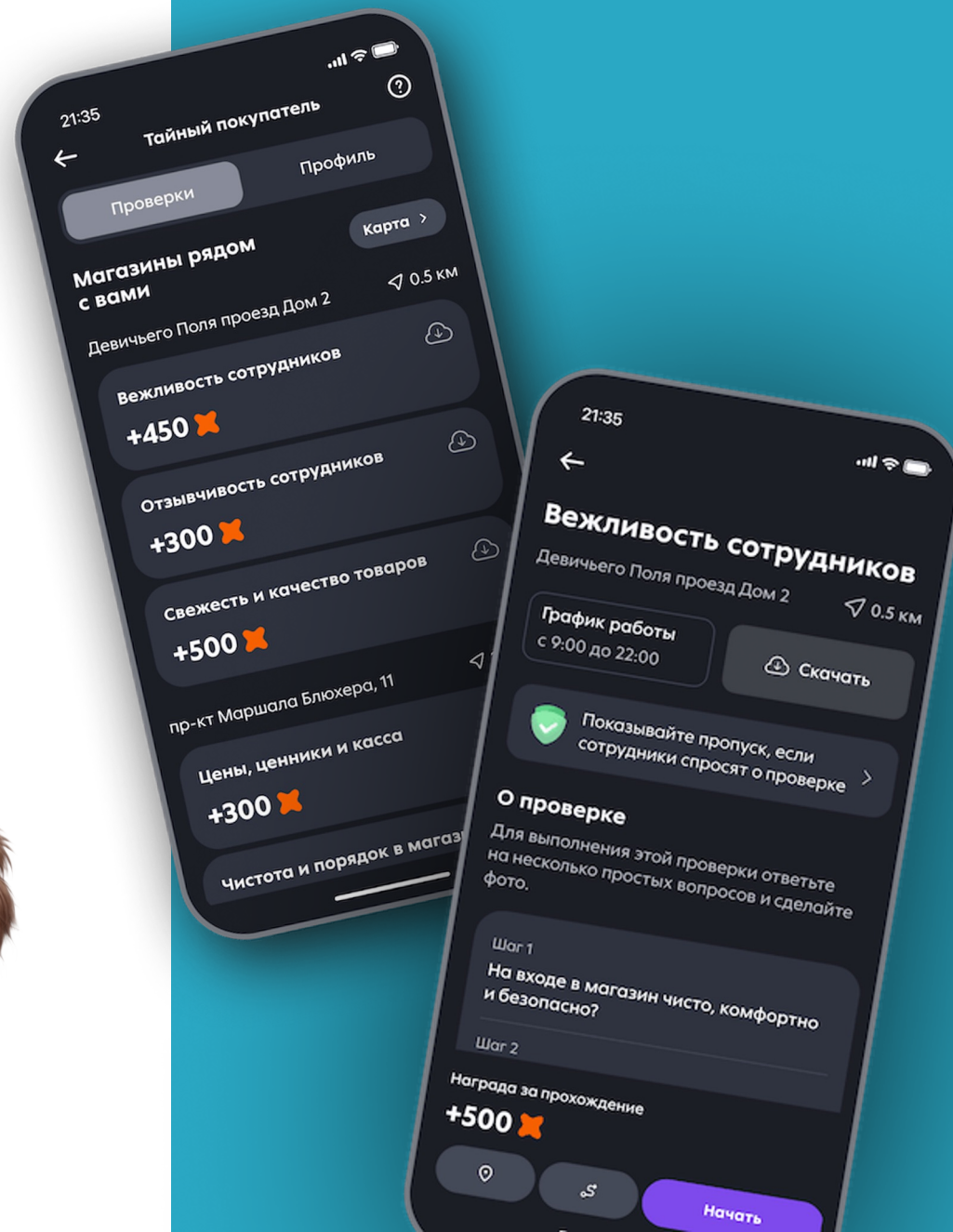
React

+

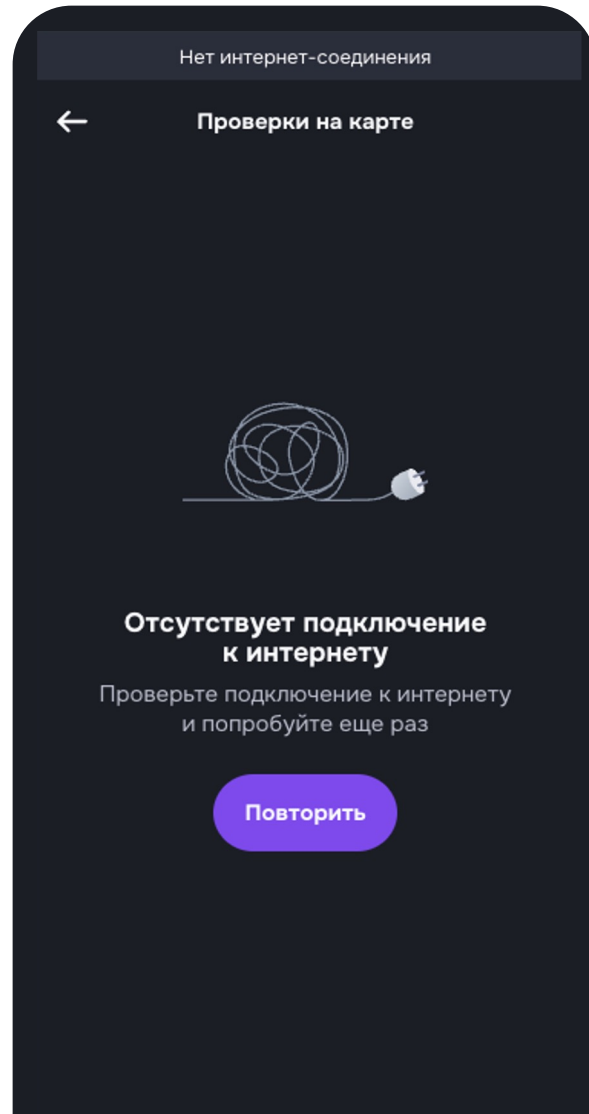
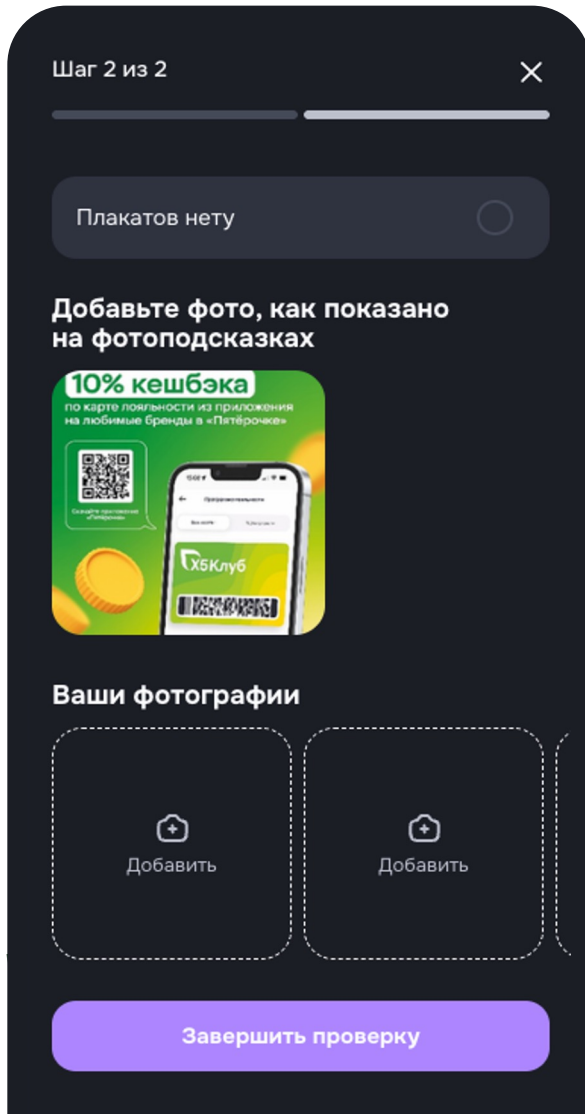
Web API

