

SmartKeeper

**Подготовка нового приложения для публикации
в Google Play.**

**Программное обеспечение «Система
электронных кошельков SmartKeeper».**

Руководство пользователя

АННОТАЦИЯ

Описание подготовки нового приложения для публикации в Google Play

Версия: 1.2

Руководство актуально для ПО «Система электронных кошельков SmartKeeper» версий 3.5x и выше



Подготовка нового приложения для публикации в Google Play.
Программное обеспечение
«Система электронных кошельков SmartKeeper».
Руководство пользователя

2008–2022 ООО «Софт-Лоджик», г. Барнаул, Россия

Данный документ входит в комплект поставки программных продуктов.

Права использования данного документа предусмотрены соответствующим лицензионным договором.

ООО «Софт-Лоджик»

656006, г. Барнаул, Малахова ул., дом 146в

Тел: (3852) 72-27-27

© *Soft-logic*

Web: <https://soft-logic.ru/>

Mail: info@soft-logic.ru

ОГЛАВЛЕНИЕ

ИСТОРИЯ ИЗМЕНЕНИЙ.....	4
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.....	4
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.2.....	4
1 ИДЕНТИФИКАТОР ПРИЛОЖЕНИЯ.....	5
2 СОЗДАНИЕ И НАСТРОЙКА АККАУНТА FIREBASE.....	6
3 СОЗДАНИЕ КЛЮЧА ДЛЯ GOOGLE КАРТ.....	14
4 СОЗДАНИЕ ОГРАНИЧЕННОГО КЛЮЧА ДЛЯ GOOGLE КАРТ.....	18
5 НАСТРОЙКА НА СТОРОНЕ АДМИНИСТРИРОВАНИЯ КОШЕЛЬКА.....	25
6 ПОДПИСАНИЕ СБОРКИ *-RELEASE-UNSIGNED.APK ПЕРЕД ПУБЛИКАЦИЕЙ.....	27
6.1 СОЗДАНИЕ СЕРТИФИКАТА.....	27
6.2 ПОДПИСАНИЕ ПРИЛОЖЕНИЯ.....	29

ИСТОРИЯ ИЗМЕНЕНИЙ**ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1**

Дата публикации: 19.02.2021.

Изменение	Раздел
Дополнения в документации к ранее имевшемуся функционалу:	
Добавлен раздел, описывающий процесс подписания приложения сертификатом для публикации в Google Play	6

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.2

Дата публикации: 04.02.2022.

Изменение	Раздел
Дополнения в документации к ранее имевшемуся функционалу:	
Обновлен раздел, описывающий процесс подписания приложения	6.2

1 ИДЕНТИФИКАТОР ПРИЛОЖЕНИЯ

Придумайте **id** приложения — уникальный идентификатор приложения в Google Play, состоящий, как правило, из трёх слов на английском языке (допускается как больше, так и меньше). Слова должны быть разделены точками, например: **ru.softlogic.smartkeeper.**

2 СОЗДАНИЕ И НАСТРОЙКА АККАУНТА FIREBASE

Для создания и настройки аккаунта Firebase необходим **id** приложения, придуманный на шаге 1.

Перейдите на страницу <https://console.firebase.google.com> и выполните следующие действия:

1. Добавьте проект, нажав на соответствующую кнопку (рисунок 2.1).

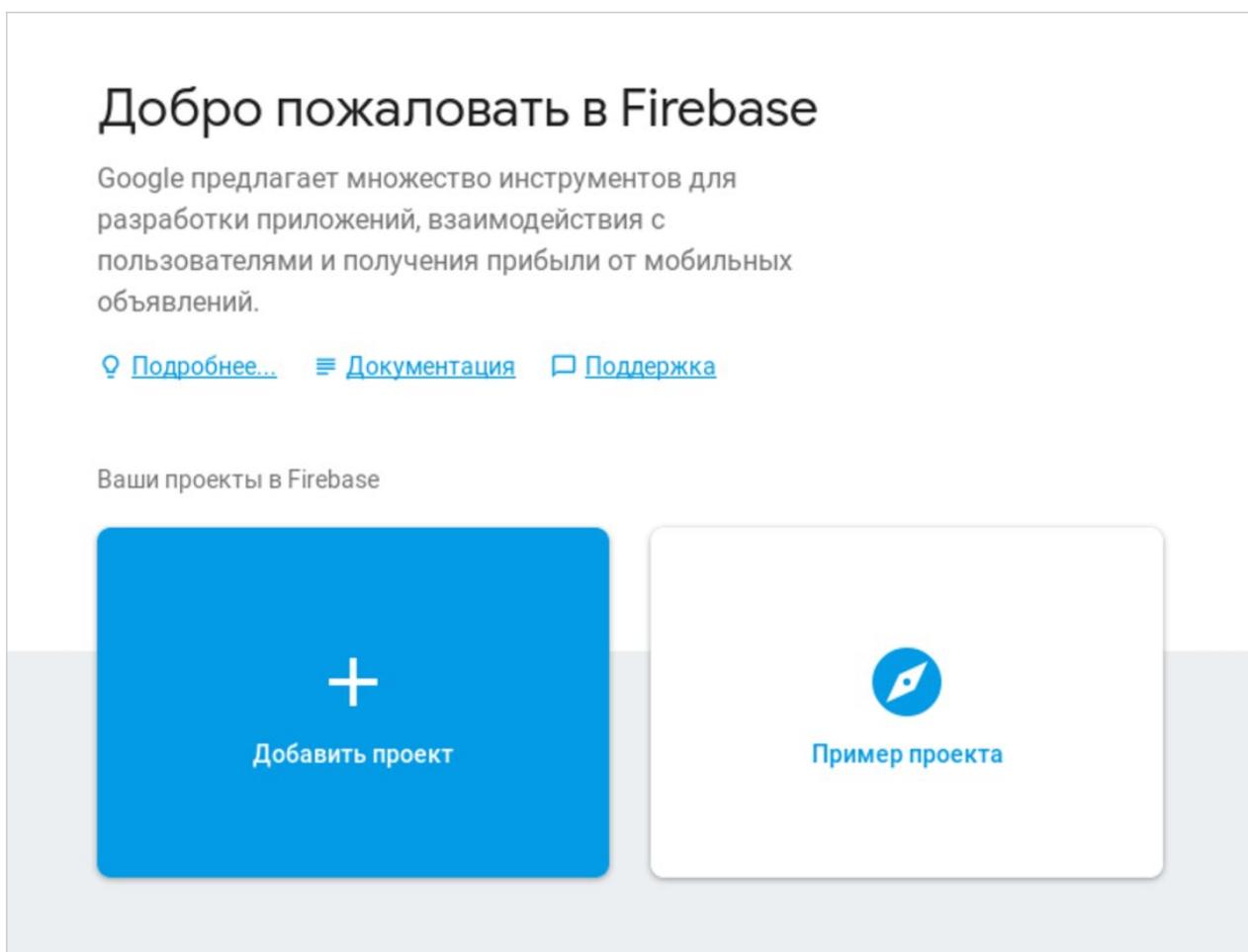


Рисунок 2.1 — Создание проекта

2. Заполните все необходимые поля в зависимости от местоположения (рисунок 2.2).

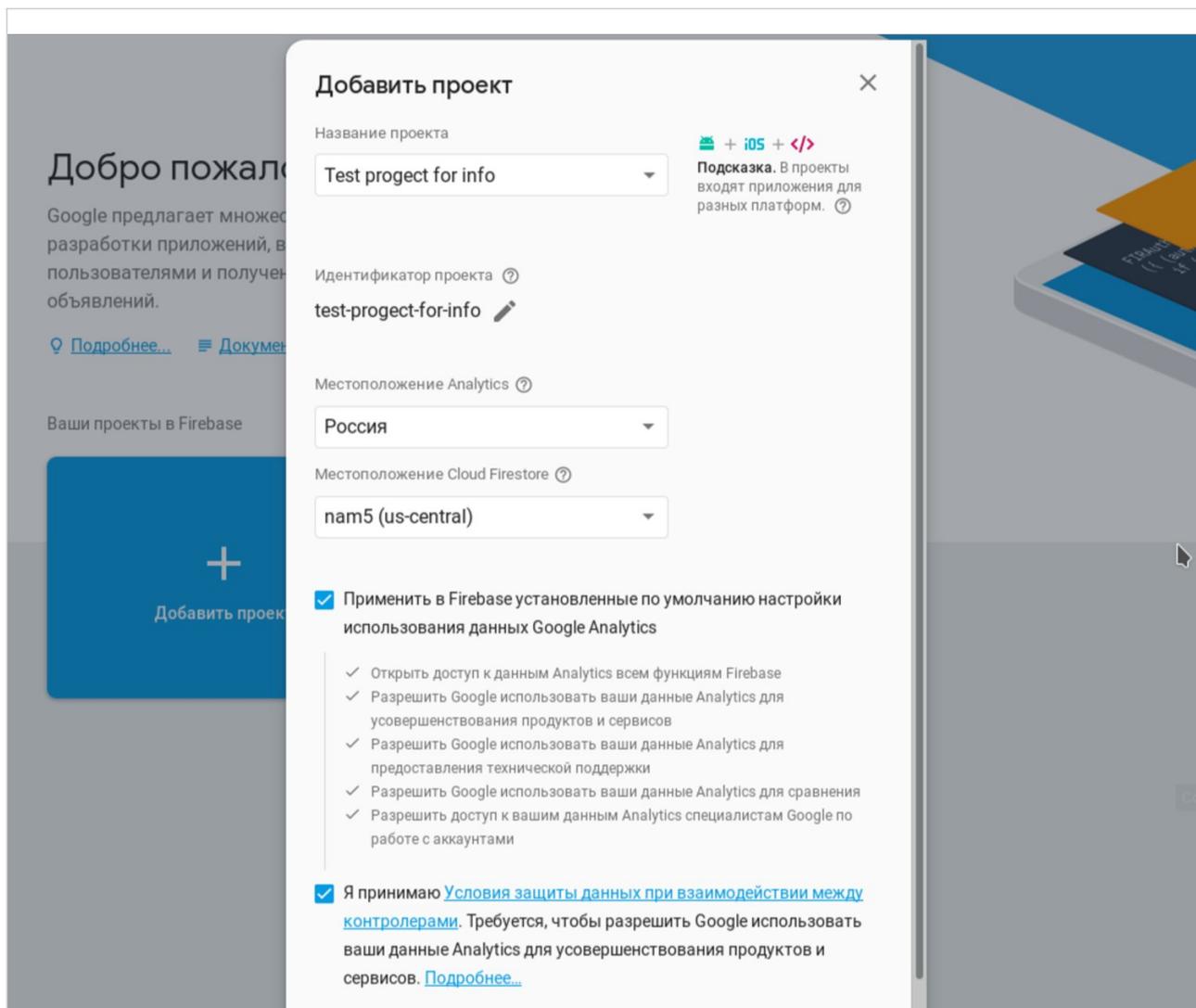


Рисунок 2.2 — Заполнение данных для нового проекта

3. После создания проекта добавьте приложение, для которого будут использоваться push-уведомления. В созданном проекте в левом верхнем углу перейдите в настройки (значок шестеренки) и нажмите **«Настройки проекта»** (рисунок 2.3).

