Demo

RandomUserApp

Korak 1

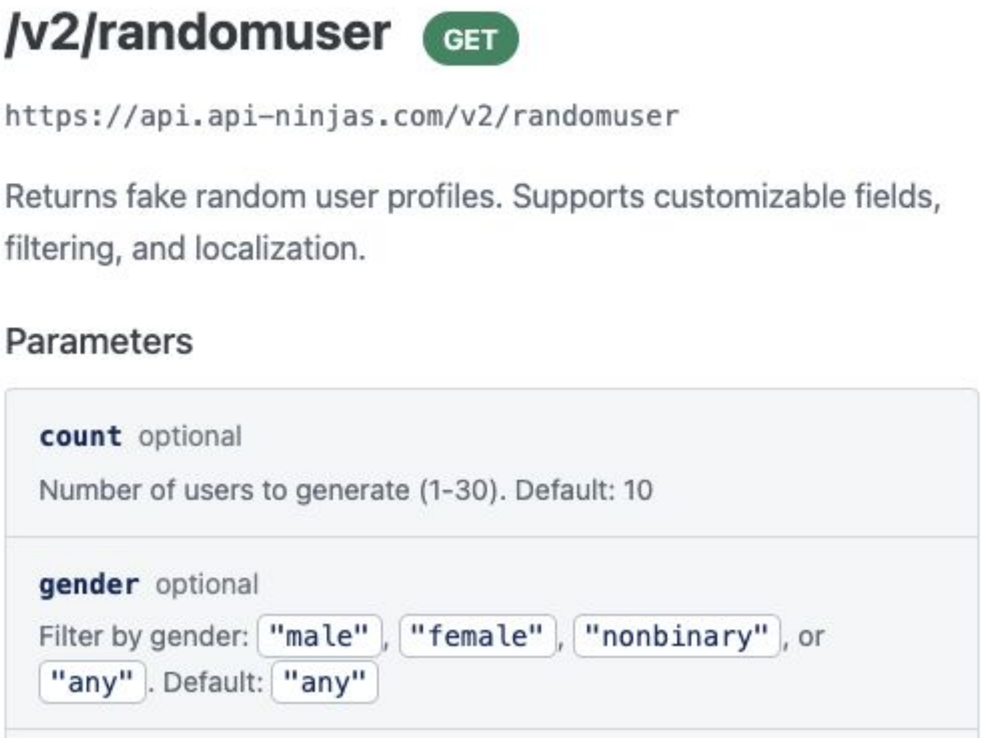
U AndroidManifest.xml dodati prije <application> tag-a:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Dodati Retrofit i Gson dependencies u build.gradle (Module: app)

```
implementation "com.squareup.retrofit2:retrofit:2.11.0"
implementation "com.squareup.retrofit2:converter-gson:2.11.0"

// coroutines
implementation "org.jetbrains.kotlinx:kotlinx-coroutines-android:1.8.1"
```

Random User API



/v2/randomuser GET

`https://api.api-ninjas.com/v2/randomuser`

Returns fake random user profiles. Supports customizable fields, filtering, and localization.

