МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра автоматизированных систем управления

Расчётно-графическое задание

по дисциплине: «Современные технологии разработки программного обеспечения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Выполнил:** |  | **Проверил:** |  | |
| Студент | *Чибисов И.А.* | Преподаватель | *Эстрайх И.В.*  *ст. преподаватель* | |
| Факультет | *АВТФ* |  |  | |
| Направление (специальность) подготовки | *09.04.03 – Прикладная информатика* | Балл: \_\_\_\_\_\_\_\_\_ | |  |
| Группа | *АПМ-22* | Оценка *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | | |
| Шифр | *010594214* |  |  | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| подпись | | подпись | | |
| Дата сдачи: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г. | Дата защиты: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г | |

Новосибирск

2023

Цель работы: создать приложение конвертер валют с использованием api, что бы получать всегда корректный курс валют.

Для создания конвертера мной была выбрана Java.

Создадим пустой проект.

Структура получилась следящая:

Главный класс MainActivity который отвечает за отображение и вызывает методы для получения данных о конвертации валют от класса CurrencyAPI.

CurrencyAPI - класс, который отвечает за получение курсов валют с помощью API, с открытого ресурса openexchangerates.org.

Currency - класс, который представляет отдельную валюту, включает в себя ее название, код.

Из важных моментов, для получения доступа в интернет нам нужна добавить разрешение в AndroidManifest.xml

Далее т.к данные из api мы будем получать в json формате в мы build.gradle добавили пару расширений это retrofit2 и gson.

Создавали макет для нашего приложения, о выглядит следущим образом, рисунок 1.

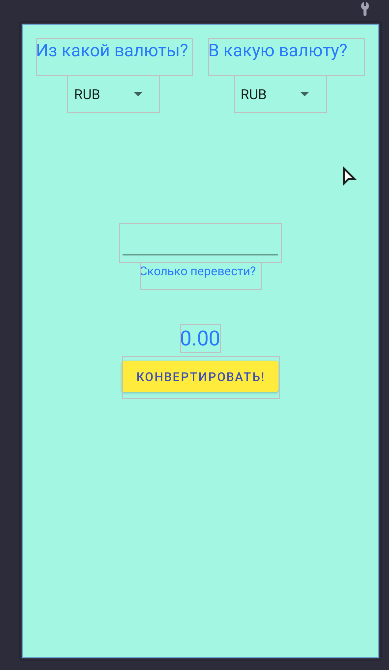


Рисунок 1 – Макет

Написали класс который будет отправлять запрос и получать ответ из api CurrencyAPI.

public class CurrencyAPI {  
  
 public static final String *API\_KEY* = "877e0a7d2ac74399b95d3ba6957f70fc"**;** private static Retrofit *retrofit* = null**;** public interface CurrencyService {  
 @GET("latest.json")  
 Call<Currency> getExchangeRates(@Query("app\_id") String app\_id)**;** }  
  
 public static Retrofit getClient() {  
  
 if (*retrofit* == null) {  
 *retrofit* = new Retrofit.Builder()  
 .baseUrl("https://openexchangerates.org/api/")  
 .addConverterFactory(GsonConverterFactory.*create*())  
 .build()**;** }  
 return *retrofit***;** }  
}

Так же написали реализацию класса Currency который берет переданные пользователем валюты, а позже забирает из api ответа нужную валюту.

public class Currency {  
  
 @SerializedName("rates")  
 @Expose  
 private Map<String**,** Double> rates**;** public Map<String**,** Double> getRates() {  
 return rates**;** }  
}

Прейдём к классу MainActivity который является основам в проекте, первым делом в нем мы собираем всё что передал пользователь.

Были добавлены проверки на то задал ли пользователь null и не выбрал ли одинаковые валюты.

if (fromCurrency.equals(toCurrency)) {  
 Toast.*makeText*(MainActivity.this**,** "Заданы одинаковые валюты"**,** Toast.*LENGTH\_SHORT*).show()**;** return**;**}  
  
if (TextUtils.*isEmpty*(etAmount.getText().toString())) {  
 Toast.*makeText*(MainActivity.this**,** "Задайте значение > 0"**,** Toast.*LENGTH\_SHORT*).show()**;** return**;**}

Дальше мы получаем ответ от api в следующем формате, тело запроса и все курсы валют, и делаем их небольшую обработку, рисунки 2-3.