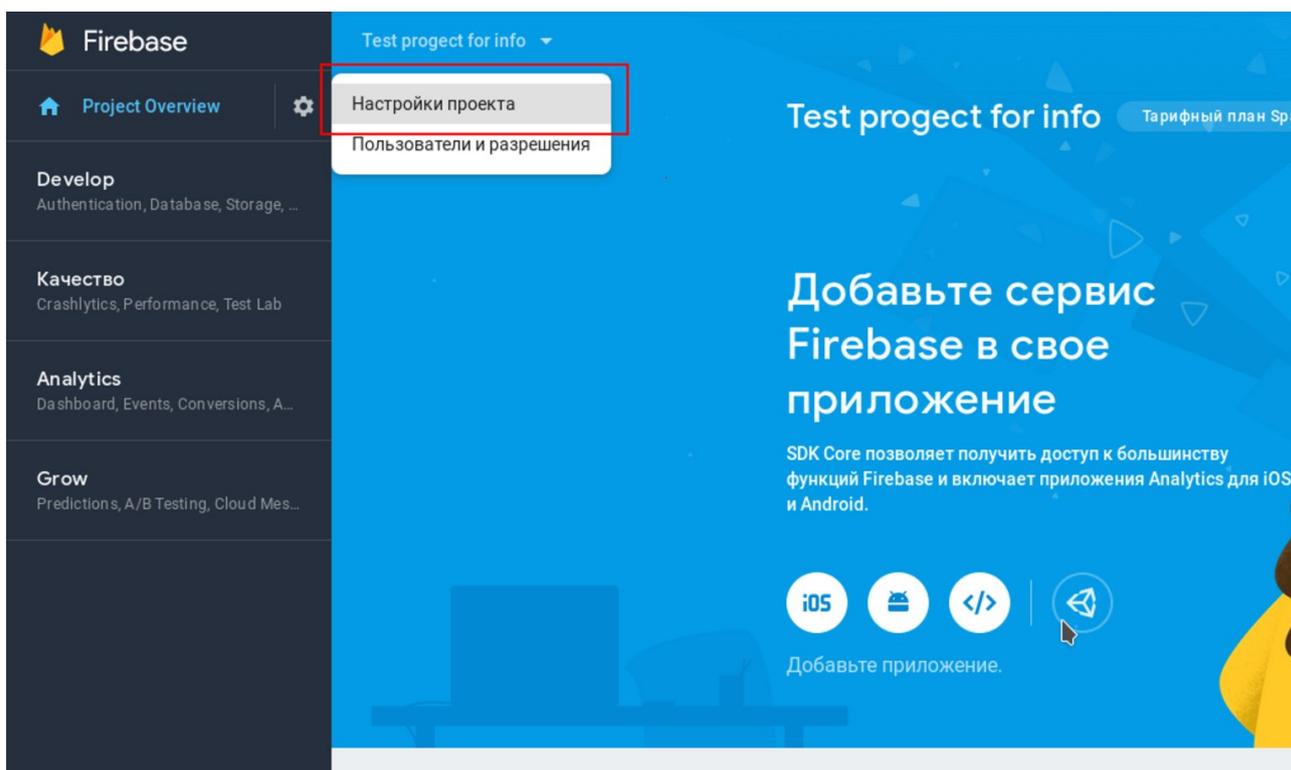


Рисунок 2.3 — Настройки проекта

4. В настройках выберите вкладку «Общие» и спуститесь в самый низ страницы до блока «Ваши приложения», нажмите на значок **Android** (рисунок 2.4). Откроется экран добавления приложения.

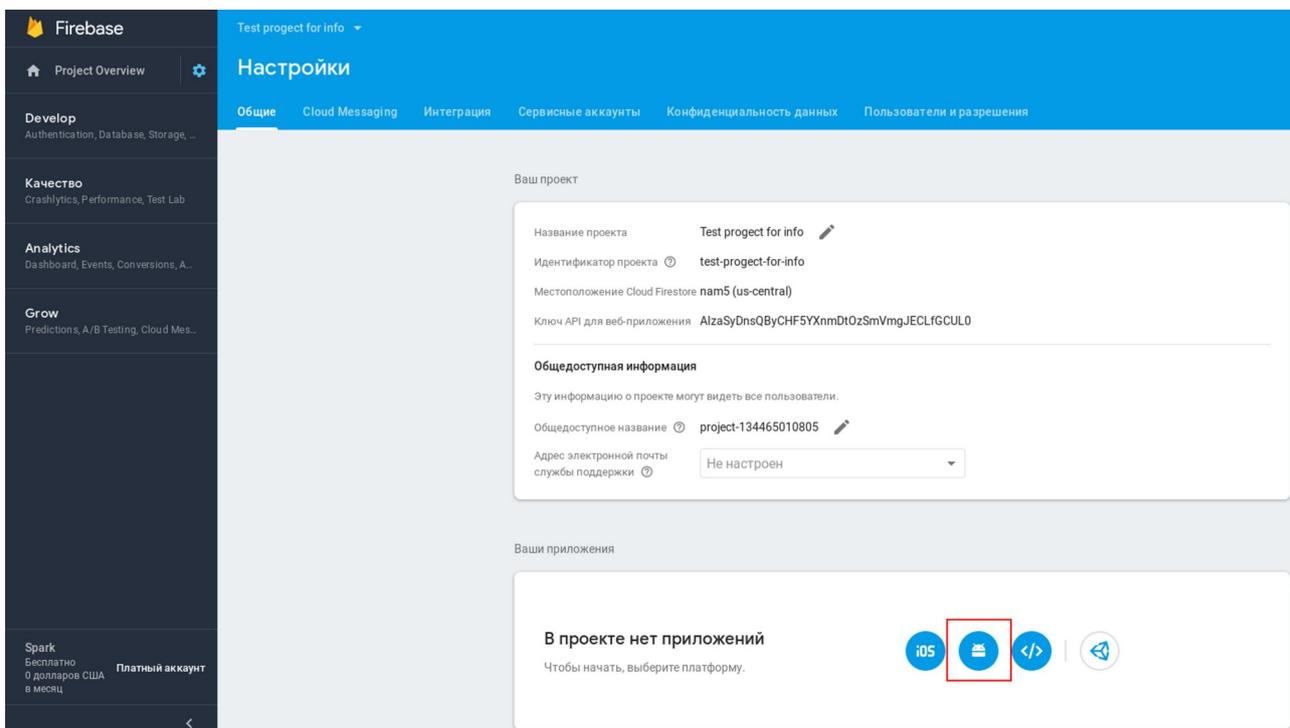
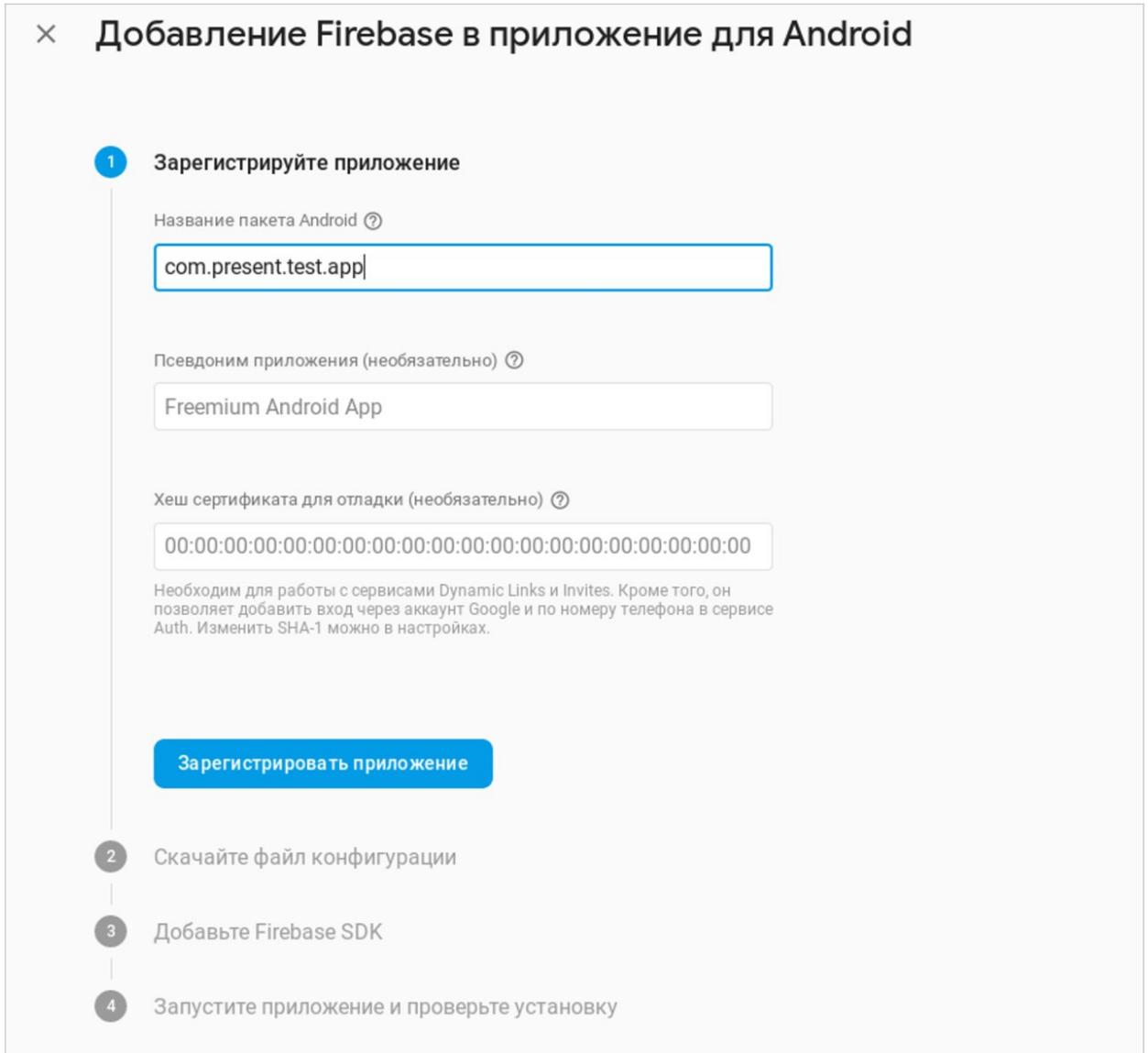


Рисунок 2.4 — Приложения проекта

5. Заполните все обязательные поля, укажите **id** приложения, придуманный на шаге 1 (рисунок 2.5).