+

Native
Bridges



Фотографии в КТП — это **ключевой функционал**



Проблема:

необходима надёжная работа с фото в оффлайн-режиме



Фоновая отправка фото при восстановлении соединения

Почему Workbox Background Sync?

- ✧ Готовое решение для фоновых запросов
- ✧ Сохраняет запросы в IndexedDB
- ✧ Есть возможность расширять/переопределять функционал



Workbox

Production-ready service worker libraries and tooling.



Workbox Background Sync



```
import {BackgroundSyncPlugin} from 'workbox-background-sync';
import {registerRoute} from 'workbox-routing';
import {NetworkOnly} from 'workbox-strategies';

const bgSyncPlugin = new BackgroundSyncPlugin('myQueueName', {
  maxRetentionTime: 24 * 60,
});

registerRoute(
  /\api\/.*\/*.json/,
  new NetworkOnly({
    plugins: [bgSyncPlugin],
  }),
  'POST'
);
```

Background Sync API работает не везде

Chrome	Edge *	Safari	Firefox	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mobile *	UC Browser for Android	Android Browser *
109									
112					16.7				
131					17.7				
136	136				18.3				
137	137		139		18.4				
138	138	18.5	140	138	18.5	28	80	15.5	138
139		26.0	141		26.0				
140		TP	142						
141			143						

Chrome Mobile WebView 78.0.3904
Chrome Mobile WebView 74.0.3729
Chrome Mobile WebView 74.0.3729
Chrome Mobile WebView 74.0.3729
Chrome Mobile WebView 72.0.3626
Chrome Mobile WebView 72.0.3626
Chrome Mobile WebView 70.0.3538
Chrome Mobile WebView 70.0.3538
Chrome Mobile WebView 68.0.3440
Chrome Mobile WebView 66.0.3359

Workbox Background Sync

Минусы

- ✗ Поддержка Android/iOS WebView
- ✗ Обновление токена
- ✗ Различные реакции на разные HTTP-статусы
- ✗ Экспоненциальный рост паузы перед повтором запроса



Custom Workbox Background Sync



```
import { Queue } from 'workbox-background-sync';  
// ...  
const queue = new Queue(queueName, {  
  maxRetentionTime,  
  onSync: onQueueSync,  
  // This flag switch workbox-background-sync to  
  // the mode, when BackgroundSync API is not implemented or disabled  
  // ("UnknownError: Background Sync is disabled." use case)  
  forceSyncFallback: true,  
});
```



```
window.addEventListener('online', () => {  
  const sw = navigator?.serviceWorker;  
  sw?.controller?.postMessage({  
    type: FALLBACK_SYNC_EVENT,  
  });  
});
```

Сторона Service Worker



```
self.addEventListener('message', async (event) => {  
  // Fallback for iOS, cuz not support Background Sync API  
  https://caniuse.com/background-sync  
  // and some Android WebView's have "UnknownError: Background Sync is disabled."  
  if (event?.data?.type === FALLBACK_SYNC_EVENT) {  
    await onQueueSync({ queue });  
  }  
  //...
```


Custom Workbox Background Sync

Настройка Workbox Background Sync для совместимости с iOS и Android WebView

🔥 Средний ⌚ 8 мин 👁 1.1K

Тutorial

- <https://habr.com/ru/articles/915896>



Custom Workbox Background Sync

Плюсы

- ✓ Повторы упавших запросов с фото
- ✓ Поддержка Android/iOS WebView
- ✓ Обновление авторизационного токена и работа с актуальным аккаунтом
- ✓ Различные реакции на разные HTTP-статусы
- ✓ Экспоненциальный рост задержки перед повтором запроса

① Будем отправлять ваше задание в фоновом режиме. Следите за статусом в профиле



Всё ещё есть минусы

Минусы

- ✗ Тяжело мониторить ошибки в офлайн на стороне ServiceWorker
- ✗ Фоновая отправка — скудная обратная связь — плохой UX
- ✗ Много неочевидных потерь данных, которые сложно отследить
- ✗ Помимо оффлайна есть ещё просто плохой Интернет (2G)
- ✗ Дублирование данных в IndexedDB →



Неудобный формат данных

The screenshot shows the Chrome DevTools Application tab. On the left, the 'Storage' section is expanded, and 'workbox-background-sync' > 'requests' is selected. A red box highlights the 'queueName' property. The main pane shows a table of requests with one entry. A red box highlights the 'queueName' property in the request's metadata, and another red box highlights the 'body' property, which is an 'ArrayBuffer(688994)'.

