

**DELIVERY  
CLUB  
TECH**



## Logistics

про логистов, курьеров и доставку

## Vendor

про интеграции с нашими партнерами

## R&D

решают наукоёмкие задачи

## Internal

call-центр и поддержка

## Platform

улучшение архитектуры и платформы в целом

## Offline Experience

клиентское направление, которое развивает офлайн-сценарии в приложении

## Consumer

это все про пользовательские продукты: сайт и мобильные приложения

## Grocery

про доставку продуктов из магазинов



**habr: Продуктовая  
трансформация в  
Delivery Club Tech**

# Выходим на рынок Huawei, *Или как мы адаптировали приложение для работы с HMS*

---

Павел Борзиков  
Георгий Гигаури





# НУАВЕІ ПЕРЕСТАЕТ РАСПРОСТРАНЯТЬ УСТРОЙСТВА С GOOGLE PLAY SERVICES



# Аналоги

GOOGLE



HUAWEI



# Собственные сервисы Huawei



**GMS**

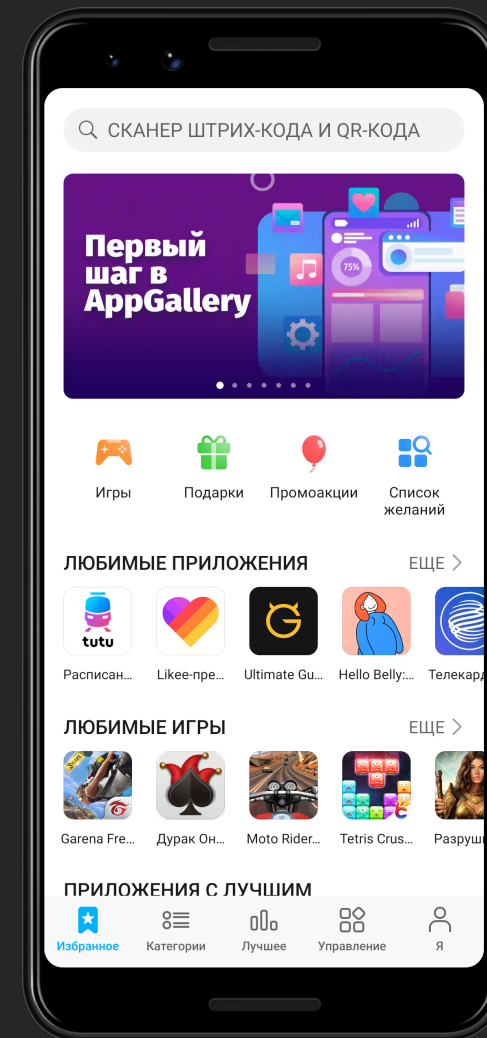
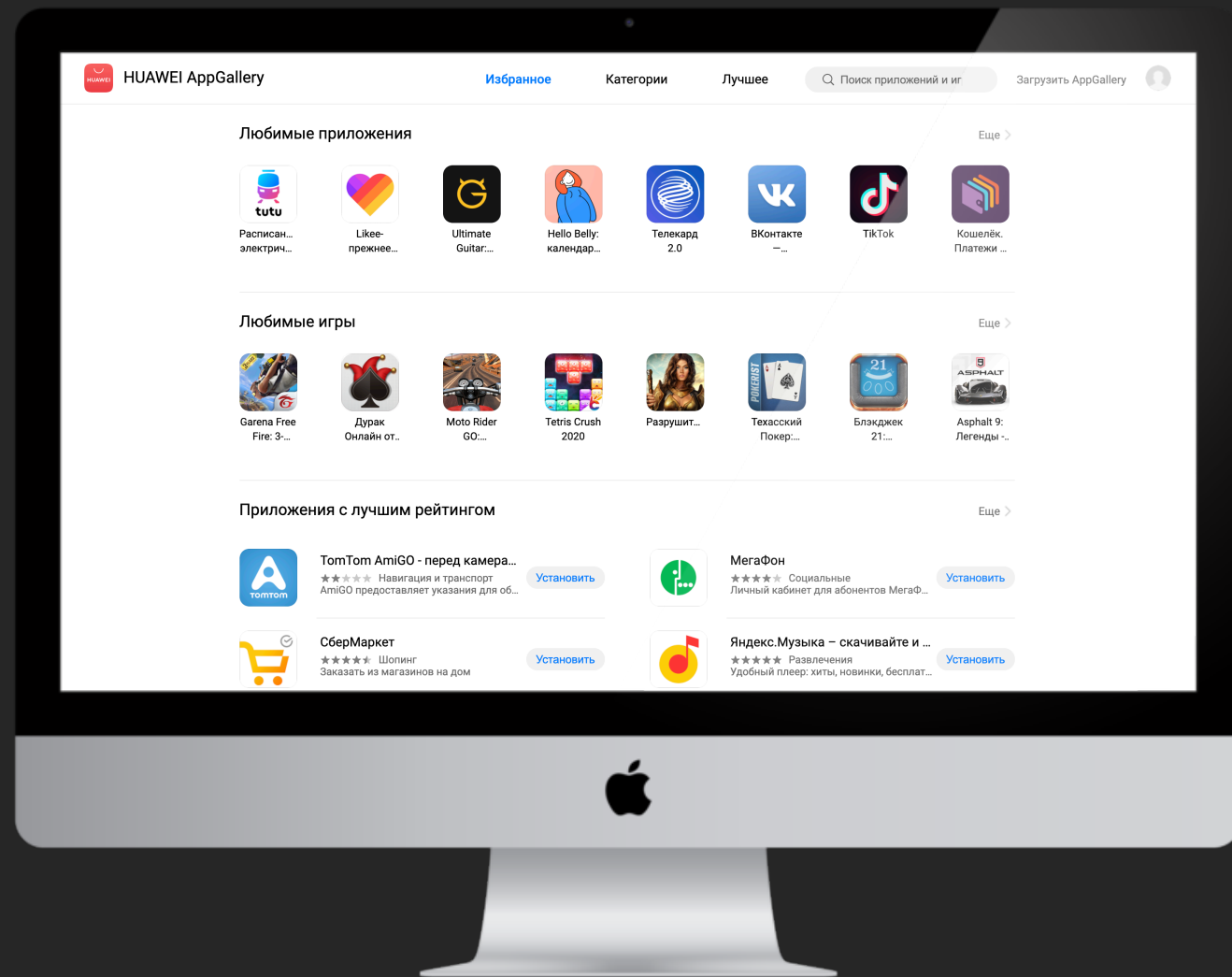
**Google Mobile Services**



**HMS**

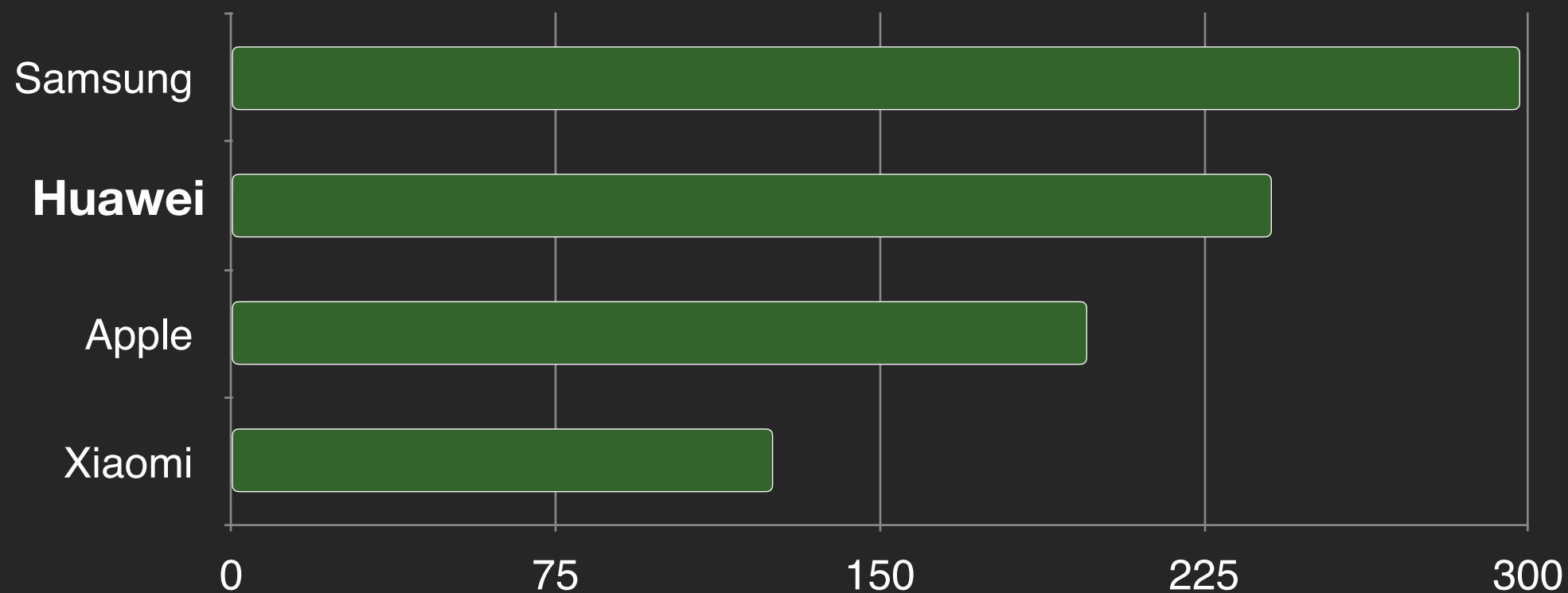
**Huawei Mobile Services**

# Huawei AppGallery



# Популярность Huawei

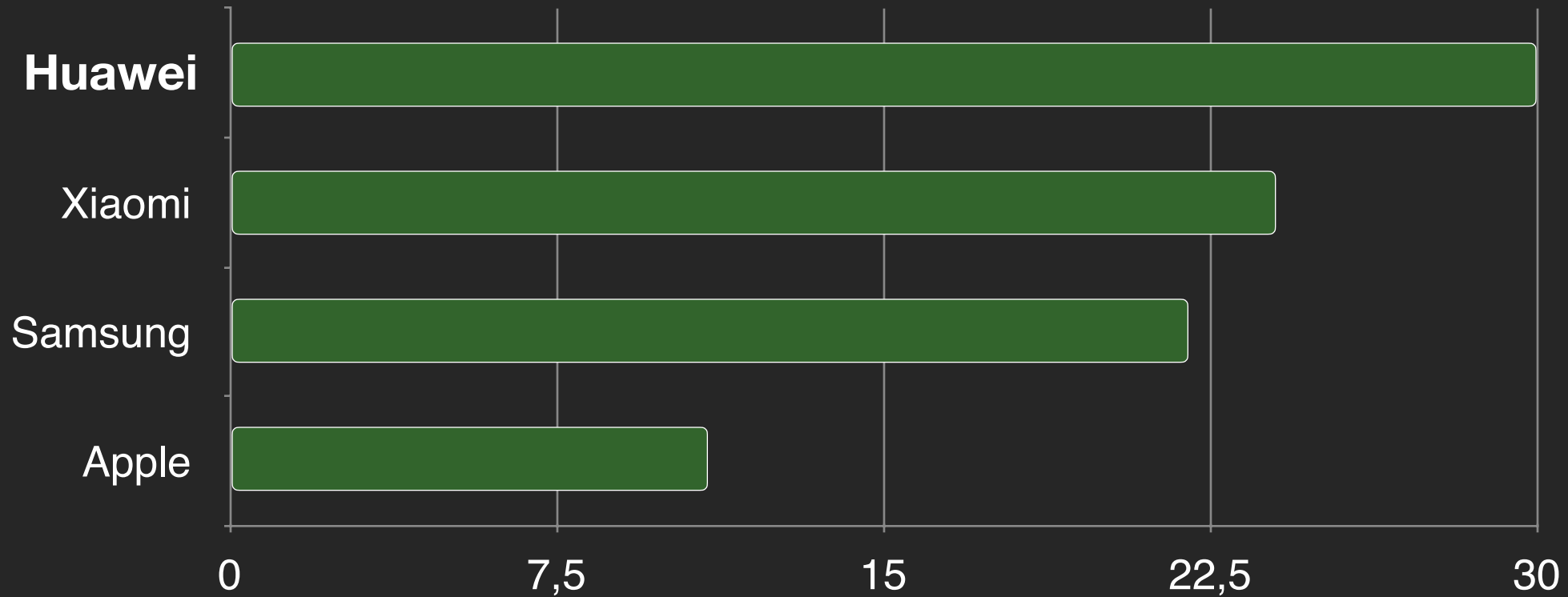
Количество проданных устройств в мире, за 2019 год  
(миллионов)



