uzum

WebView как способ интеграции между сервисами

Максим Лавренюк

Обо мне

- Frontend Tech Lead в Uzum Tezkor
- Основной стек: React + TS
- Очень люблю метрики, оптимизации и тесты

О чем сегодня поговорим

- Использование WebView как способ интеграции между сервисами
- Ограничения WebView
- Тонкости веб-разработки под мобильные клиенты
- Не буду говорить о Cordova
- Не скажу, какая классная технология
- Не скажу, какая плохая технология

Uzum Tezkor - сервис быстрой доставки из кафе, ресторанов и магазинов

Более 98%

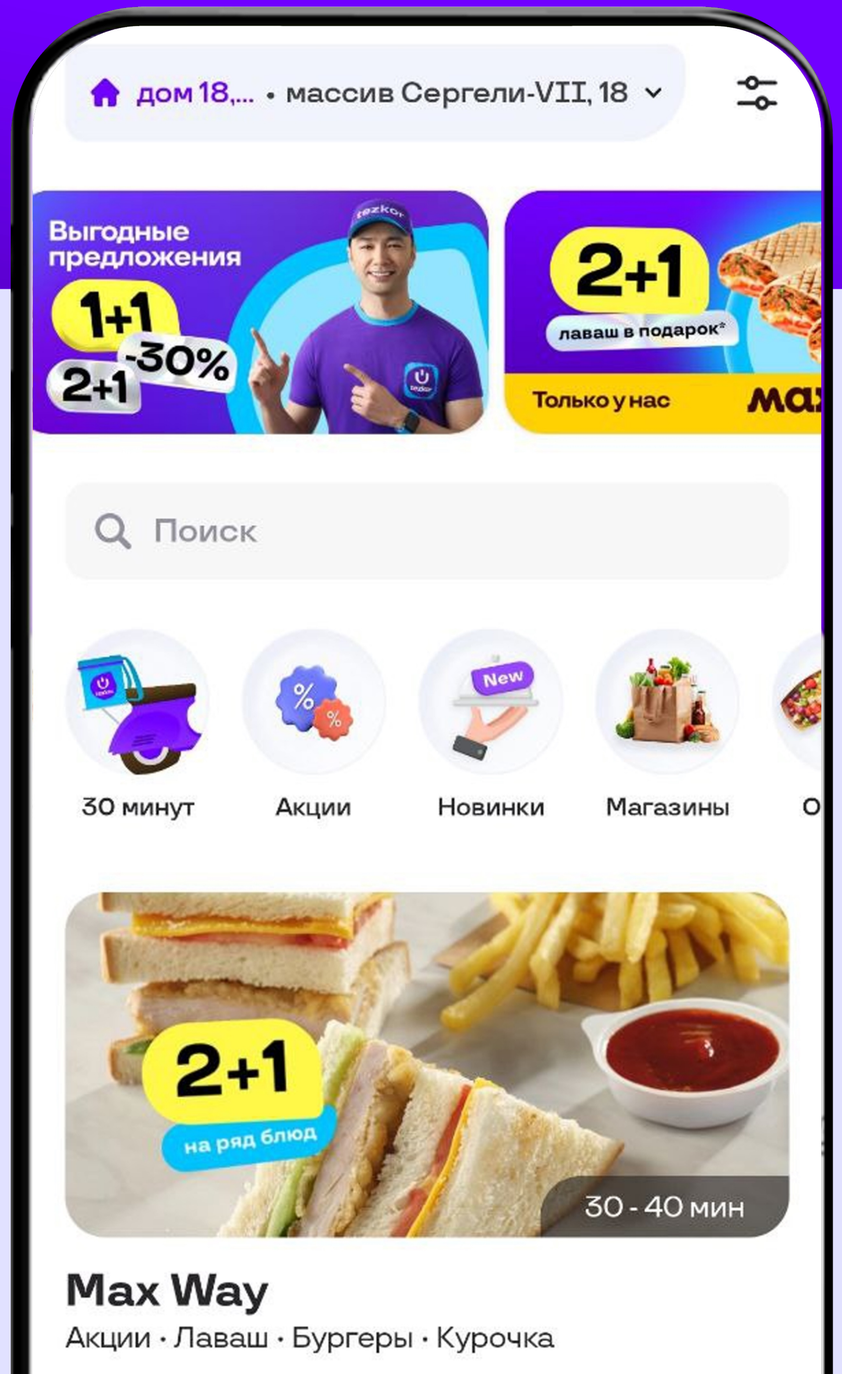
заказов доставляются за 30 мин

> 1000

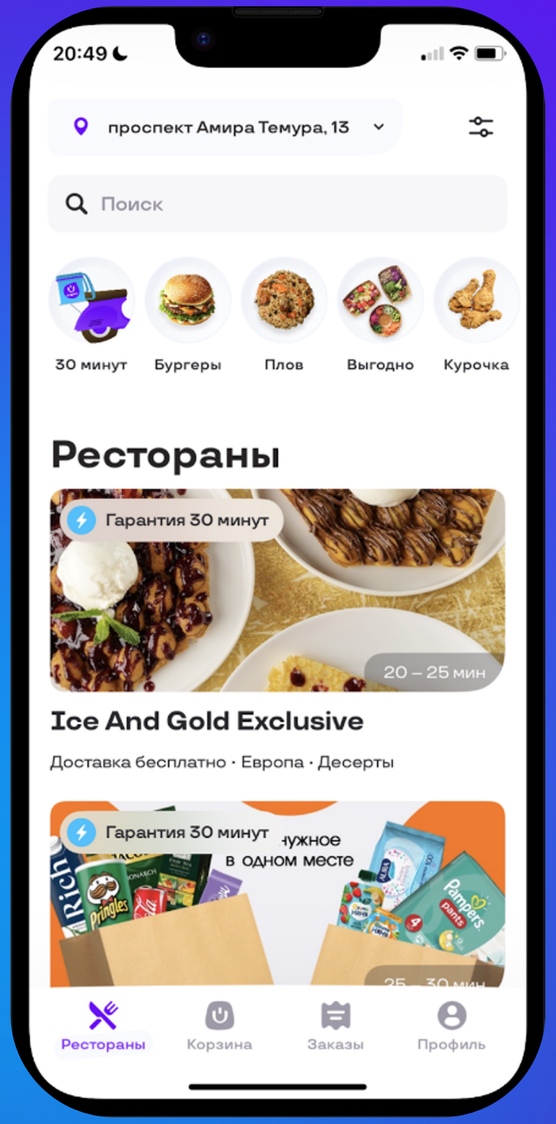
активных ресторанов и магазинов

Май 2023 года — запуск сервиса

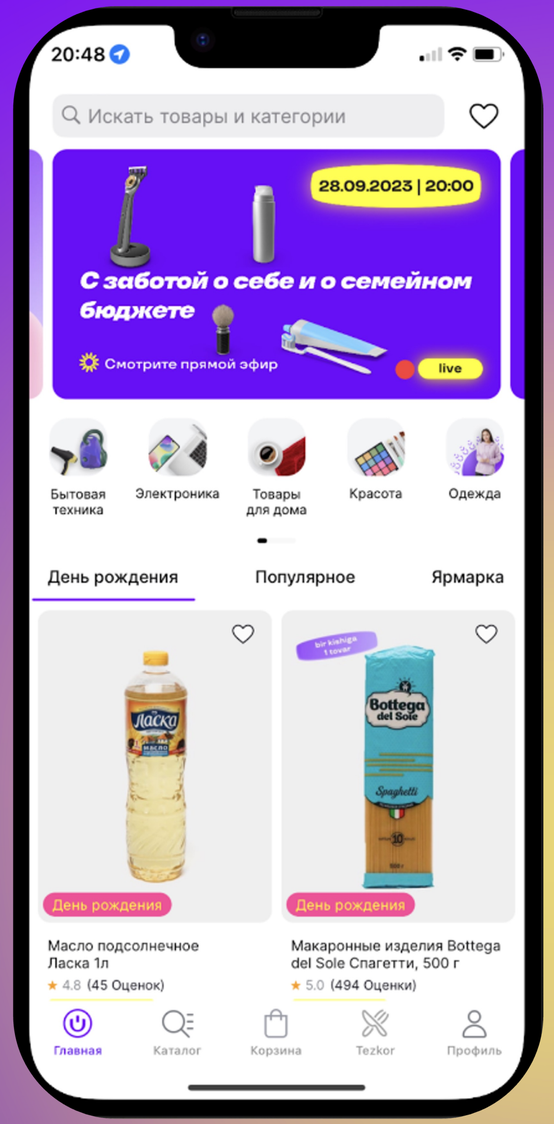
Амбиция — покрыть весь Узбекистан



Uzum Tezkor



Uzum Market



Зачем это бизнесу?