× **Добавление Firebase в приложение для Android**

1 Зарегистрируйте приложение

Название пакета Android ⓘ

Псевдоним приложения (необязательно) ⓘ

Хеш сертификата для отладки (необязательно) ⓘ

Необходим для работы с сервисами Dynamic Links и Invites. Кроме того, он позволяет добавить вход через аккаунт Google и по номеру телефона в сервисе Auth. Изменить SHA-1 можно в настройках.

Зарегистрировать приложение

2 Скачайте файл конфигурации

3 Добавьте Firebase SDK

4 Запустите приложение и проверьте установку

Рисунок 2.5 — Регистрация приложения

6. Скачайте `google-service.json`. Данный файл необходимо передать службе технической поддержки для дальнейшего добавления в проект (рисунок 2.6). Оставшиеся шаги пропустите.

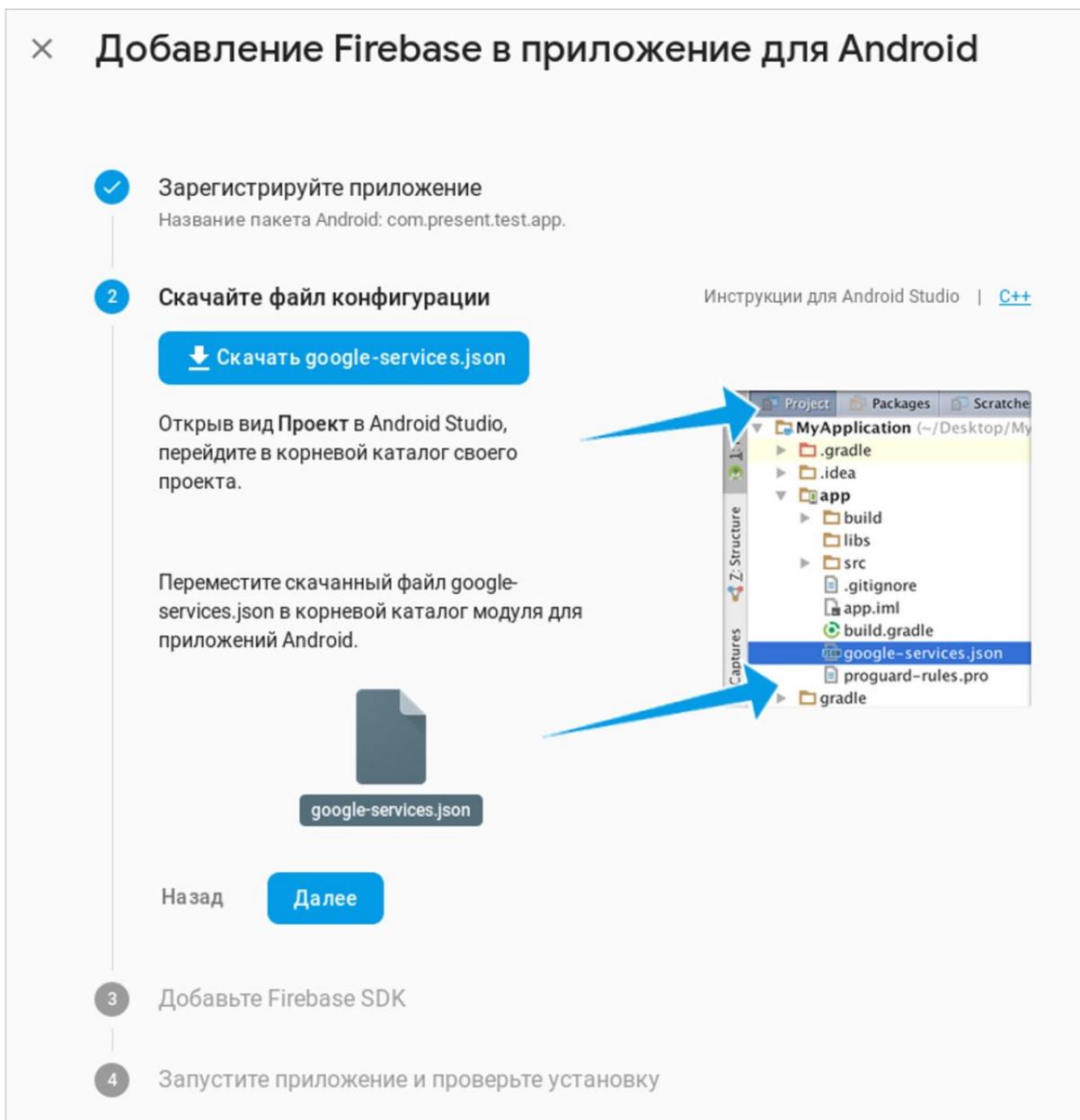


Рисунок 2.6 — `google-service.json`

Если по какой-либо причине файл `google-service.json` был утерян, его можно восстановить: для этого в настройках на вкладке «Общие» в блоке «Ваши приложения» выберите соответствующее приложение и нажмите кнопку `google-service.json` (рисунок 2.7).

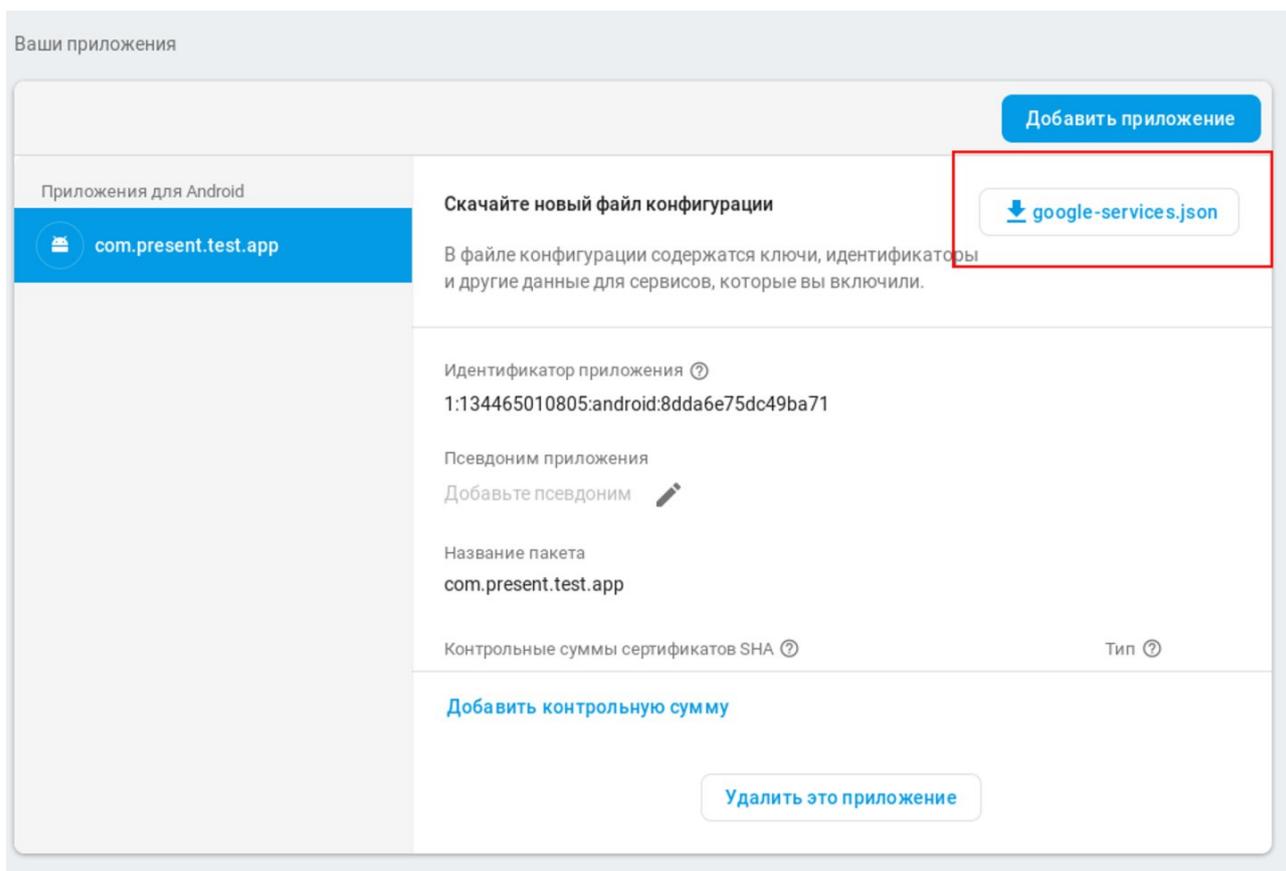


Рисунок 2.7 — Восстановление google-service.json

7. Из настроек на вкладке Cloud Messaging необходимо передать службе технической поддержки значение из поля **«Ключ сервера»** (рисунок 2.8). Если ключ отсутствует, создайте его, нажав на кнопку **«Добавить ключ сервера»**: он сгенерируется автоматически.