Источник: [canalys.com](https://www.canalys.com)

# Популярность Huawei

Количество проданных устройств в России, Q2 2020 года (%)



Источник: [canalys.com](https://www.canalys.com)

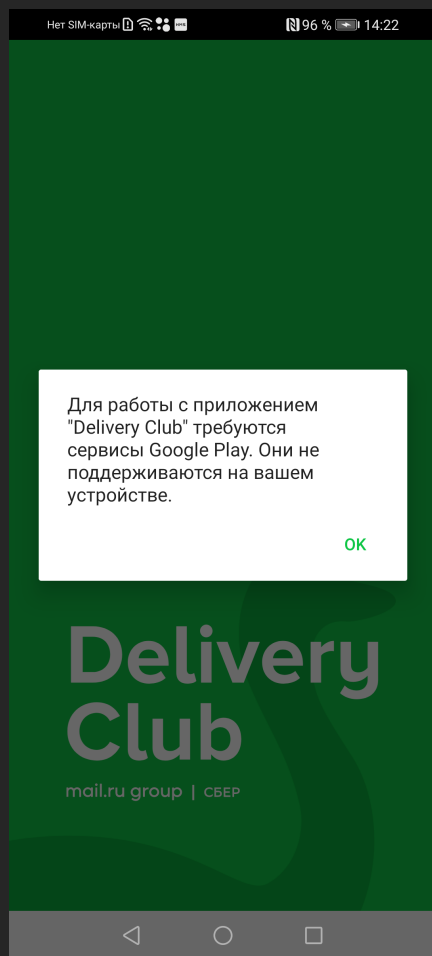
# План действий

- 1 ВЫЯВИТЬ БЛОКЕРЫ ДЛЯ ЗАПУСКА НА УСТРОЙСТВЕ БЕЗ GMS
- 2 ПОЧИНИТЬ ВСЕ ЧТО ОКАЗАЛОСЬ СЛОМАННЫМ
- 3 ДОБАВИТЬ В ПРОЕКТ ДОПОЛНИТЕЛЬНЫЕ ИНСТРУМЕНТЫ ОТ HMS
- 4 ПРОТЕСТИРУЕМ
- 5 ОПУБЛИКУЕМ .. 6 СДЕЛАЕМ ВЫВОДЫ

# Шаг №1: Выявление блокеров



# Google Services в Delivery Club.



```
private fun checkPlayServices() {
    val googleApiAvailability = GoogleApiAvailability.getInstance()
    val errorCode = googleApiAvailability.isGooglePlayServicesAvailable(this)

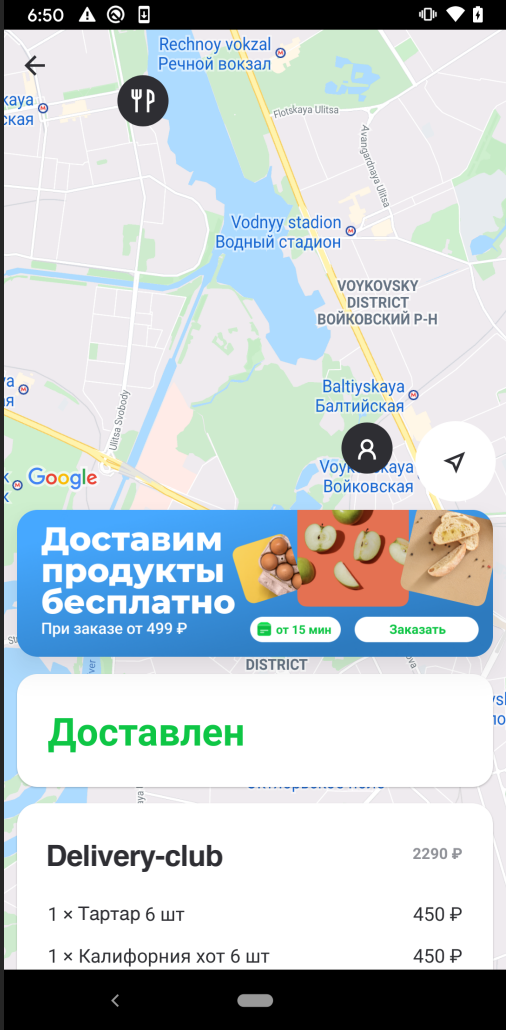
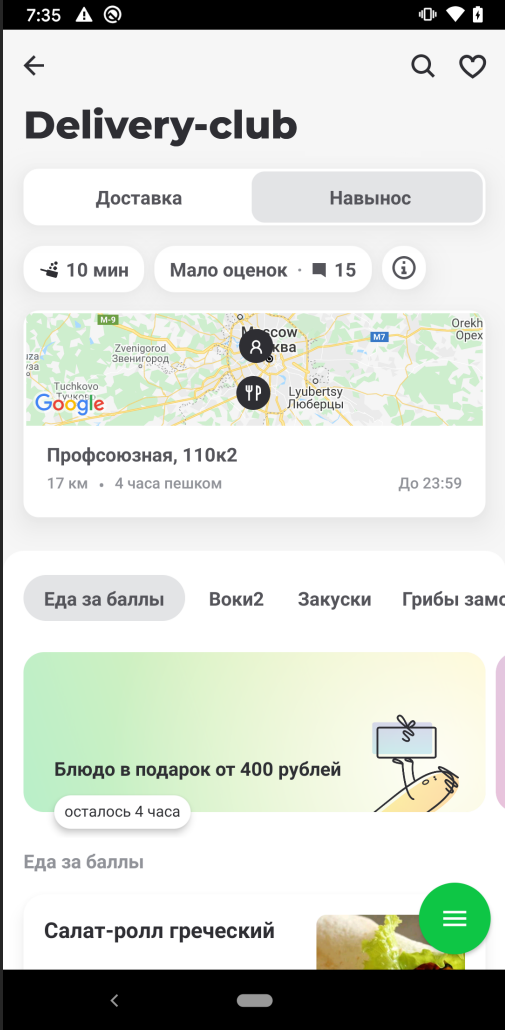
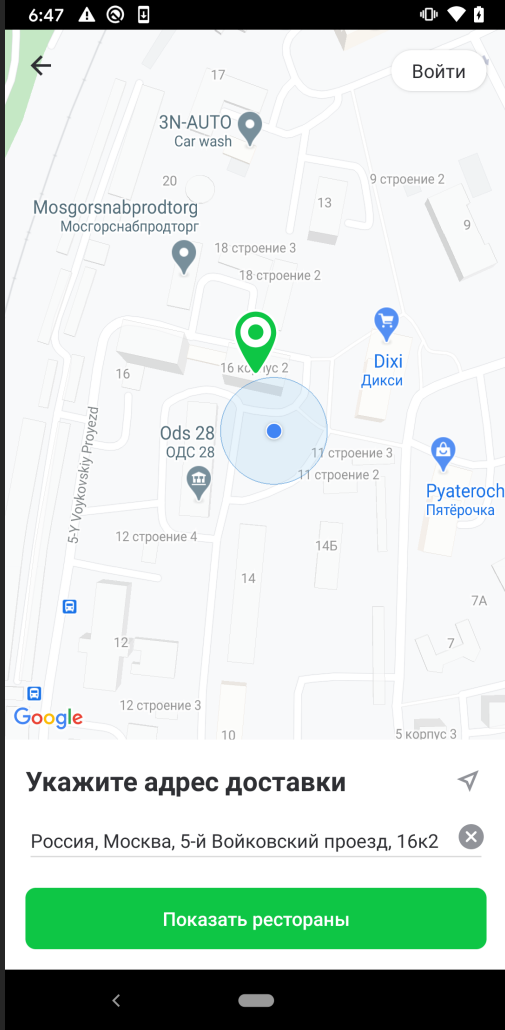
    if (errorCode == ConnectionResult.SUCCESS) {
        setPlayServicesChecked()
        return
    }

    googleApiAvailability
        .getErrorDialog(
            this,
            errorCode,
            BaseActivity.REQUEST_CODE_PLAY_SERVICES_CODE,
            DialogInterface.OnCancelListener {
                Log.d("GoogleApiAvailability", "ErrorDialog cancelled")
            }
        )
        .show()
}
```

# Google Services в Delivery Club. Карты.

GMS

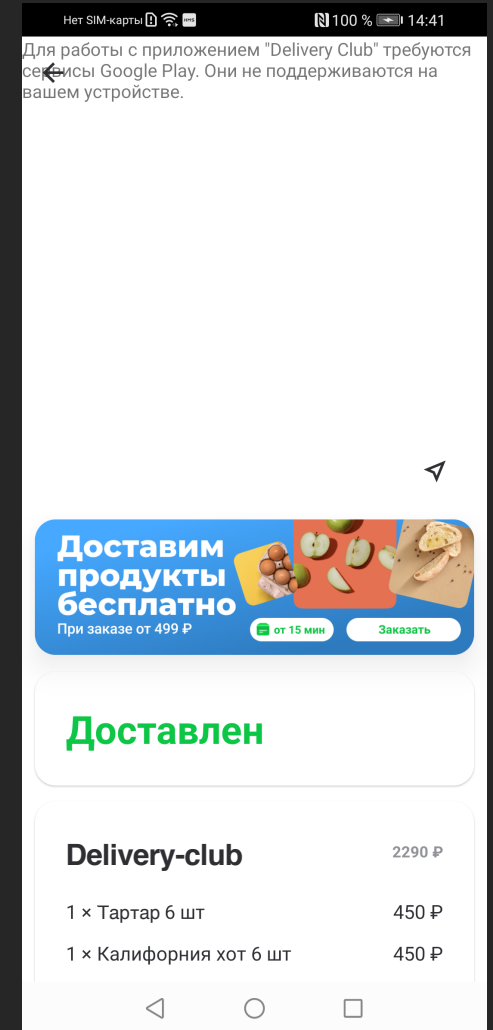
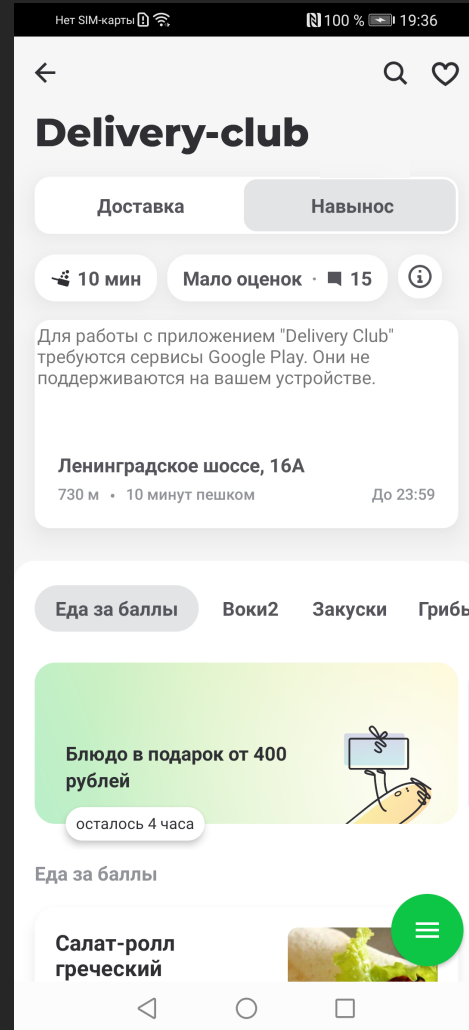
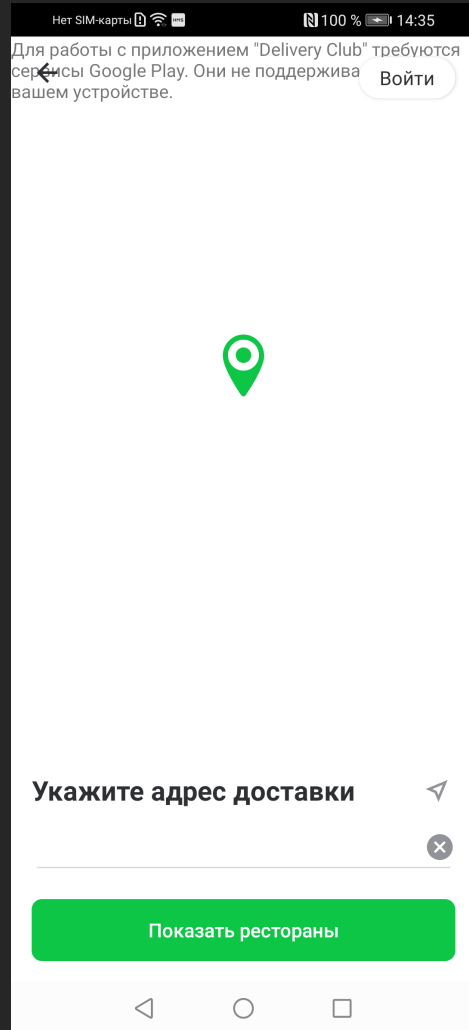
ON



