Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие 22\_2**

**Сетевые взаимодействия - работа с API.**»

Выполнил: Молотова Виктория Анатольевна

Группа: ПР-31

Преподаватель: Мирошниченко Г. В.

2023

**Цель:**

1. Научиться работать API
2. Повторить работу с форматом данных JSON и базой ROOM на языке Kotlin.

**Задание (вариант 1 – Numbers API ( http://numbersapi.com )):**

Разработать мобильное приложение, информация в приложении должна быть запрошена через API и сохраняться в room.

Приложение должно содержать:

* приветственный экран с авторизацией пользователя, проверкой;
* ввода на пустоту и сохранением информации о пользователе через механизм SharedPreferences;
* содержать форму ввода данных;
* экран для вывода сохраненной информации, с использованием библиотеки ROOM, возможностью добавления, удаления и редактирования введенных данных, с сохранением в базе данных (ROOM или SQL);
* стилизация приложения в соответствии с вариантом;
* изменить название, тему и иконку приложения (должны соответствовать названию);
* предусмотреть портретную ориентацию/

**Входные данные:**

* username (String) – введенный пользователем логин;
* password (String) – введенный пользователем пароль;
* NumberText (String) – введенный пользователем цифра для факта.

**Выходные данные:**

* Список RecyclerView, в котором отображаются все загруженные факты;
* Сообщения пользователю об ошибках.

**Листинг программы:**

Dao

package com.example.pract20\_2\_molotova  
import androidx.room.\*  
import androidx.room.Dao  
import kotlinx.coroutines.flow.Flow  
  
@Dao  
interface Dao {  
 //запись  
 @Insert  
 fun insertItem(item: Item)  
  
 //получение всех  
 @Query("SELECT \* FROM items")  
 fun getAllItem(): Flow<List<Item>>  
  
 @Delete  
 suspend fun deleteItem(item: Item)  
  
 @Update  
 suspend fun updateItem(item: Item)  
}

InfoScreen

package com.example.pract20\_2\_molotova  
  
import android.content.Intent  
import android.os.Bundle  
import android.util.Log  
import android.view.View  
import android.widget.AdapterView  
import android.widget.EditText  
import android.widget.Spinner  
import android.widget.TextView  
import androidx.appcompat.app.AppCompatActivity  
import com.android.volley.toolbox.StringRequest  
import com.android.volley.toolbox.Volley  
import java.util.Calendar  
import androidx.appcompat.app.AlertDialog  
import java.text.SimpleDateFormat  
  
class InfoScreen : AppCompatActivity() {  
 var YearAndDate:String = "date"  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_info\_screen*)  
  
  
  
 }  
  
  
fun findDataAndYear(){  
 val shapeSpinner = findViewById<Spinner>(R.id.*vibor*)  
  
 shapeSpinner.*onItemSelectedListener* = object : AdapterView.OnItemSelectedListener {  
 override fun onItemSelected(parent: AdapterView<\*>?, view: View?, position: Int, id: Long) {  
 when (shapeSpinner.*selectedItem*.toString()) {  
 "Дата" -> {  
 YearAndDate = "date"  
 }  
 "Год" -> {  
 YearAndDate = "year"  
 }  
 }  
  
 }  
 override fun onNothingSelected(parent: AdapterView<\*>?) {  
  
 }  
 }  
  
}  
  
  
 private fun getResualt(number: String, search: String)  
 {  
 findDataAndYear()  
  
 val url = "http://numbersapi.com/$number/$search"  
  
 val queue = Volley.newRequestQueue(this)  
  
 val stringRequest = StringRequest(  
 com.android.volley.Request.Method.*GET*, url,  
 **{** response **->** val textInfoSet : TextView= findViewById(R.id.*TextInfoSet*)  
 //val txt = JSONObject(response)  
 textInfoSet.*text* = response.toString()  
  
 //запись в бд  
  
 //тип  
 val type: String = "year"  
  
 //число  
 val numberTextElement: EditText= findViewById(R.id.*EditNumber*)  
 val numberText = numberTextElement.*text*.toString()  
  
  
 //факт  
 val fact = response.toString()  
  