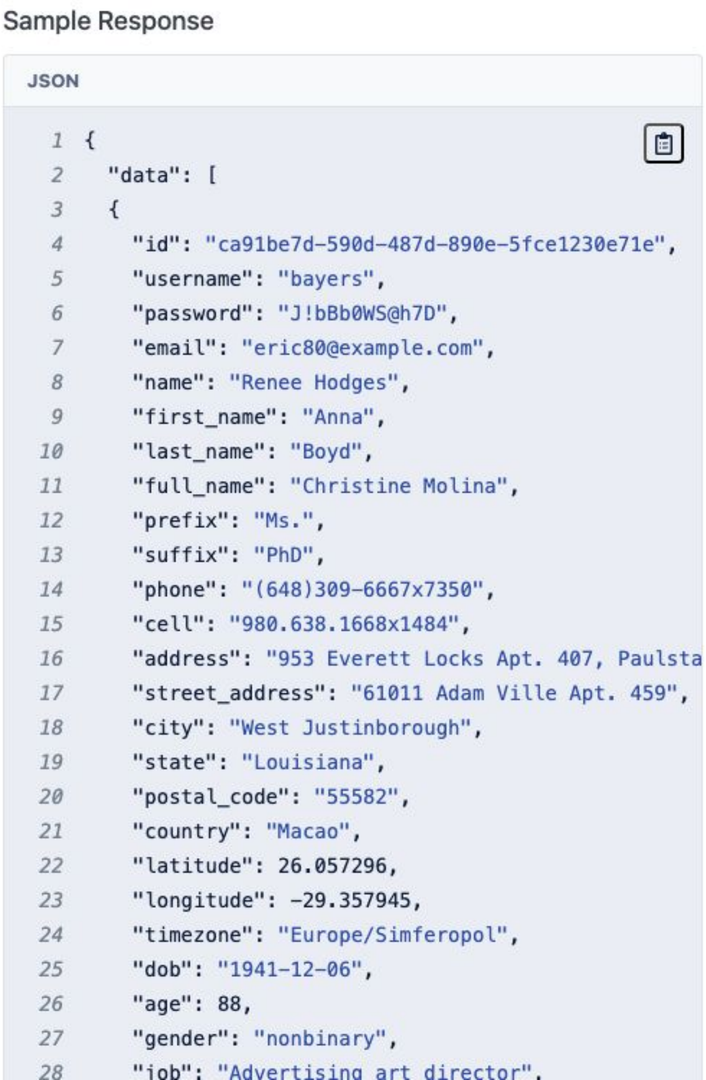
Parameters

count optional

Number of users to generate (1-30). Default: 10

gender optional

Filter by gender: `"male"`, `"female"`, `"nonbinary"`, or `"any"`. Default: `"any"`



Sample Response

```json
{
  "data": [
  {
    "id": "ca91be7d-590d-487d-890e-5fce1230e71e",
    "username": "bayers",
    "password": "J!bBb0WS@h7D",
    "email": "eric80@example.com",
    "name": "Renee Hodges",
    "first_name": "Anna",
    "last_name": "Boyd",
    "full_name": "Christine Molina",
    "prefix": "Ms.",
    "suffix": "PhD",
    "phone": "(648)309-6667x7350",
    "cell": "980.638.1668x1484",
    "address": "953 Everett Locks Apt. 407, Paulsta
    "street_address": "61011 Adam Ville Apt. 459",
    "city": "West Justinborough",
    "state": "Louisiana",
    "postal_code": "55582",
    "country": "Macao",
    "latitude": 26.057296,
    "longitude": -29.357945,
    "timezone": "Europe/Simferopol",
    "dob": "1941-12-06",
    "age": 88,
    "gender": "nonbinary",
    "job": "Advertising art director",
```

