

Чат на KOTLIN с подключением firebase

Артем Бровкин

Лашко Григорий

Группа 22207

Авторизация

Chat ▾

Authentication

Users

Sign-in method

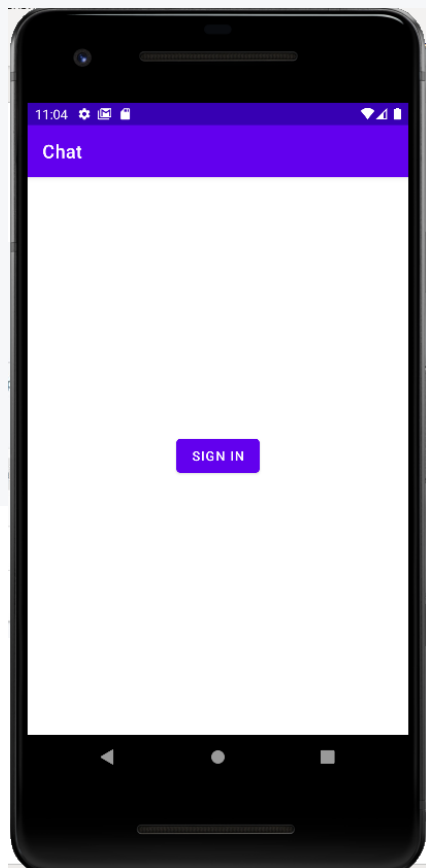
Templates

Usage



Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication

[Get started](#)



Search by email address, phone number, or user UID

[Add user](#)



Identifier

Providers

Created ↓

Signed In

User UID

zetofof@gmail.com



Jan 17, 2022

Jan 17, 2022

hrBio4k6JragLtGEWiHAYXZmi1V2

Rows per page: 50 ▾

1 - 1 of 1



Авторизация

- Была добавлена с помощью firebase
- Готовая вставка из ассистента ругается на default_web_client_id, но несмотря на это работает
- Алгоритм прост - при нажатии на SIGN IN всплывает окошко с вашими гугл аккаунтами с помощью которых можно войти

```
23
24 override fun onCreate(savedInstanceState: Bundle?) {
25     super.onCreate(savedInstanceState)
26     binding = ActivitySignInBinding.inflate(layoutInflater)
27     setContentView(binding.root)
28     auth = Firebase.auth
29     launcher = registerForActivityResult(ActivityResultContracts.StartActivityForResult()) { it: Ac
30     val task = GoogleSignIn.getSignedInAccountFromIntent(it.data)
31     try {
32         val account = task.getResult(ApiException::class.java)
33         if(account != null){
34             firebaseAuthWithGoogle(account.idToken!!)
35         }
36     } catch (e: ApiException){
37         Log.d(tag: "MyLog", msg: "Api exception")
38     }
39 }
40 binding.bSignIn.setOnClickListener { it: View?
41     signInWithGoogle()
42 }
43 checkAuthState()
44 }
45 private fun getClient(): GoogleSignInClient{
46     val gso = GoogleSignInOptions
47         .Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
48         .requestIdToken(getString(R.string.default_web_client_id))
49         .requestEmail()
50         .build()
51
52     return GoogleSignIn.getClient(activity: this, gso)
53 }
54 private fun signInWithGoogle(){
55     val signInClient = getClient()
56     launcher.launch(signInClient.signInIntent)
57 }
58 private fun firebaseAuthWithGoogle(idToken: String){
59     val credential = GoogleAuthProvider.getCredential(idToken, accessToken: null)
60     auth.signInWithCredential(credential).addOnCompleteListener { it: Task<AuthResult!>
61     if(it.isSuccessful){
62         Log.d(tag: "MyLog", msg: "Google signIn done")
63         checkAuthState()
64     } else {
65         Log.d(tag: "MyLog", msg: "Google signIn fail")
66     }
67 }
```

Акти
Чтобы

Авторизация

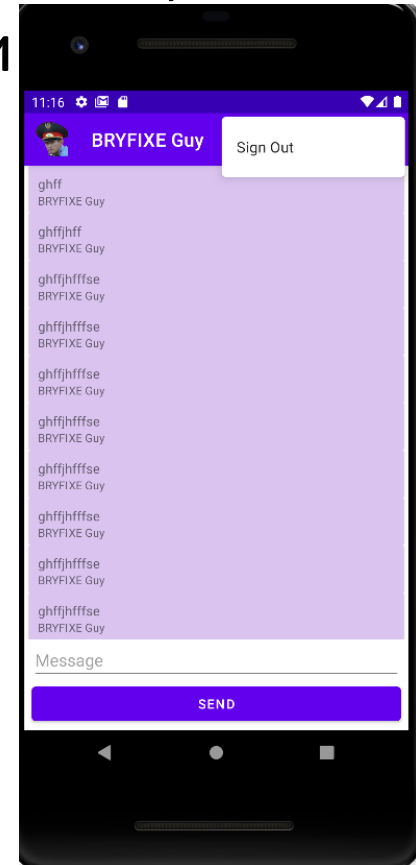
- Были добавлены простые проверки на API exserptoin и проверка на уже авторизованного пользователя
- С вашего аккаунта берется фотография и ваш никнейм, с фотографией помог PICASSO где мы подключили его и с помощью получили битмап и перевели его в битмап дравабл произвели метод Thread так как на основном потоке это занимает много time

```
private fun setUpActionBar(){
    val ab = supportActionBar
    Thread{
        val bMap = Picasso.get().load(auth.currentUser?.photoUrl).get()
        val dIcon = BitmapDrawable(resources, bMap)
        runOnUiThread {
            ab?.setDisplayHomeAsUpEnabled(true)
            ab?.setHomeAsUpIndicator(dIcon)
            ab?.title = auth.currentUser?.displayName
        }
    }.start()
}
```