uzum market

**Uzum Market —
№1 маркетплейс
в Узбекистане.**

Если мы внедрим
Uzum Tezkor
в это приложение,
то получим заказы
без вложений в рекламу,
**а это более 13,5 млн
установок приложения
Uzum Market только в
2023 году.**

Какие способы интеграции мы рассматривали

B2B-интеграция

Какие способы интеграции мы рассматривали

интеграция нативного приложения как SDK

Какие способы интеграции мы рассматривали

интеграция через WebView

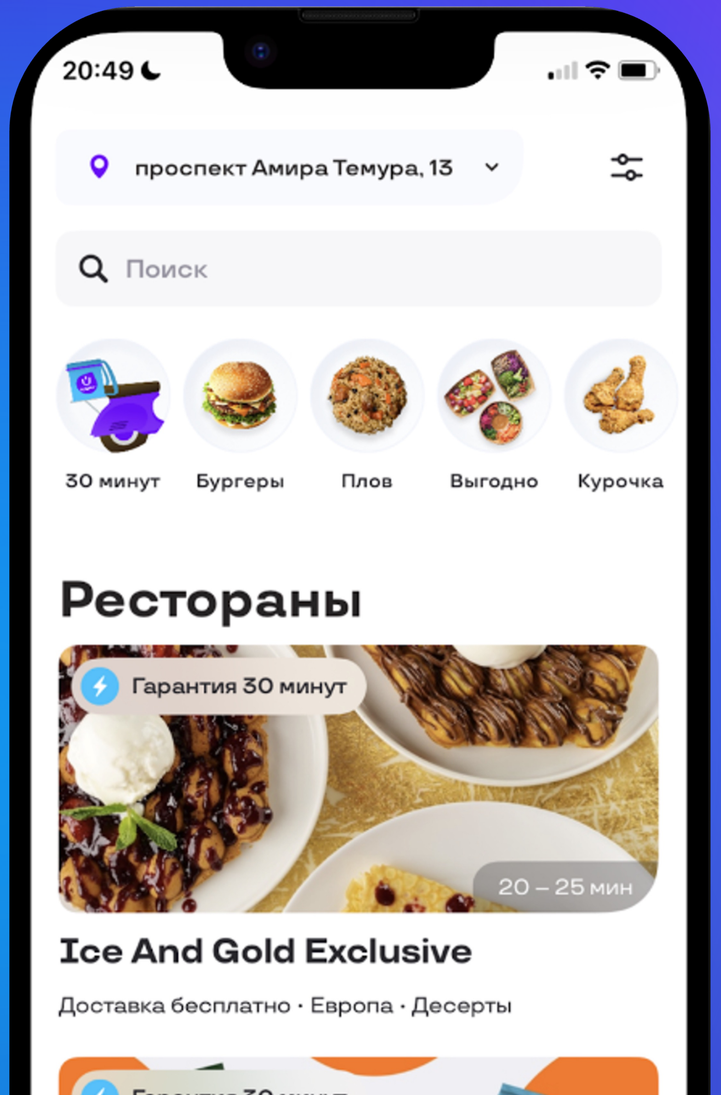
uzum tezkor

Backend

uzum market

Backend

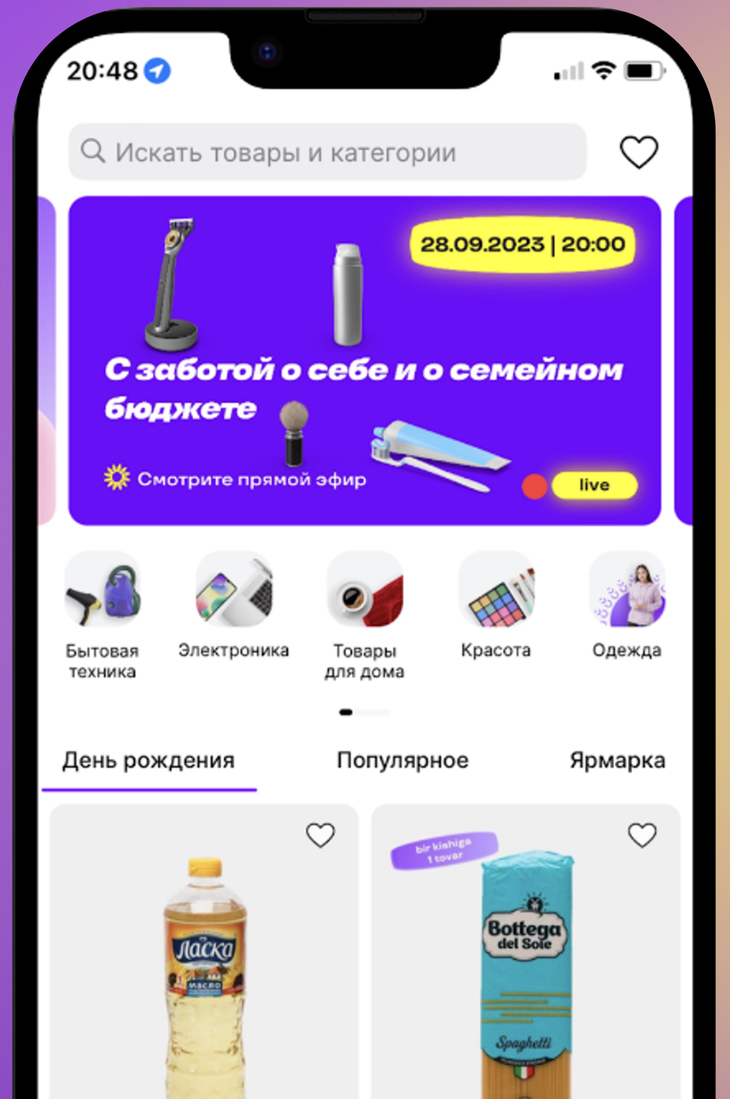
uzum tezkor


20:49
проспект Амира Темура, 13
Поиск
30 минут
Бургеры
Плов
Выгодно
Курочка
Рестораны
Гарантия 30 минут
20 – 25 мин
Ice And Gold Exclusive
Доставка бесплатно · Европа · Десерты


SDK

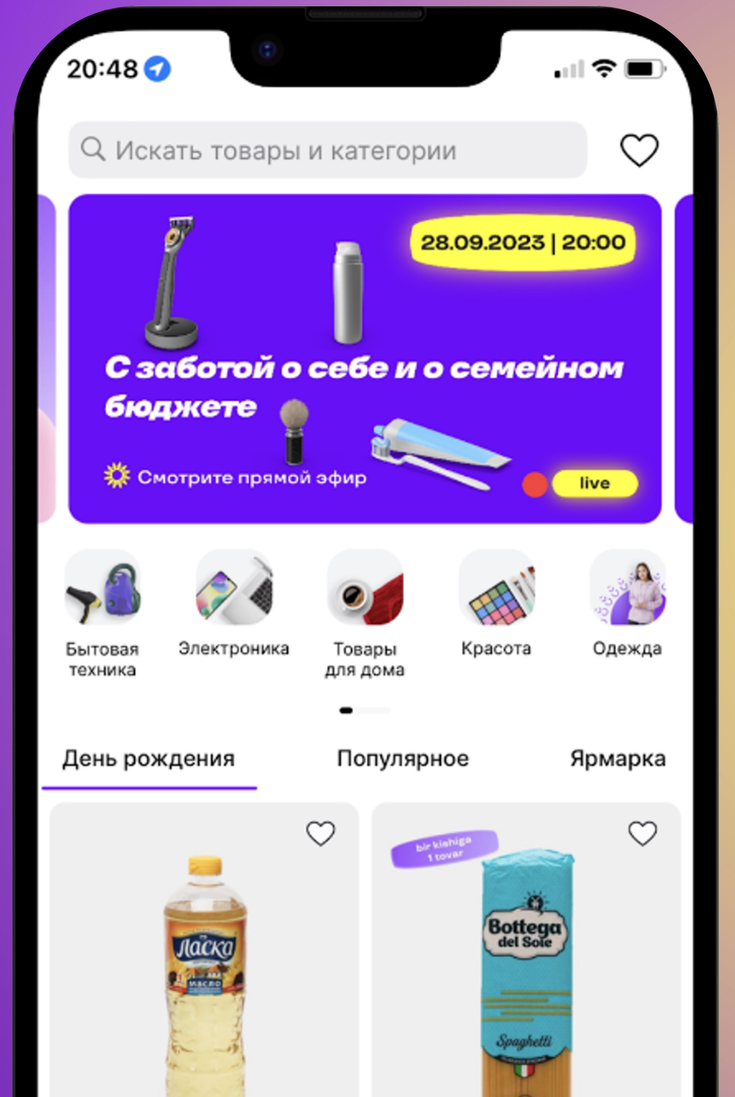
uzum market

SDK



Интеграция через WebView

uzum market



Что такое WebView?