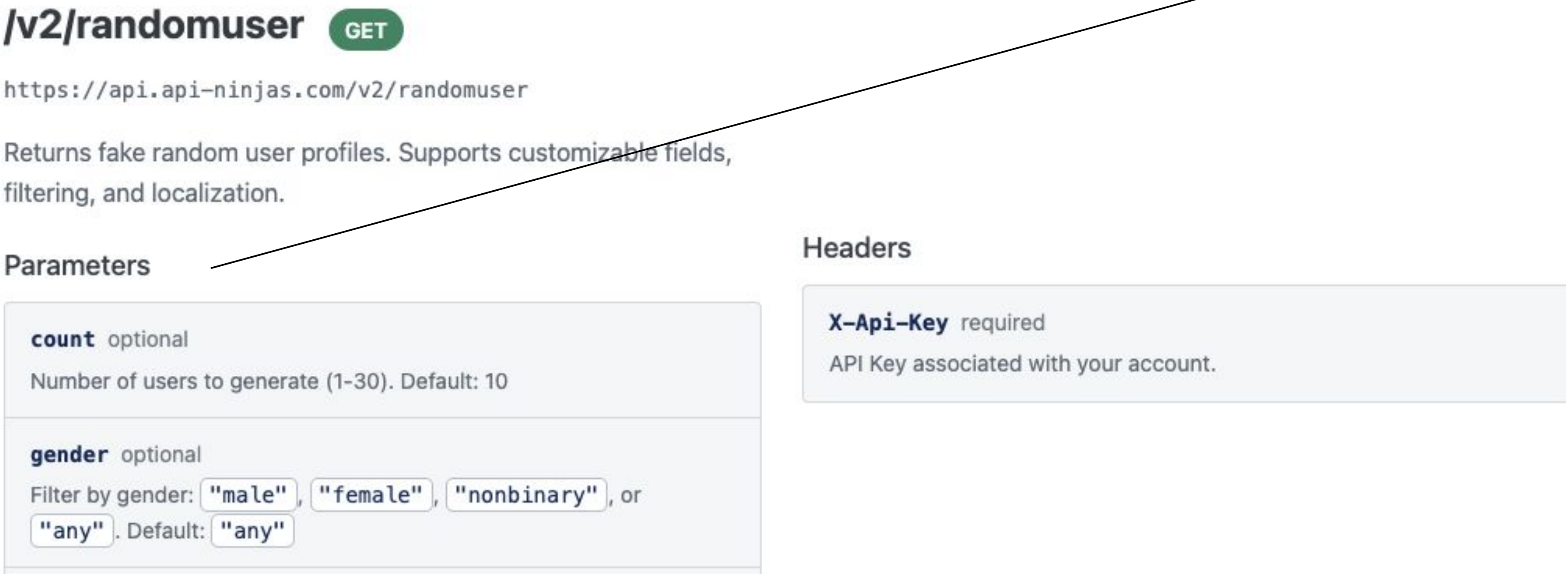
Random User API

https://api.api-ninjas.com/v2/randomuser?count=2

endpoint

Base URL

query



/v2/randomuser GET

`https://api.api-ninjas.com/v2/randomuser`

Returns fake random user profiles. Supports customizable fields, filtering, and localization.

Parameters

`count` optional

Number of users to generate (1-30). Default: 10

`gender` optional

Filter by gender: `"male"`, `"female"`, `"nonbinary"`, or `"any"`. Default: `"any"`

Headers

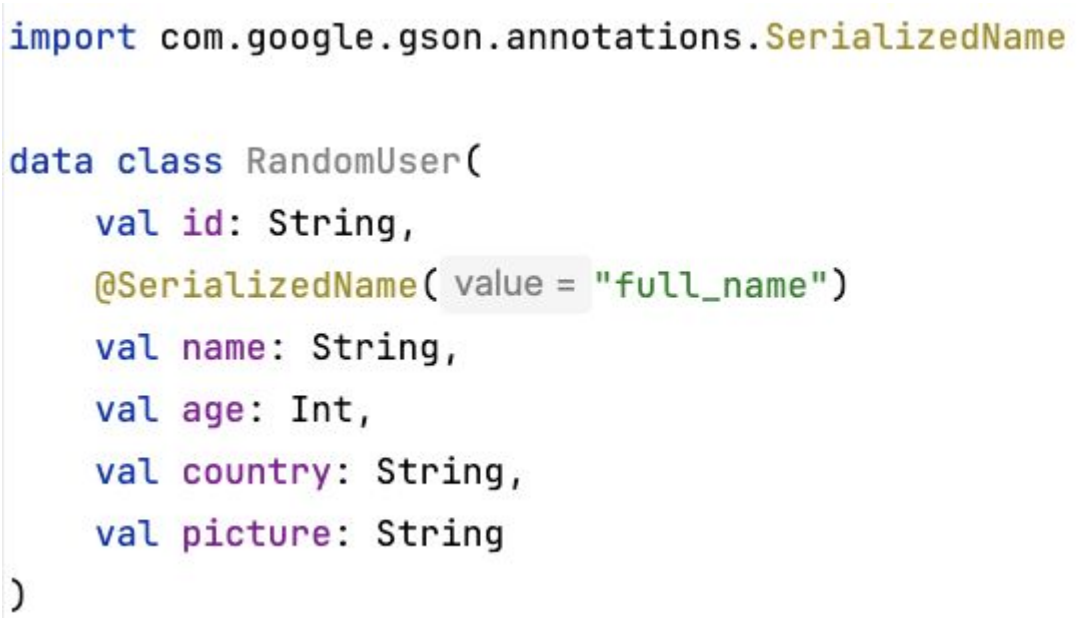
`X-Api-Key` required

API Key associated with your account.

Korak 2

- Kreiranja data class za jednog korisnika samo za sljedeća polja
    - id
    - username
    - email
    - full_name
    - country
    - picture