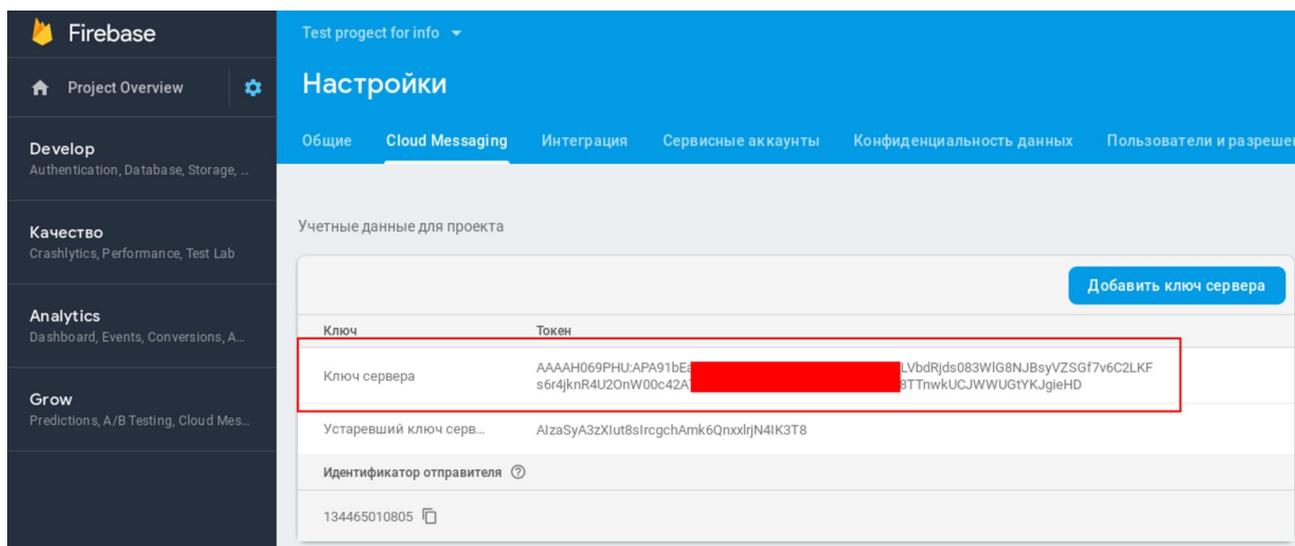


Рисунок 2.8 — Ключ сервера

3 СОЗДАНИЕ КЛЮЧА ДЛЯ GOOGLE КАРТ

Откройте страницу <https://cloud.google.com/mapsplatform>, чтобы создать платный ключ для Google Карт. Для создания бесплатного ограниченного ключа перейдите в раздел 4.



Внимание!

Не нужно создавать оба ключа. Нужно создать либо платный ключ для Google Карт, как это описано в текущем разделе, либо бесплатный, — как это описано в разделе 4.

Для использования сервиса Google Maps зарегистрируйтесь и оплатите тариф карт.

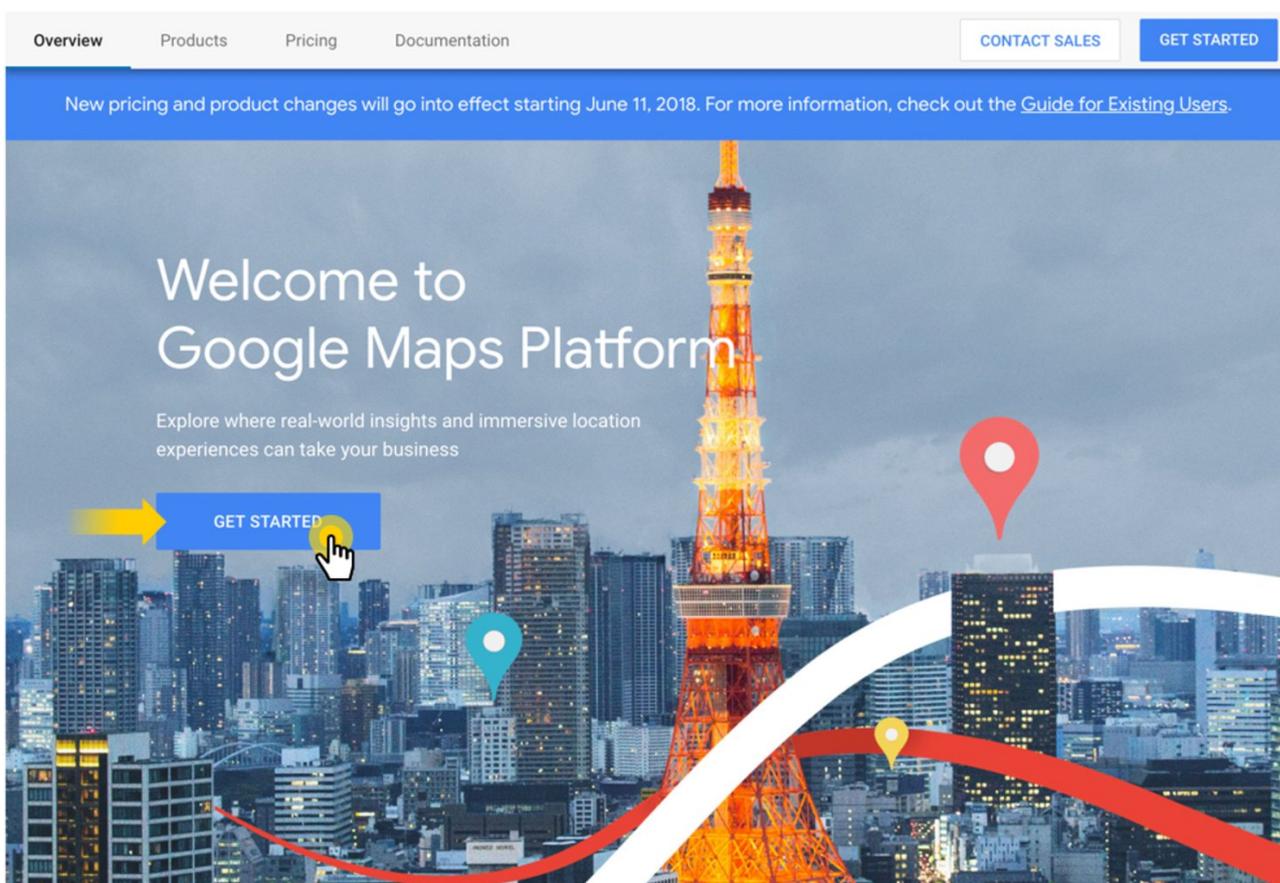


Рисунок 3.1 — Google Maps

Далее выберите проект **firebase**, созданный в разделе 2.

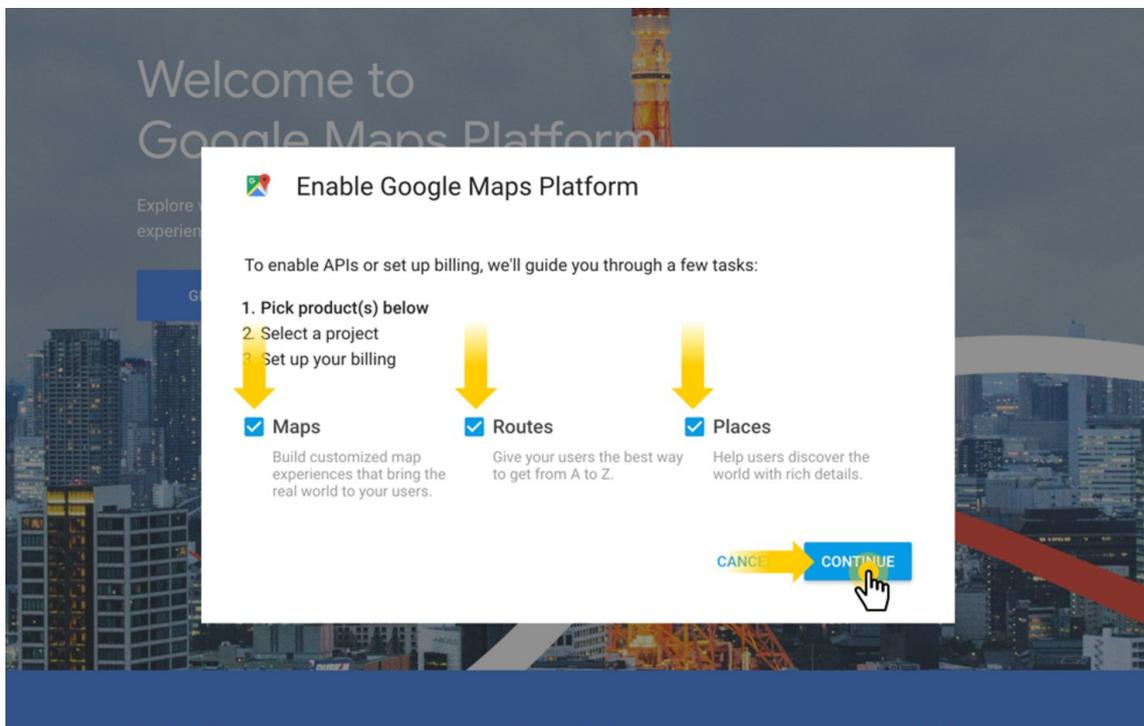


Рисунок 3.2 — Создание аккаунта на Google Maps

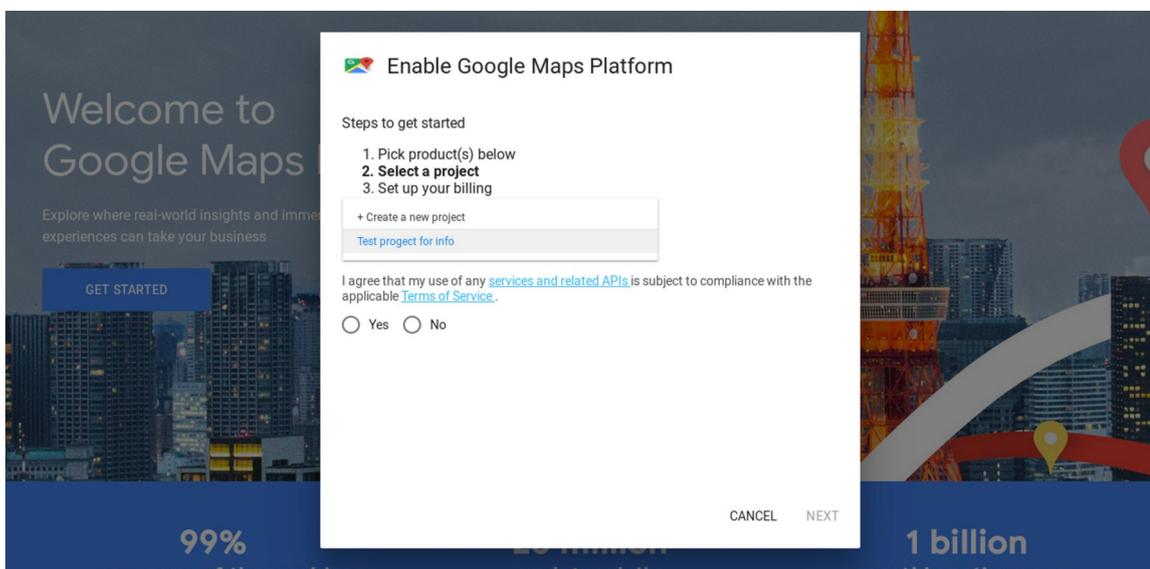


Рисунок 3.3 — Выбор проекта firebase

Создайте платёжный аккаунт (рисунок 3.4).