# Google Services в Delivery Club. Карты.

GMS

OFF



# Google Services в Delivery Club. Push.

```
class ImplementationMessagingService : FirebaseMessagingService() {  
  
     override fun onNewToken(token: String) {  
        settingsManager.setPushToken(token)  
    }  
  
    override fun onMessageReceived(message: RemoteMessage) {  
        notificationManager.handlePushMessage(message)  
    }  
}
```

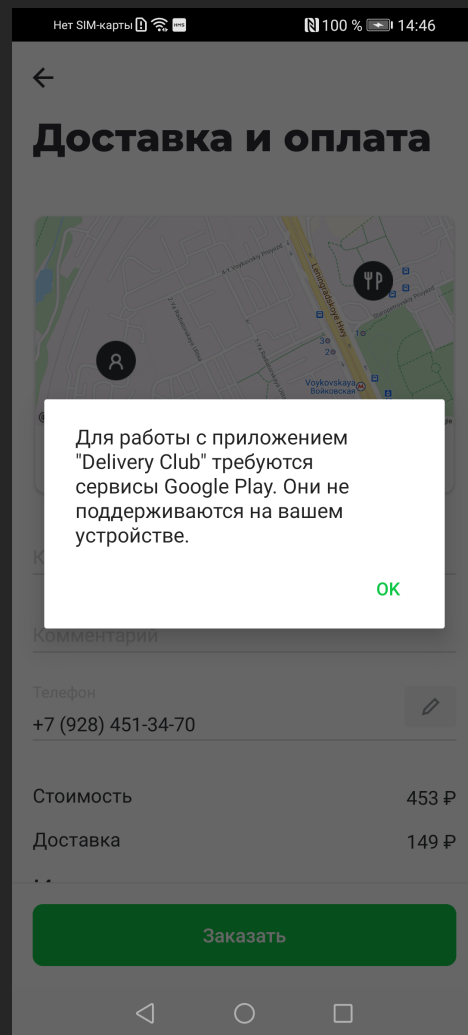
logcat:

```
18304-18571/com.deliveryclub E/FirebaseInstanceId: Google Play services missing or without correct
```

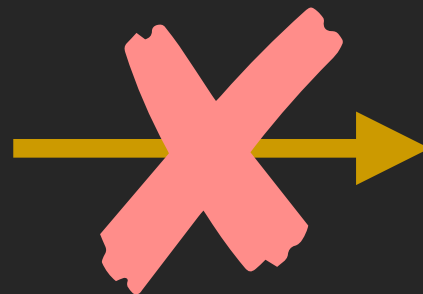
# Google Services в Delivery Club. ChromeTabs.

**ActivityNotFoundException**

# Google Services в Delivery Club. Google Pay.



# Google Services в Delivery Club. Google Pay.



4?

# Google Services в Delivery Club. Итого.

1

Карты;

2

Push-уведомления;

3

ChromeTabs.





# HMS Core

## App Services

Account Kit  
Ads Kit  
Analytics Kit  
Awareness Kit  
Drive Kit  
Game Service  
Health Kit  
Identity Kit  
In-App Purchases  
Location Kit  
Map Kit  
Push Kit  
Site Kit  
Wallet Kit  
...

## AI

ML Kit  
HiAI Foundation  
HiAI Engine  
HiAI Service

## Media

Audio Engine  
Audio Kit  
Camera Engine  
Image Kit  
Panorama Kit  
Video Kit  
WisePlay DRM

## Smart Device

CaaS Engine  
Cast Engine  
DeviceVirtualization  
OneHop Engine  
Share Engine

## Graphics

Accelerate Kit  
AR Engine  
Computer Graphics Kit  
GameTurbo Engine  
Scene Kit

## System

hQUIC Kit  
Nearby Service  
Wireless Kit

## Security

FIDO  
Safety Detect  
DataSecurity  
LocalAuthentication

# Замена GoogleMap



# Замена GoogleMap

## Google

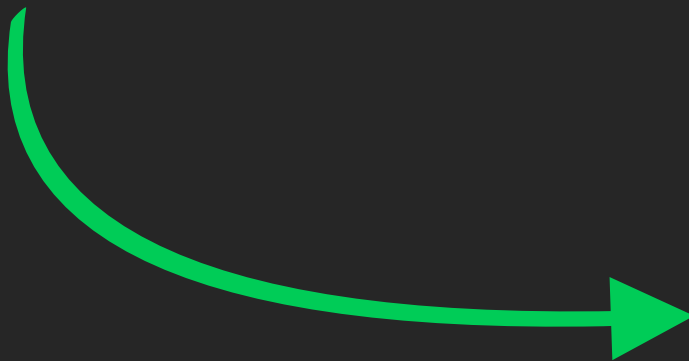
```
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.GoogleMapOptions;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.MapView;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.CircleOptions;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.LatLngBounds;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
```

## Huawei

```
import com.huawei.hms.maps.HuaweiMap;
import com.huawei.hms.maps.HuaweiMapOptions;

import com.huawei.hms.maps.CameraUpdateFactory;
import com.huawei.hms.maps.MapView;
import com.huawei.hms.maps.OnMapReadyCallback;
import com.huawei.hms.maps.model.BitmapDescriptorFactory;
import com.huawei.hms.maps.model.CircleOptions;
import com.huawei.hms.maps.model.LatLng;
import com.huawei.hms.maps.model.LatLngBounds;
import com.huawei.hms.maps.model.Marker;
import com.huawei.hms.maps.model.MarkerOptions;
```



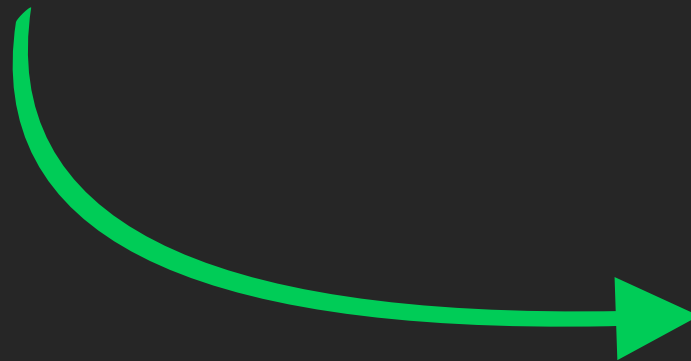
# Замена GoogleMap

## Google

```
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.GoogleMapOptions;  
  
import com.google.android.gms.maps.CameraUpdateFactory;  
import com.google.android.gms.maps.MapView;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.model.BitmapDescriptorFactory;  
import com.google.android.gms.maps.model.CircleOptions;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.LatLngBounds;  
import com.google.android.gms.maps.model.Marker;  
import com.google.android.gms.maps.model.MarkerOptions;
```

## Huawei

```
import com.huawei.hms.maps.HuaweiMap;  
import com.huawei.hms.maps.HuaweiMapOptions;  
  
import com.huawei.hms.maps.CameraUpdateFactory;  
import com.huawei.hms.maps.MapView;  
import com.huawei.hms.maps.OnMapReadyCallback;  
import com.huawei.hms.maps.model.BitmapDescriptorFactory;  
import com.huawei.hms.maps.model.CircleOptions;  
import com.huawei.hms.maps.model.LatLng;  
import com.huawei.hms.maps.model.LatLngBounds;  
import com.huawei.hms.maps.model.Marker;  
import com.huawei.hms.maps.model.MarkerOptions;
```



# Проблемы создания работы с Huawei Map

# Deprecated методы Google Map SDK

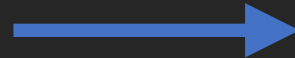
```
LocationServices.FusedLocationApi.getLastLocation(googleApiClient);
```



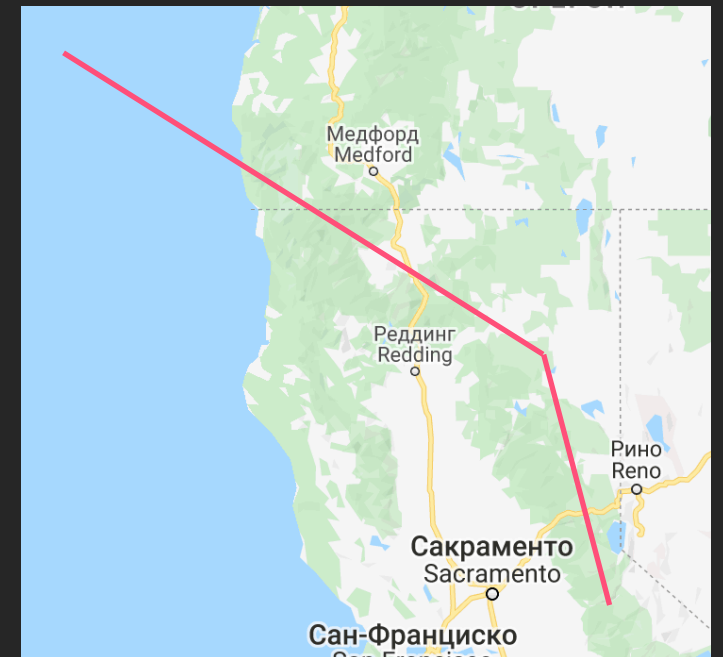
```
LocationServices.getFusedLocationProviderClient(...)  
    .getLastLocation()  
    .addOnSuccessListener(...)
```

# PolyUtil. Polyline Decode

"\_p~iF~ps|U\_u7LnnqC\_mqNvxq`@"



```
[  
  (38.5, -120.2),  
  (40.7, -120.95),  
  (43.252, -126.45300000000000002)  
]
```





# PolyUtil. Polyline Decode



`com.google.maps.android.PolyUtil`



???

Home > Products > Google Maps Platform



## Encoded Polyline Algorithm Format

[Send feedback](#)

Polyline encoding is a lossy compression algorithm that allows you to store a series of coordinates as a single string. Point coordinates are encoded using signed values. If you only have a few static points, you may also wish to use the interactive [polyline encoding utility](#).

# PolyUtil. Polyline Decode



# HWLocation

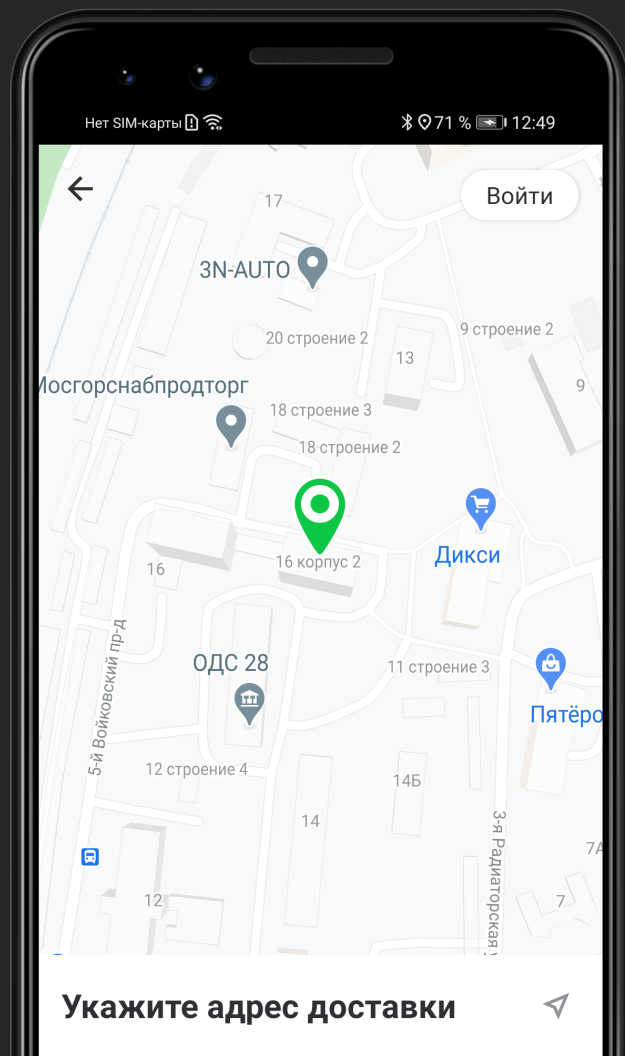
`com.huawei.hms.location.HWLocation(+ android.location.Location)`

Name	Description	Value	Value ( <i>setNeedAddress(true)</i> )
latitude	Latitude of a location.	<b>55.818401</b>	<b>55.818401</b>
longitude	Longitude of a location.	<b>37.489275</b>	<b>37.489275</b>
countryCode	Country code.	<b>null</b>	<b>RU</b>
countryName	Country name.	<b>null</b>	<b>Россия</b>
state	Administrative region.	<b>null</b>	<b>Центральный Федеральный Округ</b>
city	City of the current location.	<b>null</b>	<b>Москва</b>
street	Street of the current location.	<b>null</b>	<b>125171 Россия, Центральный Федеральный Округ, 3-я Радиаторская улица, Москва, вл11с3</b>
postalCode	Postal code of the current location.	<b>null</b>	<b>125171</b>
...			