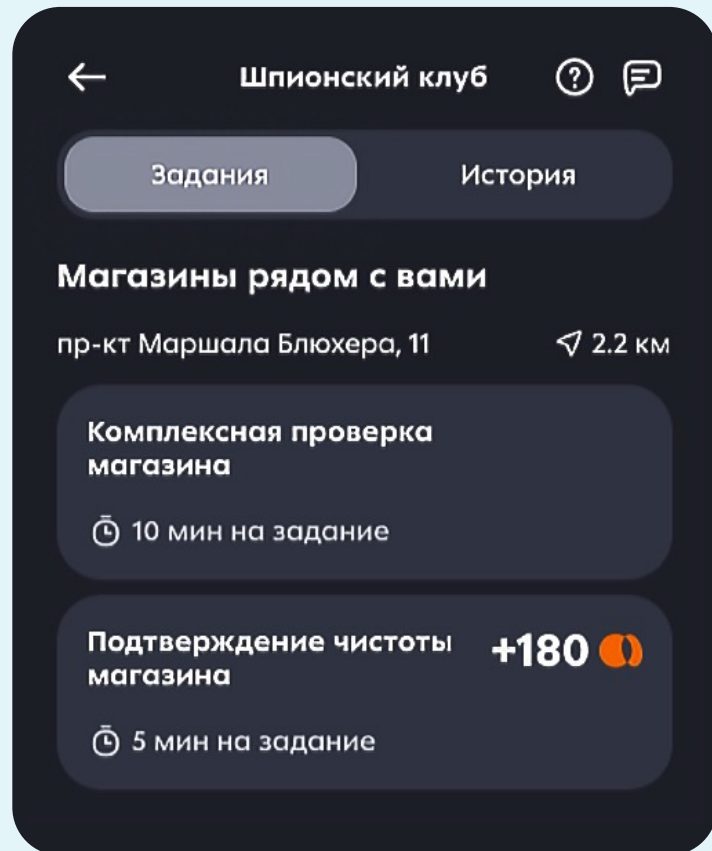
#	Ke...	Value
0	1	<pre>{requestData: {...}, timestamp: 1753084853619, metadata: {...}, id: 1, metadata: {isFromError: false, queueName: "ApiRequests", requestData: {body: ArrayBuffer(688994), cache: "default", credentials: "same-origin", headers: {accept: 'application/json', authorization: 'Bearer ...', integrity: "", keepalive: false, method: "POST", mode: "cors", redirect: "follow", referrer: "http://localhost:8000/", referrerPolicy: "strict-origin-when-cross-origin", url: "https://.../api/mystery-shopper", timestamp: 1753084853619}}</pre>



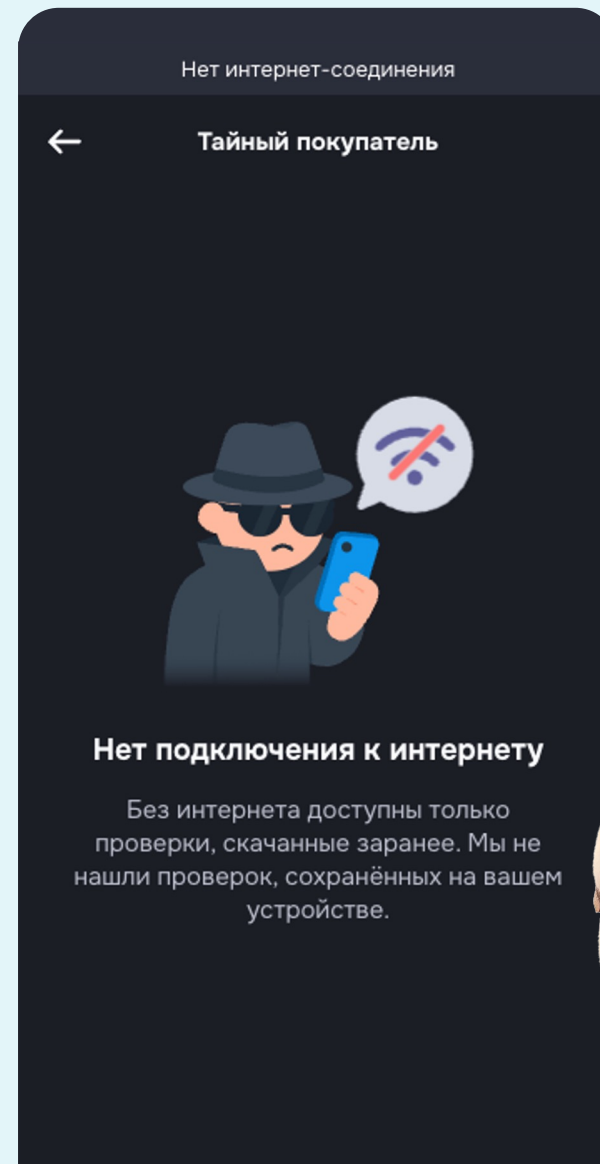
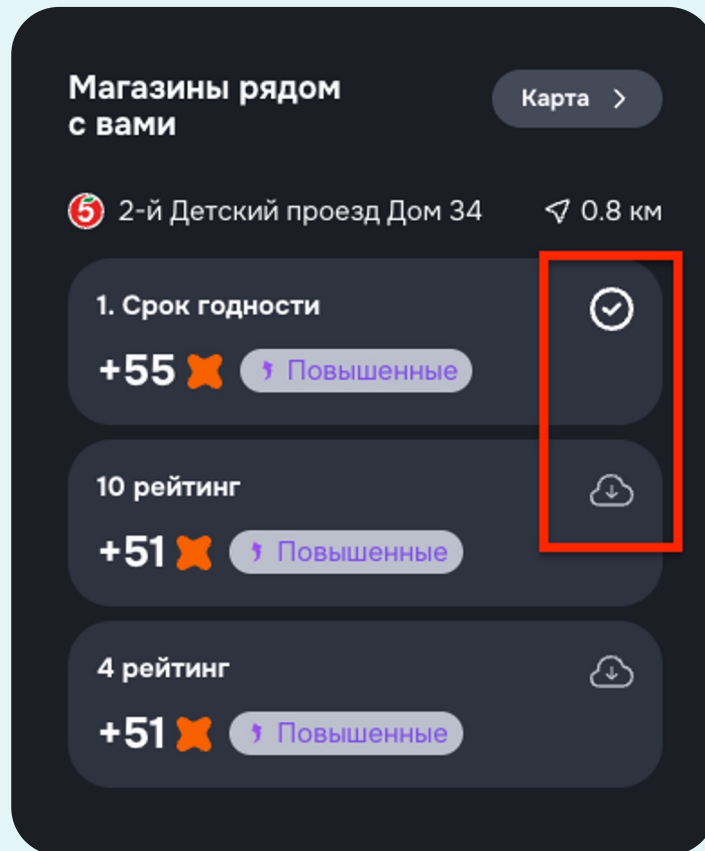
Мы подумали вместе...
и решили!