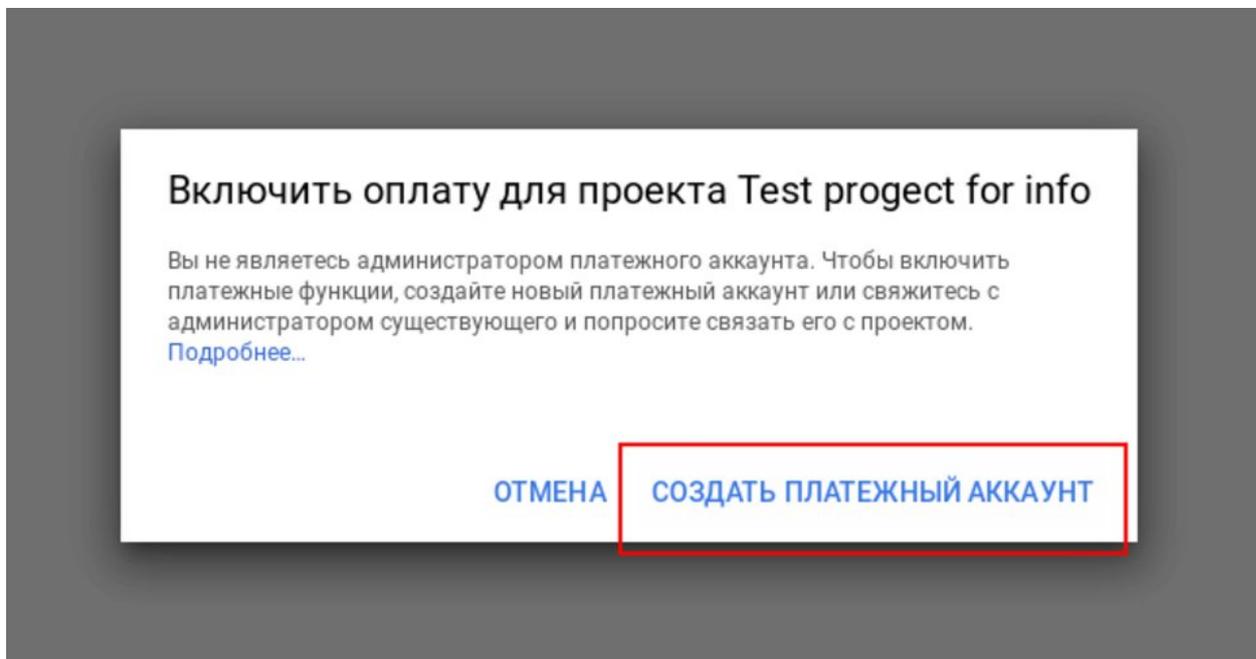


Рисунок 3.4 — Создание платёжного аккаунта

В открывшемся окне скопируйте API KEY и передайте его службе технической поддержки (рисунок 3.5, 3.6).

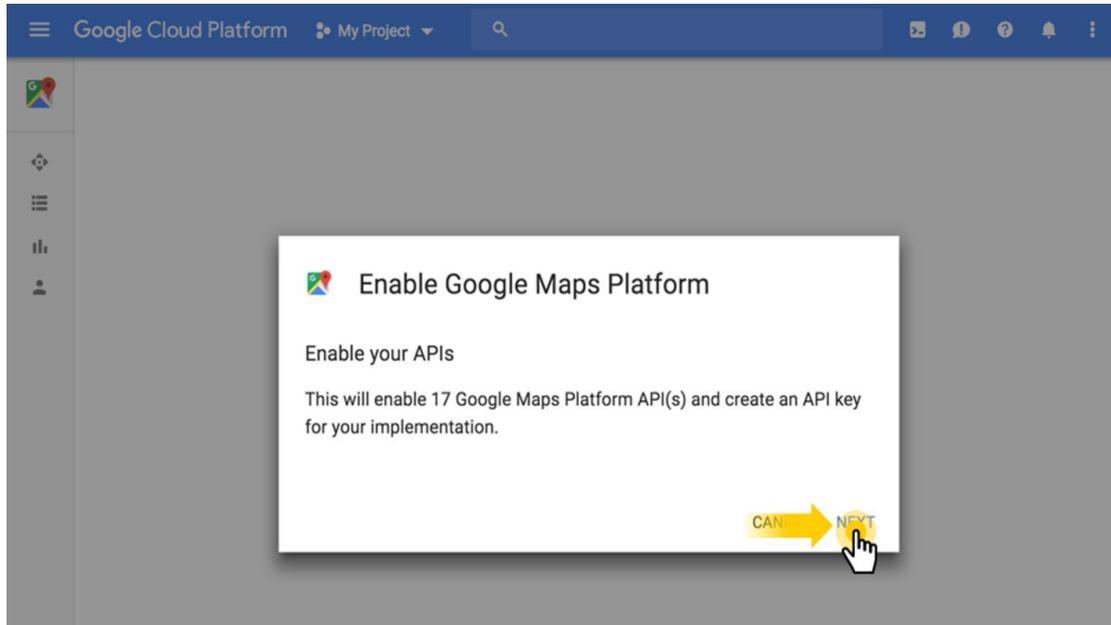


Рисунок 3.5 — Включение Google Maps

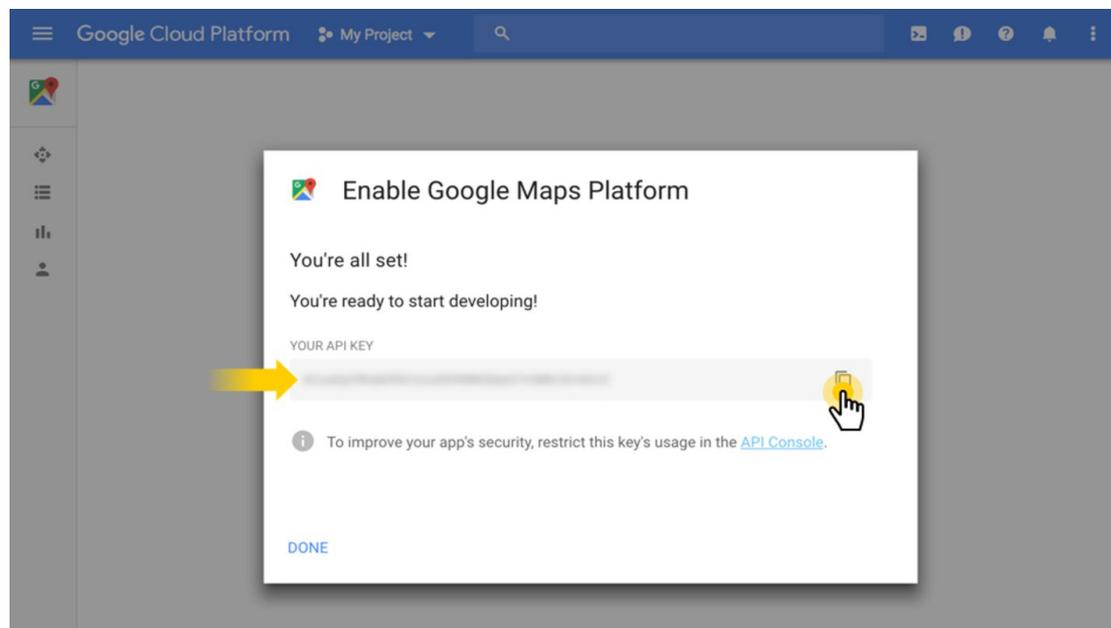


Рисунок 3.6 — Получение API KEY

4 СОЗДАНИЕ ОГРАНИЧЕННОГО КЛЮЧА ДЛЯ GOOGLE КАРТ

Для ограниченного использования сервиса Google Maps перейдите в [КОНСОЛЬ разработчика](#) и выберите проект (рисунки 4.1, 4.2).