 //сама запись  
 val db = MainDb.getDb(this)  
 val item = Item(null, type, number, fact)  
 Thread**{** db.getDao().insertItem(item)  
 **}**.start()  
  
  
  
 Log.d("MyLog", "Volley: $response")  
 **}**,  
 **{** error **->** Log.d("MyLog", "Volley error: ${error.message}")  
 **}**)  
  
 queue.add(stringRequest)  
  
 }  
  
  
 fun cheakYear(yearStr: String): Boolean {  
 //текущий год  
 val calendar = Calendar.getInstance()  
 val year = calendar.get(Calendar.*YEAR*)  
 //проверка  
 return yearStr.*toIntOrNull*()?.*let* **{ it** in 1..year**}** ?: false  
 }  
 fun cheakDate(dateStr: String): Boolean {  
 val dateFormat = SimpleDateFormat("MM/dd")  
 dateFormat.*isLenient* = false // Отключаем режим допускающий нестрогую проверку  
  
 return try {  
 val date = dateFormat.parse(dateStr)  
 date != null  
 } catch (e: Exception) {  
 false  
 }  
 }  
  
  
 //нажатие кнопки  
  
 fun ButFindInfo(view: View) {  
  
 findDataAndYear()  
 val NumberText = findViewById<EditText>(R.id.*EditNumber*)  
  
 if((YearAndDate == "date"&& cheakDate(NumberText.*text*.toString())))  
 {  
 getResualt(NumberText.*text*.toString(), YearAndDate)  
 }  
 else{  
 if((YearAndDate == "year"&& cheakYear(NumberText.*text*.toString())))  
 {  
 getResualt(NumberText.*text*.toString(), YearAndDate)  
 }  
 else{  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Ошибка")  
 builder.setMessage("Неверно введенные данные")  
 builder.setPositiveButton("OK") **{** dialog, \_ **->** dialog.dismiss()  
 **}** val alertDialog = builder.create()  
 alertDialog.show()  
 }  
  
 }  
 val b: Boolean = cheakDate(NumberText.*text*.toString())  
 val t: String = NumberText.*text*.toString() + " " + b.toString() + " " + YearAndDate + "дата ппустая"  
 Log.d("MyLog", t)  
  
  
  
 }  
  
 fun Perehod(view: View) {  
 val intent = Intent(this, StatisticsScreen::class.*java*)  
 startActivity(intent)  
 }  
}

Item

package com.example.pract20\_2\_molotova  
  
import androidx.room.ColumnInfo  
import androidx.room.Entity  
import androidx.room.PrimaryKey  
  
@Entity (tableName = "items")  
  
//коллоны  
data class Item(  
  
 //ид  
 @PrimaryKey(autoGenerate = true)  
 var id: Int? = null,  
  
 //тип  
 @ColumnInfo(name = "type")  
 var type: String,  
  
 //число  
 @ColumnInfo(name = "number")  
 var number: String,  
  
 //факт  
 @ColumnInfo(name = "fact")  
 var fact: String  
 )

Item\_list\_row

package com.example.pract20\_2\_molotova  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
  
class item\_list\_row : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_item\_list\_row*)  
 }  
}

ItemAdapter

package com.example.pract20\_2\_molotova  
  
import android.content.Context  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.BaseAdapter  
import android.widget.Button  
import android.widget.ImageButton  
import android.widget.TextView  
  
class ItemAdapter(  
 context: Context,  
 private val itemList: List<Item>,  
 private val onDeleteClick: (Item) -> Unit,  
 private val onEditClick: (Item) -> Unit  
) : BaseAdapter() {  
 private val inflater: LayoutInflater = LayoutInflater.from(context)  
  
 override fun getCount(): Int {  
 return itemList.size  
 }  
  
 override fun getItem(position: Int): Any {  
 return itemList[position]  
 }  
  
 override fun getItemId(position: Int): Long {  
 return position.toLong()  
 }  
  
  
  
  
 override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View {  
 val view: View  
 val holder: ViewHolder  
  
 if (convertView == null) {  
 view = inflater.inflate(R.layout.*activity\_item\_list\_row*, parent, false)  
 holder = ViewHolder(  
 view.findViewById(R.id.*titleTextView*),  
 view.findViewById(R.id.*descriptionTextView*),  
 view.findViewById(R.id.*editButton*),  
 view.findViewById(R.id.*deleteButton*)  
 )  
 view.*tag* = holder  
 } else {  
 view = convertView  
 holder = view.*tag* as ViewHolder  
 }  
  
 val item = itemList[position]  
 holder.titleTextView.*text* = item.number // Отображаем "number" в заголовке  
 holder.descriptionTextView.*text* = item.fact  
  