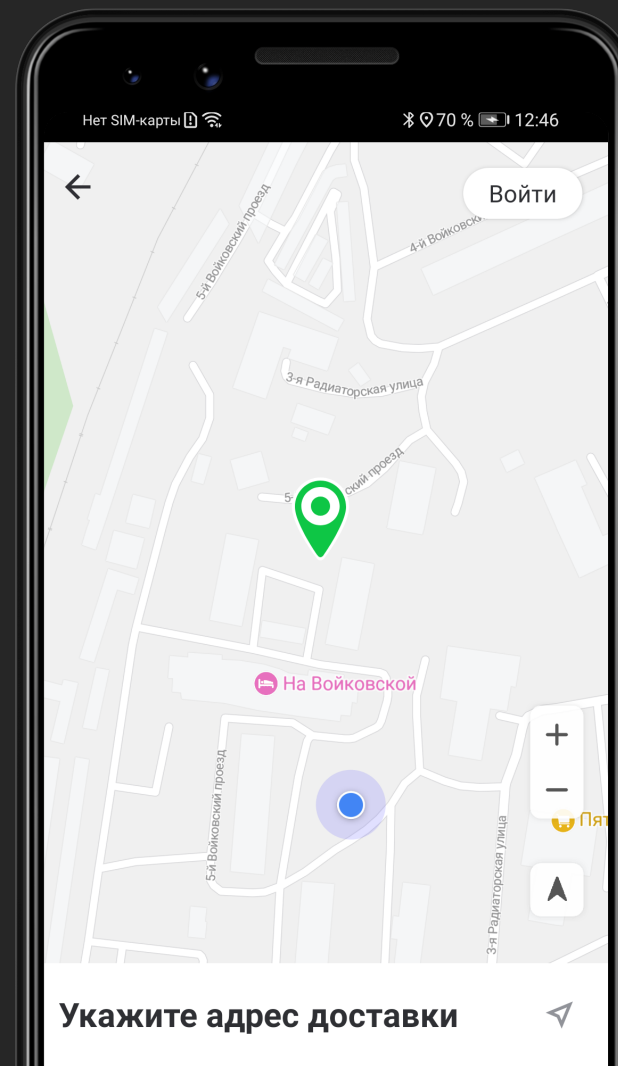
`INTERVAL)`  
`PRIORITY_HIGH_ACCURACY)`

# Замена GoogleMap

## GoogleMap

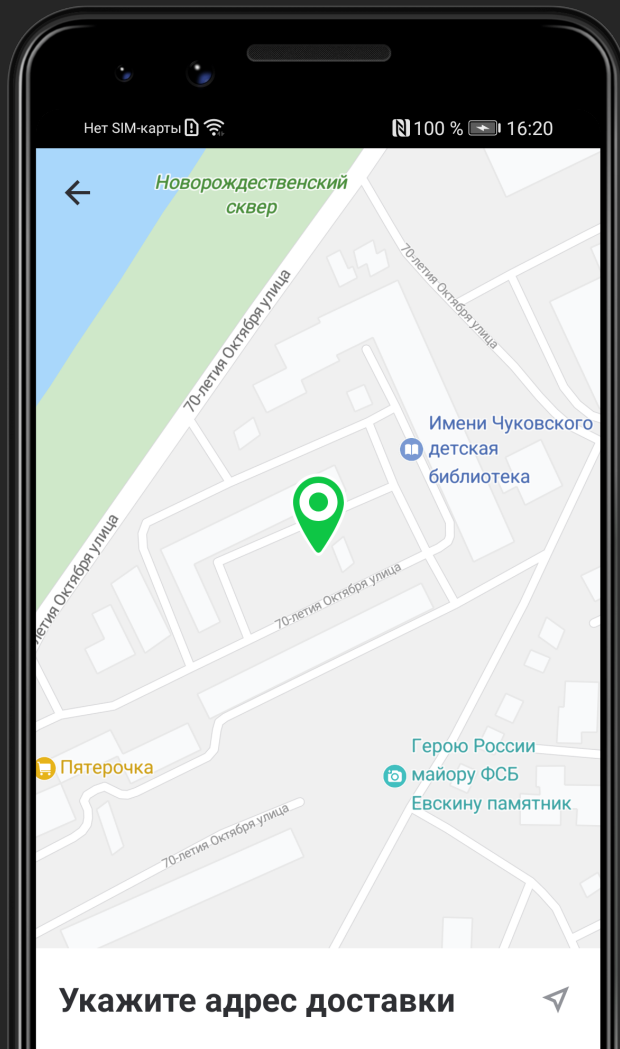


## Huawei Map

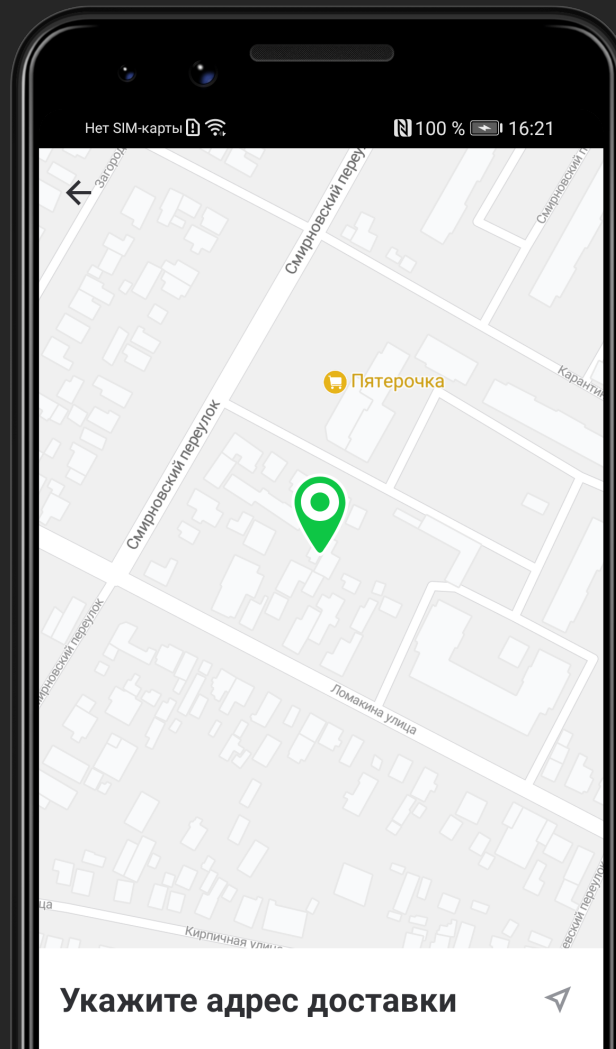


# Карты Huawei в городах России

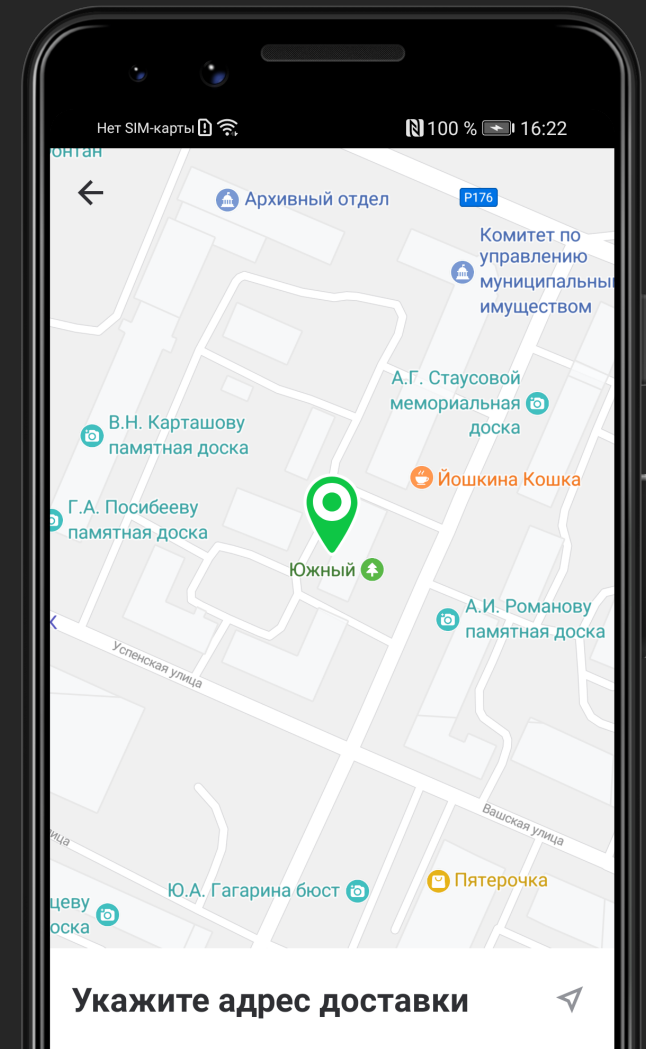
## Краснодар



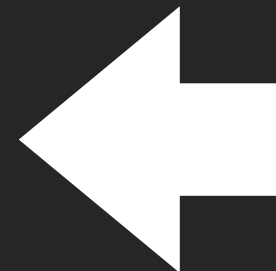
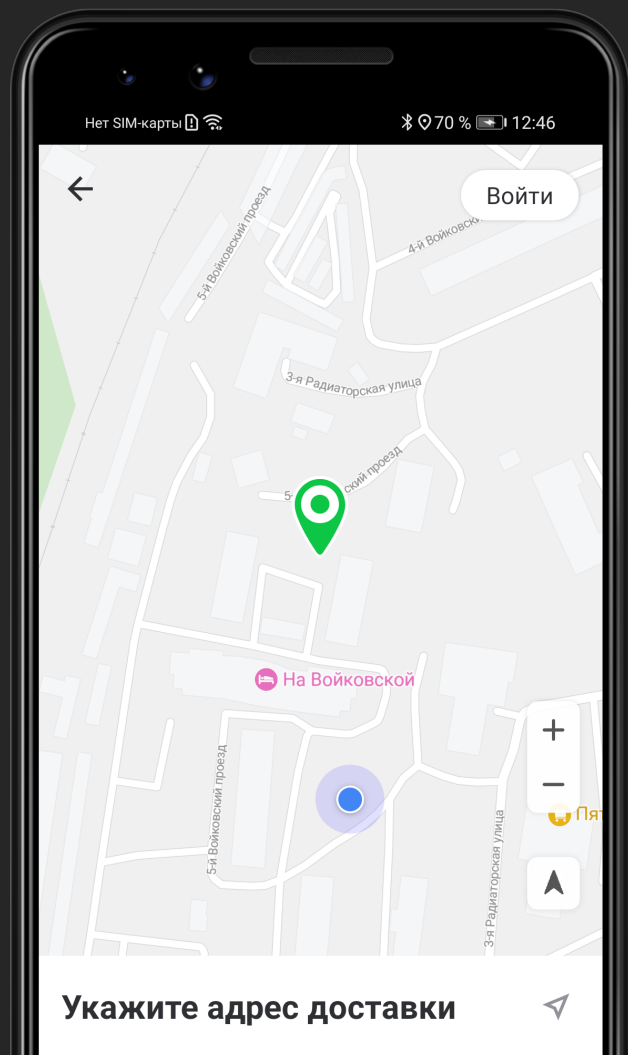
## Таганрог



## Йошкар-Ола



# Замена GoogleMap



# Как реализовать поддержку 2х карт?

# IMapWidget

```
public interface IMapWidget {  
    void animateCamera(...);  
    void setListener(OnMapEventListener listener);  
    void setMapPadding(...);  
    MapMarker addMarker(...);  
    ...  
}
```

```
interface OnMapEventListener {  
    void onMapScrollBegin();  
    void onMapScrollCancel();  
    void onMapScrollFinish();  
    void onMapReady();  
    void onMarkerClick(GeoPoint geoPoint);  
    ...  
}
```



# Объекты

`com.google.android.gms.maps.model.LatLng`

`com.huawei.hms.maps.model.LatLng`



`com.google.android.gms.maps.model.LatLng`  
`com.huawei.hms.maps.model.LatLng`

# Custom Map View

`<com.google.android  
.gms.maps.MapView/>`



`<MapView/>`



`IMapWidget`



`<com.huawei  
.hms.maps.MapView/>`



`<HuaweiMapView/>`

# Custom Map View

Обертка для отрисовки реализаций карт

Пример добавления карты:

```
class MapWrapper : FrameLayout() {  
  
    fun setupMap(widget: IMapWidget) {  
        removeAllViews()  
        addView(widget as View)  
    }  
}
```

```
override fun onCreateView(...) {  
    ...  
    val map: IMapWidget = MapFactory.createMap()  
    viewMapWrapper.setupMap(map)  
    ...  
}
```

# Custom Map View

*ConstraintLayout*

*MapWrapper*

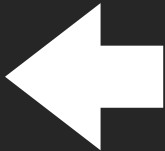


...