Предварительное скачивание и Offline

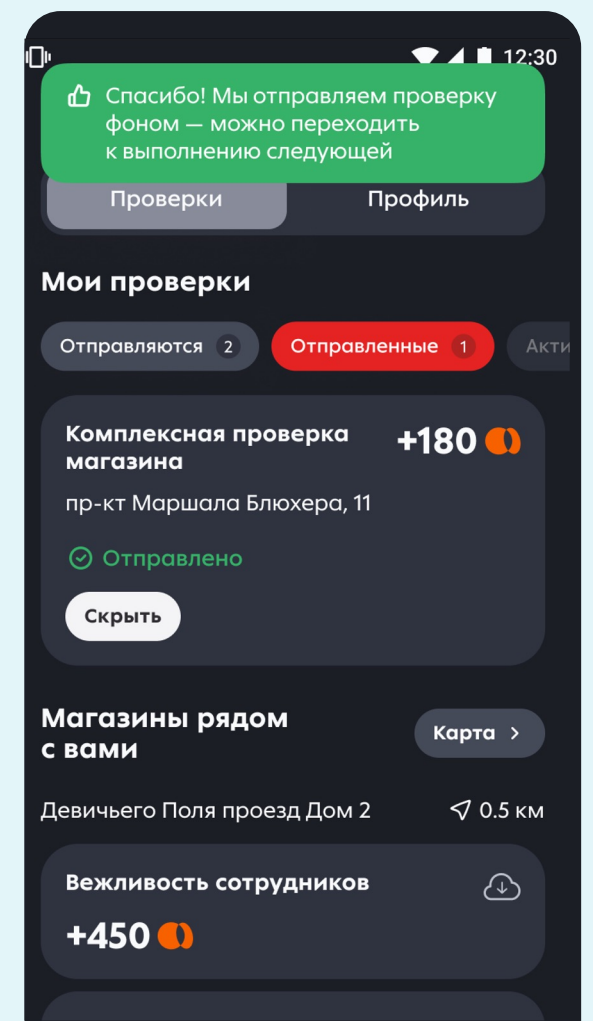
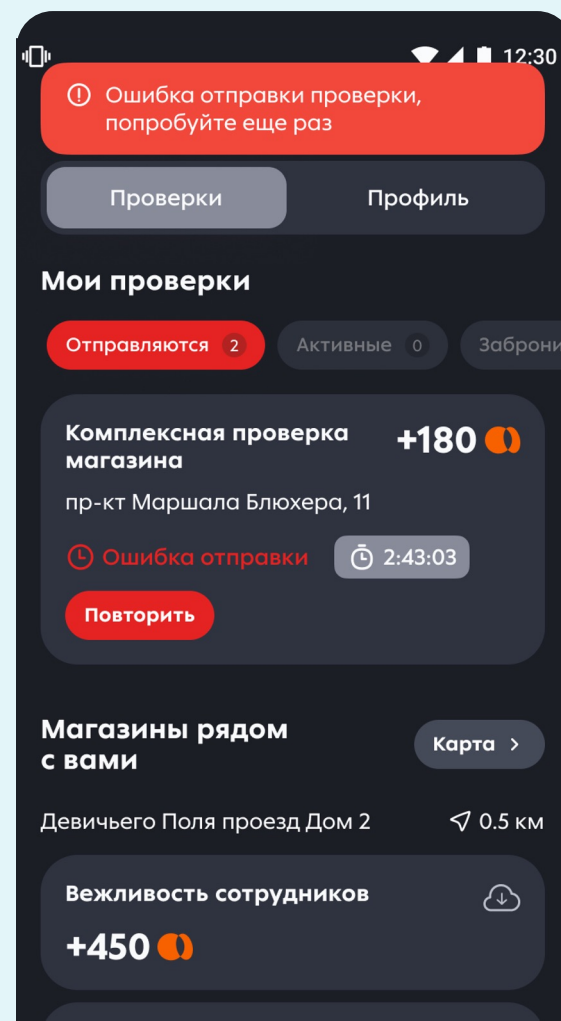
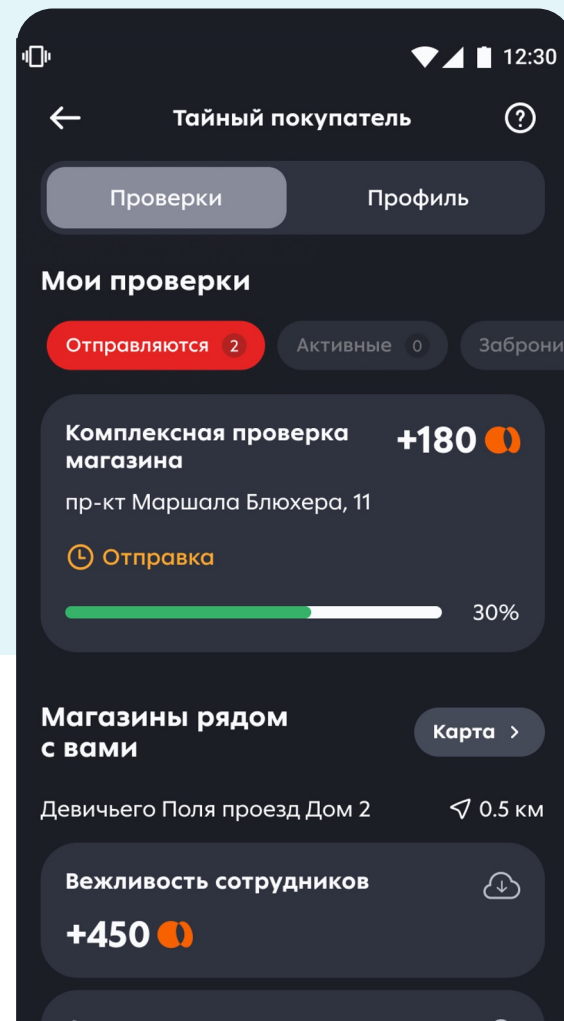
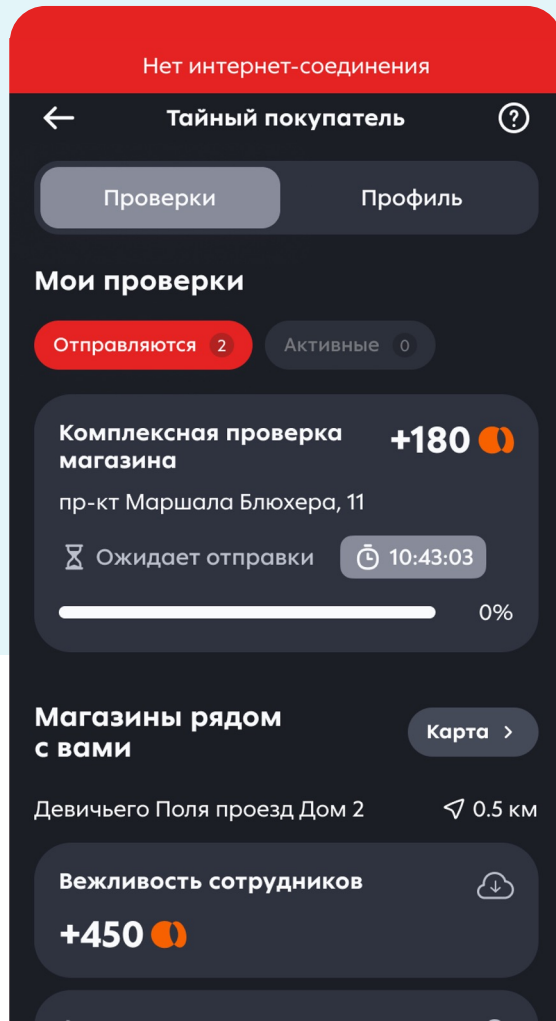
Было



