

Яндекс Такси

8. Firebase

Кроссплатформенная разработка мобильных приложений на Flutter

Сергей Кольцов, разработчик

План

- | Что такое Firebase, возможности
- | Cloud Firestore — хранение данных
- | Cloud Messaging — отправка пушей
- | Пример проекта

Что такое Firebase

Платформа с инструментами для работы над мобильным приложением

Низкий порог входа

Масштабируемость решения

Что такое Firebase

Разработка

Crash-rate, производительность, тестирование на девайсах, аутентификация, база данных, ML.

Аналитика

События, поведение, активность, возвращаемость, группировка пользователей.

Remote

Удаленная конфигурация приложения, уведомления, A/B-тестирование

Cloud Firestore

NoSQL база данных

(→ Collection → Document) → Data

Обновление в реальном времени

Кэширование

Realtime Database vs Cloud Firestore

Модель данных

RD: Один большой json

CF: Коллекции документов

Масштабирование

RD: > 200к подключений / 1к операций — требуется разделение БД

CF: Масштабируется автоматически

Ценообразование

RD: Пропускная способность и хранение

CF: Количество выполняемых операций (чтение/запись/удаление)

Firestore Cloud Messaging (FCM)

- Push-уведомления пользователям

- Таргетинг

- Firestore-console или собственный сервер

Настройка проекта (Flutter)

Добавить зависимость firebase_core

При инициализации приложения выполнить
`await Firebase.initializeApp()`

Создать проект в Firebase Console

› На этом этапе:

- Android не собирается из-за multidex
- iOS ругается на отсутствие GoogleService-Info.plist.

Настройка проекта (Android)

Создать Android-приложение в Firebase Console

- › Указать packageName
- › Скачать google-services.json
- › Положить **google-services.json** в [project]/android/app/
- › Пропускаем пункт Add Firebase SDK

Настройка проекта (Android)

Добавить classpath в [project]/android/build.gradle

```
dependencies {  
    // other classpaths  
    // ...  
    classpath 'com.google.gms:google-services:4.3.4'  
}
```

Настройка проекта (Android)

Добавить в конец файла [project]/android/app/build.gradle

```
// Add at the bottom of a file
```

```
apply plugin: 'com.google.gms.google-services'
```

Настройка проекта (Android)

Только если нужна поддержка `minSdkVersion < 21`:
Включить `multidex` в файле `[project]/android/app/build.gradle`

```
android {  
    defaultConfig {  
        // ...  
        minSdkVersion 16  
        targetSdkVersion 30  
        multiDexEnabled true  
    }  
}  
  
dependencies {  
    implementation 'com.android.support:multidex:1.0.3'  
}
```

Настройка проекта (iOS)

Создать iOS-приложение в Firebase Console

- › Указать Bundle ID (При создании flutter-проекта snake_case в названии преобразуется в lowerCamelCase в Bundle ID)
- › Скачать GoogleService-Info.plist
- › Открыть папку [project]/ios в Xcode
- › Открыть Runner
- › File → Add files to Runner
- › Выбрать файл GoogleService-Info.plist, убедиться, что стоит галочка “Copy items if needed”

Настройка проекта (Web)

Создать Web-приложение в Firebase Console

- › Скопировать код из шага “Add Firebase SDK”
- › Добавить скопированный код в файл [project]/web/index.html

```
<html>

...

<body>

  <!-- Add copied code here -->

  <script src="main.dart.js" type="application/javascript"></script>

</body>

</html>
```

Flutter + Cloud Firestore

Добавить зависимость cloud_firestore

Create database в Firebase Console → Cloud Firestore

Firestore.instance.collection() — доступ к коллекции

collection.add(json) — добавления документа со сгенерированным id

collection.doc(id).set(json) — изменение или создание документа со своим id

collection.doc(id).delete() — удаление документа

Flutter + Cloud Firestore

Security Rules

Чтение разрешено, запись запрещена

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read: if true;
      allow write: if false;
    }
  }
}
```


Flutter + Cloud Firestore

Security Rules

Чтение и запись только для авторизованных пользователей

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

Flutter + Cloud Firestore

Security Rules

Чтение и запись только для владельца данных

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /counters/{id} {
      allow read, write: if request.auth != null && request.auth.uid ==
resource.data.author;
    }
  }
}
```

Flutter + Firebase Cloud Messaging

Добавить зависимость firebase_messaging

Android: если используем Flutter Android Embedding V2 (Flutter Version ≥ 1.12) — дополнительных настроек не требуется

iOS: требуется активный аккаунт Apple Developer, FCM интегрируется с APNs

Web: пока не поддерживается

Яндекс Такси

Спасибо

Сергей Кольцов

Разработчик



ringov@yandex-team.ru