 // Обработка нажатий на кнопку "Редактировать"  
 holder.editButton.setOnClickListener **{** onEditClick(item)  
 **}** // Обработка нажатий на кнопку "Удалить"  
 holder.deleteButton.setOnClickListener **{** onDeleteClick(item)  
 **}** return view  
 }  
  
 private class ViewHolder(  
 val titleTextView: TextView,  
 val descriptionTextView: TextView,  
 val editButton: ImageButton,  
 val deleteButton: ImageButton  
 )  
}

MainActivity

package com.example.pract20\_2\_molotova  
  
import android.app.AlertDialog  
import android.app.Dialog  
import android.content.Intent  
import android.content.SharedPreferences  
import android.os.Bundle  
  
  
import android.view.View  
import android.widget.EditText  
import android.widget.Toast  
import androidx.appcompat.app.AppCompatActivity  
import androidx.fragment.app.DialogFragment  
import androidx.fragment.app.FragmentTransaction  
  
  
class MainActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 val sharedPreferences = getSharedPreferences("MyPrefs", *MODE\_PRIVATE*)  
 val editor: SharedPreferences.Editor = sharedPreferences.edit()  
  
 editor.putString("username", "11111")  
 editor.putString("password", "11111")  
 editor.apply()  
 }  
  
 fun nextScreen(view: View) {  
 val usernameEditText = findViewById<EditText>(R.id.*login*)  
 val passwordEditText = findViewById<EditText>(R.id.*password*)  
  
 val log :String = usernameEditText.*text*.toString()  
 val pas:String = passwordEditText.*text*.toString()  
  
 if (usernameEditText.*text*.toString().*isEmpty*() || passwordEditText.*text*.toString().*isEmpty*()) {  
  
 Toast.makeText(this@MainActivity, "Введите логин и пароль", Toast.*LENGTH\_SHORT*).show()  
 }  
 else{  
  
 val sharedPreferences = getSharedPreferences("MyPrefs", *MODE\_PRIVATE*)  
  
 val savedUsername = sharedPreferences.getString("username", "")  
 val savedPassword = sharedPreferences.getString("password", "")  
  
  
 if (log == savedUsername && pas == savedPassword) {  
 val intent = Intent(this@MainActivity, InfoScreen::class.*java*)  
 startActivity(intent)  
 }  
 else{  
 class MyDialogFragment : DialogFragment() {  
  
 override fun onCreateDialog(savedInstanceState: Bundle?): Dialog {  
 return *activity*?.*let* **{** val builder = AlertDialog.Builder(**it**)  
 builder.setTitle("")  
 .setMessage("Пароль неверный")  
 .setCancelable(true)  
  
 .setNegativeButton("Ок") **{** dialog, id **->** dialog.cancel()  
 **}** builder.create()  
 **}** ?: throw IllegalStateException("Activity cannot be null")  
 }  
  
 }  
 val myDialogFragment = MyDialogFragment()  
 val manager = *supportFragmentManager* //myDialogFragment.show(manager, "dialog")  
 val transaction: FragmentTransaction = manager.beginTransaction()  
 myDialogFragment.show(transaction, "dialog")  
  
 }  
  
 }  
 }  
}

MainDb

package com.example.pract20\_2\_molotova  
  
import android.content.Context  
import androidx.room.Database  
import androidx.room.Room  
import androidx.room.RoomDatabase  
  
@Database(entities = [Item::class], version = 1)  
abstract class MainDb : RoomDatabase() {  
  
 abstract fun getDao(): Dao  
  
 companion object{  
 fun getDb(context: Context): MainDb{  
 return Room.databaseBuilder(  
 context.*applicationContext*, MainDb::class.*java*, "test.db"  
 ).build()  
 }  
 }  
}