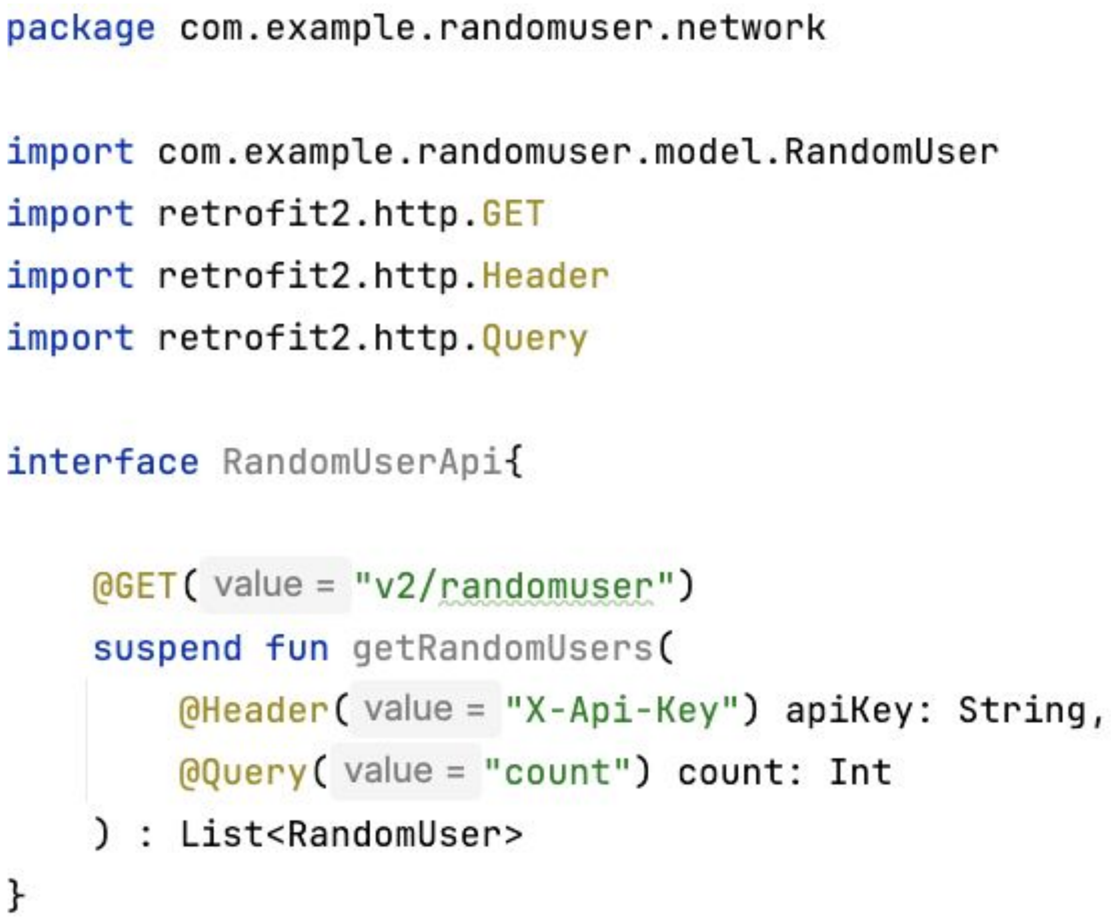
model/RandomUser.kt



```
import com.google.gson.annotations.SerializedName

data class RandomUser(
    val id: String,
    @SerializedName( value = "full_name")
    val name: String,
    val age: Int,
    val country: String,
    val picture: String
)
```

Korak 3

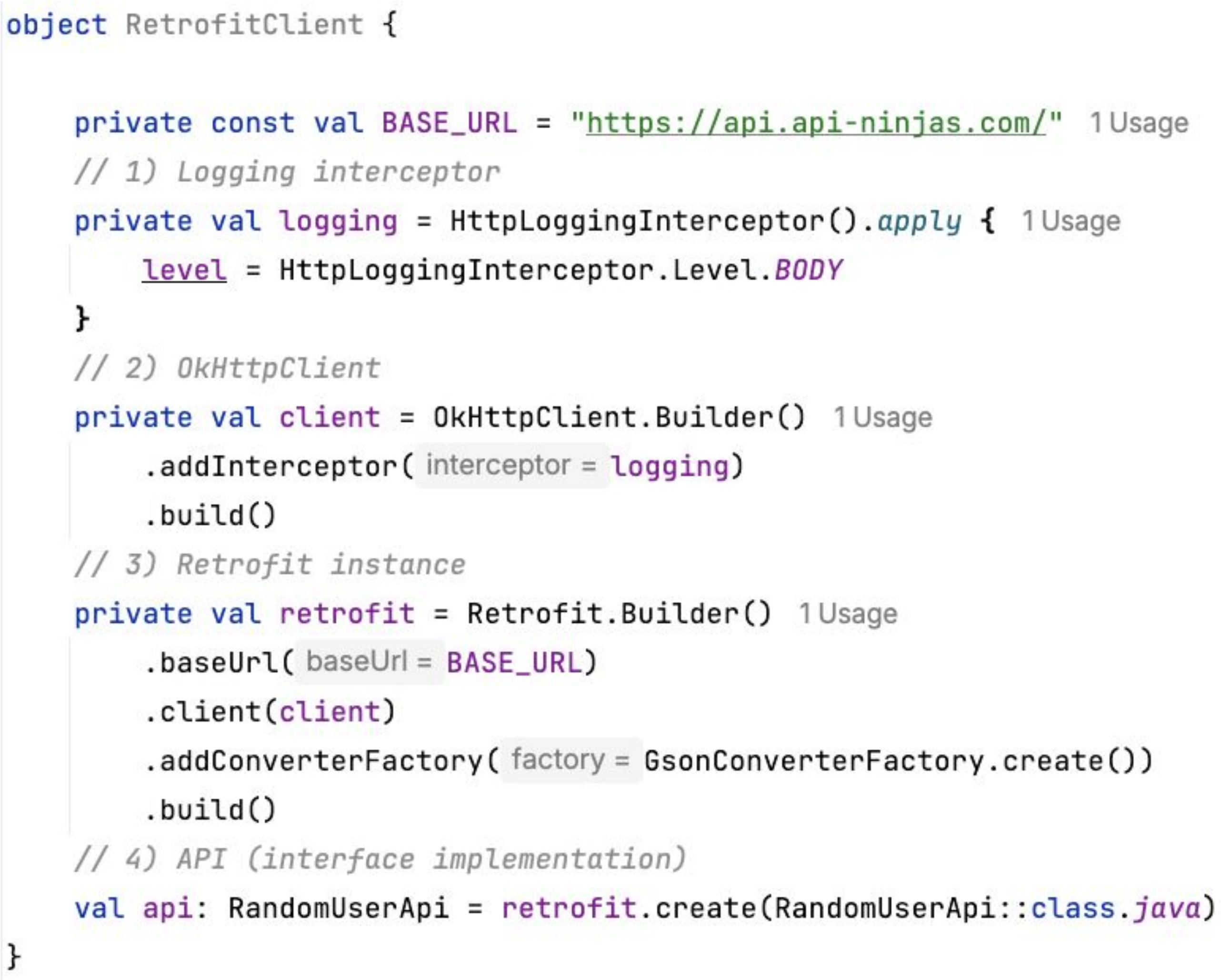
Unutar network paketa kreirati RandomUserApi.kt



```
package com.example.randomuser.network

import com.example.randomuser.model.RandomUser
import retrofit2.http.GET
import retrofit2.http.Header
import retrofit2.http.Query

interface RandomUserApi{

    @GET( value = "v2/randomuser")
    suspend fun getRandomUsers(
        @Header( value = "X-Api-Key") apiKey: String,
        @Query( value = "count") count: Int
    ) : List<RandomUser>
}
```