`MapFactory.createMap()`

GoogleMapView

HuaweiMapView



# ViewStub

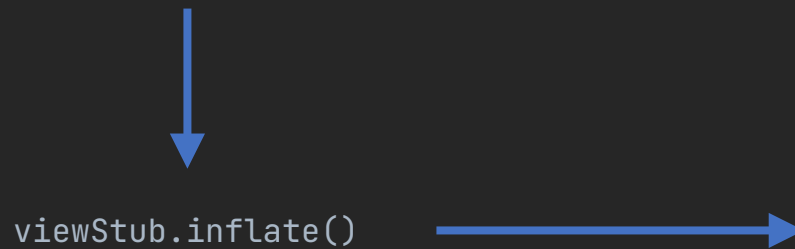
```
<LinearLayout>
```

```
    <Button  
        android:id="@+id/button"  
        android:text="Открыть ViewStub" />  
  
    <ViewStub  
        android:id="@+id/viewStub"  
        android:inflatedId="@+id/inflate"  
        android:layout="@layout/viewstub_layout" />
```

```
</LinearLayout>
```

## viewstub\_layout.xml

```
<FrameLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
  
    <TextView  
        android:text="Привет из StubView!" />  
  
</FrameLayout>
```



## LinearLayout

Открыть ViewStub

## FrameLayout

Привет из StubView!

# StubView

```
/**
 * Inflates the layout resource identified by {@link #getLayoutResource()}
 * and replaces this StubbedView in its parent by the inflated layout resource.
 *
 * @return The inflated layout resource.
 */
public View inflate() {
    final ViewParent viewParent = getParent();

    if (viewParent != null && viewParent instanceof ViewGroup) {
        if (mLayoutResource != 0) {
            final ViewGroup parent = (ViewGroup) viewParent;
            final View view = inflateViewNoAdd(parent);
            replaceSelfWithView(view, parent);
            private void replaceSelfWithView(View view, ViewGroup parent) {
                final int index = parent.indexOfChild(this);
                parent.removeViewInLayout(this);

                mInflatedViewRef = new WeakReference<>(view);
                if (mInflateListener != null) {
                    mInflateListener.onInflate(this, view);
                }
                final ViewGroup.LayoutParams layoutParams = getLayoutParams();
                if (layoutParams != null) {
                    parent.addView(view, index, layoutParams);
                } else {
                    parent.addView(view, index);
                }
            }

            return view;
        }
        ...
    }
}
```

# Custom Map View

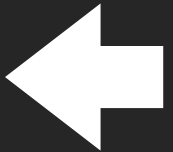
*ConstraintLayout*

...

MapFactory.createMap()

GoogleMapView

HuaweiMapView



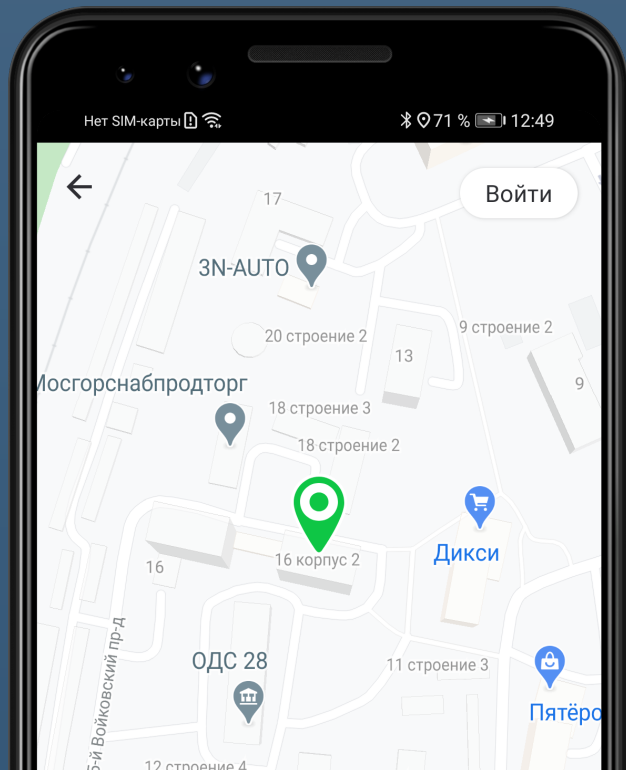
# Когда подменять карту?



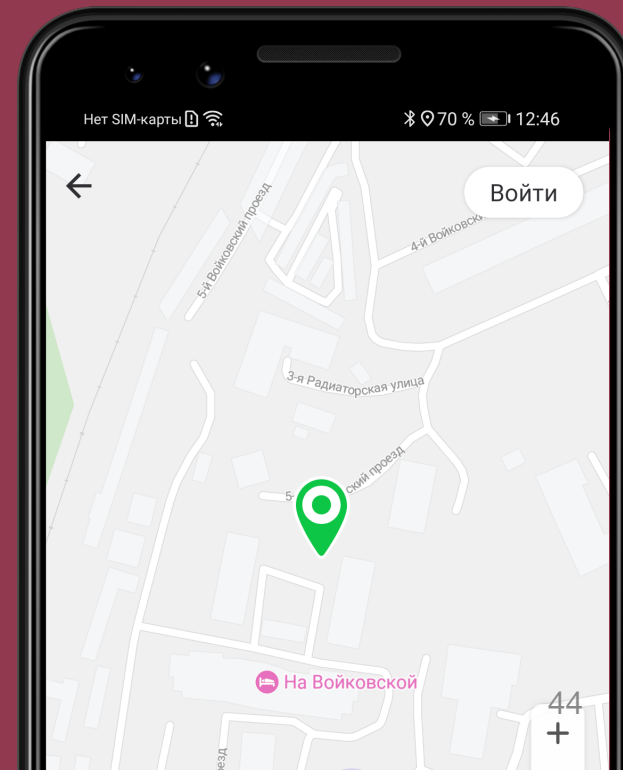
Google Play



+ 500 kb



AppGallery





# Когда подменять карту?

```
fun Context.getMobileServiceSource(): MapSource {  
    val googleApi = GoogleApiAvailability.getInstance()  
    if (googleApi.isGooglePlayServicesAvailable(this) == com.google.android.gms.common.ConnectionResult.SUCCESS) {  
        return MapSource.GOOGLE  
    }  
  
    val huaweiApi = HuaweiApiAvailability.getInstance()  
    if (huaweiApi.isHuaweiMobileServicesAvailable(this) == com.huawei.hms.api.ConnectionResult.SUCCESS) {  
        return MapSource.HMS  
    }  
}
```

# EMUI Version-Dependent Features

No.	Kit	Platform-Dependent Feature (OS or Hardware)	Required EMUI Version (HUAWEI/ HONOR Phones and Tablets)	Required Android Version (Non-Huawei Phones)
8	Map Kit	Running as a sub app in developers app	EMUI 5.0 or later	Not supported

[https://developer.huawei.com/consumer/en/doc/development/HMS-Guides/emui\\_version\\_dependent\\_features](https://developer.huawei.com/consumer/en/doc/development/HMS-Guides/emui_version_dependent_features)

# EMUI Version history

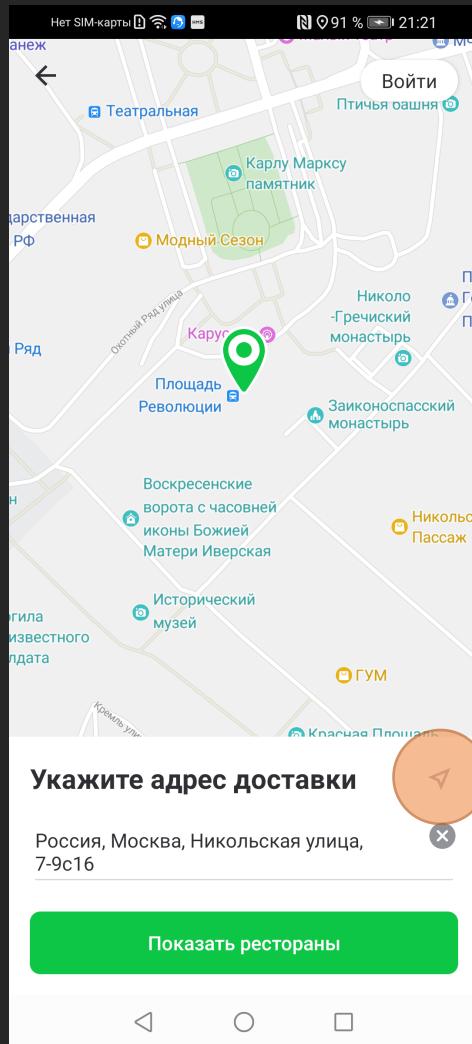
Version	Android version	Year of release
Emotion UI 1.x	Android 2.3 - 4.3	2012
Emotion UI 2.x	Android 4.2 - 4.4	2013
EMUI 3.x	Android 4.4 - 5.1	2014
EMUI 4.x	Android Marshmallow (6.0)	2015
EMUI 5.x	Android Nougat (7.x)	2016
EMUI 8.x <sup>[A]</sup>	Android Oreo (8.x)	2017
EMUI 9.x	Android Pie (9.0)	2018
EMUI 10.x	Android 10 (10.0)	2019
EMUI 11.x	Android 11 (11.0)	2020

\* Статистика распространения EMUI не найдена(

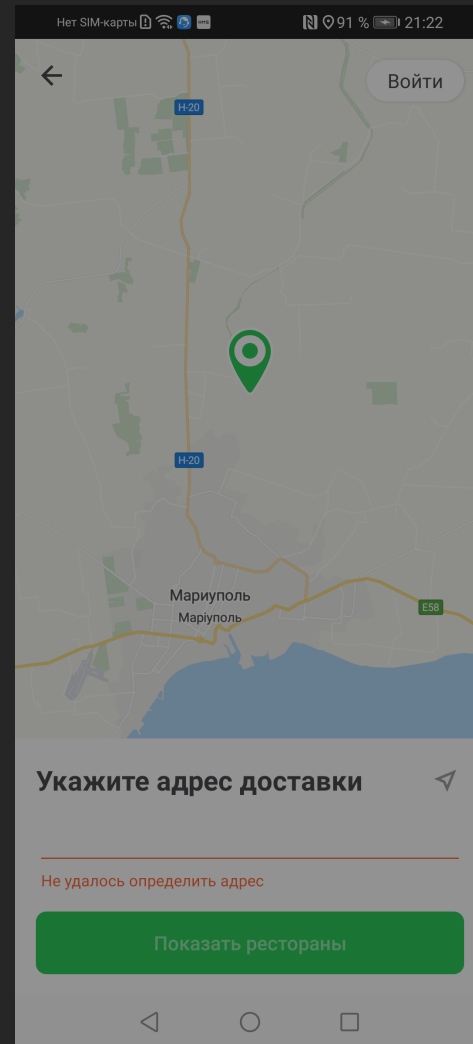
\*\* EMUI 8.0 был запущен без EMUI 6.0 и 7.0, чтобы соответствовать версии Android.