механизм открытия браузера внутри iOS/Android-приложений

не связано с React Native/Flutter

Плюсы

возможность развернуть в полноценный сайт

слабая зависимость команд разработки друг от друга

независимые релизные циклы

достаточно быстрая реализация

Минусы

очень тяжело добиться такого же
UX, как в нативном приложении

часть функционала придется реализовывать
в нативном приложении

Хотим в итоге

Нативное приложение

WebView

Подготовка

Как общаться нативному и веб-приложению?

- JSBidge
- URL: searchParams

JSBridge. iOS

- userContentController
- evaluateJavaScript

JSBridge. Android

- `addJavascriptInterface`
- `class JSBridge`
- `evaluateJavascript`

Подробнее о внутренностях:

JSBridge. Web

```ts
interface Window {
    webkit?: {
        messageHandlers: {
            // в iOS добавили через userContentController
            webViewIpcManager?: {
                // через это нативное приложение сообщение из WebView
                postMessage(message: string): void;
            };
        };
    };
    // в evaluateJavascript указывали какую функцию вызвать для оповещения WebView
    invokeWebViewIpcManager: InvokeWebViewIpcManager;
    JSBridge?: {
        // В Android приложении указывали через что получить сообщение из WebView
        handleWebViewIpcManager(event: string, payload?: string): void;
    };
}
```

URL: searchParams

```
// iOS
let url = URL(string: "https://awesome-webview.com?cool=true")!
mWebKitView.load(URLRequest(url: url))

// Android
private val url = "https://awesome-webview.com?cool=true"

override fun onCreate(savedInstanceState: Bundle?) {
    webView.loadUrl(url)
}
```

Как проверять интеграцию, пока ее нет?

в идеале иметь абстрагированный пакет с WebView

если такого нет, сделайте себе iOS/Android mock приложения для тестирования работы WebView.

Какие могут быть ограничения?

Некоторые Web API, требующие доступов, по умолчанию отключены:

- `navigator.geolocation`
- `window.open`
- `navigator.share` и др.

Доступы

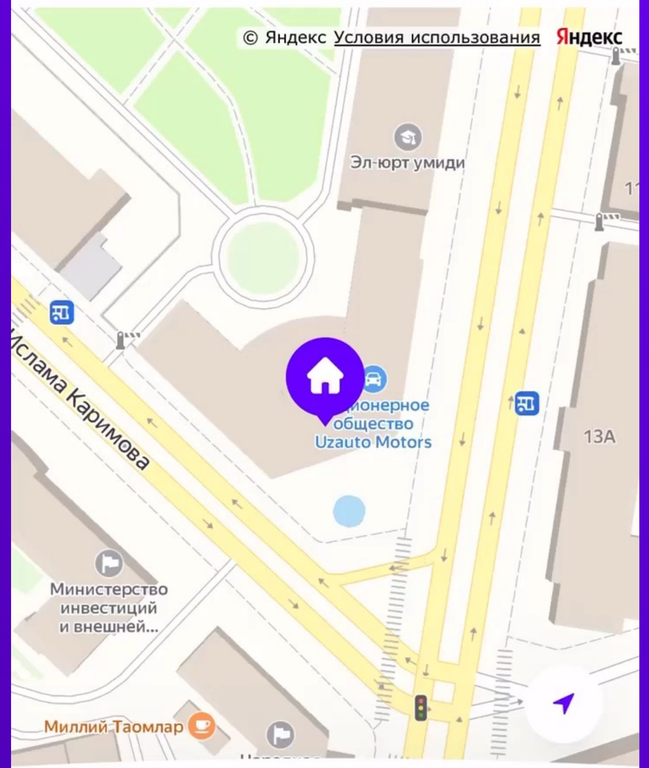
Доступы

- запрашивая доступы через Web API, вы «светите» домен
- чтобы использовать доступы через Web API, WebView должно быть предварительно настроено в нативном коде
- выданные доступы в WebView изолированы от выданных доступов в нативном приложении/web

19:41

Новый адрес

Введите адрес

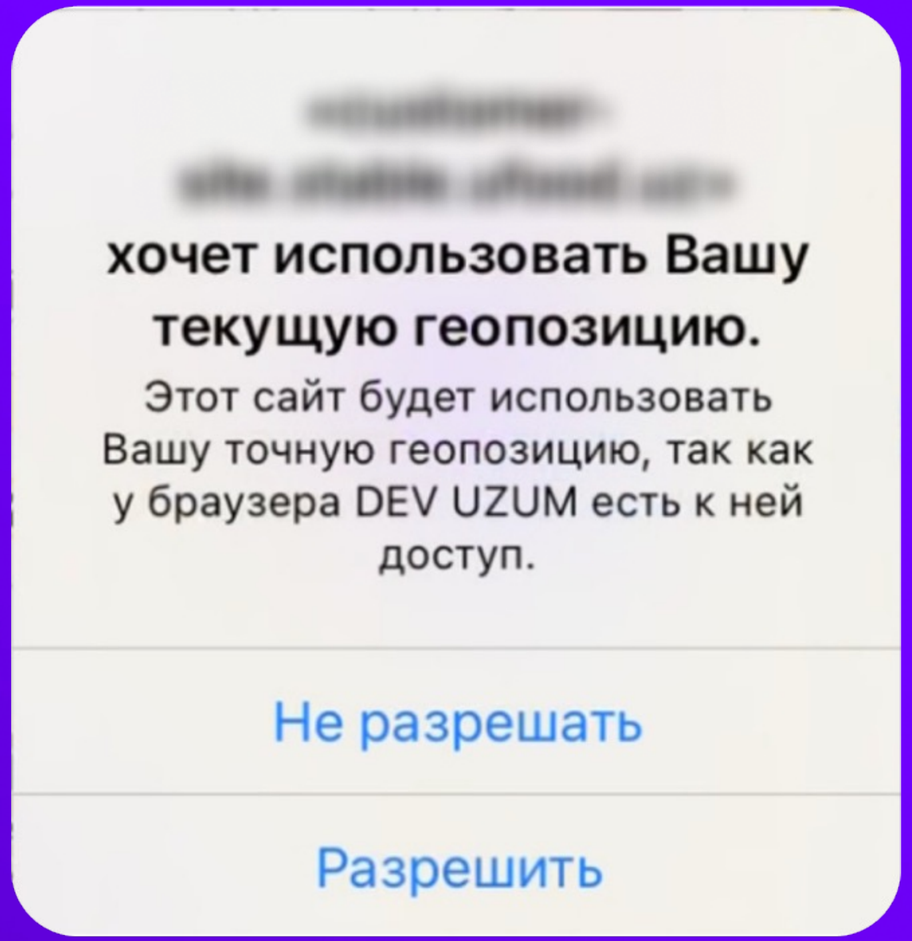


проспект Амира Темура, 13

Подтвердить локацию

Уточнить номер дома

```
navigator.geolocation.getCurrentLocation()
```



```
function getGeolocation() {
    if (isWebView()) {
        return JSBridgeProvider.invoke('getGeolocation')
    } else {
        return new Promise((resolve, reject) => {
            navigator.geolocation.getCurrentPosition(
                resolve,
                reject
            )
        })
    }
}
```

Требования к Geolocation

Android:

- Manifest.permission.ACCESS_COARSE_LOCATION
- Manifest.permission.ACCESS_FINE_LOCATION
- WebChromeClient#onGeolocationPermissionsShowPrompt

Открытие новых вкладок

```
window.open('http://example.com')
```

Требования к открытию новых вкладок

- Android: setJavaScriptCanOpenWindowsAutomatically
- iOS: javaScriptCanOpenWindowsAutomatically

Что еще отключено по умолчанию?

iOS

Android

Изоляция доступов

- Выданные Web API доступы на сайт не учитываются, если открыть этот же сайт в WebView.
- Выданные доступы в нативном приложении не учитываются при запросе доступов в WebView — пользователь все-таки увидит просьбу дать разрешение.
- Часть доступов требуют наличия соответствующих прав в приложении.

