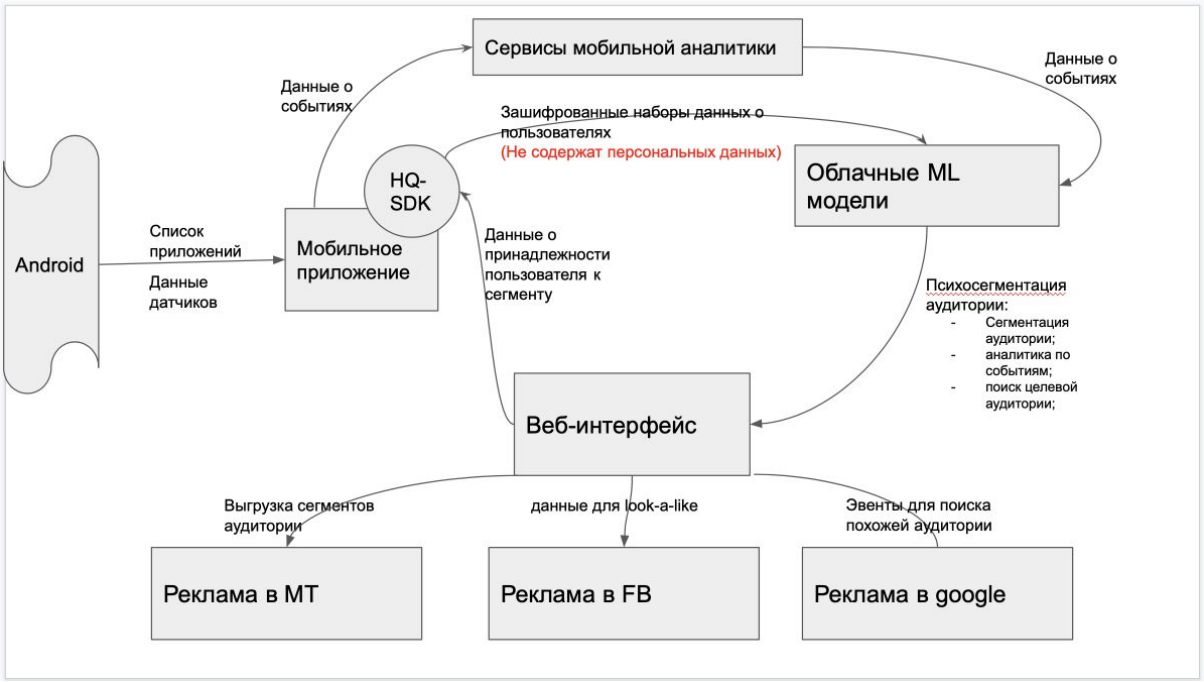
HUMANTEQ.

Инструкция по интеграции HQ SDK в мобильное приложение для фреймворка expo.io

Общая схема работы продукта

HQ-Monitor состоит из следующих частей:

- SDK, которое встраивается в мобильное приложение и отвечает за сбор и первичную обработку данных.
- ML моделей, которые обрабатывают полученную от SDK информацию в нашем облаке, опираясь на данные, собранные нашей научной командой.
- Веб-интерфейса, в котором отображается информация и который позволяет настраивать взаимодействие с различными внешними сервисами.



Порядок работ для интеграции SDK в мобильное приложение

Для интеграции системы аналитики пользователей с использованием психографики необходимо провести следующие действия:

1) Добавить в код мобильного приложения на android код нашего SDK
2) Настроить эвенты для работы для дальнейшей сегментации пользователей

Интеграция SDK в мобильное приложения

Выполните следующие действия

```
1. npm install -g expo-cli
2. cd /projects/project_name/
3. expo eject
```

Далее интеграция похожа на процесс, описанный в гайде:

4. Добавить в /projects/project_name/android/build.gradle

```
buildscript {
  repositories {
    ...
    maven { url 'https://dl.bintray.com/humanteq/HQM/' }
  }
  ...
}

allprojects {
  repositories {
  ...
  maven { url 'https://dl.bintray.com/humanteq/HQM/' }
  }
}
```

5. Добавить в /projects/project_name/android/app/build.gradle:

```
dependencies {
  ...

  implementation "io.humanteq.hqsdk:hqsdk-core:1.0.6"
  implementation "io.humanteq.hqsdk:hqsdk-apps:1.0.4"
  implementation "io.humanteq.hqsdk:hqsdk-files:1.0.4"
  implementation "io.humanteq.hqsdk:hqsdk-haptic:1.0.4"
  implementation "io.humanteq.hqsdk:hqsdk-location:1.0.4"
  implementation "io.humanteq.hqsdk:hqsdk-sensors:1.0.4"
}
```

io.humanteq.hqsdk:hqsdk-core - обязателен, остальные модули опциональны.

6. Изменить в /projects/project_name/android/app/build.gradle compileSdkVersion и targetSdkVersion:

```
android {
  compileSdkVersion 28

  ...
  defaultConfig {

  ...

    targetSdkVersion 28

  ...
  }
}
```

7. Добавить /projects/project_name/android/gradle.properties:

```
...
android.useAndroidX=true
android.enableJetifier=true
```

8. Добавить в /projects/project_name/android/app/src/main/java/path/to/your/MainApplication.java следующий код (debug и модули на усмотрение разработчиков):

```
  @Override
  public void onCreate() {
    super.onCreate();

    HQSdk.enableDebug(true);
    HQSdk.init(this, "sdk-api-key", false, new HqmCallback<HqmApi>() {

      @Override
      public void onSuccess(HqmApi hqmApi) {
        HQSdk.start(new SensorsCollector(MainApplication.this));
        HQSdk.start(new HapticCollector());
        HQSdk.start(new InstalledApplicationsCollector());
        HQSdk.start(new ActivityCollector());
        HQSdk.start(new LocationCollector());
        HQSdk.start(new ApplicationsStatisticsCollector());
        HQSdk.start(new FileStatisticsCollector());
      }

      @Override
      public void onError(@NotNull Throwable throwable) {

      }
    } );
  }
```

9. Синхронизировать, собрать проект (вручную, а не через expo.io).

Инициализация

Модули

Сбор статистики по приложениям

1. Подключить зависимость **implementation 'ru.livli:swsdk-apps:0.0.1-alpha17'**
2. Начать сбор:

```
instance.start(new InstalledApplicationsCollector(context)); // сбор установленных
приложений
```

```
instance.start(new ApplicationsStatisticsCollector(context)); // сбор статистики использования
```

Дополнительные методы

- `SWSdk.setGender(String gender); // Задать пол клиента (опционально, повышает точность работы SDK)`
- `SWSdk.setBirthyear(int birthyear ); // Задать год рождения (опционально, повышает точность работы SDK)`
- `SWSdk.setId(String opaqueId); // передает внутренний ID пользователя, необходим для работы с API и сопоставления пользователя в HQ SDK с внутренними системами.`

Работа с событиями

Вариант 1

Подключить приложение к Firebase на тарифе “Blaze” , фиксировать в нем интересные события.
Подключить BiqQuery. (Аналитика будет проходить в ручном режиме).

Вариант 2

Фиксировать интересующие события с помощью метода `SWSdk.logEvent` Hq Sdk.

```
Map<String, Object> map = new HashMap<>();
        map.put("property1", "test1");
        map.put("property2", "test2");
SWSdk.logEvent("event_name", map)
```