# «Карта не работает»

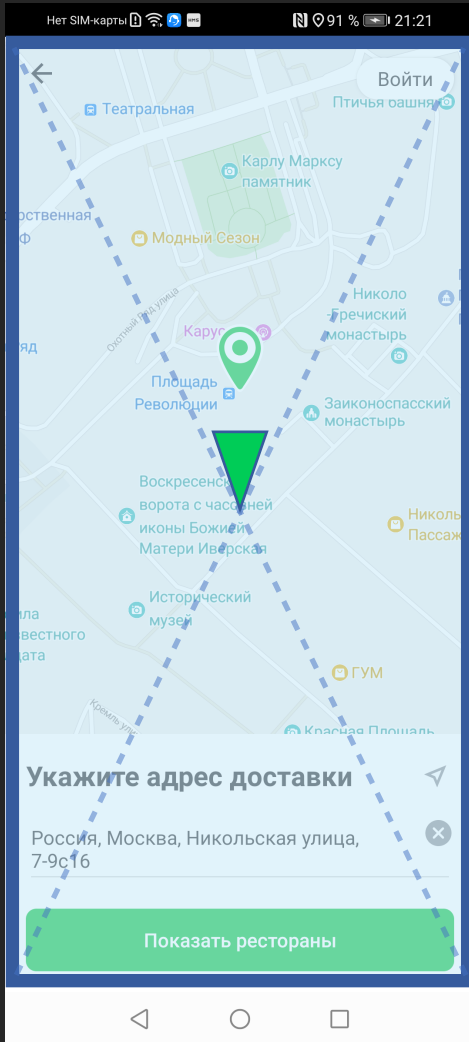
55.819207, 37.493424



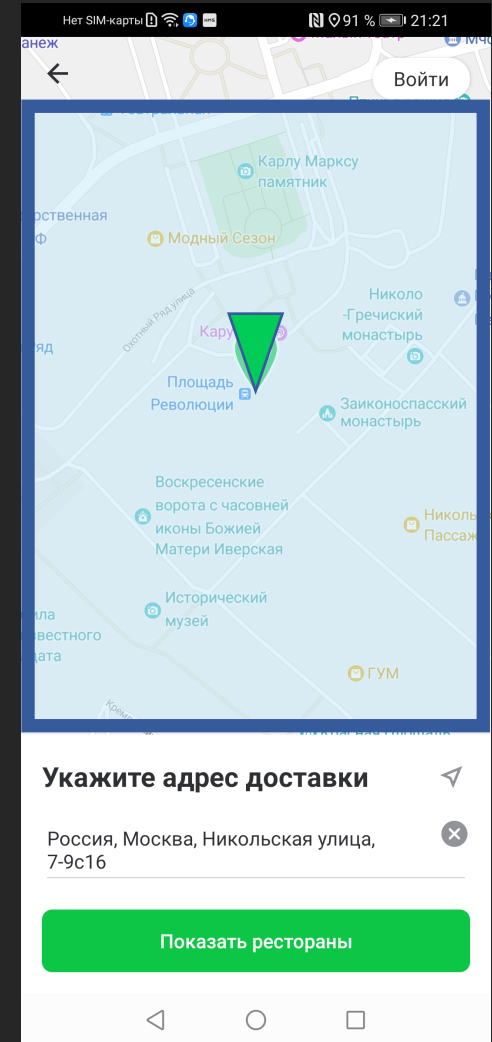
47.187447, 37.593137



# Padding



```
map.setPadding(0, 200, 0, 500)
```



# Варианты решения

1 НЕ ИСПОЛЬЗОВАТЬ ПАДДИНГИ;

2 ИСПОЛЬЗОВАТЬ АНИМИРОВАННОЕ ПЕРЕМЕЩЕНИЕ С ЗУМОМ;

```
val zoom = map.cameraPosition.zoom  
map.animateCamera(CameraUpdateFactory.newLatLngZoom(position), zoom))
```

3 ИСПОЛЬЗОВАТЬ ПЕРЕМЕЩЕНИЕ БЕЗ АНИМАЦИИ.

```
moveCamera  
map.animateCamera(CameraUpdateFactory.newLatLngZoom(position))
```

# Замена карты. Итого.

1

Для поддержки нескольких карт необходимо писать обертку (а так же для базовых моделей);

2

Google Map и Huawei Map не идентичны;

3

Переключение карт происходит в зависимости от текущей системы;

4

Выключили анимацию перемещений для Huawei Map.

# Google Services в Delivery Club. Итого.



1 Карты;

2

Push-уведомления;

3

ChromeTabs.



# Замена push service

```
Было: class ImplementationMessagingService : FirebaseMessagingService() {  
  
    override fun onNewToken(token: String) {  
        val commonApi = getComponentFactory().get(CommonApi::class.java)  
        commonApi.settingsManager().setPushToken(token)  
    }  
  
    override fun onMessageReceived(message: RemoteMessage) {  
        message.from?.let { from ->  
            val appManagersComponent = getComponentFactory().get(AppManagersApi::class.java)  
            appManagersComponent.pushManager().handle(message.data)  
        }  
    }  
}
```

# Замена push service

Было:

```
FirebaseMessaging.getInstance().token.addOnCompleteListener { task ->
    if (task.isSuccessful) {
        val token = task.result
    }
}
```

# Замена push service

Стало: `class ImplementationHuaweiMessagingService : HmsMessageService() {`

```
    override fun onNewToken(token: String?) {
```

```
        val commonApi = getComponentFactory().get(CommonApi::class.java)
```

```
        commonApi.settingsManager().setPushToken(token)
```

```
    }
```

```
    override fun onMessageReceived(message: RemoteMessage?) {
```

```
        message?.let {
```

```
            val appManagersComponent = getComponentFactory().get(AppManagersApi::class.java)
```

```
            appManagersComponent.pushManager().handle(it.dataOfMap)
```

```
        }
```

```
    }
```

# Замена push service

Стало: `val token = HmsInstanceId.getInstance(context)  
 .getToken(appId, com.huawei.hms.push.HmsMessaging.DEFAULT_TOKEN_SCOPE)`

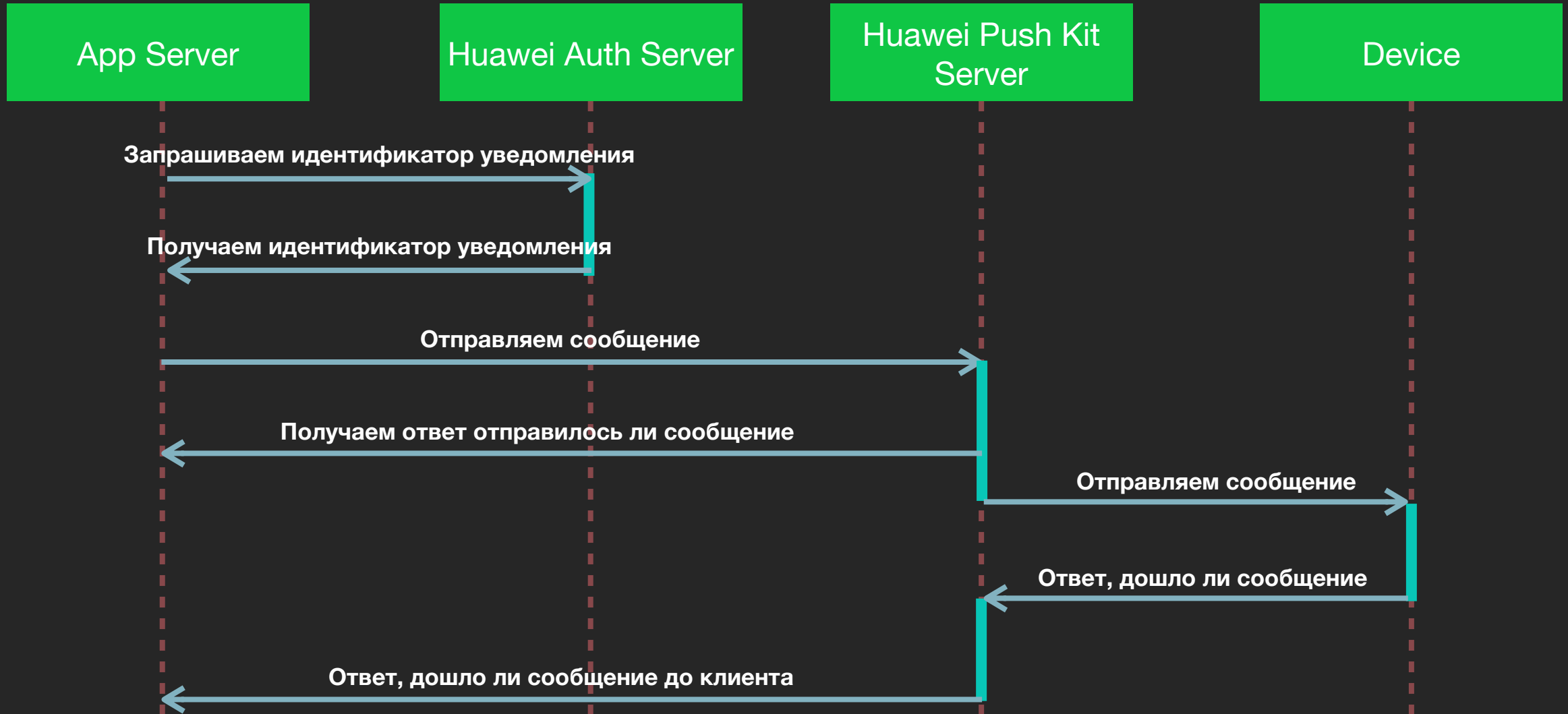
```
public static final String DEFAULT_TOKEN_SCOPE = "HCM";
```

# Как отправлять пуши

POST [https://push-api.cloud.huawei.com/v1/\[appid\]/messages:send](https://push-api.cloud.huawei.com/v1/[appid]/messages:send)  
Authorization: Bearer CgB6e3x9pzAkPRPDoctuJ...

```
{
  "validate_only": false,
  "message": {
    "data": "{ 'param1': 'value1', 'param2': 'value2' }",
    "notification": {
      "title": "Title 1",
      "body": "Body 1"
    },
    "android": {
      "notification": {
        "title": "Android Title 1",
        "body": "Android Body 1",
        "click_action": {
          "type": 1,
          "intent": "#Intent;compo=com.rvr/.Activity;S.W=U;end"
        }
      }
    }
  },
  "token": [
    "client_token_1",
    "client_token_2"
  ]
}
```

# Как отправлять пуши



# HUAWEI Push Kit

Отправка сообщений на устройства целевой аудитории в режиме реального времени.

Примечание. Если ваше приложение не добавлено в проект, Push Kit нельзя будет включить. Если место хранения данных не указано, отправка сообщений по теме, группе устройств или типу Web Push Agent будет недоступна. Если функция расширенного анализа не включена, тест A/B, прогнозы, сообщения для аудиторий и отчеты будут недоступны.

Включить



## Текст сообщения

Скрыть ^

\* Имя ?

0/40

\* Тип ?

Уведомление  Информация

\* Режим отображения

Стандартный стиль  Большой текст

Заголовок push-окна

0/40

\* Заголовок сообщения

0/40

\* Текст сообщения

0/1024

Кнопки

Добавить кнопку

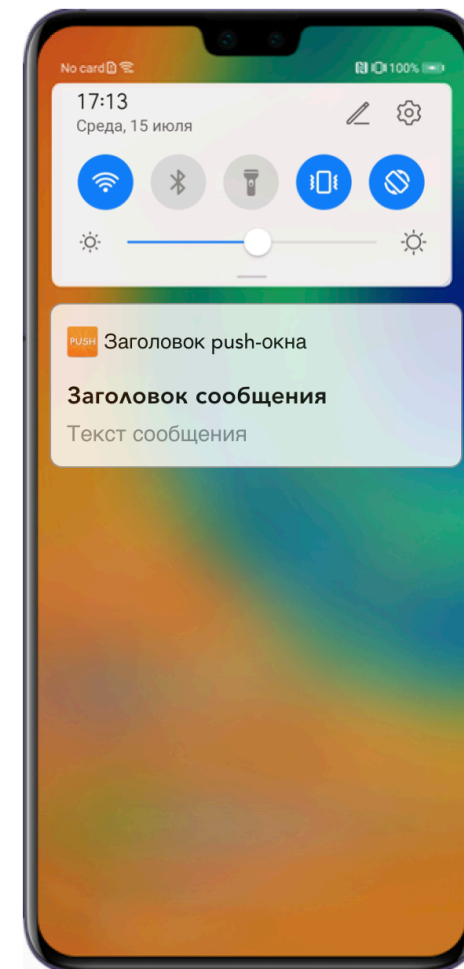
\* Действие

Открыть приложение



\* Страница приложения

Главная страница





# HMS Push. Итого.

1

Изменения как на Backend так и на клиенте минимум;

2

На клиента отдельные приемники токена и данных, обработка общая.

3

На сервере вся подготовка остается прежней, меняется только адрес отправки.

4

Google и Huawei имеют удобные консоли для формирования и отправки уведомления.

# Google Services в Delivery Club. Итого.



Карты;



Push-уведомления;



ChromeTabs.