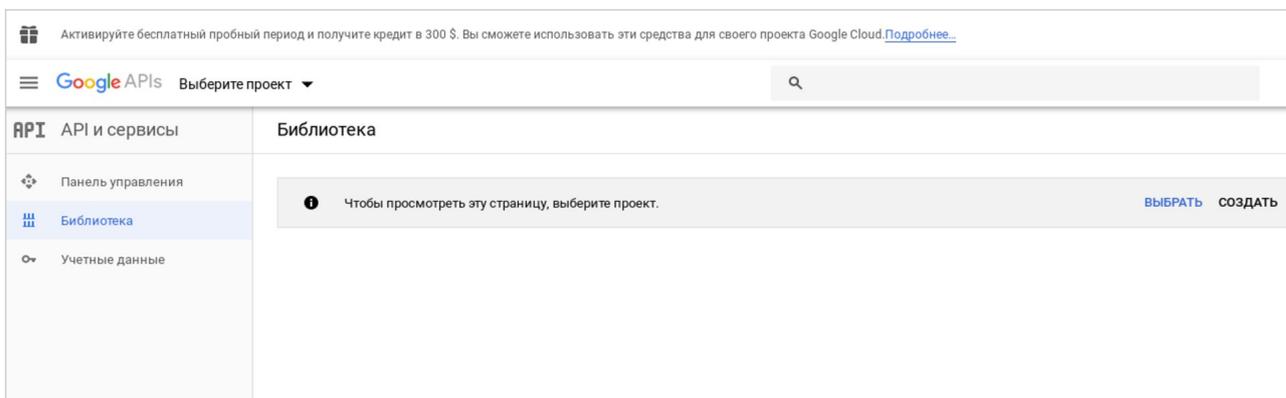


Рисунок 4.1 — Выбор проекта. Часть 1

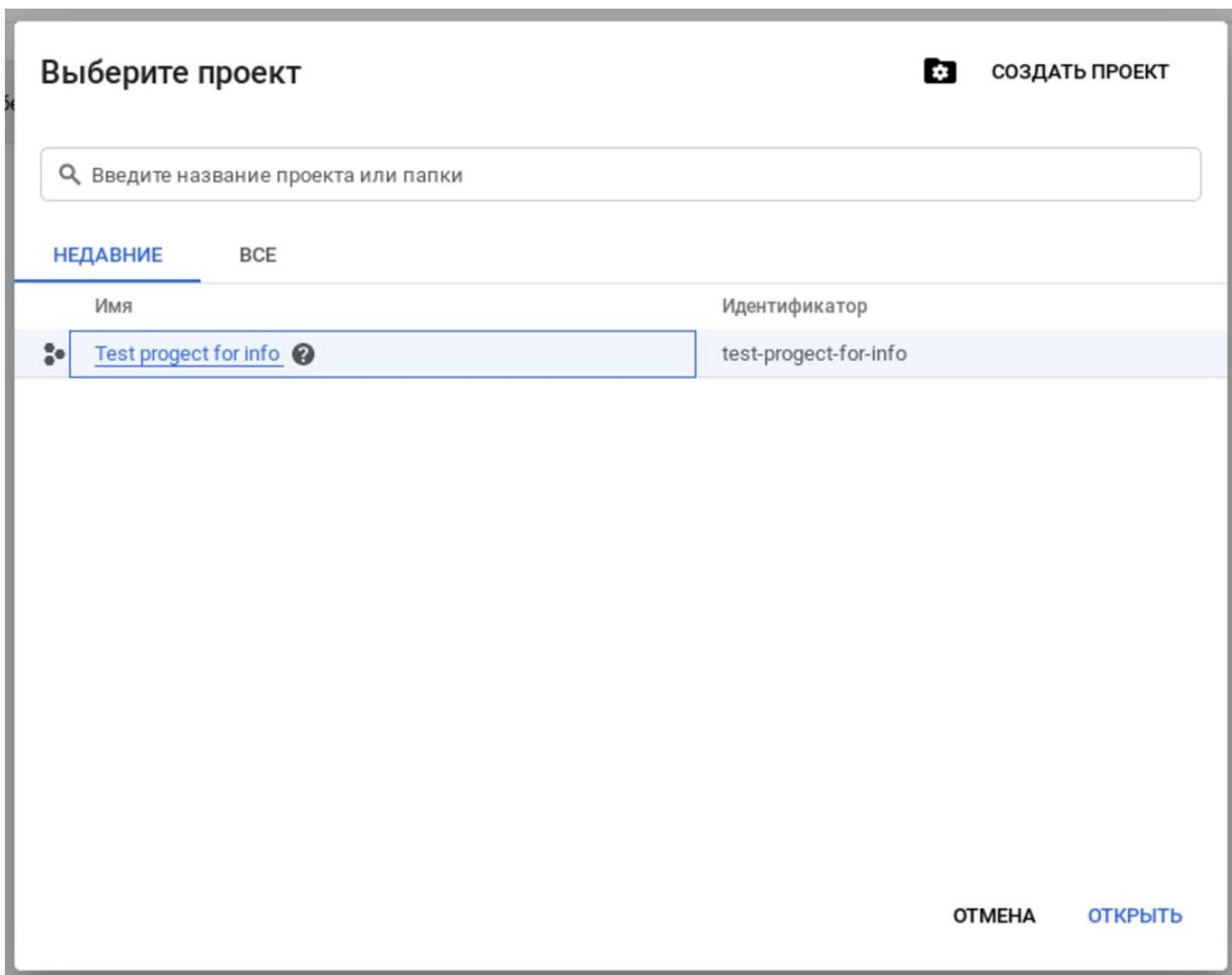


Рисунок 4.2 — Выбор проекта. Часть 2

Затем в библиотеке API выберите **Maps SDK for Android** (рисунок 4.3) и активируйте его (рисунок 4.4).

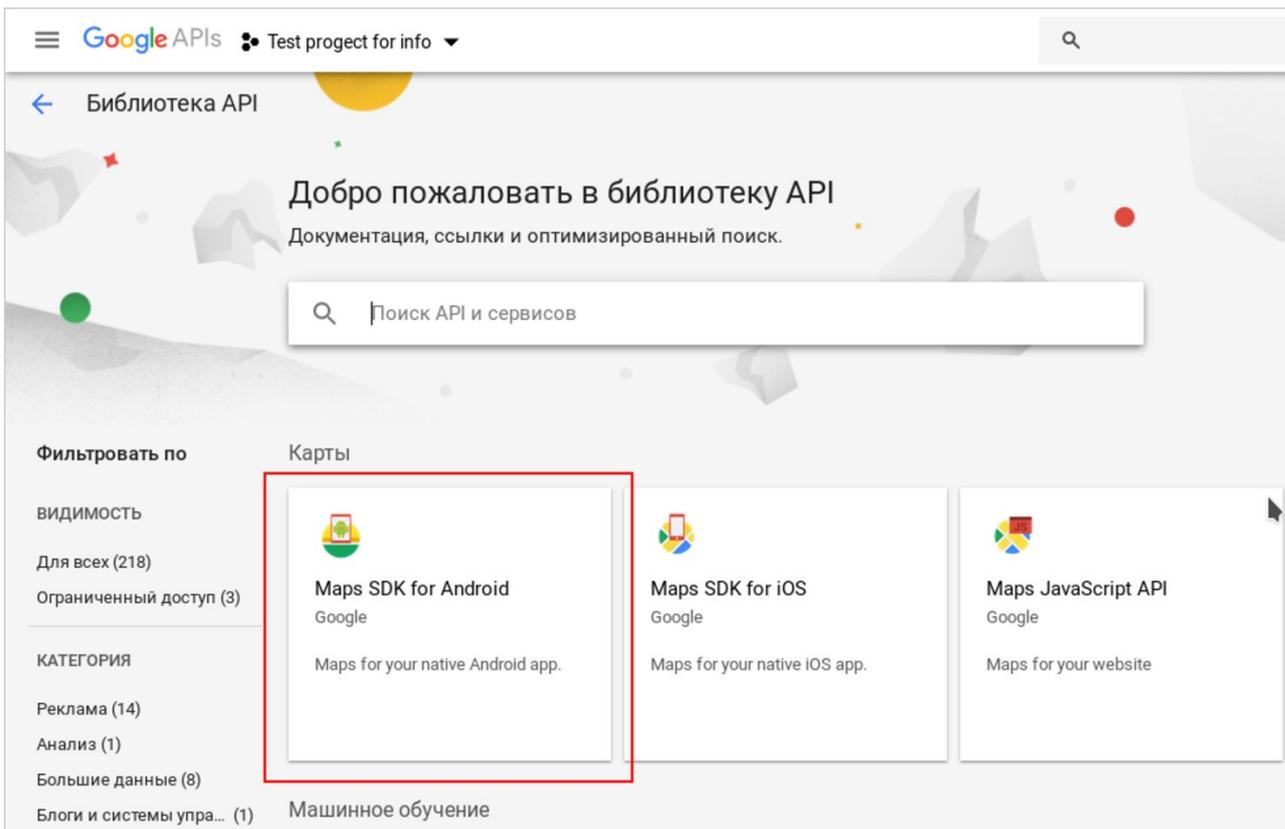


Рисунок 4.3 — Выбор Maps SDK for Android

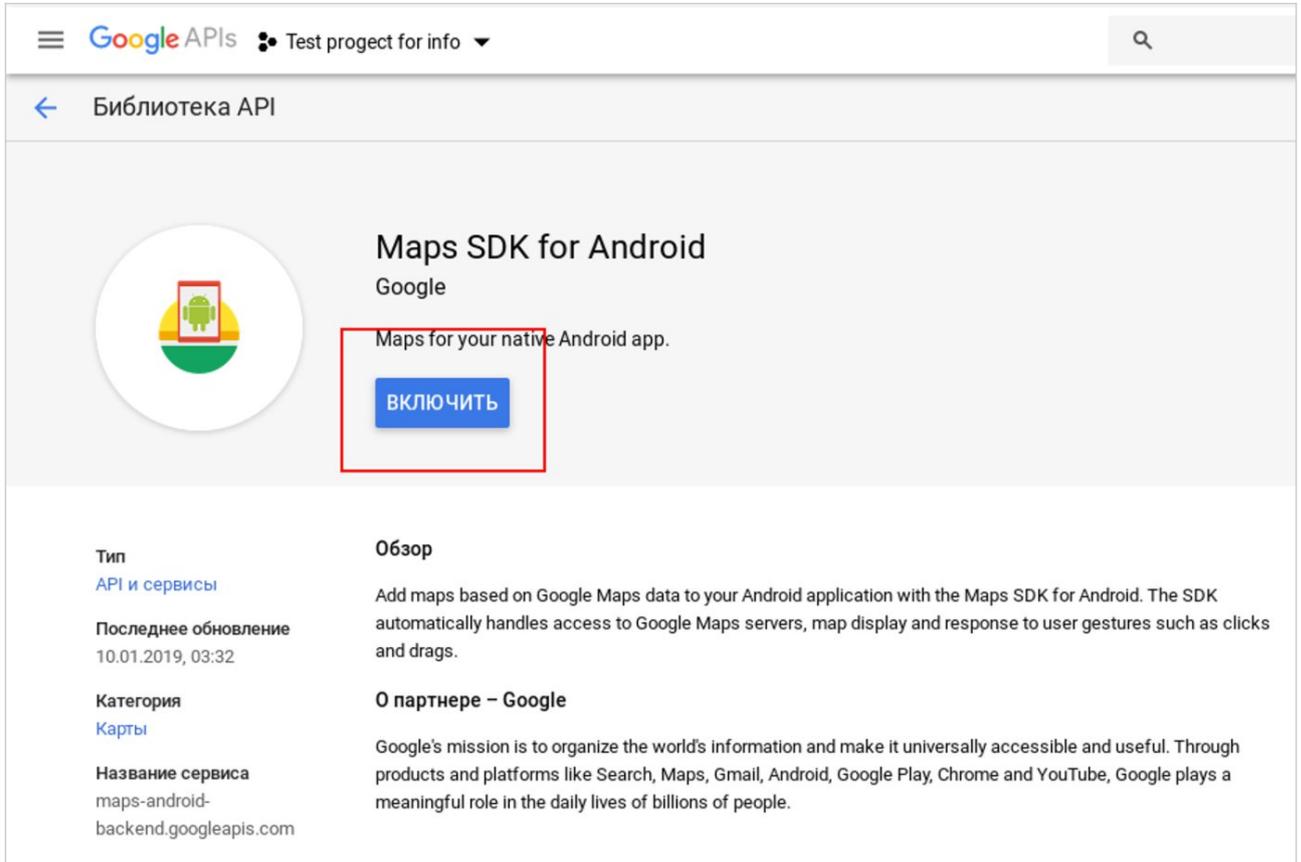


Рисунок 4.4 — Активация Maps SDK for Android

На вкладке «Общая информация» выберите **«Учётные данные»** и перейдите на страницу создания ключа API, нажав на слово **«здесь»**, как показано на рисунке 4.5.