Стало



Отправка проверки с фото в КТП после офлайн



Работа с фото при плохом соединении



```
this.timerId = setTimeout(() => {  
  if (!this.payload) {  
    runInAction(() => {  
      this.isBadConnection = true;  
    });  
  }  
}, CHECK_TIMEOUT);  
this.payload = await requestHealthCheck({}, REQUEST_TIMEOUT);  
  
runInAction(() => {  
  this.isBadConnection = false;  
});  
}
```

```
const connectionType = window.navigator.connection?.effectiveType;  
this.isBadConnection =  
  connectionType === 'slow-2g' || connectionType === '2g';
```


Поддержка Blob в IndexedDB

§ 2.3. Values

Each record is associated with a **value**. User agents must support any [serializable object](#). This includes simple types such as [String](#) primitive values and [Date](#) objects as well as [Object](#) and [Array](#) instances, [File](#) objects, [Blob](#) objects, [ImageData](#) objects, and so on. Record [values](#) are stored and retrieved by value rather than by reference; later changes to a value have no effect on the record stored in the database.

Record [values](#) are [Records](#) output by the [StructuredSerializeForStorage](#) operation.

steps_photos |  Enter a SQL expression to filter results (use Ctrl+Space)

Grid		123 step_done_id	ABC hashCode	☒ is_duplicate
1		6,875	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	[v]
2		6,874	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	[v]
3		6,874	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	[v]
4		6,875	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	[v]

 Приложенные фото

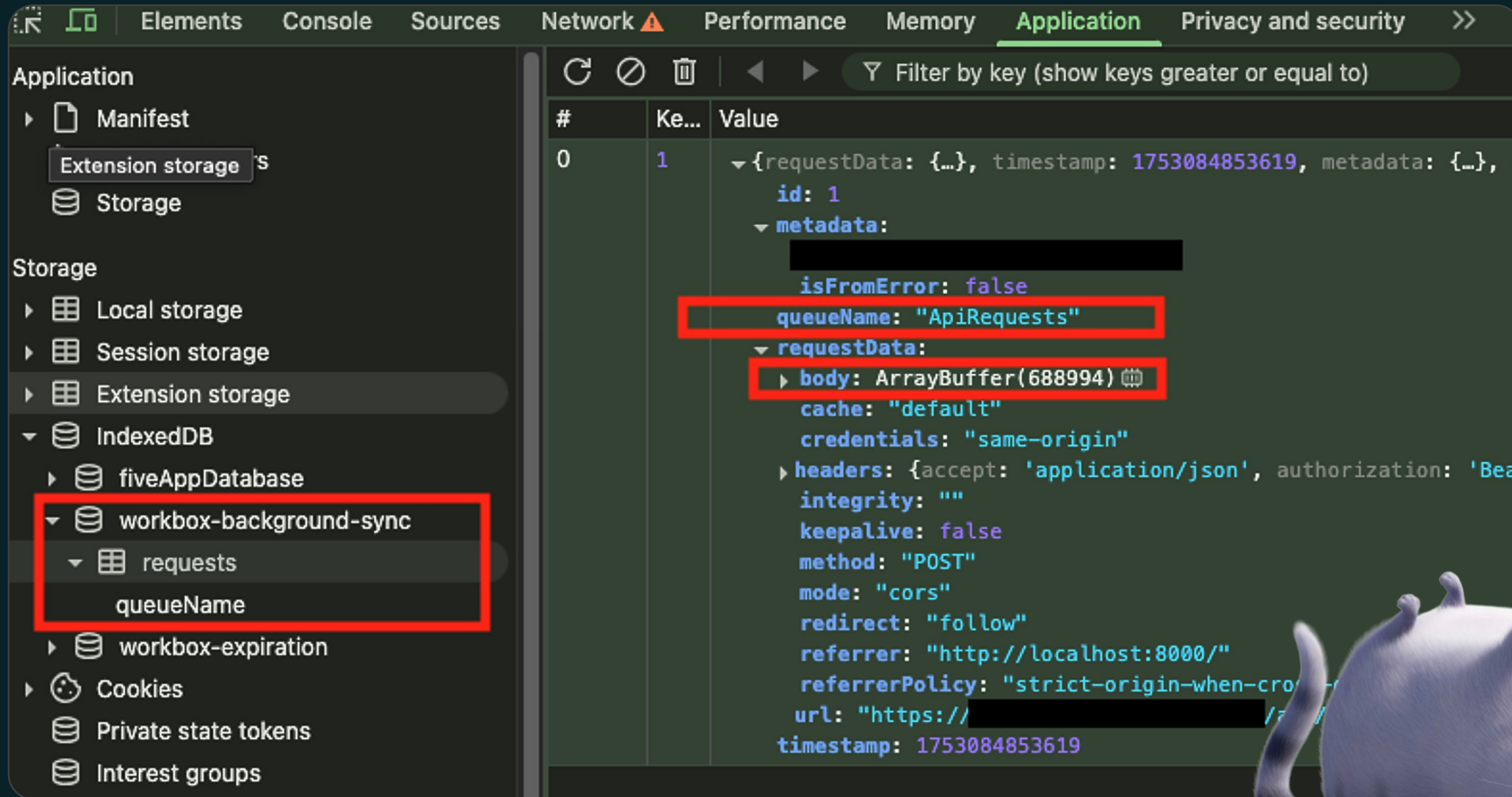
 ПОДСКАЗКА



2014
год



Как было в Workbox?



