

**Кенчошвили Вадим Викторович**

студент

ФГБОУ ВО «Государственный университет морского  
и речного флота им. адмирала С.О. Макарова»

г. Санкт-Петербург

## **СОЗДАНИЕ ОНЛАЙН-ЧАТА НА ЯЗЫКЕ JAVA**

*Аннотация:* статья отражает основные элементы для написания чата для *Android* на языке *Java*. В основном рассмотрена необходимость подключения к сервису *Firebase*.

*Ключевые слова:* Java, Firebase, чат, библиотека.

### **Введение**

В настоящее время существует огромное множество онлайн-чатов, и у каждого из них есть свои плюсы и минусы. В данной статье будут рассмотрены некоторые особенности языка Java, которые необходимы для создания онлайн-чата с базовым функционалом. Актуальность данной работы объясняется тем, что начинающие программисты на Java не знают, что начать изучать для написания онлайн-чата, а данная статья объяснит многие аспекты его написания.

### **Онлайн-чат**

Перед началом разработки приложения необходимо подключить сервис *Firebase*. *Firebase* – это специальный сервис от *Google*, который способен решить большинство потребностей мобильного разработчика. Данный сервис предоставляет почти все, что необходимо для создания полноценного онлайн-чата. Основное, что необходимо, – это подключить систему *Realtime Database*, которая позволяет обмениваться сообщениями между пользователями в реальном времени. Также при желании можно подключить систему аутентификации «*Authentication*» и систему обмена различными файлами помимо текстовых «*Storage*». Также имеются дополнительные возможности *Firebase*, но они больше как вспомогательные для приложения, и нет необходимости в их установке.

После подключения Firebase необходимо создать layout файлы для графического отображения сообщений. Для базового онлайн-чата (без регистрации и отправки файлов) необходимо создать две страницы. Первая страница необходима чтобы пометить область для вывода сообщений и ввода текста. Вторая страница необходима, чтобы на первой странице сообщения выводились не сплошным списком, а имели свои границы и сообщения пользователей не сливались друг с другом. По итогу две эти страницы помещаются на одну и выглядят следующим образом (рисунок 1):

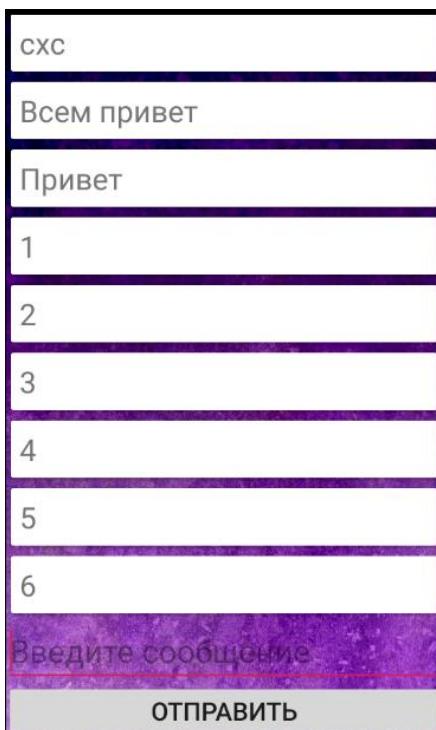


Рис. 1. Страница «Окно чата»

Для такого отображения сообщений и получения возможности просматривания старых сообщений необходимо установить библиотеку RecyclerView и CardView.

Сама логика отправки и получения сообщения прописывается в Activity файле MainActivity. Данный код производит вход в базу данных и производит обмен данными, а также сохраняет копию данной базы данных в своей базе. Код без импортированных библиотек представлен ниже.

```
public class MainActivity extends AppCompatActivity {  
    private static int MAX_MESSAGE_LENGTH = 150;
```

2 <https://interactive-plus.ru>

Содержимое доступно по лицензии Creative Commons Attribution 4.0 license (CC-BY 4.0)

```
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("messages");
EditText mEditTextMessage;
Button mSendButton;
RecyclerView mMessagesRecycler;
ArrayList<String> messages = new ArrayList<>();
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mSendButton = findViewById(R.id.send_message_b);
    mEditTextMessage = findViewById(R.id.message_input);
    mMessagesRecycler = findViewById(R.id.messages_recycler);
    mMessagesRecycler.setLayoutManager(new LinearLayoutManager(this));
    final DataAdapter dataAdapter = new DataAdapter(this, messages);
    mMessagesRecycler.setAdapter(dataAdapter);
    mSendButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String msg = mEditTextMessage.getText().toString();
            if (msg.equals("")){
                Toast.makeText(getApplicationContext(), "Введите сообщение!", Toast.LENGTH_SHORT).show();
                return;
            }
            if (msg.length()== MAX_MESSAGE_LENGTH){
                Toast.makeText(getApplicationContext(), "Слишком длинное сообщение!", Toast.LENGTH_SHORT).show();
                return;
            }
        }
    });
}
```

```
myRef.push().setValue(msg);
mEditTextMessage.setText("");
}
});
myRef.addChildEventListener(new ChildEventListener() {
@Override
public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable
String s) {
String msg = dataSnapshot.getValue(String.class);
messages.add(msg);
dataAdapter.notifyDataSetChanged();
mMessagesRecycler.smoothScrollToPosition(messages.size());
}
@Override
public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable
String s) {
}
@Override
public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {
}
@Override
public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable
String s) {
}
@Override
public void onCancelled(@NonNull DatabaseError databaseError) {
}
});
}
}
}
```

## Перспективы развития

Созданный онлайн-чат можно в дальнейшем развивать во многих направлениях и менять по своему предпочтению. Также, если данное приложение продолжить развивать, то оно может соперничать с другими популярными онлайн-чатами.

## Вывод

Разработанный онлайн-чат выполняет только функцию передачи и получения сообщений и в данный момент имеет ограниченный функционал. Но основная цель данной статьи – показать основные направления и возможности создания онлайн-чата – была выполнена, и, если продолжить развивать представленные системы, можно создавать приложения для больших компаний.

## *Список литературы*

1. Firebase [Электронный ресурс] / Режим доступа:  
<https://www.firebaseio.google.com>
2. Руководство по Java [Электронный ресурс] / Режим доступа:  
<https://docs.oracle.com/javase/tutorial/index.html>