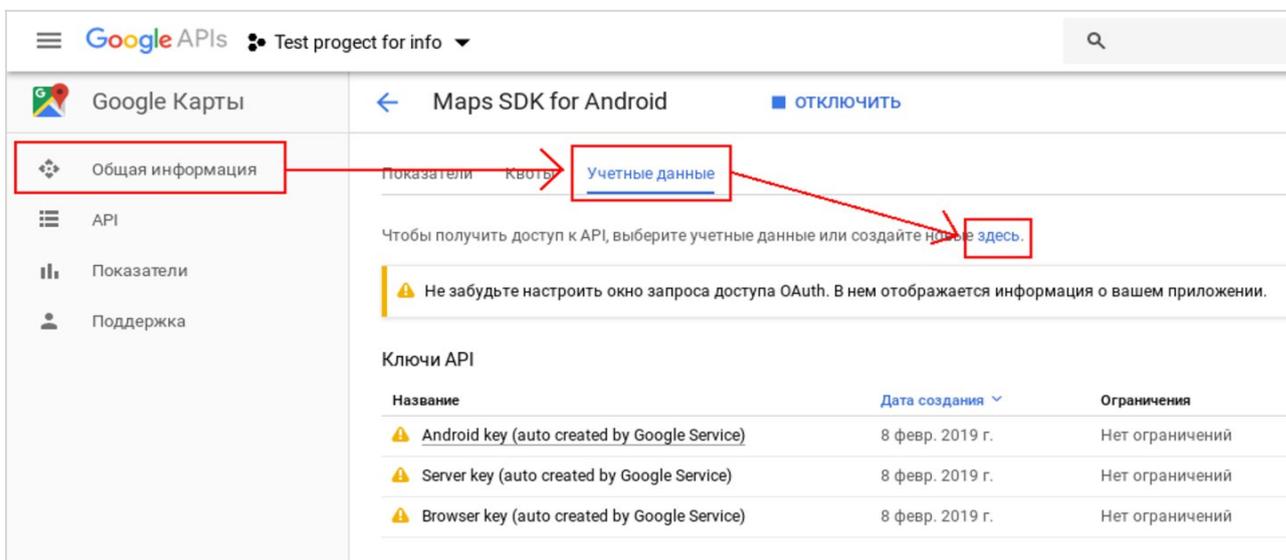


Рисунок 4.5 — Создание ключа API

В выпадающем списке выберите пункт «**Ключ API**» (рисунок 4.6).

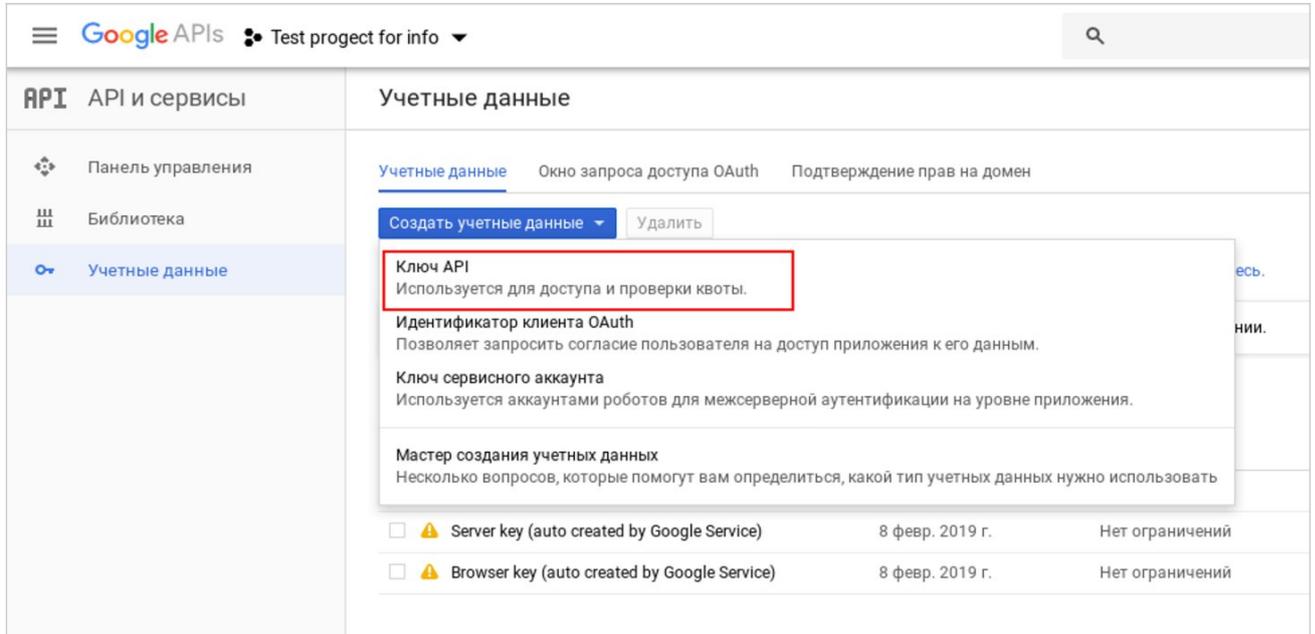


Рисунок 4.6 — Ключ API

После генерации передайте ключ API технической поддержке. Сгенерированный ключ автоматически добавится в список доступных ключей (рисунок 4.7).

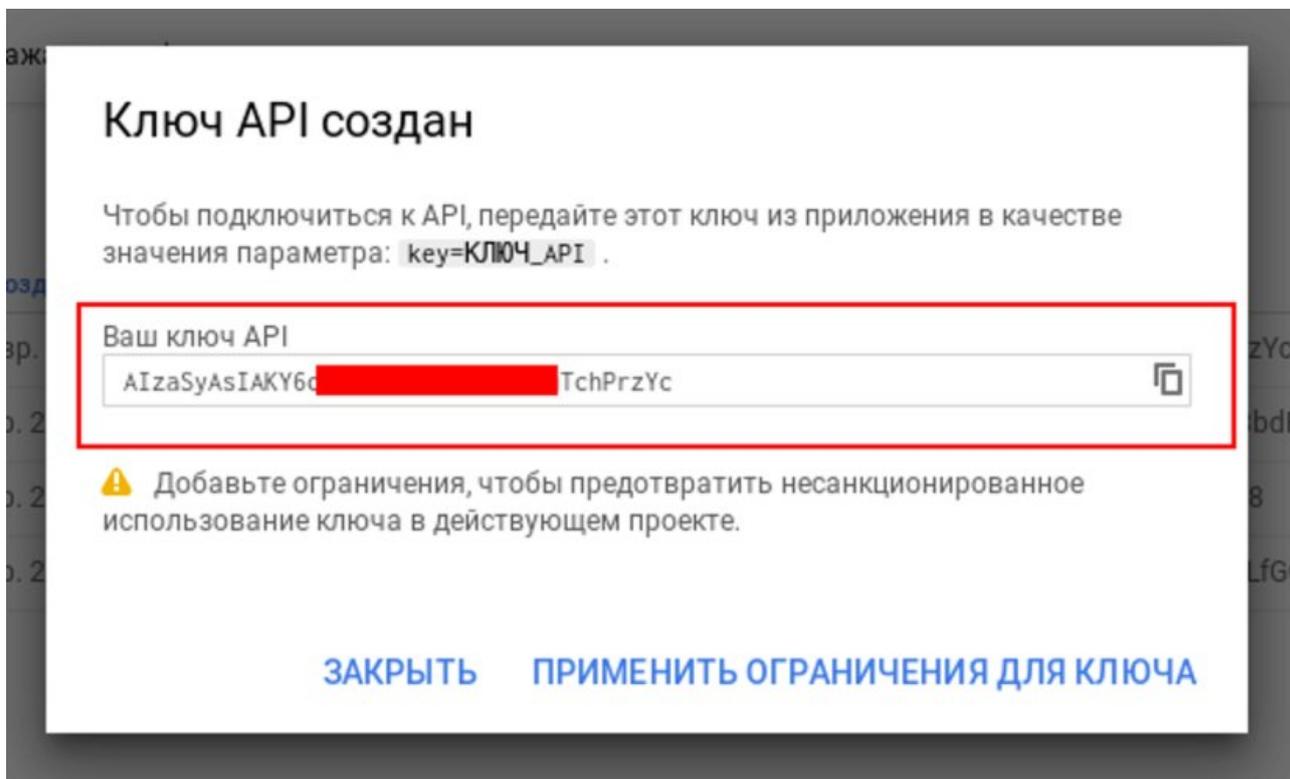


Рисунок 4.7 — Просмотр сгенерированного ключа

5 НАСТРОЙКА НА СТОРОНЕ АДМИНИСТРИРОВАНИЯ КОШЕЛЬКА

Для настройки PUSH-уведомлений перейдите в раздел «Транспорты сообщений» (рисунок 5.1).