Korak 4

Unutar network paketa kreirati RetrofitClient



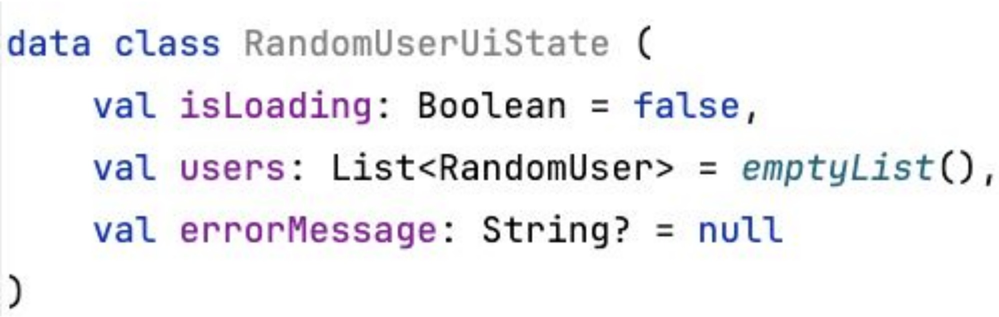
```
object RetrofitClient {

    private const val BASE_URL = "https://api.api-ninjas.com/"  1 Usage
    // 1) Logging interceptor
    private val logging = HttpLoggingInterceptor().apply {  1 Usage
        level = HttpLoggingInterceptor.Level.BODY
    }
    // 2) OkHttpClient
    private val client = OkHttpClient.Builder()  1 Usage
        .addInterceptor( interceptor = logging)
        .build()
    // 3) Retrofit instance
    private val retrofit = Retrofit.Builder()  1 Usage
        .baseUrl( baseUrl = BASE_URL)
        .client(client)
        .addConverterFactory( factory = GsonConverterFactory.create())
        .build()
    // 4) API (interface implementation)
    val api: RandomUserApi = retrofit.create(RandomUserApi::class.java)
}
```

Korak 5. Kreiranje ViewModel-a koji poziva API

Dodati u dependencies

implementation "androidx.lifecycle:lifecycle-viewmodel-ktx:2.8.6"