The screenshot shows the Chrome DevTools Application tab. On the left, the 'Storage' section is expanded, and 'workbox-background-sync' is selected, showing a 'requests' queue with a 'queueName' property. On the right, the 'Network' tab shows a list of requests. The first request is selected, and its details are shown in the right pane. The request is a POST to 'https://...' with a timestamp of 1753084853619. The request data is highlighted with a red box, showing 'body: ArrayBuffer(688994)'. The metadata is also highlighted with a red box, showing 'queueName: "ApiRequests"'. The request headers are also visible, including 'accept: application/json', 'authorization: 'Bearer...', 'integrity: ''', 'keepalive: false', 'method: POST', 'mode: cors', 'redirect: follow', 'referrer: http://localhost:8000/', 'referrerPolicy: strict-origin-when-cross-origin', and 'url: https://...'.

#	Key	Value
0	1	<pre>{requestData: {...}, timestamp: 1753084853619, metadata: {...}, id: 1, metadata: {isFromError: false, queueName: "ApiRequests"}, requestData: {body: ArrayBuffer(688994), cache: "default", credentials: "same-origin"}, headers: {accept: 'application/json', authorization: 'Bearer...', integrity: '', keepalive: false, method: "POST", mode: "cors", redirect: "follow", referrer: "http://localhost:8000/", referrerPolicy: "strict-origin-when-cross-origin", url: "https://..."}, timestamp: 1753084853619}</pre>



Хранение фото в IndexedDB: от Blob к ArrayBuffer



```
export const blobToArrayBuffer = (  
  blob: Blob,  
) : Promise<string | ArrayBuffer | null> => {  
  return new Promise((resolve, reject) => {  
    const reader = new FileReader();  
    reader.addEventListener('loadend', () => {  
      resolve(reader.result);  
    });  
    reader.addEventListener('error', reject);  
    reader.readAsArrayBuffer(blob);  
  });  
};
```


Добавление фото: MediaDevices vs Media Capture



```
<input
  id={cameraId}
  type="file"
  accept="image/*"
  capture="environment"
  onChange={onClick}
  hidden
/>
```

The **capture** content attribute is an [enumerated attribute](#) whose state specifies the [preferred facing mode](#) for the media capture mechanism.

The attribute's keywords are user, environment, and the empty string, which map to the *user*, *environment*, and the *implementation-specific* state respectively. The semantics of the states *user* and *environment* mirror the similarly named enumeration values defined in [VideoFacingModeEnum](#).

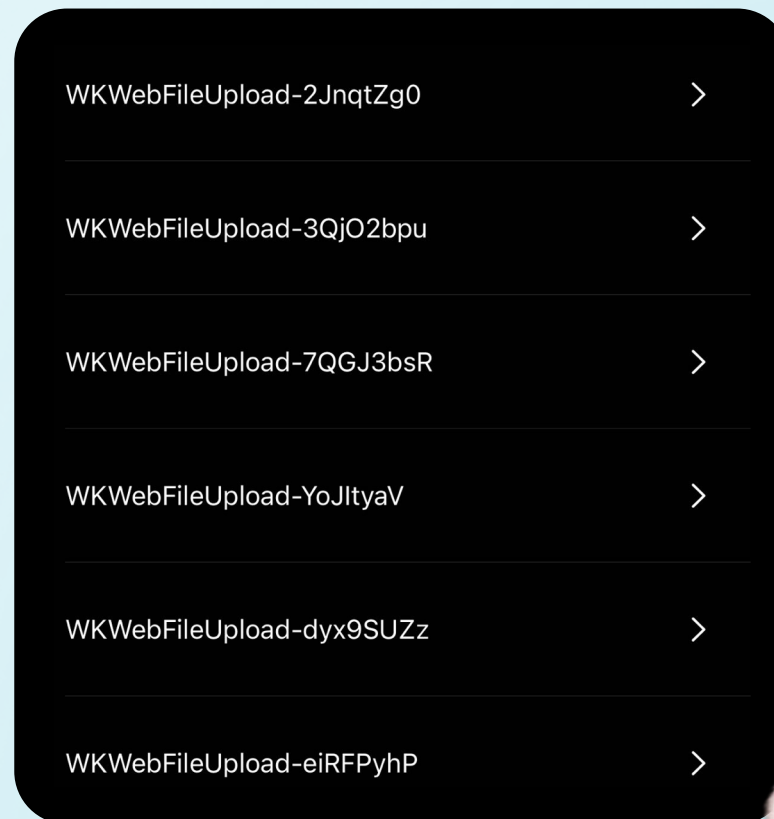
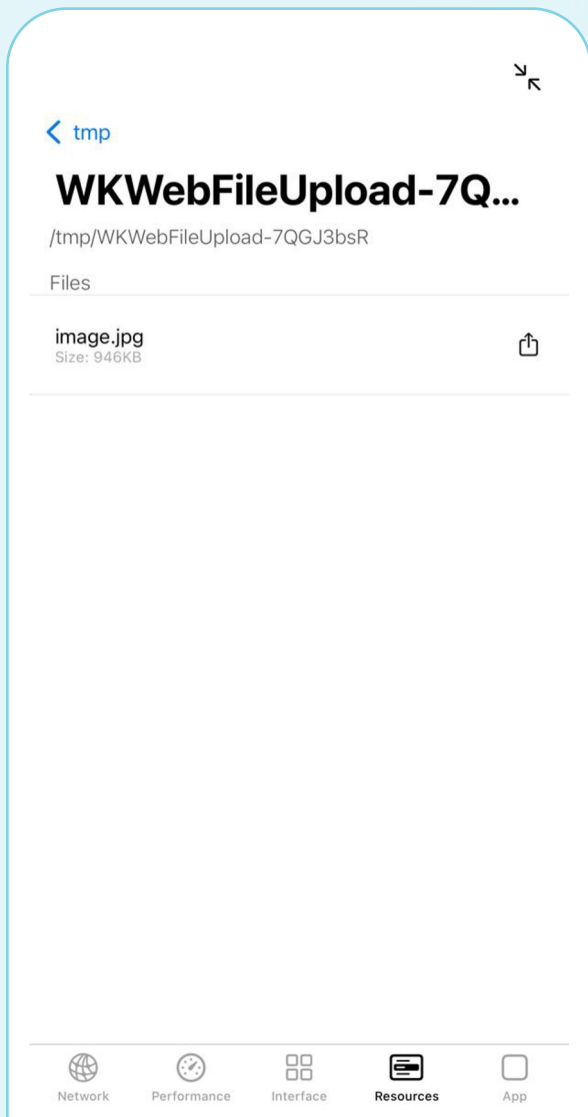
А память-то растёт...