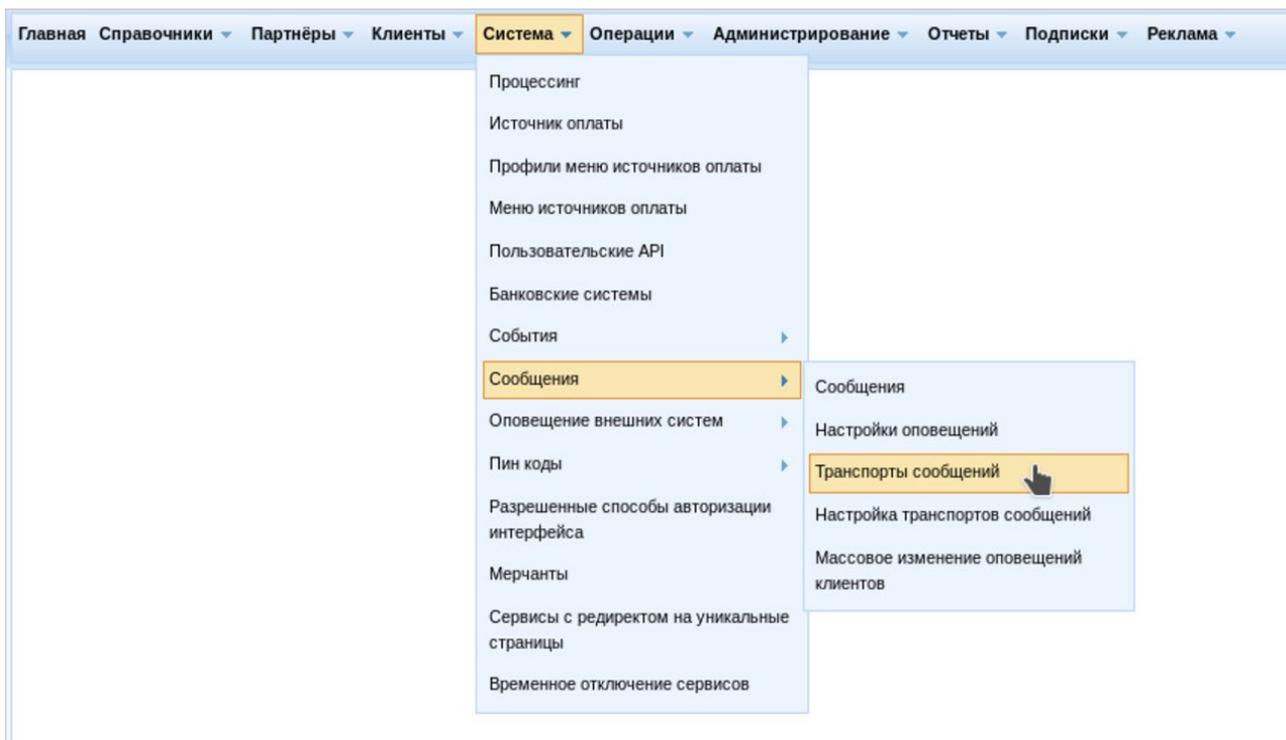
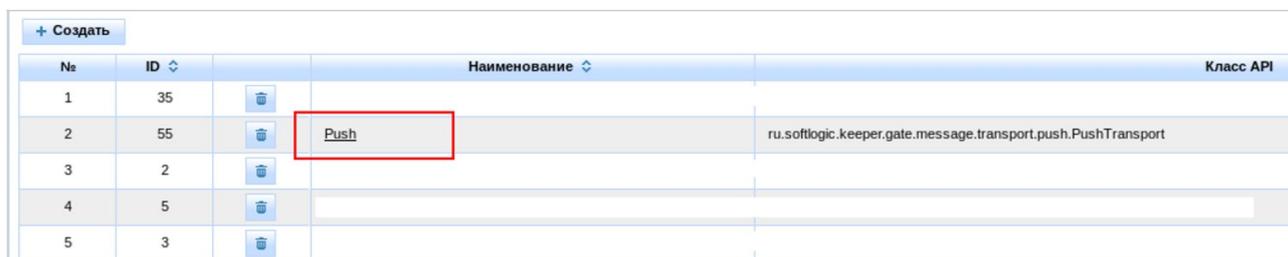


Рисунок 5.1 — Транспорты сообщений

Откройте элемент Push для редактирования (рисунок 5.2).



№	ID	Наименование	Класс API
1	35		
2	55	Push	ru.softlogic.keeper.gate.message.transport.push.PushTransport
3	2		
4	5		
5	3		

Рисунок 5.2 — Редактирование Push



Измените значение параметра **android.googleServerKey** на значение ключа сервера из раздела настроек Cloud Messaging (рисунок 2.8), а значение параметра **android.url** – на <https://fcm.googleapis.com/fcm/send>.

Пересоберите приложение кошелька после всех настроек.

6 ПОДПИСАНИЕ СБОРКИ *-RELEASE-UNSIGNED.APK ПЕРЕД ПУБЛИКАЦИЕЙ

6.1 СОЗДАНИЕ СЕРТИФИКАТА

Для публикации Android-приложения в Google Play необходимо иметь сертификат. Чтобы его сгенерировать, выполните следующую команду в командной строке:

```
keytool -genkey -v -keystore <имя_сертификата>.keystore -alias  
<название_алиаса> -keyalg RSA -keysize 2048 -validity 10000
```

Например, если вы хотите получить **production_sign.keystore** с алиасом **production_alias**, то команда будет иметь вид:

```
keytool -genkey -v -keystore production_sign.keystore -alias production_alias  
-keyalg RSA -keysize 2048 -validity 10000
```

Keytool — утилита, которая входит в состав Java (папка *jre/bin*), скачать пакет можно по [ссылке](#).

После выполнения команды введите все требуемые данные: пароль, название организации и другие (рисунок 6.1).

```
[root@localhost bin]# keytool -genkey -v -keystore production_sign.keystore -alias production_alias -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  Ivanov Ivan
What is the name of your organizational unit?
  [Unknown]:  Test inc
What is the name of your organization?
  [Unknown]:  Test inc
What is the name of your City or Locality?
  [Unknown]:  Barnaul
What is the name of your State or Province?
  [Unknown]:  Altai
What is the two-letter country code for this unit?
  [Unknown]:  RU
Is CN=Ivanov Ivan, OU=Test inc, O=Test inc, L=Barnaul, ST=Altai, C=RU correct?
  [no]:  yes

Generating 2 048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10 000 days
    for: CN=Ivanov Ivan, OU=Test inc, O=Test inc, L=Barnaul, ST=Altai, C=RU
Enter key password for <production_alias>
    (RETURN if same as keystore password):
Re-enter new password:
[Storing production_sign.keystore]

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srckeystore production_sign.keystore -destkeystore production_sign.keystore -deststoretype pkcs12".
```

Рисунок 6.1 — Создание сертификата

**Внимание!**

Полученный сертификат ***.keystore** создаётся только один раз — перед первой публикацией приложения. Его нужно сохранить в надёжном месте для подписания следующих версий приложения: если подписать приложение другим сертификатом и попытаться опубликовать в Google Play, то в действительности будет отказано.

6.2 ПОДПИСАНИЕ ПРИЛОЖЕНИЯ

Перед публикацией приложения необходимо подписать его с помощью ключа ***.keystore**, полученного на шаге 6.1.

Для начала проверьте, что файл *.apk не подписан:

```
apksigner verify my_application.apk
```

apksigner — утилита, которая находится в составе SDK, начиная с версии 24.0.3 (папка *SDK/build-tools/XX.X.X*).

```
[victor@localhost ~]$ apksigner verify my_application.apk
DOES NOT VERIFY
ERROR: No JAR signatures
[victor@localhost ~]$ █
```

Рисунок 6.2 — Проверка наличия подписи у файла *.apk

Подпишите приложение. При подписи необходимо указать параметр **--v2-signing-enabled true**:

```
apksigner sign --v2-signing-enabled true --ks <имя_сертификата>.keystore
--ks-key-alias <название_алиаса> <название_файла_приложения>.apk
```

Например, для подписания приложения **my_application.apk** (рисунок 6.3) с помощью ключа **production_sign.keystore**, имеющего алиас **production_alias**, требуется выполнить команду:

```
apksigner sign --v2-signing-enabled true --ks production_sign.keystore --
ks-key-alias production_alias my_application.apk
```

```
[victor@localhost ~]$ apksigner sign --v2-signing-enabled true --ks production_sign.keystore --ks-key-alias production_alias my_application.apk
Keystore password for signer #1:
[victor@localhost ~]$ █
```

Рисунок 6.3 — Пример подписи приложения ключом *.keystore

Проверьте, что файл *.apk удалось подписать (предупреждения можно игнорировать):

```
apksigner verify my_application.apk
```



```
[victor@localhost ~]$ apksigner verify my_application.apk  
[victor@localhost ~]$ █
```

Рисунок 6.4 — Проверка наличия подписи

Если файл подписан, сообщение «DOES NOT VERIFY. ERROR: No JAR signatures» пропадет.

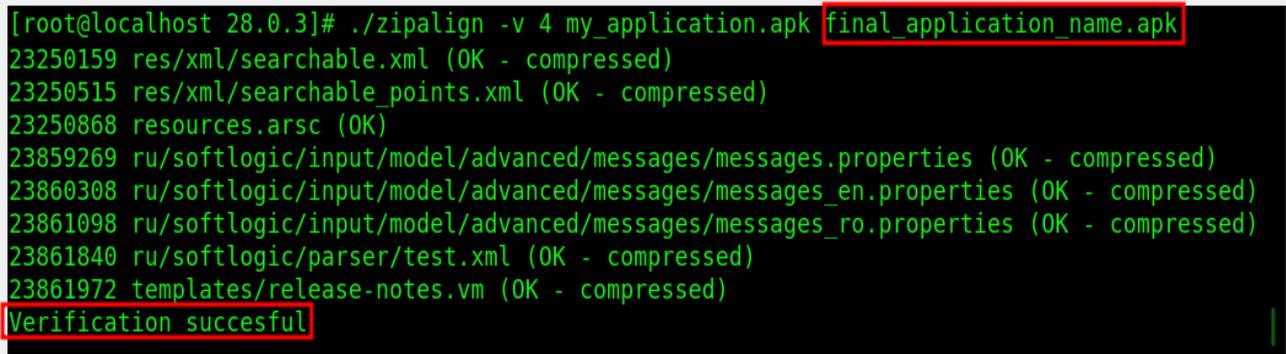
Далее подпишите и сожмите приложение с помощью утилиты **zipalign**:

```
zipalign -v 4 <имя_файла_приложения>.apk <имя_файла_для_публикации>.apk
```

Например, для подписания приложения **my_application.apk** и получения файла **final_application_name.apk** для публикации в Google Play требуется выполнить следующую команду (рисунок 6.5):

```
zipalign -v 4 my_application.apk final_application_name.apk
```

zipalign — утилита, которая является частью инструментария Android Studio (SDK/build-tools/<актуальная_версия>), скачивание доступно по [ссылке](#).



```
[root@localhost 28.0.3]# ./zipalign -v 4 my_application.apk final_application_name.apk  
23250159 res/xml/searchable.xml (OK - compressed)  
23250515 res/xml/searchable_points.xml (OK - compressed)  
23250868 resources.arsc (OK)  
23859269 ru/softlogic/input/model/advanced/messages/messages.properties (OK - compressed)  
23860308 ru/softlogic/input/model/advanced/messages/messages_en.properties (OK - compressed)  
23861098 ru/softlogic/input/model/advanced/messages/messages_ro.properties (OK - compressed)  
23861840 ru/softlogic/parser/test.xml (OK - compressed)  
23861972 templates/release-notes.vm (OK - compressed)  
Verification successful
```

Рисунок 6.5 — Подписание приложения с помощью утилиты **zipalign**