Проблемы UX

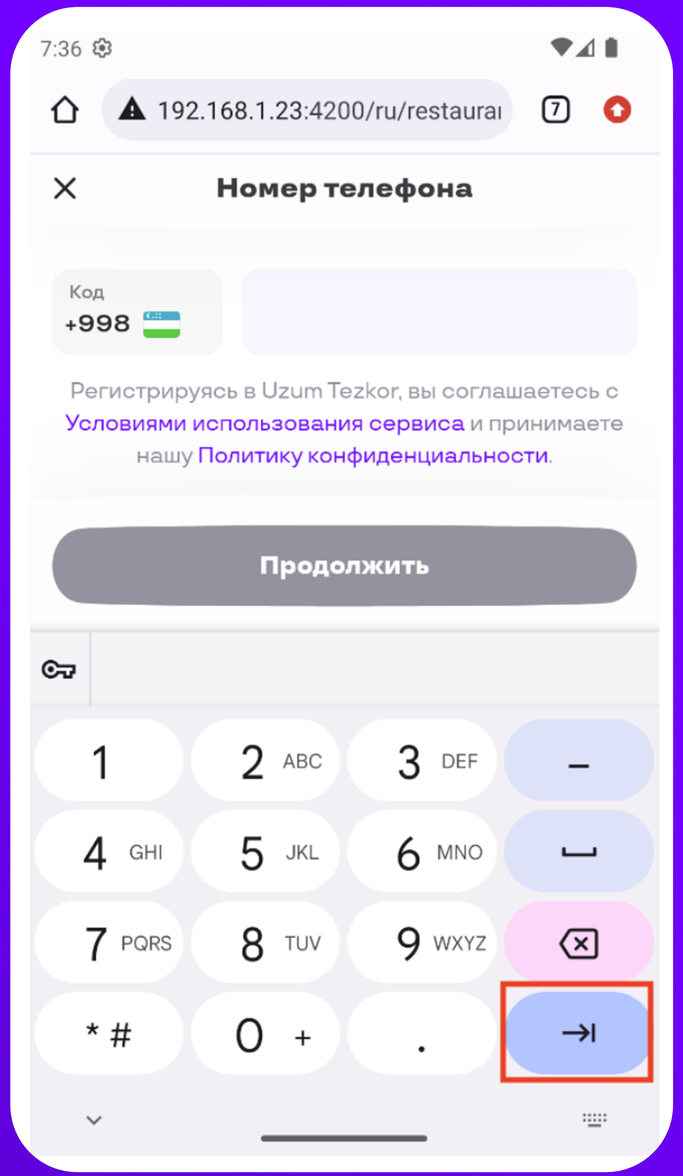
<input />

- <input /> **обязательно** должен лежать в <form />
- ввод с экранной клавиатуры **имеет** свои **особенности**

<input />. Enterkeyhint

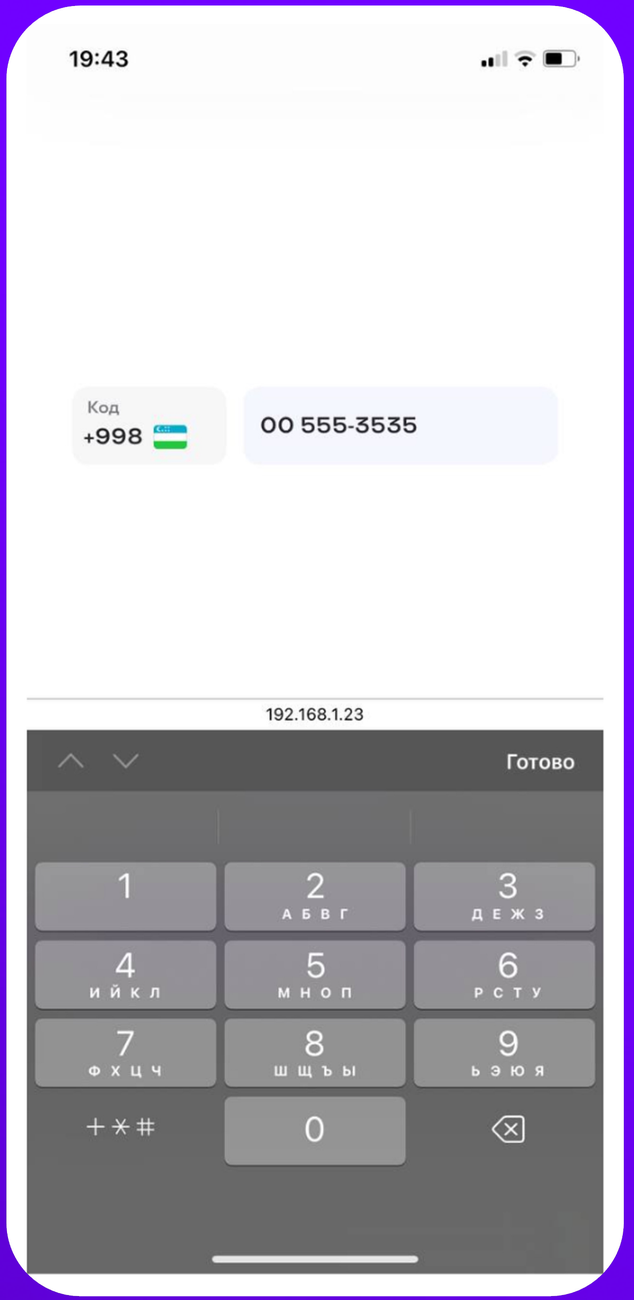
Enterkeyhint — атрибут, определяющий какую метку действия отображать для клавиши ввода на экранных клавиатурах.

Зависит от OS и type атрибута.



<input />. Enterkeyhint

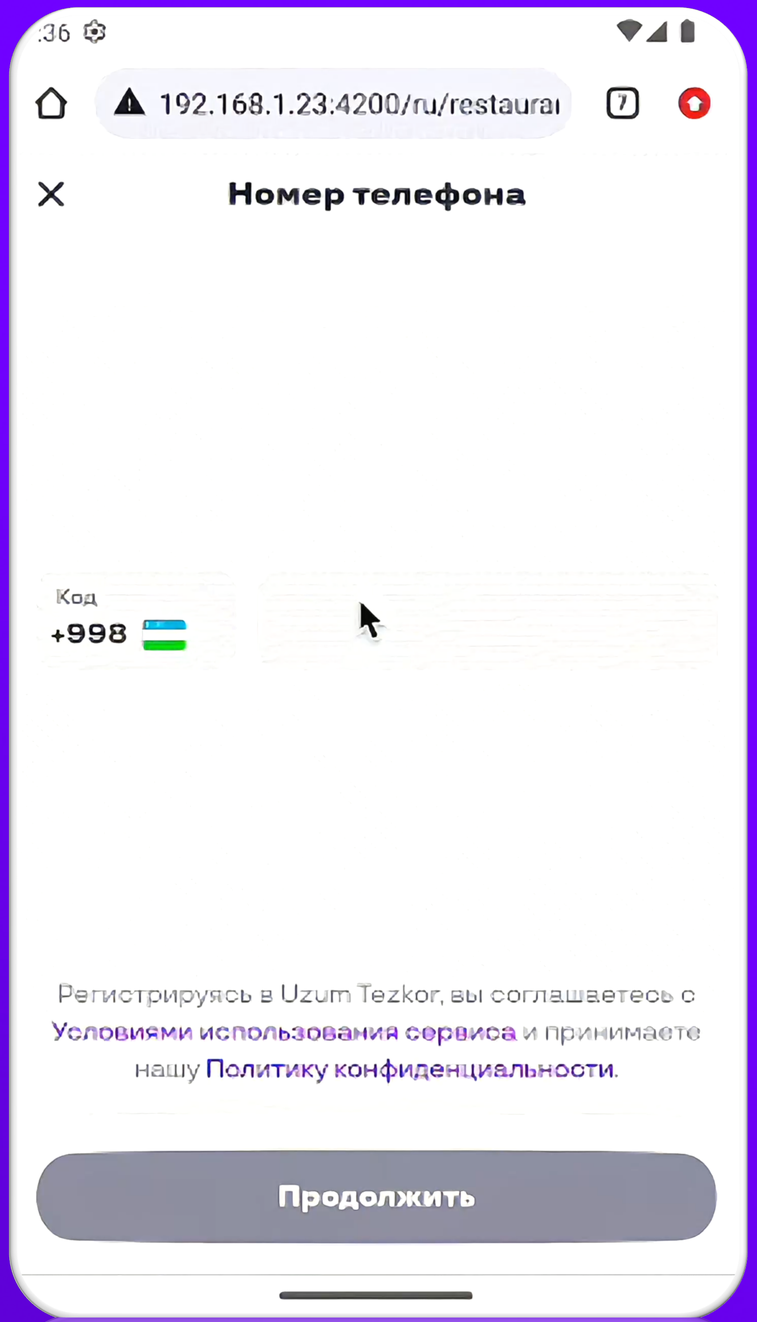
Enterkeyhint
на iOS отсутствует



<input />. Enterkeyhint

Если <input /> не обернут <form />, то Android Chrome попробует найти соседний <input /> и сфокусироваться на нем.