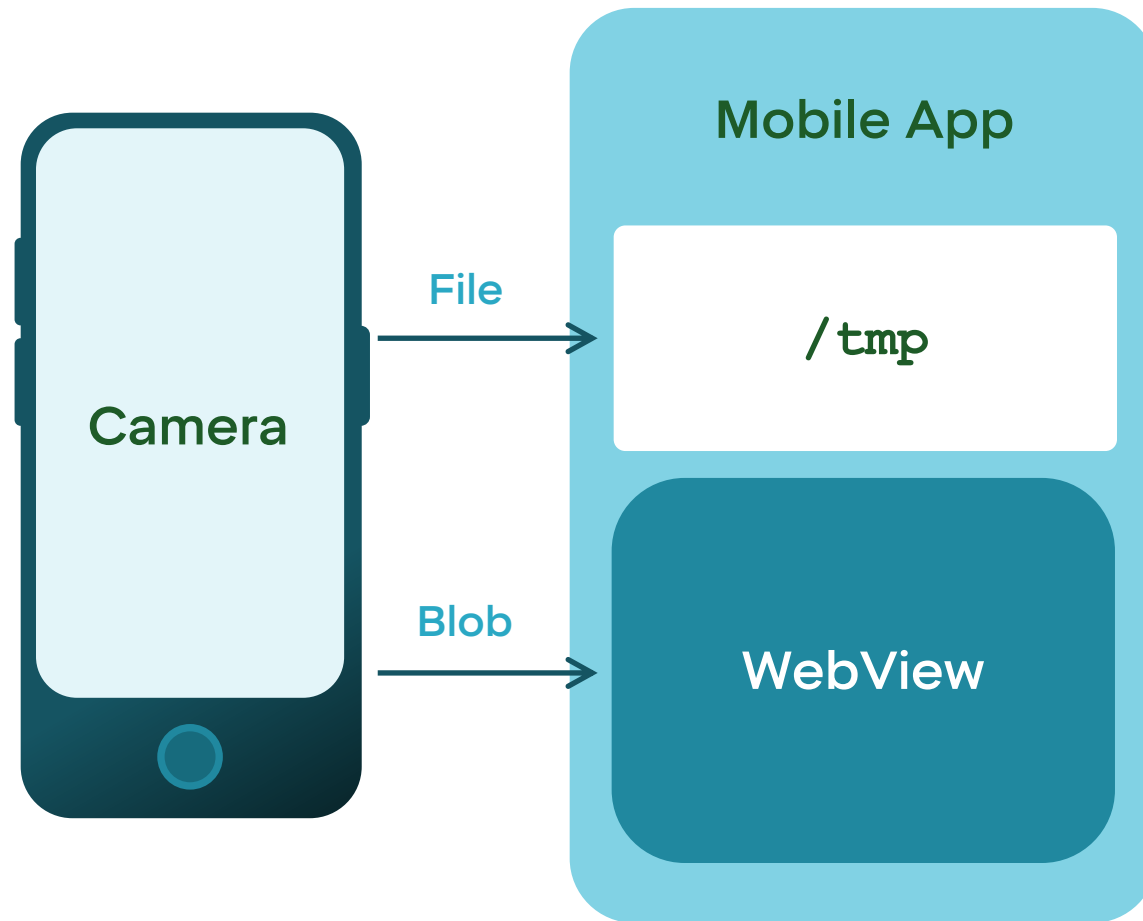
	5ка Debug 3.6.1
Использовано места	
Приложение	204 МБ
Данные	64,61 МБ
Кэш	6,84 МБ
Всего	276 МБ

	5ка Debug 3.6.1
Использовано места	
Приложение	204 МБ
Данные	92,93 МБ
Кэш	6,85 МБ
Всего	304 МБ