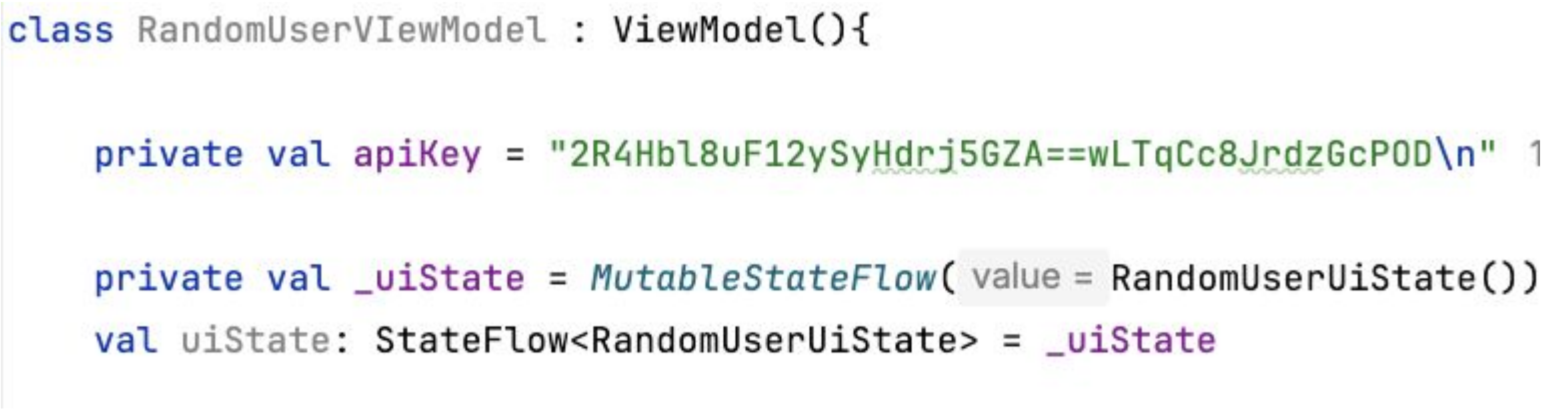
Unutar ui paketa, kreirati RandomUserViewModel.kt i dodati



```kotlin
data class RandomUserUiState (
    val isLoading: Boolean = false,
    val users: List<RandomUser> = emptyList(),
    val errorMessage: String? = null
)
```

Korak 5. Kreiranje ViewModel-a koji poziva API

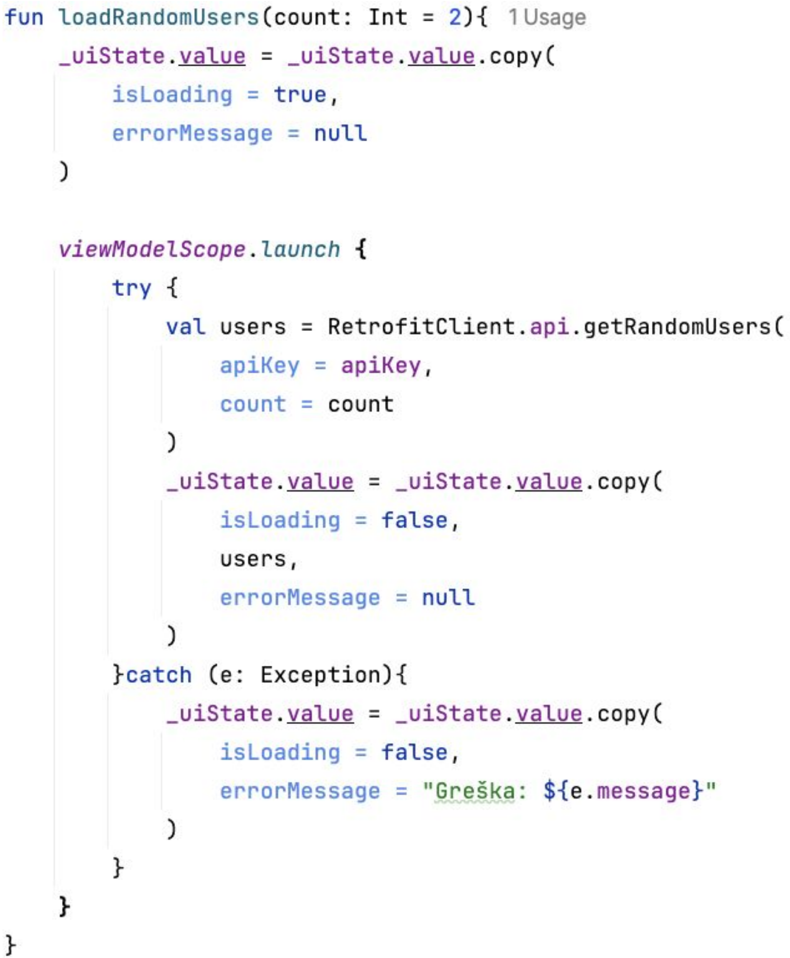
Unutar ui paketa, kreirati RandomUserViewModel.kt i dodati



```
class RandomUserVIewModel : ViewModel(){

    private val apiKey = "2R4Hbl8uF12ySyHdrj5GZA==wLTqCc8JrdzGcPOD\n"  1

    private val _uiState = MutableStateFlow( value = RandomUserUiState())
    val uiState: StateFlow<RandomUserUiState> = _uiState
```

