

# Курс:

## Программирование под Android с нуля

Тренер:

Пономарев Алексей

[oleksii\\_android@web-academy.com.ua](mailto:oleksii_android@web-academy.com.ua)

Лекция 17

HTTP, Firebase



# Firebase

<https://console.firebase.google.com/>

- Database
- File storage
- Hosting
- Notifications


# Создание проекта

## Создать проект



Название проекта

Web-academy

Страна/регион 

Украина



Данные Firebase Analytics нужны, чтобы сделать другие сервисы Firebase и Google лучше. Проверить, к каким именно сведениям Firebase Analytics есть доступ у других сервисов, можно в любое время. [Подробнее...](#)

ОТМЕНА

СОЗДАТЬ ПРОЕКТ

# Добавление приложения, ч. 1

## Добавление Firebase в приложение для Android



1

Введите данные  
приложения

2

Скопируйте файл  
конфигурации

3

Добавьте код к  
файлу build.gradle

Название пакета ?

ua.com.webacademy

Package name приложения

Хеш сертификата для отладки (необязательно) ?

45:D0:E8:39:5D:AB:F7:36:B4:CB:70:43:77:B5:55:53:39:5B:3C:84

Требуется для поддержки Dynamic Links Invites и Google Sign-In в Auth. Изменить хеш SHA-1 можно в "Настройках".

ОТМЕНА

ДОБАВИТЬ ПРИЛОЖЕНИЕ

скачать  
google-services.json для  
вашего приложения

# Добавление приложения, ч. 1.1

- Найти утилиту **keytool** в папке где установлена java.
- Для получения хеша сертификата выполнить команду, пароль **android**:
  - windows: *keytool -exportcert -list -v -alias androiddebugkey -keystore %USERPROFILE%\android\debug.keystore*
  - mac: *keytool -exportcert -list -v -alias androiddebugkey -keystore ~/.android/debug.keystore*
- Скопировать значение **SHA1**

# Добавление приложения, ч. 2

## Добавление Firebase в приложение для Android



1

Введите данные  
приложения

2

Скопируйте файл  
конфигурации

3

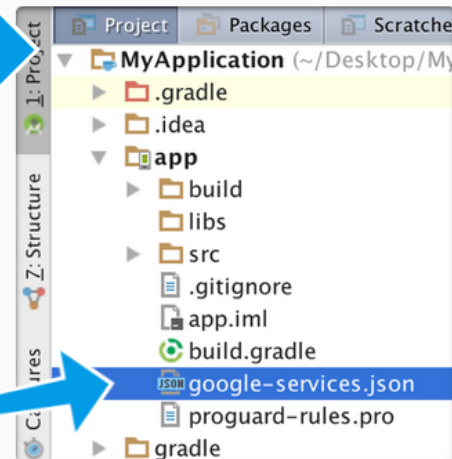
Добавьте код к  
файлу build.gradle

Открыв вид **Проект** в Android Studio, перейдете в корневой каталог своего проекта.

Переместите скачанный файл **google-services.json** в корневой каталог модуля для приложений Android.



google-services.json



Зависимости уже добавлены?  
[Закреть](#)

ПРОДОЛЖИТЬ

# Добавление приложения, ч. 3

## Добавление Firebase в приложение для Android



1


Введите данные  
приложения

2

Скопируйте файл  
конфигурации

3

Добавьте код к  
файлу build.gradle


Плагин сервисов Google для [Gradle](#)  загружает только что скачанный файл `google-services.json`. Чтобы использовать плагин, измените файлы `build.gradle`.