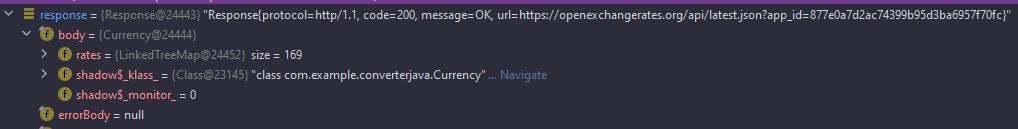


Рисунок 2 – тело ответа

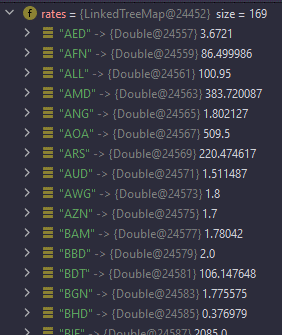


Рисунок 3 – тело ответа

Также добавил проверку нормальный ли ответ дала api и дала ли вообще.

} else {  
 Toast.*makeText*(MainActivity.this**,** "Ошибка интеграции: " + response.message()**,** Toast.*LENGTH\_SHORT*).show()**;** }  
}  
@Override  
public void onFailure(Call<Currency> call**,** Throwable t) {  
 Toast.*makeText*(MainActivity.this**,** "Ошибка со связью с API " + t.getMessage()**,** Toast.*LENGTH\_SHORT*).show()**;**}

В итоге класс выглядит так:

public class MainActivity extends AppCompatActivity {  
  
 private EditText etAmount**;** private Spinner spFromCurrency**;** private Spinner spToCurrency**;** private TextView tvResult**;** private CurrencyAPI.CurrencyService currencyService**;** @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState)**;** setContentView(R.layout.*activity\_main*)**;** etAmount = findViewById(R.id.*et\_amount*)**;** spFromCurrency = findViewById(R.id.*sp\_from\_currency*)**;** spToCurrency = findViewById(R.id.*sp\_to\_currency*)**;** tvResult = findViewById(R.id.*tv\_result*)**;** Button btnConvert = findViewById(R.id.*btn\_convert*)**;** currencyService = CurrencyAPI.*getClient*().create(CurrencyAPI.CurrencyService.class)**;** btnConvert.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 convertCurrency()**;** }  
 })**;** }  
  
 private void convertCurrency() {  
 String apiKey = CurrencyAPI.*API\_KEY***;** String fromCurrency = spFromCurrency.getSelectedItem().toString()**;** String toCurrency = spToCurrency.getSelectedItem().toString()**;** if (fromCurrency.equals(toCurrency)) {  
 Toast.*makeText*(MainActivity.this**,** "Заданы одинаковые валюты"**,** Toast.*LENGTH\_SHORT*).show()**;** return**;** }  
  
 if (TextUtils.*isEmpty*(etAmount.getText().toString())) {  
 Toast.*makeText*(MainActivity.this**,** "Задайте значение > 0"**,** Toast.*LENGTH\_SHORT*).show()**;** return**;** }  
 double amount = Double.*parseDouble*(etAmount.getText().toString())**;** Call<Currency> call = currencyService.getExchangeRates(apiKey)**;** call.enqueue(new Callback<Currency>() {  
 @Override  
 public void onResponse(Call<Currency> call**,** Response<Currency> response) {  
 if (response.isSuccessful()) {  
 Currency currency = response.body()**;** double fromRate = currency.getRates().get(fromCurrency)**;** double toRate = currency.getRates().get(toCurrency)**;** double result = (amount / fromRate) \* toRate**;** tvResult.setText(String.*format*("%.2f"**,** result))**;** } else {  
 Toast.*makeText*(MainActivity.this**,** "Ошибка интеграции: " + response.message()**,** Toast.*LENGTH\_SHORT*).show()**;** }  
 }  
 @Override  
 public void onFailure(Call<Currency> call**,** Throwable t) {  
 Toast.*makeText*(MainActivity.this**,** "Ошибка со связью с API " + t.getMessage()**,** Toast.*LENGTH\_SHORT*).show()**;** }  
 })**;** }  
}

Ошибки представлены на рисунке 4, в том числе если интернет выключен.