Проблема в том, что нажатия на «Enter» — это не тоже самое, что нажатие на enterkeyhint, значит при разработке будет тяжело отловить.



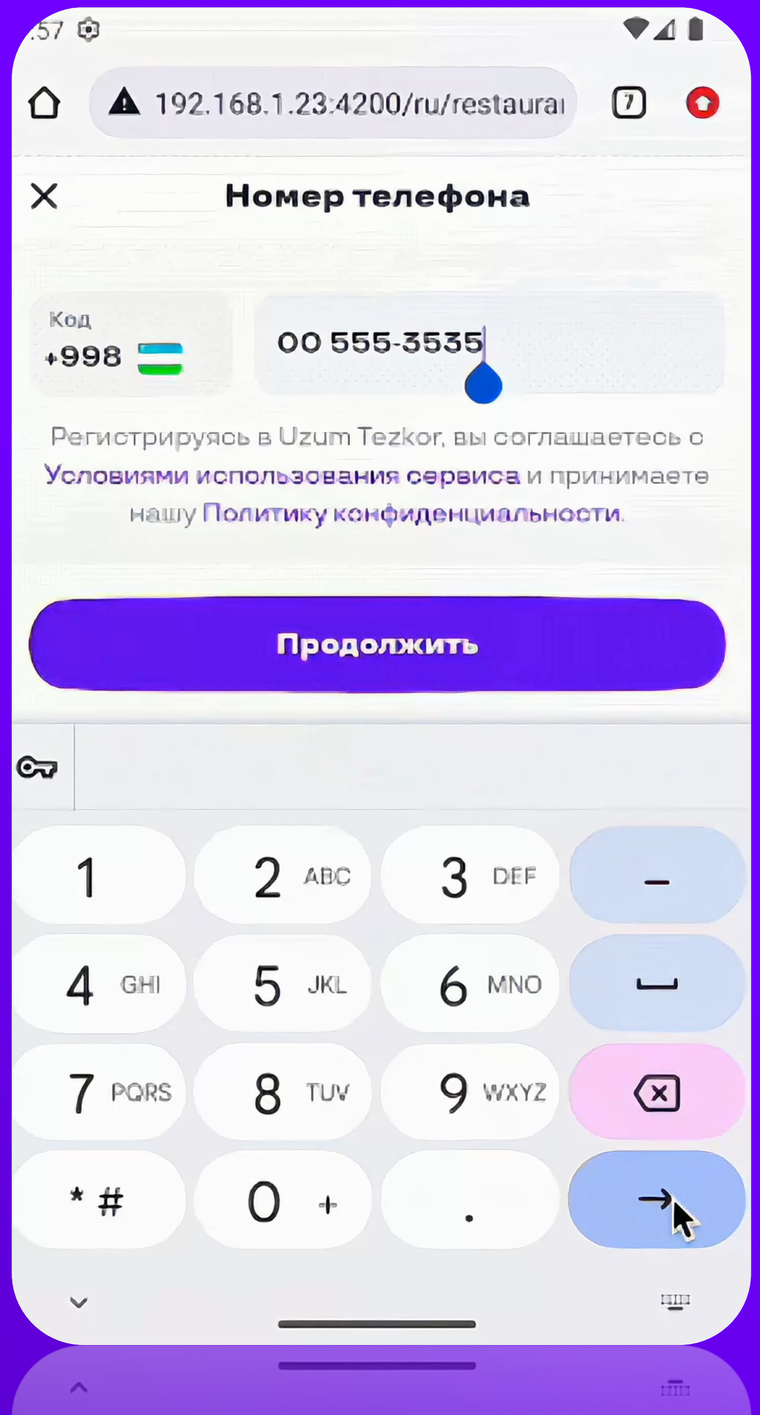
<input />. Enterkeyhint

- В Desktop браузерах такого нет:
  при разработке вы не заметите проблему.

- Проблема может не воспроизводиться
  при определенной комбинации компонентов вокруг
  <input /> (слайдеры и т.п.).

<input />. Enterkeyhint

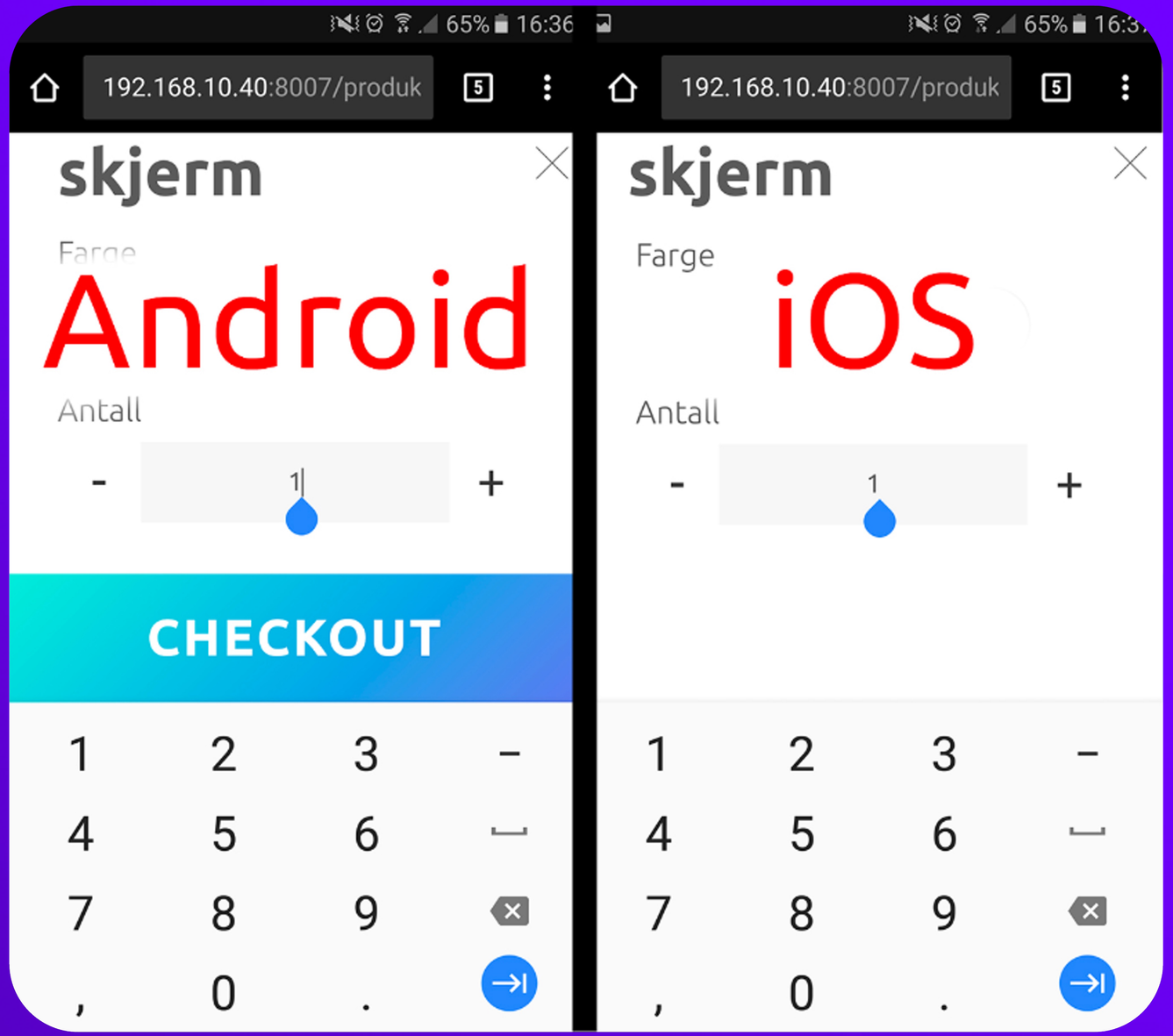
Если обернуть <input /> тегом <form />, то описанное поведение прекращается. Android Chrome не станет искать соседние <input />.