Korak 5. Kreiranje ViewModel-a koji poziva API

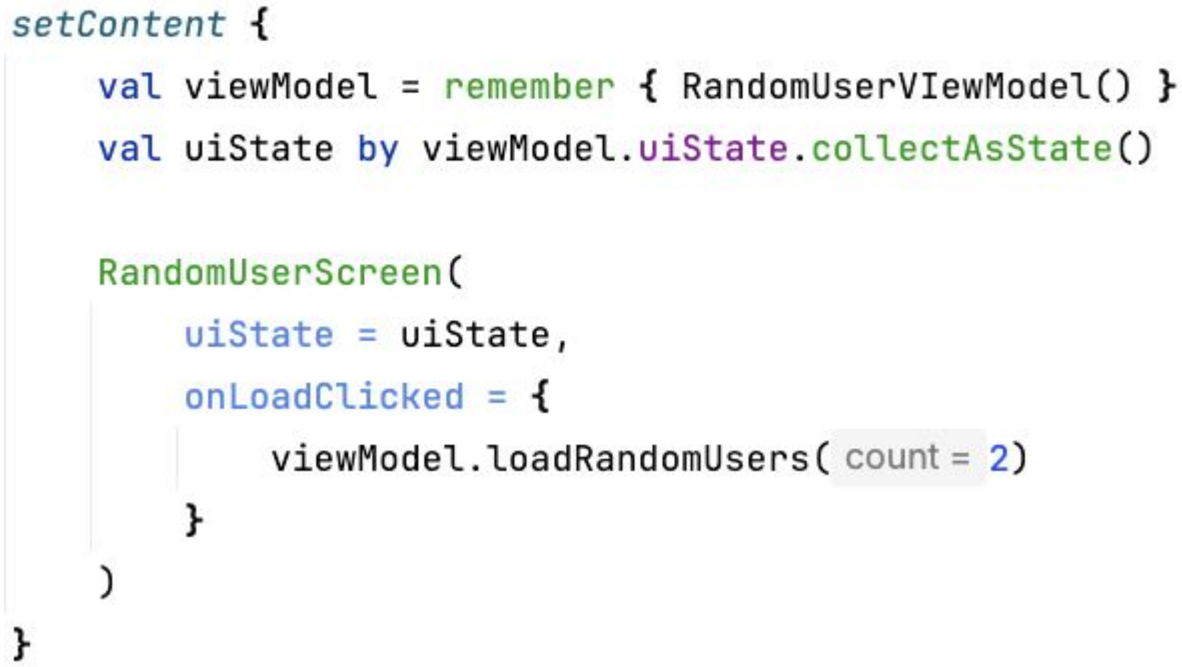
- **viewModelScope.launch** po defaultu radi na Main thread-u, ali ne blokira UI.
- **suspend** funkcije ne blokiraju thread, samo pauziraju coroutine dok ne stigne rezultat.
- **Retrofit + OkHttp** automatski izvršavaju network pozive na background thread-ovima (OkHttp Dispatcher).



```
fun loadRandomUsers(count: Int = 2){  1 Usage
    _uiState.value = _uiState.value.copy(
        isLoading = true,
        errorMessage = null
    )

    viewModelScope.launch {
        try {
            val users = RetrofitClient.api.getRandomUsers(
                apiKey = apiKey,
                count = count
            )
            _uiState.value = _uiState.value.copy(
                isLoading = false,
                users,
                errorMessage = null
            )
        }catch (e: Exception){
            _uiState.value = _uiState.value.copy(
                isLoading = false,
                errorMessage = "Greška: ${e.message}"
            )
        }
    }
}
```