StatistickScreen

package com.example.pract20\_2\_molotova  
  
import android.app.AlertDialog  
import android.content.DialogInterface  
import android.os.Bundle  
import android.view.View  
import android.widget.\*  
import androidx.appcompat.app.AppCompatActivity  
import androidx.lifecycle.asLiveData  
import kotlinx.coroutines.\*  
  
  
class StatisticsScreen : AppCompatActivity() {  
 var users = ArrayList<String>()  
  
  
 private val dataList = *mutableListOf*<Item>()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_statistics\_screen*)  
 printInfo()  
  
 }  
  
  
 private fun printInfo() {  
 val textList = findViewById<ListView>(R.id.*listView*)  
 dataList.clear()  
  
 val db = MainDb.getDb(this)  
  
 db.getDao().getAllItem().*asLiveData*().observe(this) **{** list **->** val updatedDataList = *mutableListOf*<Item>()  
 list.*forEach* **{** item **->** val text = "id ${item.id}, type ${item.type} number ${item.number} fact ${item.fact} \n"  
 // Создайте новый элемент Item и добавьте его в dataList  
 val newItem = Item(id = item.id, type = item.type, number = item.number, fact = item.fact)  
 updatedDataList.add(newItem)  
 **}** // После завершения цикла, обновите dataList  
 dataList.clear()  
 dataList.addAll(updatedDataList)  
  
 // Создайте или обновите адаптер  
 val itemAdapter = ItemAdapter(this, dataList,  
 onDeleteClick = **{** itemToDelete **->** // Здесь выполняйте удаление элемента из базы данных и обновление списка  
 val db = MainDb.getDb(this)  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** try {  
 db.getDao().deleteItem(itemToDelete)  
 // После удаления, обновите список данных  
 printInfo()  
 } catch (e: Exception) {  
 // Обработка ошибки удаления, если произошла  
 }  
 **}** printInfo()  
 printInfo()  
 **}**,  
 onEditClick = **{** position **->** // Получаю рание данные  
 val db = MainDb.getDb(this)  
 var bdNewText: Array<String> = *arrayOf*("id", "type", "number", "fact")  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** try {  
 bdNewText= *arrayOf*(position.id.*toString*(), position.type, position.number, position.fact)  
 } catch (e: Exception) {  
 // Обработка ошибки удаления, если произошла  
 }  
 **}** //изменения через AlertDiolog  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Измените данные")  
  
 // Создайте контейнер LinearLayout для размещения двух EditText  
 val layout = LinearLayout(this)  
 layout.*orientation* = LinearLayout.*VERTICAL* // Создайте два EditText для ввода данных  
 val EditNumber = EditText(this)  
 val EditFakt = EditText(this)  
  
 // Установите хинты для EditText  
 EditNumber.*hint* = "Введите дату или год"  
 EditFakt.*hint* = "Введите факт"  
 EditNumber.setText(bdNewText[2])  
 EditFakt.setText(bdNewText[3])  
 // Добавьте EditText к контейнеру  
 layout.addView(EditNumber)  
 layout.addView(EditFakt)  
  
 builder.setView(layout)  
  
 // Установите кнопку "OK" для сохранения данных  
 builder.setPositiveButton("OK") **{** dialog, \_ **->** var number = EditNumber.*text*.toString()  
 var fakt = EditFakt.*text*.toString()  
  
  
 //запись в бд  
 //проверка  
 if(number.*isEmpty*()){  
 number = bdNewText[2];  
 }  
 if(fakt.*isEmpty*()){  
 fakt = bdNewText[3];  
 }  
 //сама запись  
 val item = Item(bdNewText[0].*toInt*(), "null", number, fakt)  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** // Вызовите suspend-функцию updateItem в корутине  
 db.getDao().updateItem(item)  
 **}** printInfo()  
  
 **}** // Установите кнопку "Отмена" для закрытия диалога  
 builder.setNegativeButton("Отмена") **{** dialog, \_ **->** dialog.cancel()  
 **}** val dialog = builder.create()  
 dialog.show()  
 printInfo()  
  