# Chrome Custom Tabs - ActivityNotFoundException

```
fun Context.openLink(url: String, customTabsSession: CustomTabsSession? = null): Boolean {
    try {
        CustomTabsIntent.Builder(customTabsSession)
            .build()
            .launchUrl(this, Uri.parse(url))
        return true
    } catch (throwable: Throwable) {
        Timber.tag("Context::openLink").e(throwable, "CustomTabsIntent error on url: $url")
    }

    return openLinkInBrowser(url)
}
```

# Ссылка в Store

[https://play.google.com/store/apps/details?id=  
com.deliveryclub](https://play.google.com/store/apps/details?id=com.deliveryclub)  
<https://appgallery.huawei.com/#/app/C101189677>

# Google Services в Delivery Club. Итого.



Карты;



Push-уведомления;



ChromeTabs.

# Вопросы?

# Что еще есть в HMS Core?

# Замена аналитики

- Инициализация : `HiAnalytics.getInstance(context)`
- Копируем Firebase analytics events, если есть в проекте
- Заменяем `firebaseAnalytics.logEvent(name, bundle)` -> `huaweiAnalytics.onEvent(name, bundle)`
- Меняем системные ивенты :  
`FirebaseAnalytics.Event.BEGIN_CHECKOUT` -> `HAEventType.STARTCHECKOUT`
- Меняем системные параметры :  
`FirebaseAnalytics.Param.PRICE` -> `HAParamType.PRICE`  
`FirebaseAnalytics.Param.CURRENCY` -> `HAParamType.CURRRNAME`
- Не у каждого event и param есть аналоги в huawei, их придется заменить на кастомные



# Аналитика Итого.

1

Легко внедряется Huawei  
Analytics Kit

2

Простор в изучении  
показателей активности  
клиентов в приложении

# Crashlytics -> AGConnectCrash

```
AGConnectCrash.getInstance().enableCrashCollection(true)
```

```
AGConnectCrash.getInstance().setUserId("testuser")
AGConnectCrash.getInstance().log(Log.DEBUG, "set debug log.")
AGConnectCrash.getInstance().log(Log.INFO, "set info log.")
AGConnectCrash.getInstance().log(Log.WARN, "set warning log.")
AGConnectCrash.getInstance().log(Log.ERROR, "set error log.")
AGConnectCrash.getInstance().setCustomKey("stringKey", "Hello world")
AGConnectCrash.getInstance().setCustomKey("booleanKey", false)
AGConnectCrash.getInstance().setCustomKey("doubleKey", 1.1)
AGConnectCrash.getInstance().setCustomKey("floatKey", 1.1f)
AGConnectCrash.getInstance().setCustomKey("intKey", 0)
AGConnectCrash.getInstance().setCustomKey("longKey", 11L)
```

# Тестирование

# Тестирование



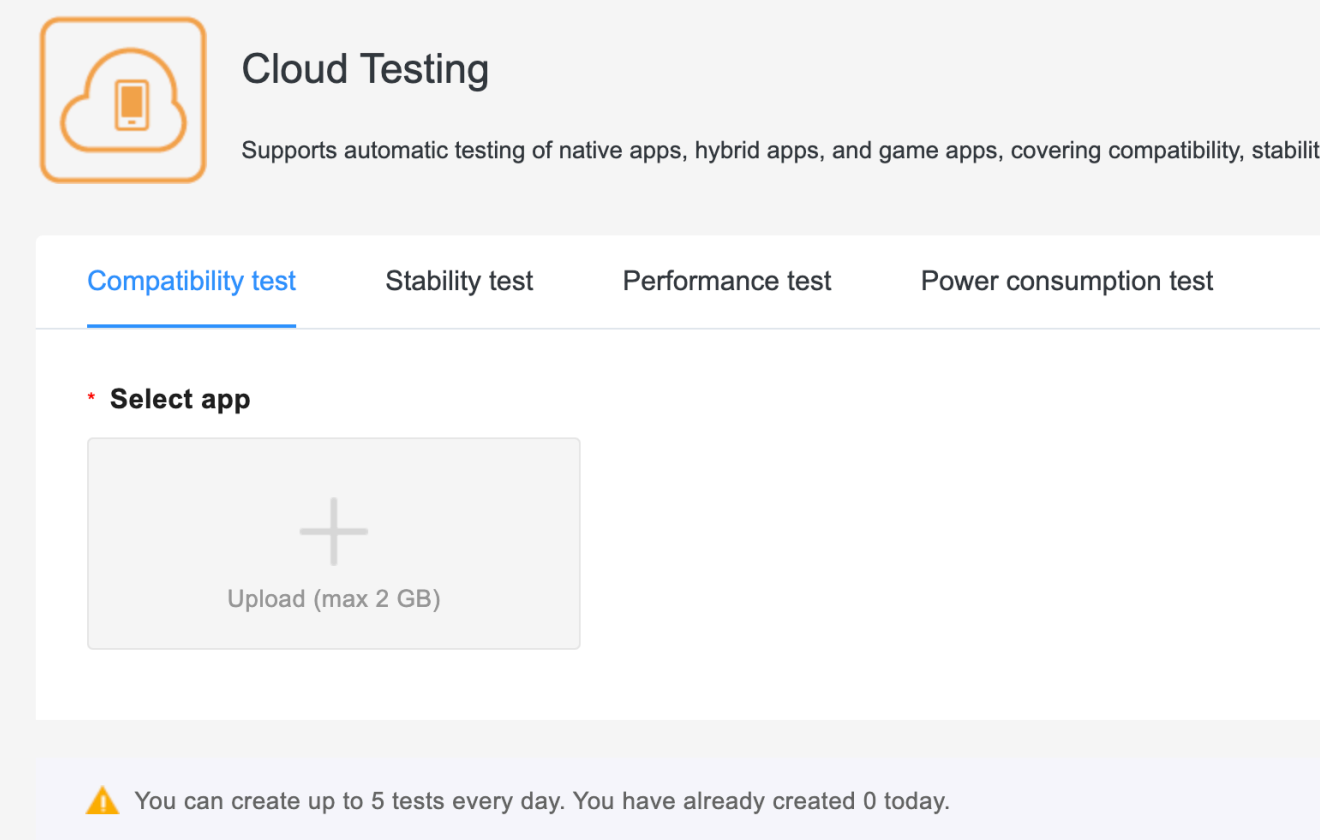
Облачное тестирование



Облачная служба отладки

# Huawei Cloud Testing

- На подобии Firebase Test Lab - Huawei сделали свое облачное тестирование (но по сути это просто monkey test, если ничего не падает тест пройдет успешно)



**Cloud Testing**  
Supports automatic testing of native apps, hybrid apps, and game apps, covering compatibility, stability, performance, and power consumption.

[Compatibility test](#)   [Stability test](#)   [Performance test](#)   [Power consumption test](#)

**\* Select app**

Upload (max 2 GB)

**⚠** You can create up to 5 tests every day. You have already created 0 today.

# Huawei Cloud Debugging

AppGallery Connect Мои проекты

Облачная отладка  
Отладка приложений на удаленных устройствах Android в режиме реального времени. [Описание службы](#)

Модели устройств Мои данные

Смартфон Планшет

Регион	Все	Russia Emulators	Russia Lab		
Серия	Все	mate	nova	p	
ОС	Все EMUI	10.0.0	9.1.1	9.1.0	9.0.0
	Все Android	10	9		
Тип устройства	Все	Эмулятор	Реальное устройство		