Экранная клавиатура

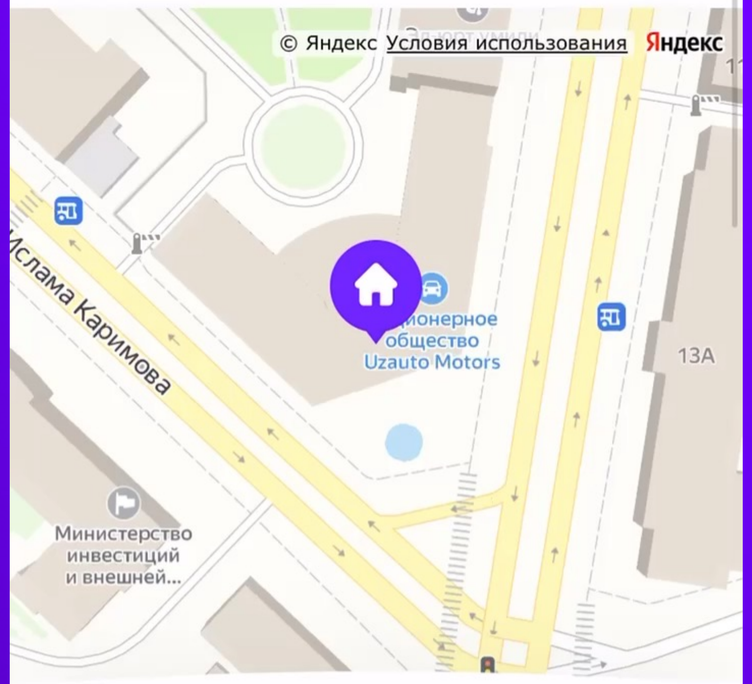
Экранная клавиатура
не сужает viewport,
а «скроллит» его до <input />
в фокусе.

Экранная клавиатура



23:43

проспект Амира Темура, 13



Название

Дом

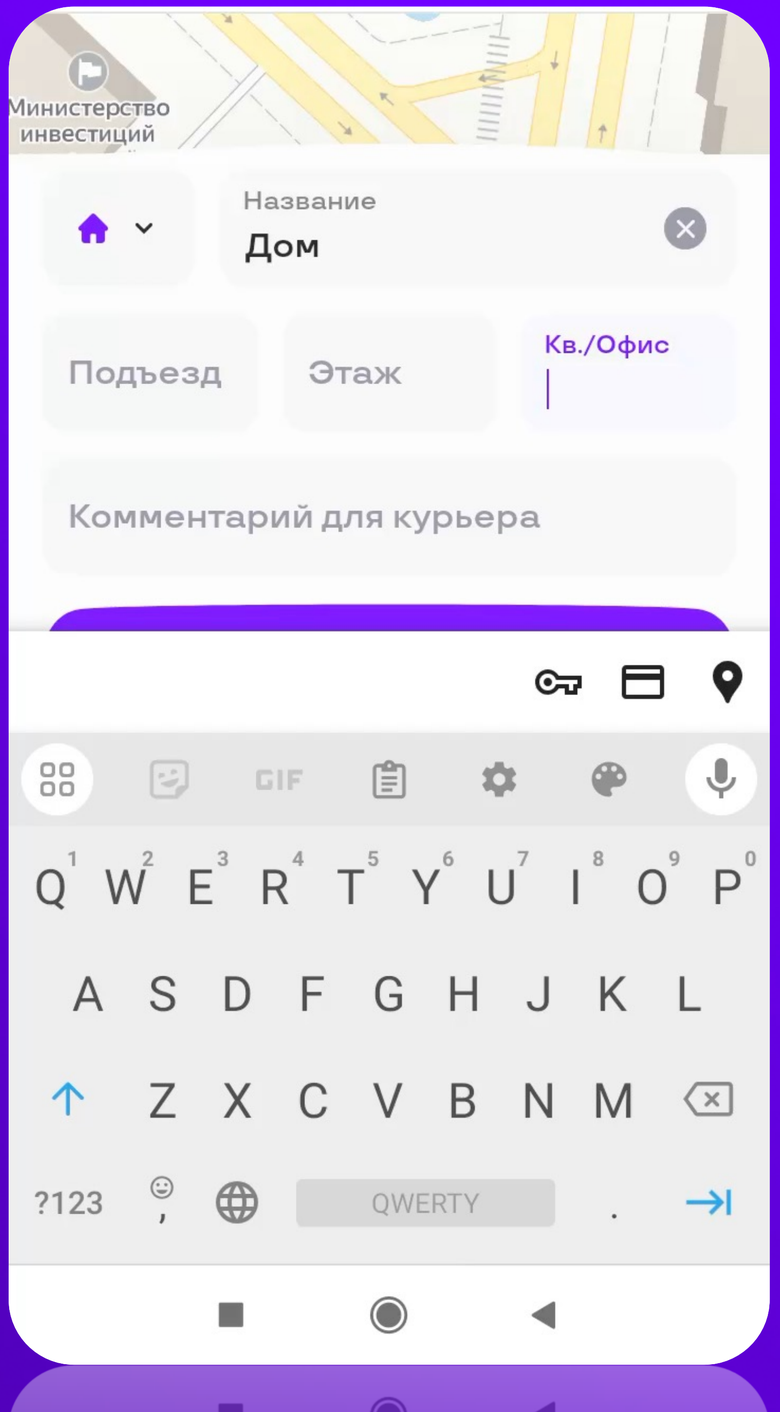
Подъезд

Этаж

Кв./Офис

Комментарий для курьера

Продолжить


Министерство инвестиций
Название
Дом
Подъезд
Этаж
Кв./Офис
Комментарий для курьера
GIF
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
?123
QWERTY

Экранная клавиатура

создает огромные проблемы UX, если внизу экрана есть контролы для взаимодействия.

Страничка в баг-трекере webkit:

Экранная клавиатура

Что об этом
думают разработчики:

```
Thank your for making
safari the next IE. dumbass
```

Экранная клавиатура

Сделано для оптимизаций:

изменение высоты, как и изменение любых размеров сильно бьет по производительности.

Единого решения проблемы не существует, зависит от верстки проекта.

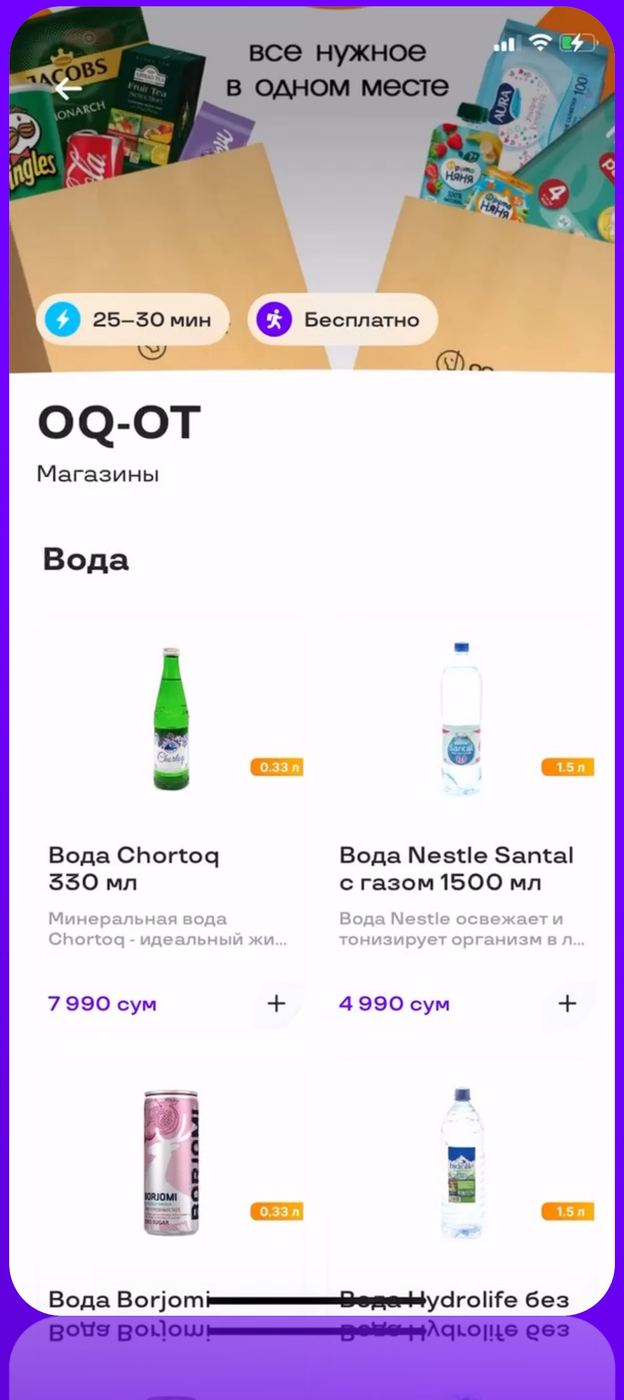
Роутинг

- пользователь будет ожидать поведение навигации как в приложении, а не как на сайте
- в части устройств для навигации могут использоваться свайпы
- состояние должно сохраняться между страницами

Роутинг

В нативных приложениях пользователь ожидает восстановление предыдущего состояния при навигации.