 **}**)  
 textList.*adapter* = itemAdapter  
  
 // Уведомите адаптер о изменениях  
 itemAdapter.notifyDataSetChanged()  
  
 **}** }  
  
 fun ButCleaAll(view: View) {  
 val db = MainDb.getDb(this)  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** // Очистите все таблицы в базе данных  
 db.clearAllTables()  
  
 // После удаления данных, выполните операции обновления UI на главном потоке (если необходимо)  
 withContext(Dispatchers.Main) **{** // Обновление UI, если необходимо  
 **}  
 }** printInfo()  
 }  
 fun AddNewButton(view: View) {  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Введите данные")  
  
 // Создайте контейнер LinearLayout для размещения двух EditText  
 val layout = LinearLayout(this)  
 layout.*orientation* = LinearLayout.*VERTICAL* // Создайте два EditText для ввода данных  
 val EditNumber = EditText(this)  
 val EditFakt = EditText(this)  
  
 // Установите хинты для EditText  
 EditNumber.*hint* = "Введите дату или год"  
 EditFakt.*hint* = "Введите факт"  
  
 // Добавьте EditText к контейнеру  
 layout.addView(EditNumber)  
 layout.addView(EditFakt)  
  
 builder.setView(layout)  
  
 // Установите кнопку "OK" для сохранения данных  
 builder.setPositiveButton("OK") **{** dialog, \_ **->** val number = EditNumber.*text*.toString()  
 val fakt = EditFakt.*text*.toString()  
  
 // Здесь можно обработать введенные данные (value1 и value2)  
 //запись в бд  
 if(number.length == 0 || fakt.length == 0){  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Не сохранено")  
 .setMessage("Необходимо заполнить все поля")  
 .setPositiveButton("ОК") **{** dialog, id **->** dialog.cancel()  
 **}** builder.create()  
 }  
 else{  
 //сама запись  
 val db = MainDb.getDb(this)  
 val item = Item(null, "null", number, fakt)  
 Thread**{** db.getDao().insertItem(item)  
 **}**.start()  
 dialog.dismiss()  
 printInfo()  
 }  
  
 **}** // Установите кнопку "Отмена" для закрытия диалога  
 builder.setNegativeButton("Отмена") **{** dialog, \_ **->** dialog.cancel()  
 **}** val dialog = builder.create()  
 dialog.show()  
  
 }  
  
  
  
}

activity\_info\_screen

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".InfoScreen"  
 android:background="@color/background">  
  
 <TextView  
 android:gravity="center"  
 android:textColor="@color/white"  
 android:id="@+id/TextInfoSet"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignBottom="@+id/EditNumber"  
 android:layout\_alignParentEnd="true"  
 android:text="@string/info"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/EditNumber"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.345" />  
  
  
 <Spinner  
 android:id="@+id/vibor"  
 style="@style/Spinner"  
 android:layout\_width="300dp"  
  
 android:layout\_height="50dp"  
 android:layout\_marginTop="80dp"  
 android:entries="@array/name"  
 android:textColor="@color/black"  
 app:layout\_constraintBottom\_toTopOf="@+id/EditNumber"  
  
 app:layout\_constraintEnd\_toEndOf="parent"  
  
 app:layout\_constraintHorizontal\_bias="0.495"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/TextInfoSet"  
 app:layout\_constraintVertical\_bias="0.153"  
 tools:ignore="MissingConstraints" />  
  
 <EditText  
 android:id="@+id/EditNumber"  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginTop="384dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.495"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/findInfo"  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/button2"  
 android:textColor="@color/white"  
 android:layout\_below="@+id/buttonScreen"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginTop="56dp"  
 android:onClick="ButFindInfo"  
 android:text="@string/infoSearch"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/EditNumber" />  
  
 <Button  
 android:textColor="@color/white"  
 android:background="@drawable/button2"  
 android:id="@+id/buttonScreen"  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/TextInfoSet"  
 android:layout\_alignParentEnd="true"  
 android:onClick="Perehod"  
 android:text="@string/checkStatick"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/findInfo"  
 app:layout\_constraintVertical\_bias="0.166" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_item\_list\_row

