

JAVA TILIDA ANDROID PLATFORMASI UCHUN REST API BILAN ISHLASH: RETROFIT VA GSON KUTUBXONALARIDAN FOYDALANISH METODOLOGIYASI

Xasanova MahliyoXon Abduhalimovna

*Andijon davlat universiteti Kompyuter injiniringi yo'nalishi 4-bosqich
talabasi.*

hasanovam634@gmail.com

Annotatsiya: Ushbu maqolada Java dasturlash tilida Android mobil ilovalarini RESTful veb-xizmatlar (REST API) bilan bog'lash masalasi Retrofit va Gson kutubxonalari misolida ko'rib chiqiladi. Zamonaviy mobil ilovalarning aksariyati serverdan ma'lumot olish yoki serverga ma'lumot yuborish uchun tarmoq orqali ishlaydi, shu sababli tarmoq qatlamini to'g'ri va samarali loyihalash dasturchi tayyorlash jarayonida muhim o'rin tutadi. Tadqiqotning maqsadi HttpURLConnection va Volley kabi an'anaviy yondashuvlar bilan solishtirganda Retrofit + Gson arxitekturasining afzalliklarini ko'rsatish hamda uni oliy ta'lim muassasalarida "Mobil ilovalar yaratish" va "Java dasturlash" fanlari doirasida o'qitish uchun bosqichma-bosqich metodik namunani taqdim etish. Tadqiqot metodologiyasi rasmiy texnik hujjatlarni tahlil qilish, ilmiy manbalarni qiyosiy o'rganish hamda amaliy dastur (case-study) loyihalashtirish usullariga asoslangan.

Kalit so'zlar: REST API, Android, Java, Retrofit, Gson, mobil ilovalar, tarmoq arxitekturasini, JSON, OkHttp, dasturiy ta'minot ta'limi.

1. Kirish

So'nggi o'n yillikda mobil ilovalar deyarli barcha sohalarda ta'lim, moliya, sog'liqni saqlash, davlat xizmatlari va boshqa yo'nalishlarda asosiy dasturiy vosita sifatida qo'llanilmoqda. Mobil ilovalarning katta qismi mustaqil, izolyatsiyalangan tizim sifatida emas, balki markaziy serverlar bilan doimiy aloqada ishlaydigan mijoz (client) sifatida loyihalanadi. Bunday aloqa odatda Representational State

Transfer (REST) arxitektura uslubiga asoslangan veb-xizmatlar (REST API) orqali amalga oshiriladi, chunki bu uslub soddaligi, HTTP protokoli bilan tabiiy mosligi va keng tarqalganligi bilan ajralib turadi. Android platformasida Java tilida tarmoq bilan ishlashning bir necha yondashuvi mavjud. Dastlabki bosqichlarda dasturchilar `HttpURLConnection` klassidan foydalanib, so'rovlarni qo'lda shakllantirishgan, javoblarni oqim (stream) sifatida o'qib, `JSONObject` yordamida qo'lda tahlil qilishgan. Bu yondashuv ko'p miqdordagi takrorlanuvchi kod (boilerplate), xatolikka moyillik va past darajadagi murakkablik bilan bog'liq bo'lgan. Keyinchalik Google tomonidan taqdim etilgan Volley kutubxonasi so'rovlar navbatini boshqarishni soddalashtirdi, biroq tur xavfsizligi va POJO modellari bilan avtomatik ishlash imkoniyati cheklangan bo'lib qoldi. Ushbu muammolarni bartaraf etish maqsadida Square kompaniyasi tomonidan ishlab chiqilgan Retrofit kutubxonasi REST API endpoint'larini oddiy Java interfeysi ko'rinishida ifodalash imkonini beradi, bu esa kodni o'qish, qo'llab-quvvatlash va sinovdan o'tkazishni sezilarli darajada yengillashtiradi. Retrofit odatda Gson kutubxonasi bilan birgalikda qo'llaniladi Gson JSON formatidagi ma'lumotlarni avtomatik ravishda Java obyektlariga (POJO) va aksincha aylantirish vazifasini bajaradi. Natijada dasturchi past darajadagi tarmoq operatsiyalari bilan emas, balki dastur mantig'i bilan shug'ullanish imkoniyatiga ega bo'ladi. Ta'lim nuqtai nazaridan, Retrofit + Gson arxitekturasini o'qitish nafaqat texnik ko'nikma sifatida, balki dasturiy ta'minot muhandisligining umumiy tamoyillari abstraksiya, interfeyslar orqali dasturlash, mas'uliyatlarni ajratish (separation of concerns) bilan tanishtirish vositasi sifatida ham ahamiyatlidir. Shu sababli ushbu maqolaning maqsadi quyidagilardan iborat: (1) Retrofit va Gson kutubxonalari asosidagi tarmoq qatlami arxitekturasini tizimli tarzda tavsiflash; (2) namunaviy Android ilovasi misolida amalga oshirish bosqichlarini bosqichma-bosqich ko'rsatish; (3) an'anaviy yondashuvlar bilan solishtirma tahlil o'tkazish; (4) mavzuni oliy ta'lim muassasalarining "Mobil ilovalar yaratish" kabi fanlarida o'qitish bo'yicha metodik tavsiyalar ishlab chiqish.