В Web с клиентским роутингом про это легко забыть.



```
function Page() {
    const [loading, setLoading] = useState(true)
    const [data, setData] = useState([])

    useEffect(() => {
        (async () => {
            const response = await fetch('/data')
            setLoading(false)
            setData(await response.json())
        })()
    }, []);

    return (
        loading
            ? (<span>Loading</span>)
            : (data.map((el) => <span key={el.id}>{el.name}</span>))
    )
}
```

Роутинг

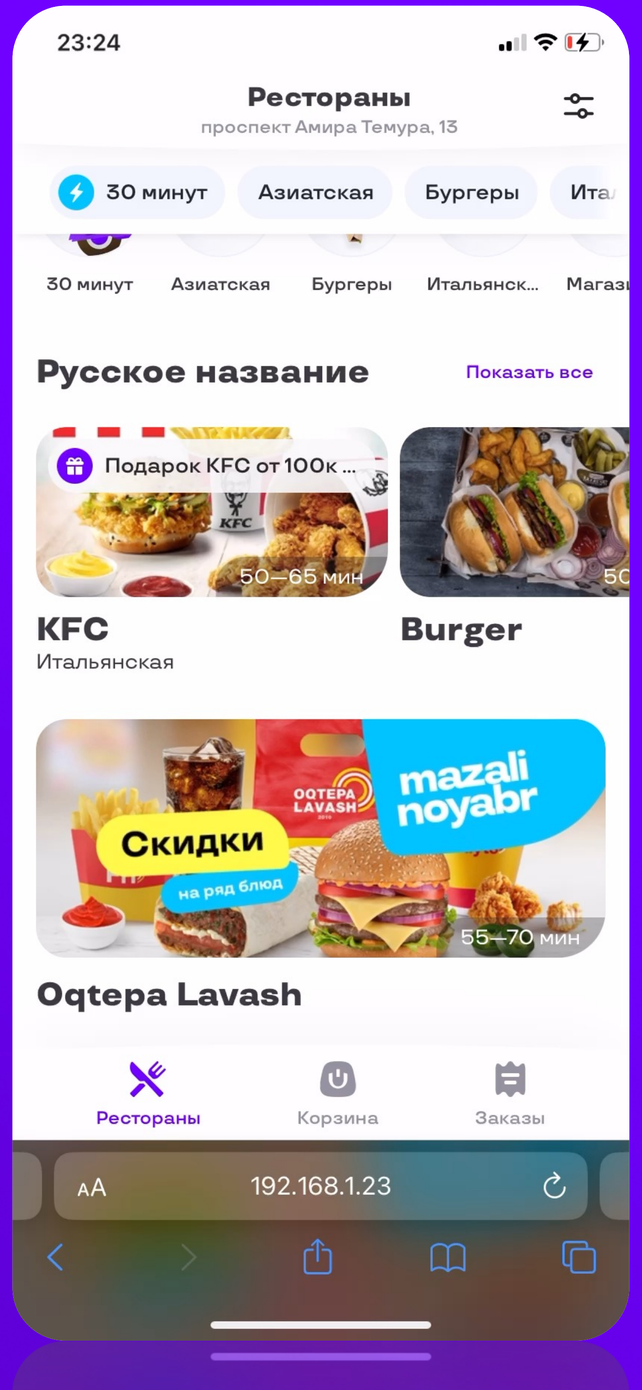
При клиентской навигации компонент монтируется еще раз.

Т.к. все данные хранились в state и процесс загрузки начинался в useEffect(), то весь рендер начинается заново.

Роутинг

Чтобы при монтировании мы работали с данными из кеша, стоит выносить их в отдельный network слой, например, с использованием

query/rtk-query.



```
function Page() {
    const {
        data,
        isLoading,
    } = useDataQuery()

    return (
        isLoading
            ? (<span>Loading</span>)
            : (data.map((el) => <span key={el.id}>{el.name}</span>))
    )
}
```

Авторизация

Авторизация

SSO

«Единый» профиль без SSO

SSO

native-app открывает ссылку на sso провайдера

→

SSO провайдер перехватывает cookie с токенами в запросе из нативного приложения и создает новые токены с той же авторизацией

→

SSO провайдер редиректит на веб-приложение, в ответе выставляя новые токены в cookie

→

Веб-приложение начинает работать с выписанными токенами

Если SSO нет

Backend нативного приложения валидирует токены

native-app открывает ссылку на web-приложение. Открывая ссылку, проставляет куки с токенами авторизации.

web-приложение при открытии проверяет, были ли установлены cookie из нативного приложения.

web-приложение идет в свой бекэнд для обмена токенов из нативного приложения на свои.

Backend web-приложения при обмене токенов декодирует токены, выясняя телефон или другой индификатор. Далее идет сверять токены в Backend нативного приложения.

Выяснив всю информацию из токена. Backend web-приложения выдает свои токены доступа, на этом этапе он уже знает гость это или авторизированный пользователь (ранее провалидировал).

Сравнение производительности

Сравнение производительности

LCP. Время показа первого экрана;

FPS в самых нагруженных участках runtime.

LCP

- LCP web vitals
- в нативных приложениях нет аналога, придется костылить самим

FPS

- нельзя брать данные из rum-пакетов, они будут учитывать fps в простое
- нет возможности использовать эмулятор Android / симулятор iOS при сравнении
- сравнивать нужно только на реальных устройствах

Время жизни WebView

Время жизни WebView

Если свернуть iOS-приложение больше чем на час, WebView выгружаются из памяти, хотя нативное приложение может оставаться в памяти.

Из-за этого слетают все сессионные данные.

Время жизни WebView

Не используйте хранилища, привязанные к сессии:

- SessionStorage
- Session Cookie

Если очень нужно, делегируйте хранение сессионных данных нативному приложению.