Авторизация выход

- Так как единожды вы уже зашли через свой гугл аккаунт в чат никаким образом через интерфейс чата вы бы не смогли выйти, разве что удалить и заново скачать, по этому мы добавили кнопочку выхода

```
45 override fun onCreateOptionsMenu(menu: Menu?): Boolean {  
46     menuInflater.inflate(R.menu.main_menu, menu)  
47     return super.onCreateOptionsMenu(menu)  
48 }  
49  
50 override fun onOptionsItemSelected(item: MenuItem): Boolean {  
51     if(item.itemId == R.id.sign_out){  
52         auth.signOut()  
53         finish()  
54     }  
55     return super.onOptionsItemSelected(item)  
56 }
```




Хранение гугл аккаунтов

Search by email address, phone number, or user UID

Add user

↺ ⋮

Identifier	Providers	Created ↓	Signed In	User UID
zetofof@gmail.com		Jan 17, 2022	Jan 17, 2022	hrBio4k6JragLtGEWiHAYXZmi1V2

Rows per page: 50 1 – 1 of 1 < >

Хранение сообщений в realtime database

Кстати для каждого сообщения был сгенерирован свой путь чтобы не сбрасывать значение текущего значения, а добавлять его

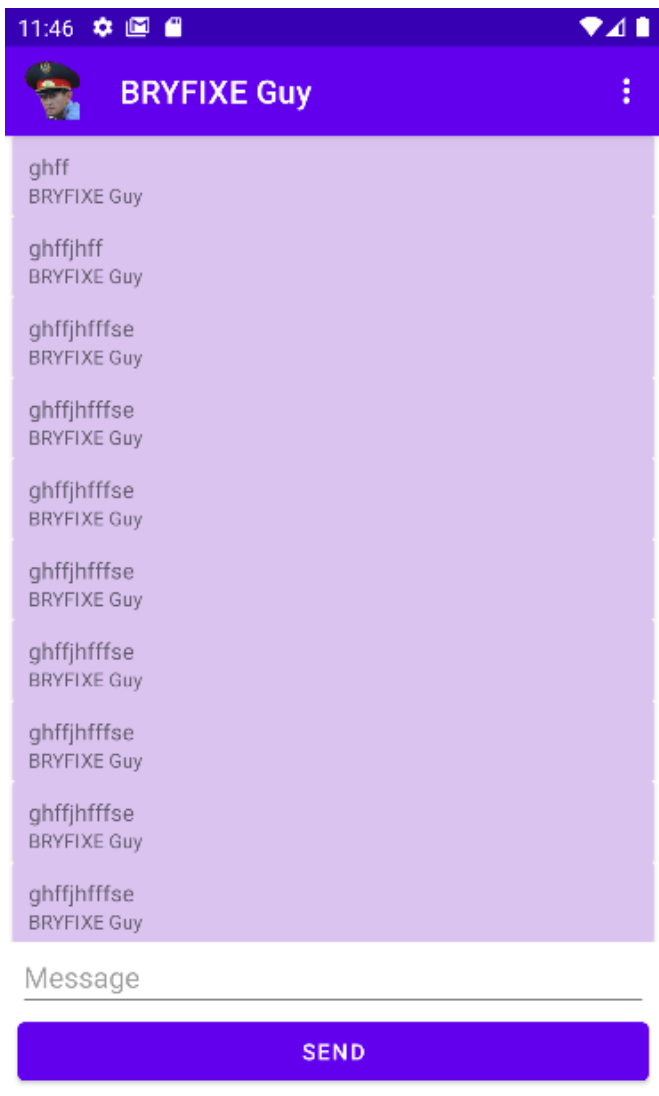
https://chat-4f766-default-rtdb.europe-west1.firebaseio.com/