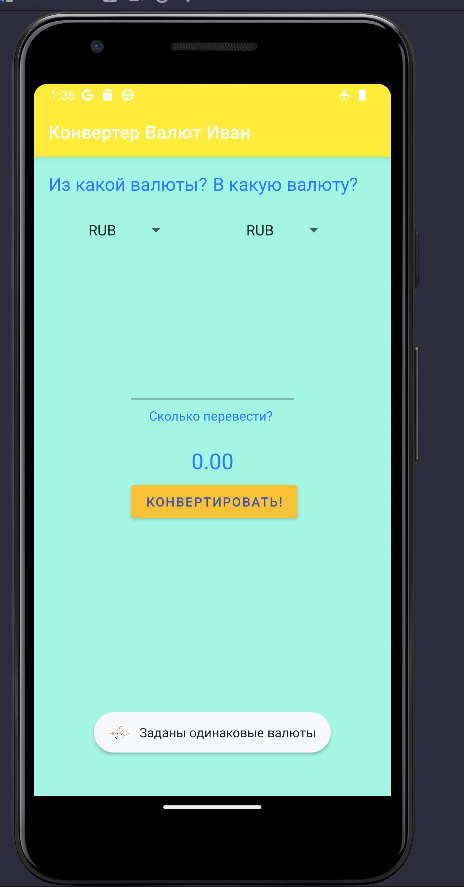
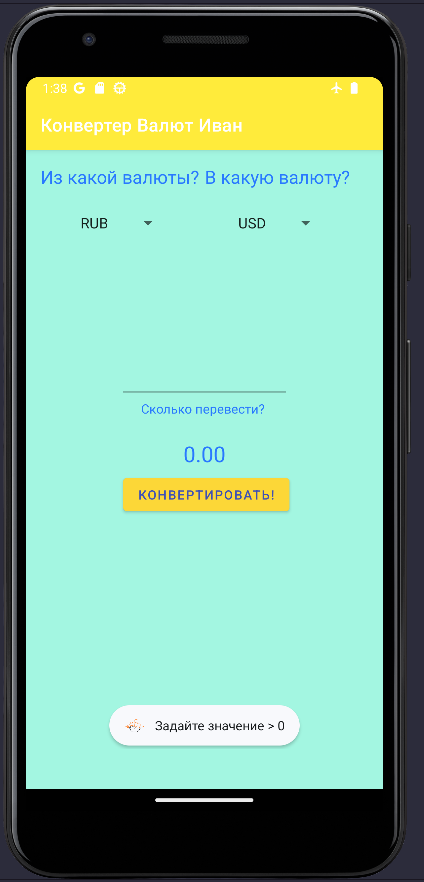
  

Рисунок 4 – Ошибки

Ниже на рисунке 5 представлена работа приложения.

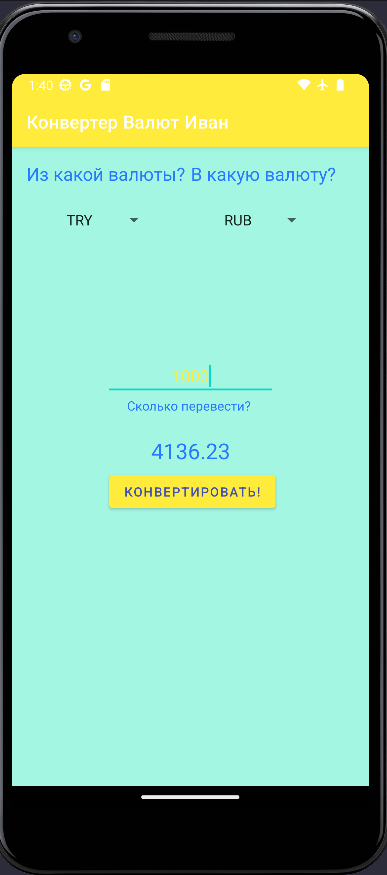


Рисунок 5 – конвертация

На рисунке 6 представлена работа на реальном устройстве, создал apk и установил на устройстве android.



Рисунок 6 – конвертация на реальном устройстве

Вывод: мы создали приложение конвертер валют с использованием api, чтобы получать всегда корректный курс валют.