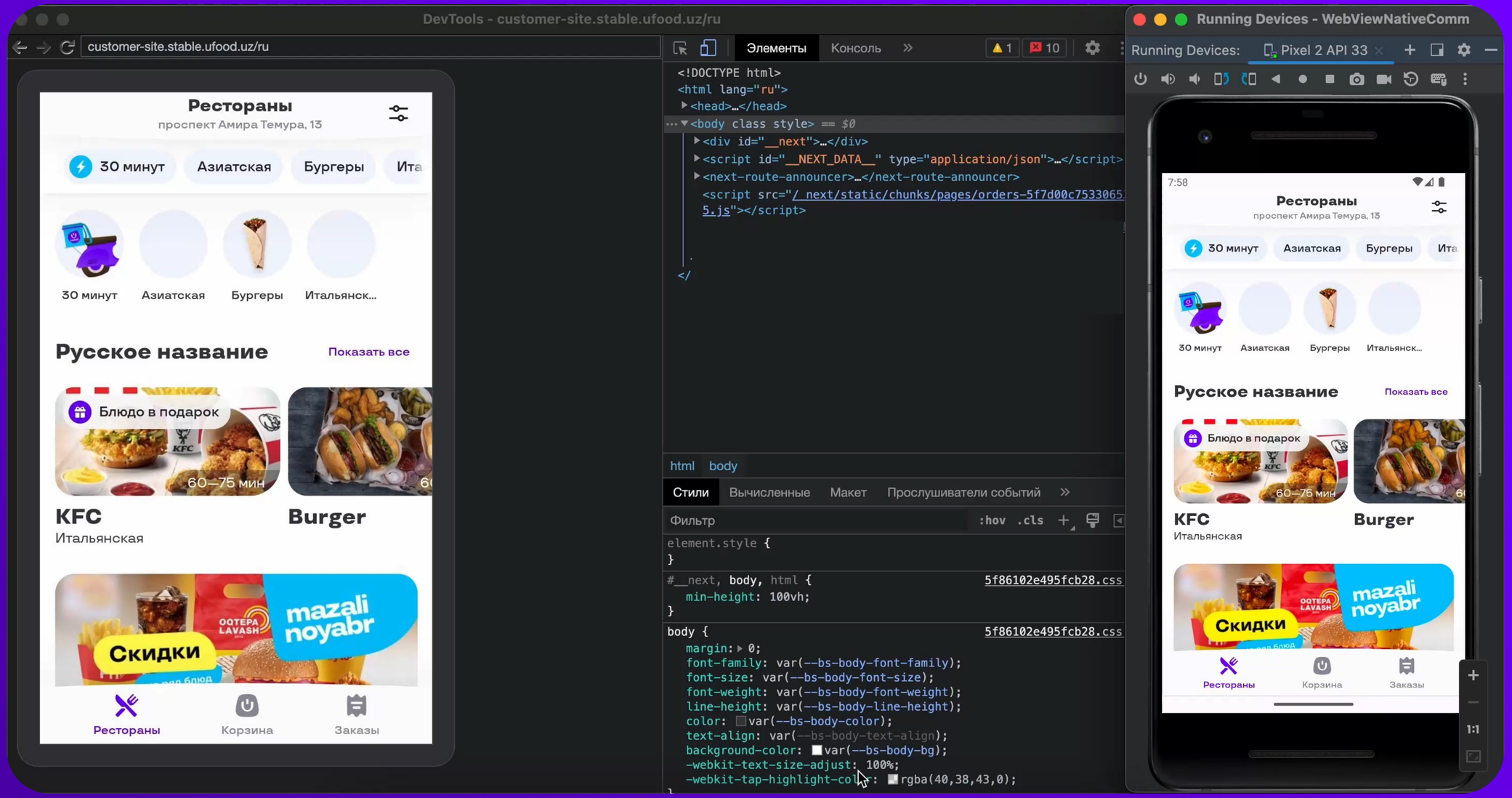
Пруфы

Отладка

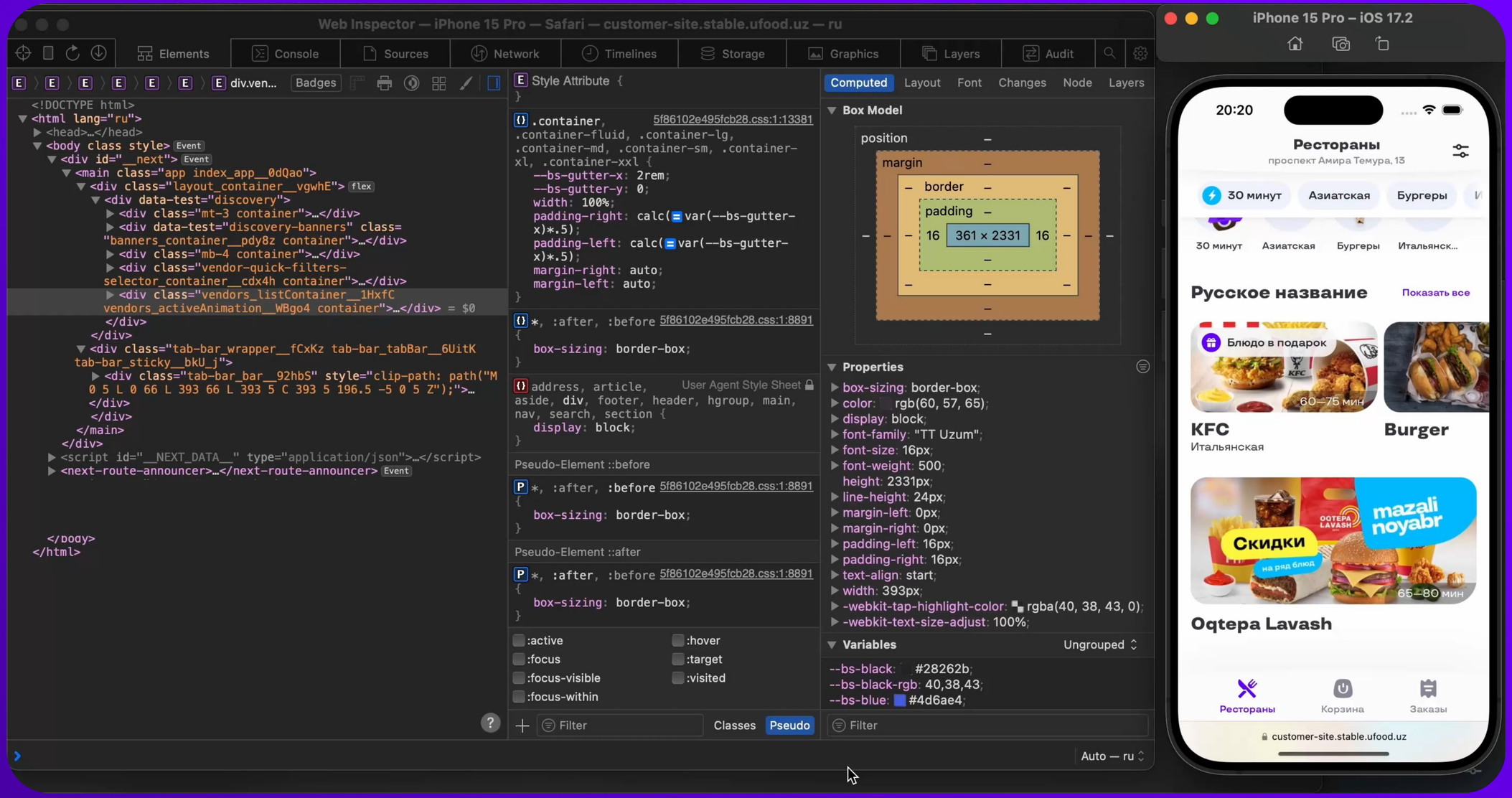
Отладка

- можно дебажить в devtools, в браузере и в WebView в нативном приложении
- работает с эмулятором Android и симулятором iOS

Android


DevTools - customer-site.stable.ufood.uz/ru
customer-site.stable.ufood.uz/ru
Рестораны
проспект Амира Темура, 13
30 минут
Азиатская
Бургеры
Ита
30 минут
Азиатская
Бургеры
Итальянск...
Русское название
Показать все
Блюдо в подарок
60–75 мин
KFC
Итальянская
Burger
Скидки
mazali noyabr
Рестораны
Корзина
Заказы
Элементы
Консоль
<!DOCTYPE html>
<html lang="ru">
<head>…</head>
<body class style> == $0
…
<script id="__NEXT_DATA__" type="application/json">…</script>
<next-route-announcer>…</next-route-announcer>
<script src="/_next/static/chunks/pages/orders-5f7d00c7533065
5.js"></script>
html body
Стили
Вычисленные
Макет
Прослушиватели событий
Фильтр
:hov .cls
element.style {
}
#__next, body, html {
min-height: 100vh;
}
5f86102e495fcb28.css
body {
margin: 0;
font-family: var(--bs-body-font-family);
font-size: var(--bs-body-font-size);
font-weight: var(--bs-body-font-weight);
line-height: var(--bs-body-line-height);
color: var(--bs-body-color);
text-align: var(--bs-body-text-align);
background-color: var(--bs-body-bg);
-webkit-text-size-adjust: 100%;
-webkit-tap-highlight-color: rgba(40,38,43,0);
Running Devices - WebViewNativeComm
Running Devices:
Pixel 2 API 33
7:58
Рестораны
проспект Амира Темура, 13
30 минут
Азиатская
Бургеры
Итальянск...
Русское название
Показать все
Блюдо в подарок
60–75 мин
KFC
Итальянская
Burger
Скидки
mazali noyabr
Рестораны
Корзина
Заказы

iOS


Web Inspector — iPhone 15 Pro — Safari — customer-site.stable.ufood.uz — ru
Elements
Console
Sources
Network
Timelines
Storage
Graphics
Layers
Audit
Computed
Layout
Font
Changes
Node
Layers
Box Model
Properties
Variables
Ungrouped
Filter
Classes
Pseudo
Auto — ru
iPhone 15 Pro – iOS 17.2
20:20
Рестораны
проспект Амира Темура, 13
30 минут
Азиатская
Бургеры
Русское название
Показать все
Блюдо в подарок
60–75 мин
KFC
Итальянская
Burger
Скидки
на ряд блюд
mazali noyabr
65–80 мин
Oqtepa Lavash
Рестораны
Корзина
Заказы
customer-site.stable.ufood.uz

Отладка

- работает с Simulator iOS / Emulator Android
- работает с реальными устройствами, подключив кабель
- основное время разработки будете дебажить как обычный сайт

Отладка

Android:

- chrome://inspect/#devices

iOS:

- Разработка => устройство => url.
- Могут быть проблемы с версиями iOS/macOS.
- Может помочь Safari Technology Preview, если обычный не подключается.

На реальном устройстве включите отладку по USB.

Тестирование

Тестирование

основные кейсы e2e можно прогнать через любимый e2e фреймворк (playwright/cypress etc);

тестировать взаимодействия нативного приложения и WebView можно через appium.

Что в итоге?

В итоге

- **Не забывайте**, что приложение будет запущено на телефоне. Телефон должен стать **основным** средством тестирования.
- В WebView **есть не все доступы**, которые есть у браузера.
- Переведите запросы, требующие доступов, на запросы к **нативному приложению** (push/geo etc).
- Помните, что использование WebView — это **компромисс** по производительности.

Спасибо за внимание!

@maksimlavrenyuk

maksim.lavrenyuk@outlook.com