chat-4f766-default-rtdb

message

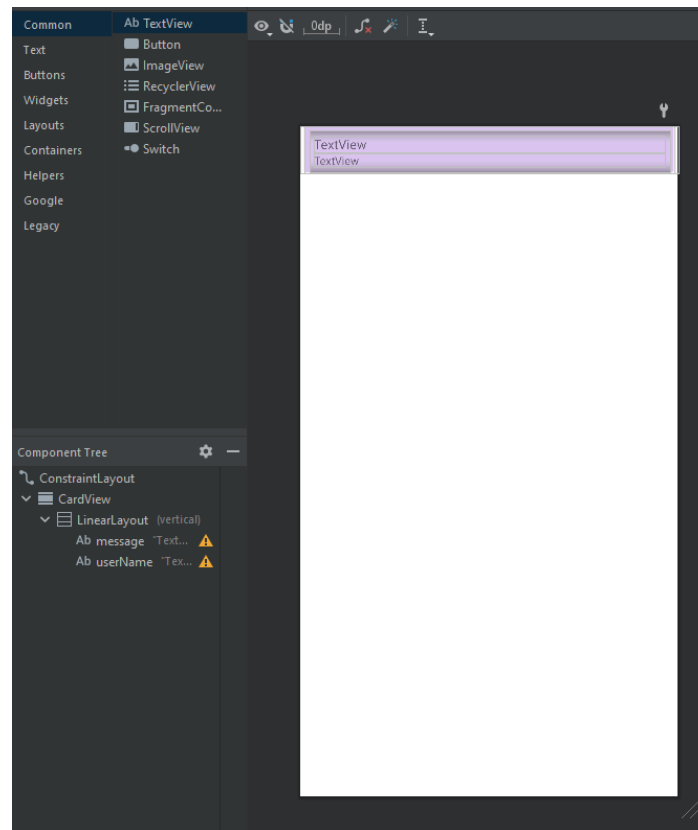
- Mt_xSUEL_m9eGdOYP4V
 - message: "ghff"
 - name: "BRYFIXE Guy"
- Mt_xT-O5YT7FS_-twRL
- Mt_xTMAoGK4Q0tlRKbv
- Mt_xTz6VM07bzI5Fq7z
- Mt_xU6zTcDBj-Tv-pJr
- Mt_xUDxBRBUGs2dn6ej
- Mt_xUK0erNNQ7q35qLI
- Mt_xUMeOlw-CvnThziJ
- Mt_xURot0IM3LRNqRYG
- Mt_xUXW16lDFHlzF79N
- Mt_xUcl5K1E5bG8sShs
- Mt_xUjnJcLFSMLS70Bm
- Mtb0xA5qWvpXHJrz2HM

Database location: Belgium (europe-west1)

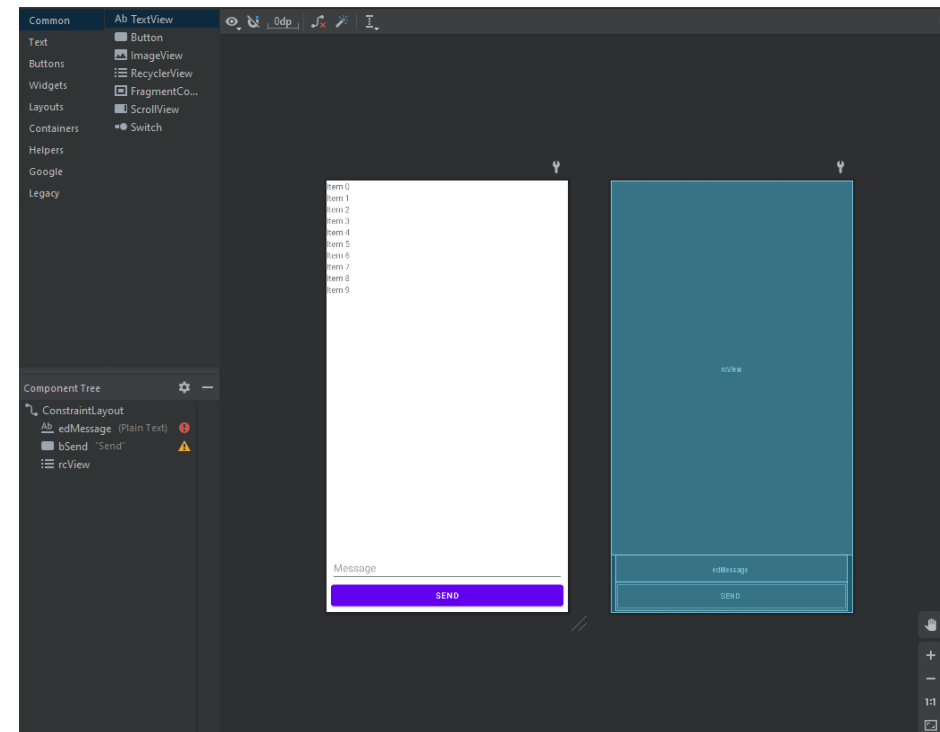


Итоговая разметка
выглядит вот так

Как видно есть
сообщение и никнейм
пользователя



Это rcView если что



По итогу мы имеем:

- Рабочий чат на 30 дней(так как это пробная версия от firebase)
- Хранения зарегистрированных пользователей и их сообщений в базе данных
- Дальше можно красть аккаунты добавив еще несколько строк кода, но можно просто дополнить его и сделать более красивым и функциональным