Все  Доступные

**nova 5i** Новое

Регион: Russia Lab

Версия Android 9 Разрешение 1080 \* 2310

Версия EMUI Модель




**P40 Lite** Новое

Регион: Russia Lab

Версия Android 9 Разрешение 720 \* 1560

Версия EMUI Модель




**Mate 30 Pro**

Регион: Russia Emulators

Версия Android 10 Разрешение 1176 \* 2400

Версия EMUI Модель




**Mate 30**

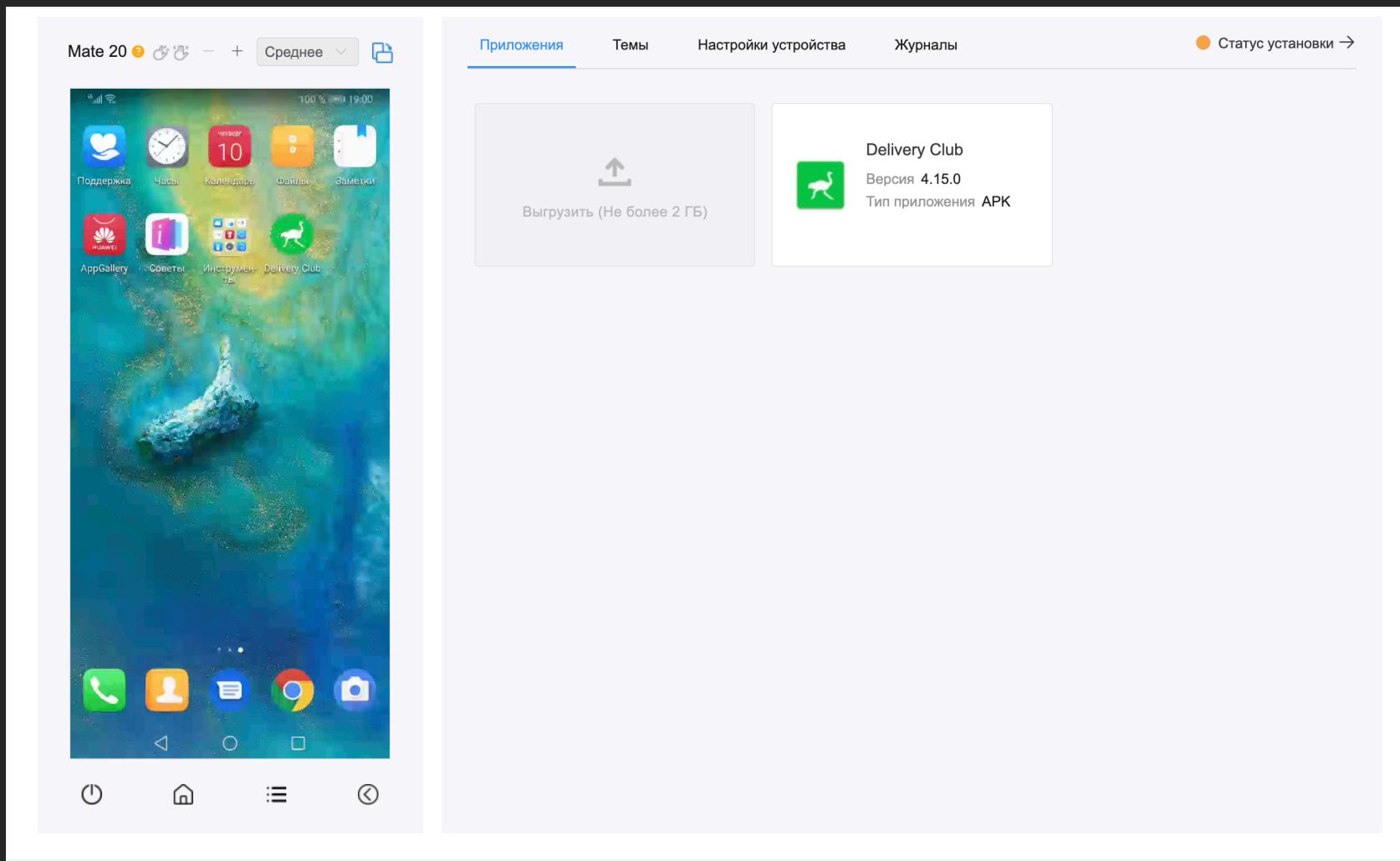
Регион: Russia Emulators

Версия Android 10 Разрешение 1080 \* 2340

Версия EMUI Модель



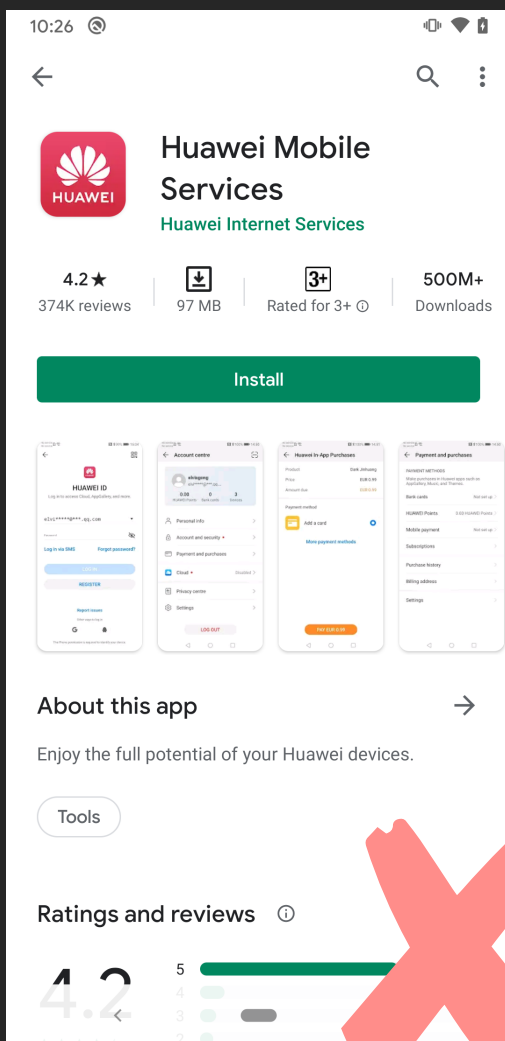
# Huawei Cloud Debugging



# Тестирование на устройстве

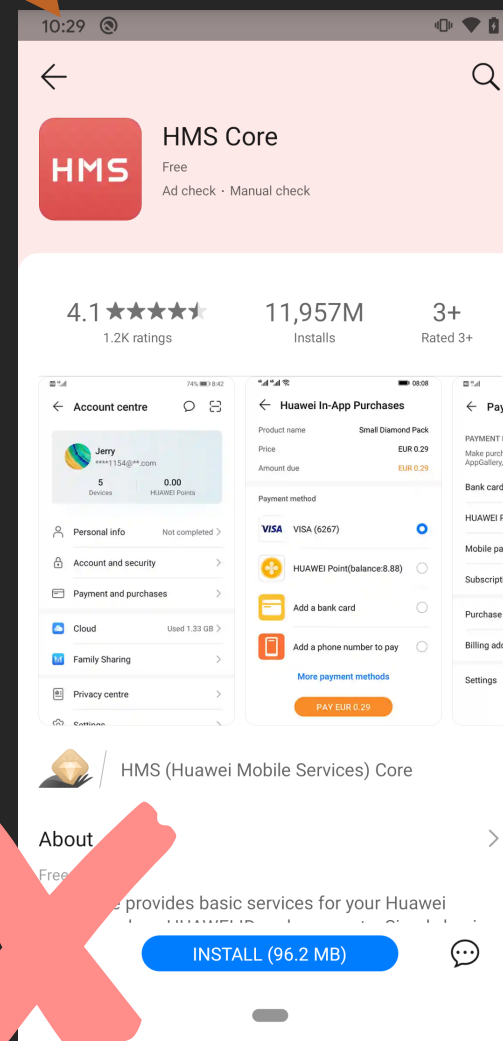
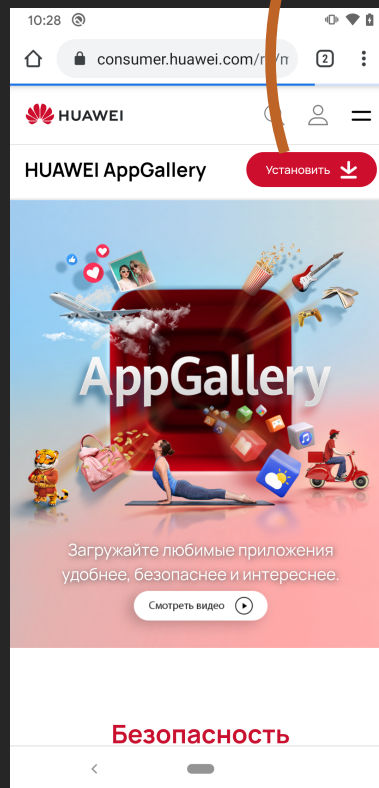
## Google Play

3.0.3.300  
Dec 2, 2019



## AppGallery

5.0.3.304  
Oct 22, 2020





# Huawei Alpha, Beta testing

AppGallery Connect Мои приложения

Распространение Управление

Публикация приложений

- Данные приложений
- Данные версии
- Черновик**
- Версия/Обновить

Службы

- Предварительная ре...
- Подпись приложения

Запрос журналов

- Журнал версий New
- Событие приложения

**Черновик**

Чтобы продолжить, укажите требуемые [сведения приложения](#).

Статус: Черновик

Комментарии проверки: --

Страны/регионы

\* Страны/регионы: Управление 198 выбрано , и "Новая ст

- Глобально
- КНР (кроме Гонконга, Макао и Тайваня)
- КНР (кроме Г...

# Huawei Alpha, Beta testing

## Applying for an Open Test

The Opening Testing service is still in beta state. To launch an open test, send an email to [agconnect@huawei.com](mailto:agconnect@huawei.com) for application.



# Тестирование

1

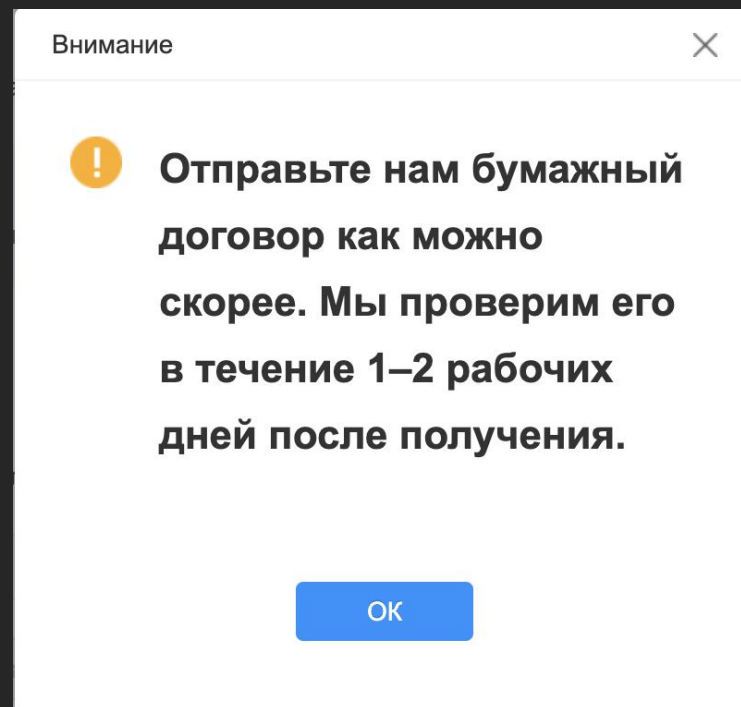
Тестовое устройство Huawei  
без GMS;

2

Тестирование приложения в  
консоли Huawei.

# Публикация приложения

- Проверка данных при регистрации может занять до 2-х недель
- Чтобы стать продавцов нужно отправить по почте “Согласие на обработку персональных данных”



# In-App Purchases

При старте приложения запрашиваем текущие покупки пользователя

```
fun getOwnedPurchases(  
    activity: Activity,  
    ownedPurchasesResultOnSuccessListener: OnSuccessListener<OwnedPurchasesResult>,  
    failureListener: OnFailureListener  
) {  
    val ownedPurchasesReq = OwnedPurchasesReq()  
  
    // priceType: 0: consumable; 1: non-consumable; 2: auto-renewable subscription  
    ownedPurchasesReq.priceType = IapClient.PriceType.IN_APP_SUBSCRIPTION  
  
    // To get the Activity instance that calls this API.  
    val task: Task<OwnedPurchasesResult> = IapClient(activity)  
        .obtainOwnedPurchases(ownedPurchasesReq)  
  
    task.addOnSuccessListener(ownedPurchasesResultOnSuccessListener)  
        .addOnFailureListener(failureListener)  
}
```

# In-App Purchases

Запрашиваем детали по товарам для продажи

```
fun loadProduct(
    context: Context,
    productInfoResultOnSuccessListener: OnSuccessListener<ProductInfoResult>,
    onFailureListener: OnFailureListener
) {
    // obtain in-app product details configured in AppGallery Connect, and then show the products
    val iapClient: IapClient = IapClient.getInstance(context)
    val task: Task<ProductInfoResult> = iapClient.obtainProductInfo(createProductInfoReq())
    task.addOnSuccessListener(productInfoResultOnSuccessListener)
        .addOnFailureListener(onFailureListener)
}

private fun createProductInfoReq(): ProductInfoReq {
    val req = ProductInfoReq()
    // 0: consumable ; 1: non-consumable ; 2: auto-renewable subscription
    req.priceType = IapClient.PriceType.IN_APP_SUBSCRIPTION

    val productIds = ArrayList<String>()
    productIds.add("PRODUCT_ID")
    req.productIds = productIds
    return req
}
```

# In-App Purchases

Открываем страницу оплаты

```
fun gotoPay(activity: Activity, productId: String, type: Int) {
    val client: IapClient = Iap.getIapClient(activity)
    val task: Task<PurchaseIntentResult> = client.createPurchaseIntent(createPurchaseIntentReq(type, productId))
    task.addOnSuccessListener { result ->
        result?.let {
            val status: Status = result.status
            if (status.hasResolution()) {
                try {
                    status.startResolutionForResult(activity, PAY_RESULT_ARG)
                } catch (exception: SendIntentException) {
                    Timber.e(exception)
                }
            } else {
                Timber.d("intent is null")
            }
        }
    }.addOnFailureListener { exception ->
        Timber.e(exception)
    }
}
```

# In-App Purchases

Обрабатываем результат оплаты

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
    super.onActivityResult(requestCode, resultCode, data)  
    if (resultCode == PAY_RESULT_ARG) {  
        val purchaseResultInfo: PurchaseResultInfo = lap.getlapClient(this).parsePurchaseResultInfoFromIntent(data)  
        when (purchaseResultInfo.returnCode) {  
            OrderStatusCode.ORDER_STATE_SUCCESS -> {  
                successResult(purchaseResultInfo)  
            }  
  
            OrderStatusCode.ORDER_STATE_CANCEL -> {  
  
            }  
  
            OrderStatusCode.ORDER_PRODUCT_OWNED -> {  
  
            }  
        }  
    }  
}
```



# In-App Purchases

Добавляем клиенту новый товар

```
private fun successResult(purchaseResultInfo: PurchaseResultInfo) {  
    val inAppPurchaseData = InAppPurchaseData(purchaseResultInfo.inAppPurchaseData)  
    val req = ConsumeOwnedPurchaseReq()  
    req.purchaseToken = inAppPurchaseData.purchaseToken  
  
    val client: IapClient = Iap.getIapClient(this)  
    val task: Task<ConsumeOwnedPurchaseResult> =  
        client.consumeOwnedPurchase(req)  
  
    task.addOnSuccessListener {  
        // Consume success  
    }.addOnFailureListener { exception ->  
        Timber.e(exception)  
    }  
}
```

# Developer Program Policies

Huawei



Google



# Документация

The screenshot shows the Huawei Developer website interface. At the top, there is a navigation bar with the Huawei logo, language options (English), and user actions (Sign in, Sign up). Below this is a secondary navigation bar with categories like DEVELOPERS, Develop, Distribute, Monetize, Solutions, Events, Programs, Community, Training, and Support. The main content area is titled 'HMS Core' and has sub-sections for Overview, Guides (selected), References, Examples, and Library. A left sidebar lists various kits: Map Kit (expanded), Android SDK, ML Kit, Nearby Service, and Panorama Kit. Under 'Map Kit', 'About the Service' is selected. The main content area features a large heading 'About the Service' followed by a 'Service Introduction' section, which states that HMS Core Map Kit is an SDK for map development covering over 200 countries. Below this is a 'Use Cases' section with three bullet points: Map display, Map interaction, and Map drawing. A 'Restrictions' section is also present. On the right side, a vertical list of links includes Service Introduction, Use Cases, Restrictions, Supported Devices, Supported Locations, Privacy Statement, and Copyright Statement. At the bottom, a table header for 'Supported Devices' is visible, with columns for Device, Device Type, and HMS Core (APK) Version.

<https://developer.huawei.com/consumer/en/doc/overview/HMS>

# Документация

1

Много codelabs;

2

Примеров кода много, все на java;

3

Migration From Google to HUAWEI;

4

Android / Cross-Platform;

# Ссылки

Гайды по интеграции  
библиотек



HUAWEI HMS CORE



In-App Purchases



# Выводы

1 ПОДДЕРЖКА В ПРИЛОЖЕНИИ HMS - ВАЖНО

2 ЗАТРАЧЕННОЕ ВРЕМЯ - НЕДЕЛЯ

3 ХОРОШАЯ ДОКУМЕНТАЦИЯ

4 УДОБНОЕ ТЕСТИРОВАНИЕ

5 БЫСТРЫЙ РОСТ СИСТЕМЫ

# Спасибо за внимание

---

[p.borzikov@corp.mail.ru](mailto:p.borzikov@corp.mail.ru)

[g.gigauri@corp.mail.ru](mailto:g.gigauri@corp.mail.ru)



