

Цели практической работы

- Применить знания о работе с Firebase к проекту.
- Закрепить на практике принципы работы с сервисом Crashlytics и уведомлениями, сообщениями от сервиса Firebase.

Что входит в задание

1. Подключить Firebase к проекту.
2. Настроить проект для получения отчётов о сбоях и ошибках, происходящих в приложении.
3. Настроить проект для получения пуш-уведомлений от сервиса Firebase.

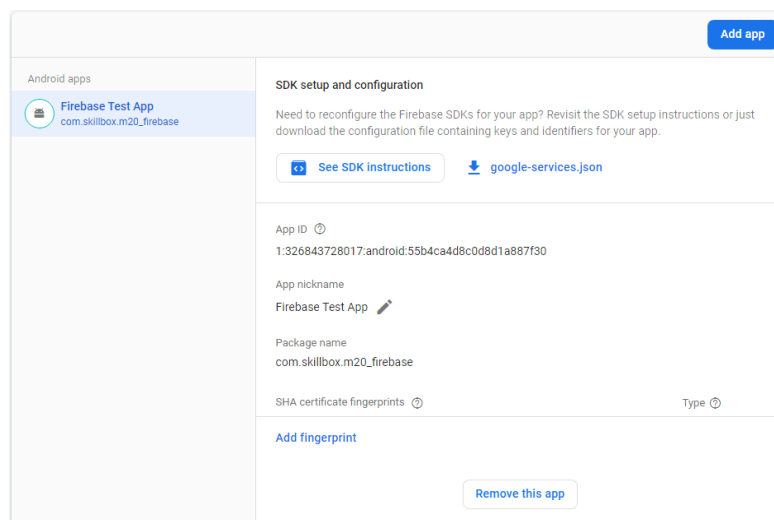
Подготовка к работе

Убедитесь, что девайс или эмулятор поддерживают google-services: без этого не будет возможности протестировать работу пуш-уведомлений. Для выполнения практической работы вы можете установить на эмулятор образ Android, включающий в себя google-services.

Что нужно сделать

Для практической работы можете использовать приложение «Достопримечательности», которое разрабатывали в прошлых модулях. Помимо самого проекта, для зачёта практической работы нужно предоставить скриншоты, которые демонстрируют работу подключённых сервисов Firebase.

1. Создайте проект [в консоли Firebase](#).
2. Добавьте к созданному проекту Android-приложение:
 - зарегистрируйте приложение в Firebase;
 - скачайте полученный файл google-services.json и добавьте его в проект, в папку app (или аналогичную; ориентиром может служить файл build.gradle уровня приложения, а не проекта);
 - если ранее вы уже регистрировали приложение, то повторно найти и скачать файл google-services.json можно в разделе Project Settings.



3. Подключите необходимые библиотеки и плагины в файлах build.gradle уровня приложения и уровня проекта в соответствии с инструкцией (найдите её в консоли разработчика при создании проекта).

Если у вас возникает ошибка, связанная с репозиторием Google, попробуйте удалить этот репозиторий из секции allProjects в файле

build.gradle уровня проекта. Если это не поможет — обратитесь за помощью в конфигурации к проверяющему куратору.

4. Подключите сервис Crashlytics к приложению в соответствии с [инструкцией](#).
5. Убедитесь, что всё подключено верно: спровоцируйте краш в приложении. Отчёт об ошибке должен появиться в консоли Firebase в разделе Crashlytics.
6. Добавьте в приложение уведомление для пользователя при совершении какого-то действия. Для этого:
 - создайте channel для уведомления;
 - создайте сам UI-компонент Notification;
 - наполните его необходимыми данными;
 - отобразите Notification с помощью NotificationManagerCompat в нужный момент.
7. Подключите сервис Messaging к приложению в соответствии с [инструкцией](#).
8. Убедитесь, что всё подключено верно: отправьте себе тестовое сообщение с помощью утилиты [Notification Composer](#) в консоли Firebase. Если сообщение не приходит:
 - Убедитесь, что приложение в background. Notification-сообщения обрабатываются автоматически, только когда приложение в background. Помните, что для обработки сообщения любого типа в foreground нужен отдельный сервис.
 - Подождите какое-то время. У сервиса FCM могут быть задержки при обработке сообщений.
9. **Выполните по желанию:**
 - Добавьте сервис для обработки пуш-сообщений. Создайте класс, который наследуется от FirebaseMessagingService.
 - Зарегистрируйте его в AndroidManifest.
 - Переопределите функцию onMessageReceived, напишите необходимый код для обработки сообщения и отображения его в виде уведомления.
 - С помощью Firebase Messaging API и сервиса Postman отправьте себе несколько сообщений и убедитесь, что они приходят, а также отображаются в виде уведомлений и когда приложение в foreground, и когда приложение в background. Пошаговую инструкцию по работе с Firebase Messaging API и сервисом Postman можно найти в презентации к занятию.

Рекомендации по выполнению

- [Создание проекта в Firebase и добавление приложения](#)
- [Подключение Crashlytics](#)
- [Подключение Messaging Service](#)
- [Обработка получаемых сообщений от FCM](#)
- [Создание channel для уведомлений](#)
- [Создание уведомления](#)

Если с какими-то пунктами задания возникнут проблемы, попросите помощи у куратора.

Что оценивается

- Выполнены все обязательные пункты задания.

- При краше приложения отчёт об ошибке приходит в сервис Crashlytics.
- В приложении реализован подход MVVM.
- Код чистый, у переменных и компонентов понятные названия, соблюдаются принципы ООП.

Как отправить работу на проверку

1. Используйте репозиторий `android_dev_1_2022`.
2. Скачайте изменения в репозитории на ваш компьютер.
3. Выполните практическую работу в папке `m20_firebase`. Отправьте коммиты в удалённый репозиторий.
4. Приложите скриншоты, на которых видно:
 - краши в Crashlytics;
 - сообщение, которое отправляется с помощью Notification Composer, и уведомление, которое отображается;
 - для задания по желанию: настройки Postman, отправляемое сообщение и отображаемое уведомление, обработанное с помощью вашего сервиса FirebaseMessagingService.