1. Файл `build.gradle` уровня проекта (`<project>/build.gradle`):

```
buildscript {  
    dependencies {  
        // Add this line  
        classpath 'com.google.gms:google-services:3.0.0'  
    }  
}
```

2. Файл `build.gradle` уровня приложения (`<project>/<app-module>/build.gradle`):

```
...  
// Add to the bottom of the file  
apply plugin: 'com.google.gms.google-services'
```

по умолчанию включает сервис *Firebase Analytics* 

3. Нажмите кнопку **Sync now** (Синхронизовать), которая появится на панели в интегрированной среде разработки:

Gradle files have changed sir

[Sync now](#)

# Разрешение на запись

Realtime Database



ДАННЫЕ

ПРАВИЛА

ИСПОЛЬЗОВАНИЕ

СИМУЛЯТОР

```
1 {  
2   "rules": {  
3     ".read": true,  
4     ".write": true  
5   }  
6 }
```



# Хранение данных

```
{
  "rooms" : {
    "room 1" : {
      "description" : "main room"
    },
    "room 2" : {
      "description" : "test room"
    }
  },
  "messages" : {
    "room 1" : {
      "-KSro9OxA20nGn9l-aWW" : {
        "message" : "message 1",
        "time" : 1474730960653
      }
    },
    "room 2" : {
      "-KSroBiNo3h7ARUW909g" : {
        "message" : "message 2",
        "time" : 1474730970153
      }
    }
  }
}
```

# Список

```
//Подключение к базе данных
```

```
DatabaseReference mDatabaseReference =  
FirebaseDatabase.getInstance().getReference();
```

```
FirebaseRecyclerAdapter<Room, RoomViewHolder> mFirebaseAdapter =  
    new FirebaseRecyclerAdapter<Room, RoomViewHolder>(  
        Room.class,  
        android.R.layout.simple_list_item_2,  
        RoomViewHolder.class,  
        mDatabaseReference.child(FIREBASE_CHILD_ROOMS)  
        .orderByKey()) {
```

```
    @Override
```

```
    protected void populateViewHolder(RoomViewHolder holder,  
    Room room, int position) {  
        //заполнение элемента  
    }
```

```
};
```

```
RecyclerView mListView = (RecyclerView) findViewById(R.id.listView);  
mListView.setAdapter(mFirebaseAdapter);
```

# Сортировка

- **orderByKey** – по ключу
- **orderByChild** – по значению

Сортировка работает только от меньшего к большему, обратной сортировки нет. Сортировка может быть только по одному полю.

```
//Вывод списка в обратном порядке
```

```
LinearLayoutManager layoutManager = new LinearLayoutManager(this);  
layoutManager.setReverseLayout(true);  
layoutManager.setStackFromEnd(true);
```

# Фильтрация

```
String filter = "main";
```

```
mDatabaseReference.child("rooms")  
    .orderByChild("description")  
    .startAt(filter)  
    .endAt(filter + "\uf8ff");
```

Поиск может быть только по одному полю, по которому идет сортировка.

# Выборка одного элемента

```
mDatabaseReference.child("rooms/room 1")
    .addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            Room room = dataSnapshot.getValue(Room.class);
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });
```

**addValueEventListener** подписывает на дальнейшие изменения в элементе.

# Одноразовая выборка

```
mDatabaseReference.child("rooms/room 1")
    .addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            Room room = dataSnapshot.getValue(Room.class);
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });
```

# Вставка/обновление по ключу

```
mDatabaseReference
```

```
.child("rooms/room 1/")
```

```
.setValue(new Room("test room"));
```

# Добавление в коллекцию

```
mDatabaseReference
```

```
.child("messages/room 1/")
```

```
.push()
```

```
.setValue(new Message("message", 14747309701531));
```



# Серверное время

mDatabaseReference

```
.child("messages/room 1/-KSRoBiNo3h7ARUW909g/time")  
.setValue(ServerValue.TIMESTAMP);
```

# Удаление

```
mDatabaseReference  
    .child("rooms/room 1/")  
    .removeValue();
```

# Спасибо за внимание

Тренер:

Пономарев Алексей

[oleksii\\_android@web-academy.com.ua](mailto:oleksii_android@web-academy.com.ua)

Лекция 17  
HTTP, Firebase