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 tools:context=".item\_list\_row"  
 android:orientation="horizontal"  
 android:background="@color/background"  
 >  
 <LinearLayout  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <TextView  
 android:textColor="@color/white"  
 android:id="@+id/titleTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:paddingStart="16dp"  
 android:paddingTop="8dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="4dp" />  
  
 <TextView  
 android:textColor="@color/white"  
 android:id="@+id/descriptionTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp" />  
 </LinearLayout>  
  
 <ImageButton  
 android:scaleType="centerCrop"  
 android:id="@+id/editButton"  
 android:background="@color/background"  
 android:src="@drawable/editbut"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp">  
  
 </ImageButton>  
 <ImageButton  
 android:scaleType="centerCrop"  
 android:id="@+id/deleteButton"  
 android:background="@color/background"  
 android:src="@drawable/mus"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp">  
  
 </ImageButton>  
  
  
</LinearLayout>

activity\_main

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"  
 android:background="@drawable/back3"  
 >  
  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/registertext"  
 android:textColor="@color/white"  
 android:textSize="30sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.156" />  
  
 <EditText  
 android:id="@+id/login"  
 android:layout\_width="300dp"  
 android:layout\_height="50dp"  
 android:layout\_marginBottom="444dp"  
 android:autofillHints=""  
 android:background="@color/white"  
 android:hint="@string/login"  
 android:inputType="textEmailAddress"  
 android:textColor="@color/black"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.495"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintVertical\_bias="0.132"  
 tools:ignore="MissingConstraints" />  
  
 <EditText  
 android:id="@+id/password"  
 android:layout\_width="300dp"  
 android:layout\_height="50dp"  
 android:layout\_marginBottom="216dp"  
 android:autofillHints=""  
 android:background="@color/white"  
 android:hint="@string/password"  
 android:inputType="textPassword"  
 android:textColor="@color/black"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.495"  
  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/login"  
 app:layout\_constraintVertical\_bias="0.191"  
 tools:ignore="MissingConstraints" />  
  
 <Button  
 android:textColor="@color/white"  
 android:id="@+id/nextMain"  
 android:layout\_width="330dp"  
 android:layout\_height="60dp"  
 android:layout\_marginTop="80dp"  
 android:background="@drawable/rounded\_button"  
 android:onClick="nextScreen"  
 android:text="@string/register"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.493"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/password"  
 tools:ignore="MissingConstraints" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_statistics\_screen

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".StatisticsScreen"  
 android:orientation="vertical"  
 android:background="@color/background">  
  
 <ListView  
  
 android:id="@+id/listView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <Button  
 android:background="@drawable/button2"  
 android:id="@+id/clearButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="очистить"  
 android:textColor="@color/white"  
 android:layout\_margin="4dp"  
 android:onClick="ButCleaAll"/>  
  
  
 <Button  
 android:textColor="@color/white"  
 android:background="@drawable/button2"  
 android:id="@+id/AddButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="добавить"  
 android:layout\_margin="4dp"  
 android:onClick="AddNewButton"/>  
 </LinearLayout>  
  
</LinearLayout>

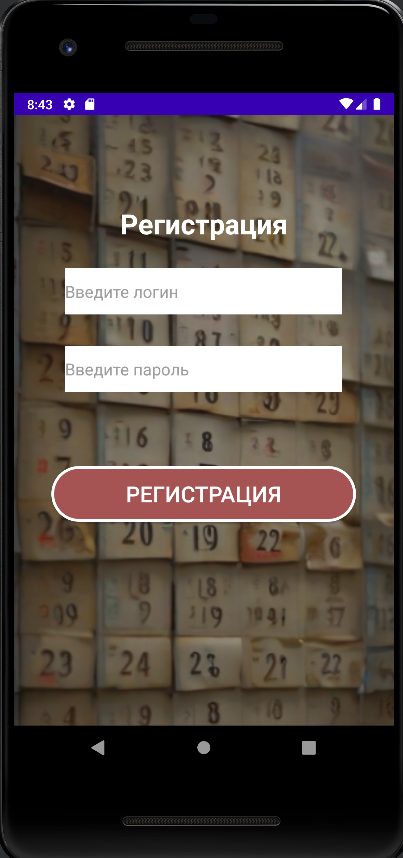
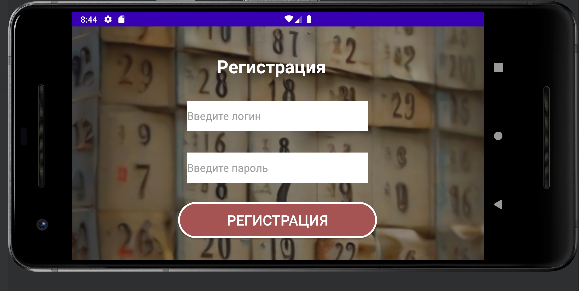
colors.xml