Korak 6: Kreiranje UI-a

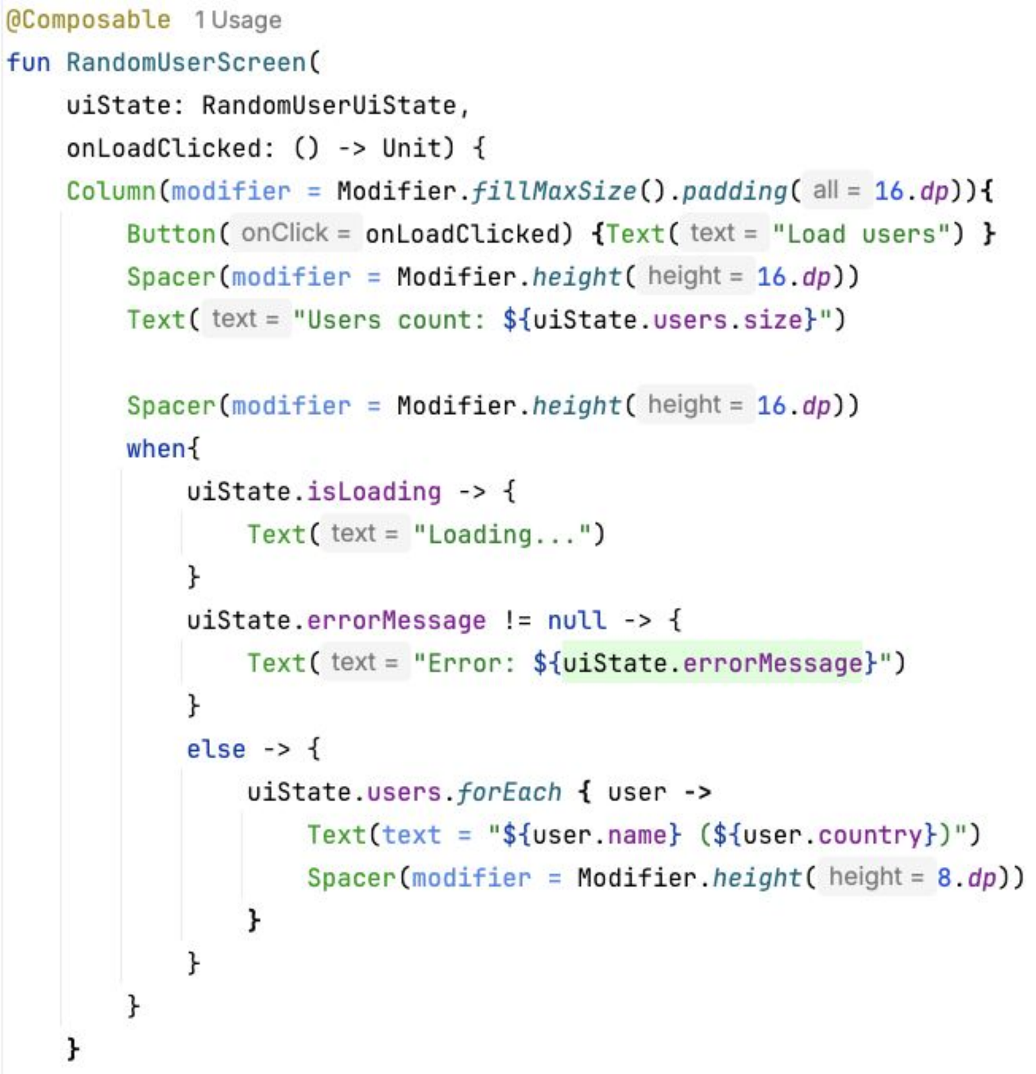
Unutar MainActivity.kt



```
setContent {
    val viewModel = remember { RandomUserVIewModel() }
    val uiState by viewModel.uiState.collectAsState()

    RandomUserScreen(
        uiState = uiState,
        onLoadClicked = {
            viewModel.loadRandomUsers( count = 2)
        }
    )
}
```

Korak 6: Kreiranje UI-a

Unutar MainActivity.kt



```
@Composable  1 Usage
fun RandomUserScreen(
    uiState: RandomUserUiState,
    onLoadClicked: () -> Unit) {
    Column(modifier = Modifier.fillMaxSize().padding( all = 16.dp)){
        Button( onClick = onLoadClicked) {Text( text = "Load users") }
        Spacer(modifier = Modifier.height( height = 16.dp))
        Text( text = "Users count: ${uiState.users.size}")

        Spacer(modifier = Modifier.height( height = 16.dp))
        when{
            uiState.isLoading -> {
                Text( text = "Loading...")
            }
            uiState.errorMessage != null -> {
                Text( text = "Error: ${uiState.errorMessage}")
            }
            else -> {
                uiState.users.forEach { user ->
                    Text(text = "${user.name} (${user.country})")
                    Spacer(modifier = Modifier.height( height = 8.dp))
                }
            }
        }
    }
```

Korak 7: Ubacivanje slika preko Coil

Dodati u dependencies

```
implementation "io.coil-kt:coil-compose:2.7.0"
```

Dodati u RandomUserScreen.kt

```
import androidx.compose.foundation.shape.CircleShape
import androidx.compose.ui.draw.clip

AsyncImage(
   model = user.picture,
   contentDescription = "Profile picture of ${user.name}",
   modifier = Modifier
      .size(120.dp)
      .clip(CircleShape)
)
```