Утечка памяти из-за нативного приложения камеры



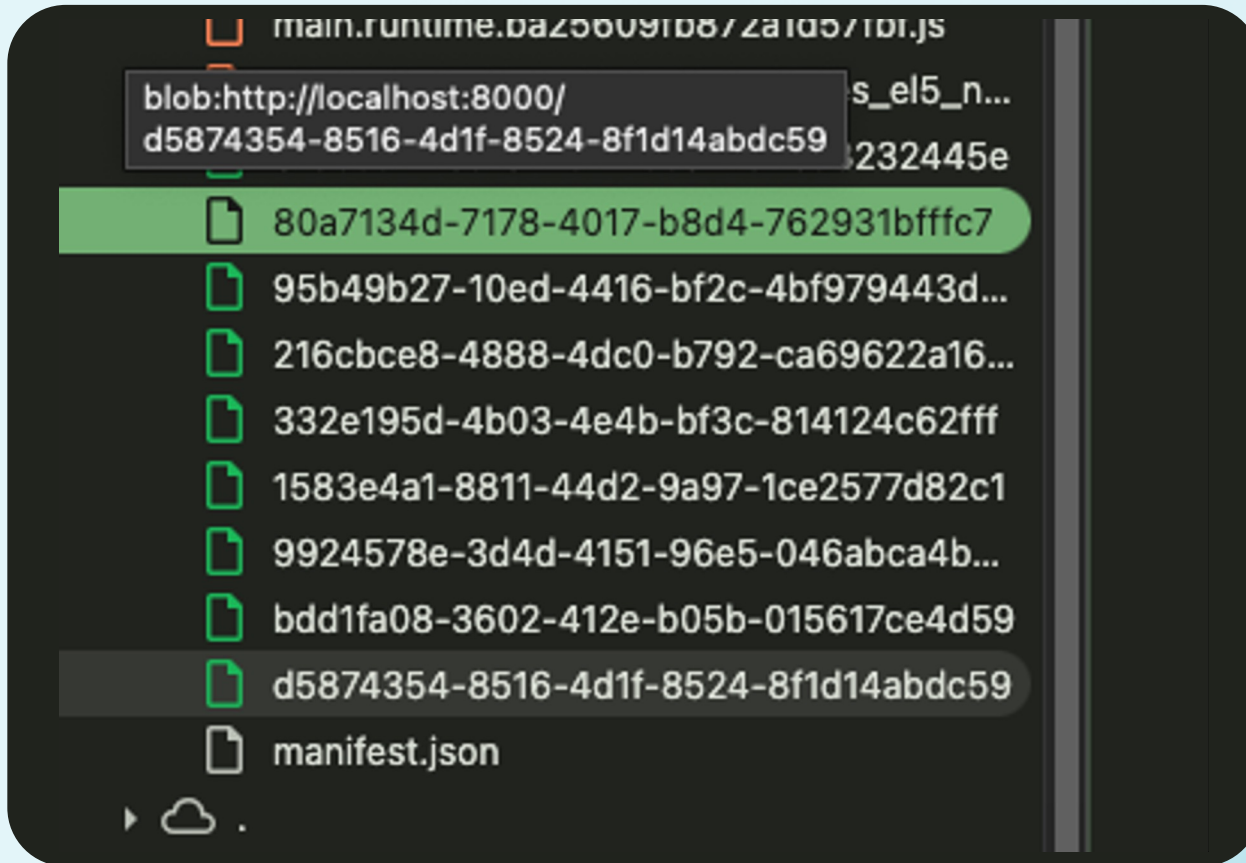
Очистка временной директории



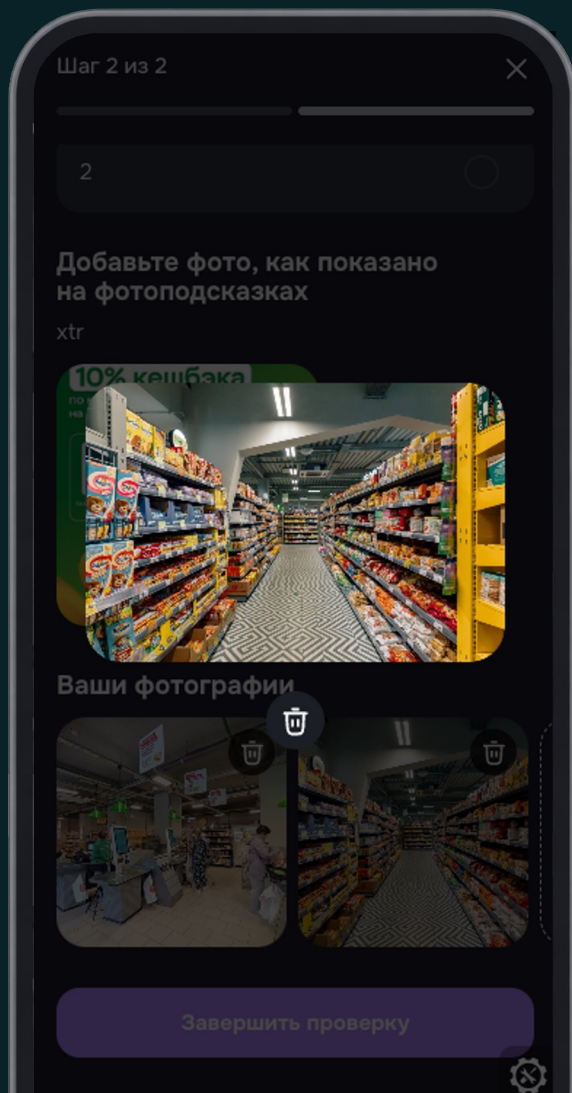
```
// iOS
window
  .webkit?
  .messageHandlers?
  .clearTemporaryStorage?
  .postMessage?.()

// Android
window
  .AndroidInterface?
  .clearTemporaryStorage?.()
```

Больше Blob-ов – больше памяти



Сжимаем фото, чтобы оно меньше весило



// **✗** Проблемный подход - двойная работа + утечки памяти

```
const objectUrl = URL.createObjectURL(file);
const img = new Image();
img.src = objectUrl;
img.onload = () => {
  // Рендерим на canvas для сжатия
  canvas.drawImage(img, 0, 0);
  // Создаем новый blob
  canvas.toBlob(compressedBlob => {
    // Еще один URL
    const compressedUrl = URL.createObjectURL(compressedBlob);
    // Утечки памяти на мобильных WebView!
  });
};
```

Меняем + Blob URL на Bitmap + HTML Canvas



```
const drawImage = async () => {
  const canvas = canvasRef.current;
  const context = canvas.getContext('2d', { alpha: false });
  imageBitmap = await createImageBitmap(photo);
  canvas.height = imageBitmap.height;
  canvas.width = imageBitmap.width;
  context.drawImage(imageBitmap, 0, 0, imageBitmap.width, imageBitmap.height);
};

useEffect(() => {
  return () => {
    imageBitmap?.close?();
  };
}, []);

// ...

return <canvas className={className} ref={canvasRef} onClick={onClick} />;
```

Сравнение



	 + Blob URL	HTML Canvas + ImageBitmap
Память	Кэшируется, но Blob URL надо ревокать	GPU-память, нужно закрывать
Декодирование	Синхронное (может тормозить)	Асинхронное (createImageBitmap)
Рендеринг	CPU/GPU (зависит от браузера)	GPU (аппаратное ускорение)
Обработка	Нет (только через CSS/Canvas)	Да (фильтры, пиксельные операции)

Резюме



Workbox Background Sync

Красив, но пока не готов



Кастомный плагин

Палка о двух концах



Offline-First

Это не про технологию, а про UX



Данные

Бетон, а не песок. ArrayBuffer даёт стабильность



Нативные мосты

Необходимая, а не желательная поддержка в WebView



Рендеринг без мусора с Canvas + Image Bitmap



Ссылки

- Изображения персонажей из мультфильма «Секретная жизнь домашних животных»
<https://www.kinopoisk.ru/film/743088>
- Документация Workbox Background Sync
<https://developer.chrome.com/docs/workbox/modules/workbox-background-sync?hl=ru>
- Проблемы с «BackgroundSync in disabled»
<https://tinyurl.com/mrymjnbj>
- Как реализовать стратегию повторных попыток экспоненциальной отсрочки
<https://tinyurl.com/3s7zcrpj>
- Настройка Workbox Background Sync для совместимости с iOS и Android WebView
<https://habr.com/ru/articles/915896/>
- Спецификация HTML Media Capture
<https://w3c.github.io/html-media-capture/#dfn-capture>
- Спецификация IndexedDB: допустимые значения
<https://w3c.github.io/IndexedDB/#value-construct>
- Проблемы с Blob в IndexedDB
<https://github.com/nolanlawson/state-of-binary-data-in-the-browser>
- Освобождение памяти Blob-объекта
<https://stackoverflow.com/a/50698058>



Спасибо! Вопросы?

Артур Басак

Руководитель команды разработки



arturbasak.dev