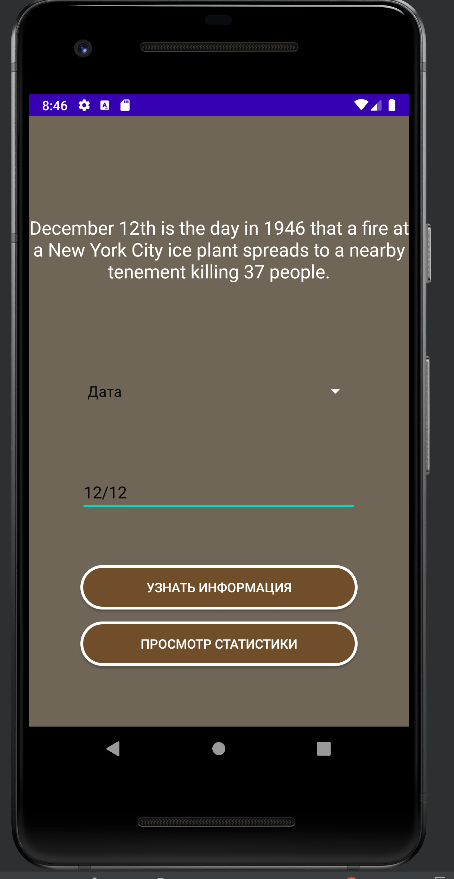
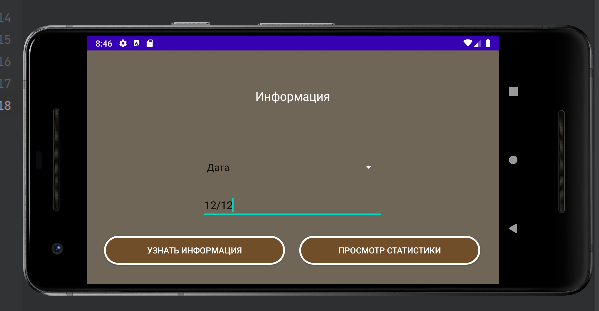
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="purple\_200">#FFBB86FC</color>  
 <color name="purple\_500">#FF6200EE</color>  
 <color name="purple\_700">#FF3700B3</color>  
 <color name="teal\_200">#FF03DAC5</color>  
 <color name="teal\_700">#FF018786</color>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
 <color name="background">#6F6658</color>  
 <color name="colorButon">#A65353</color>  
 <color name="colorButon2">#714E2A</color>  
  
</resources>

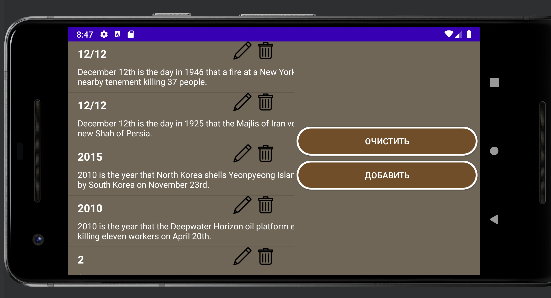
strings.xml

<resources>  
 <string name="app\_name">Случайные факты</string>  
 <string name="infoSearch">Узнать информация</string>  
 <string name="info">Информация</string>  
 <string name="checkStatick">Просмотр статистики</string>  
 <string name="login"> Введите логин</string>  
 <string name="password"> Введите пароль</string>  
 <string name="register">Регистрация</string>  
 <string name="registertext">Регистрация</string>  
 <string name="vodData">Ввести дату</string>  
 <string name="vodYear">Ввести год</string>  
  
 <string-array name="name">  
 <item>Дата</item>  
 <item>Год</item>  
  
 </string-array>  
</resources>

**Экраны (взаимодействие с приложением):**

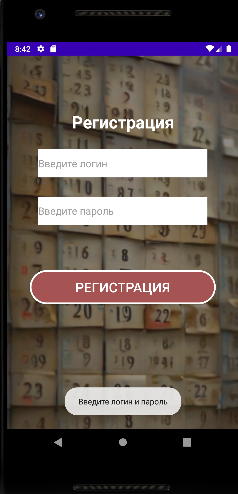
 

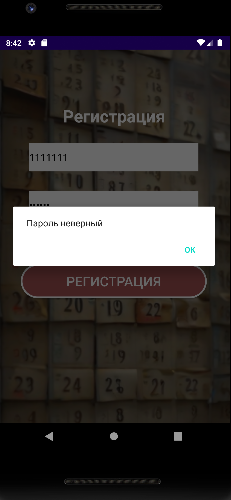


**Тестовые ситуации:**

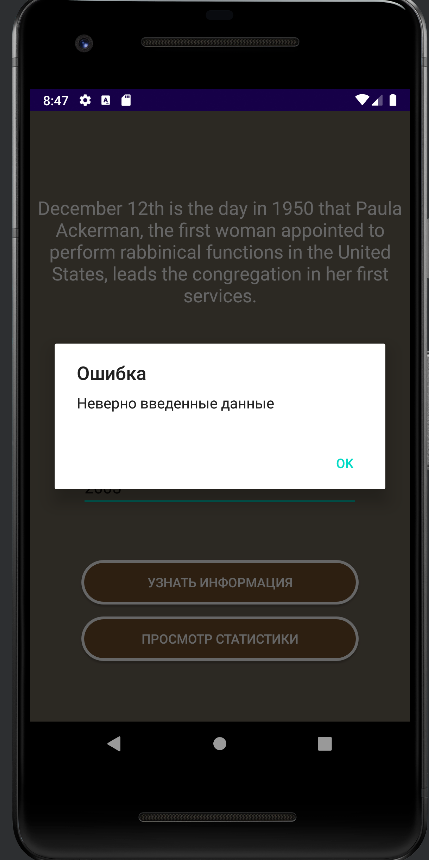
Если пользователь не будет водить пароль и/или логин, то программа сообщит об ошибке



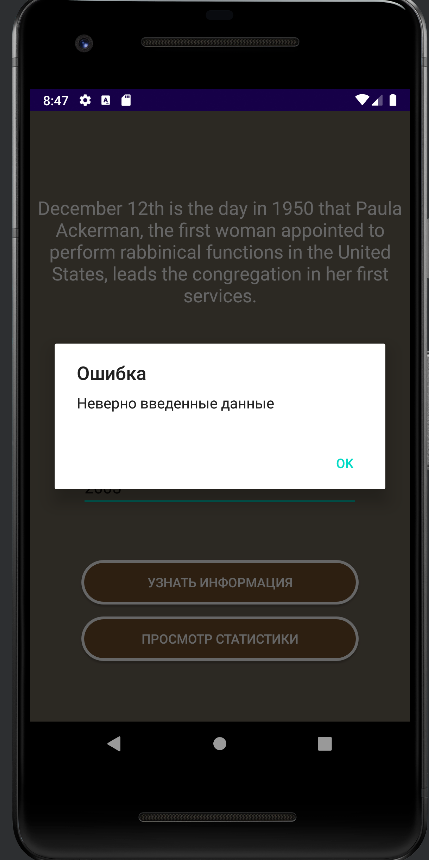
Если пользователь введет неверный пароль и/или логин то программа сообщит об ошибке



Если пользователь выберет в списке год/дату, но введет дату/год то программа сообщит об ошибке



Если пользователь введет несуществующую дату или год, программа так же сообщит об ошибке



**Вывод:** Благодаря этой практической я научилась работать с API, а так же укрепила свои знания по room и sharedpreferens