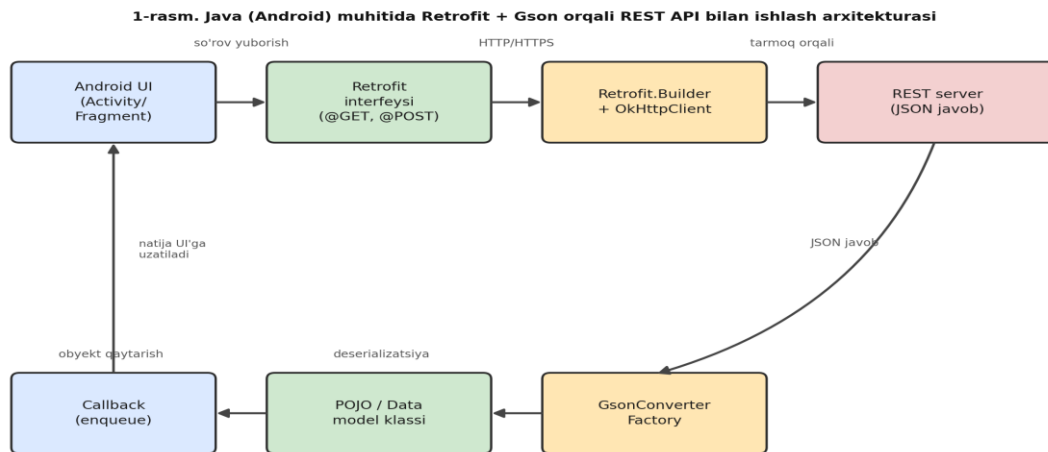
2. Tadqiqot metodologiyasi

Tadqiqotda uchta asosiy metod qo'llanildi. Birinchidan, Retrofit, OkHttp va Gson kutubxonalarining rasmiy texnik hujjatlari, shuningdek REST arxitekturasi bo'yicha asosiy ilmiy manba (Fielding, 2000) tahlil qilindi. Ikkinchidan, mavzu bo'yicha nashr etilgan ilmiy-amaliy maqolalar (Hafidz & Suhendar, 2023) va texnik ta'lim resurslari (Baeldung, 2025; GeeksforGeeks, 2025) qiyosiy kontent-tahlil usulida o'rganildi, bu esa Retrofit'ning boshqa kutubxonalarga nisbatan afzalliklarini aniqlash imkonini berdi. Uchinchidan, case-study (amaliy loyiha) metodidan foydalanib, ochiq REST API endpoint'iga murojaat qiluvchi namunaviy Android ilovasi loyihalashtirildi; bunda Android Studio muhiti, Java dasturlash tili, Retrofit 2.9.0, OkHttp va Gson kutubxonalarining barqaror versiyalari qo'llanildi. Amaliy qism quyidagi bosqichlarni o'z ichiga oladi: (a) loyihaga zarur bog'liqliklarni (dependencies) qo'shish; (b) server javobiga mos POJO model klassini yaratish; (c) API endpoint'larini tavsiflovchi Retrofit interfeysini yaratish; (d) Retrofit obyektini singleton naqshi (singleton pattern) asosida sozlash; (e) so'rovni asinxron tarzda yuborish va javobni qayta ishlash; (f) tarmoq xatoliklarini qayta ishlash mexanizmini joriy etish. Har bir bosqich quyidagi "Natijalar" bo'limida kod namunalari bilan izohlanadi.

3. Natijalar

3.1. Umumiy arxitektura

Retrofit + Gson yondashuvida ma'lumotlar oqimi aniq belgilangan qatlamlar orqali harakatlanadi: foydalanuvchi interfeysi (UI) so'rovni Retrofit interfeysiga uzatadi, Retrofit.Builder va OkHttpClient so'rovni HTTP formatida serverga yuboradi, server JSON formatida javob qaytaradi, GsonConverterFactory javobni avtomatik ravishda POJO obyektiga aylantiradi va natija Callback orqali UI qatlamiga qaytariladi. Ushbu jarayon 1-rasmda umumlashtirilgan tarzda tasvirlangan.



3.2. Loyihaga bog'liqliklarni qo'shish. Retrofit va Gson kutubxonalaridan foydalanish uchun avvalo modul darajasidagi build.gradle faylida quyidagi bog'liqliklar ko'rsatiladi:

```
dependencies {  
    implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
    implementation 'com.squareup.retrofit2:converter-gson:2.9.0'  
    implementation 'com.squareup.okhttp3:logging-interceptor:4.12.0'  
}
```

Bundan tashqari, AndroidManifest.xml faylida internetga kirish uchun ruxsat berilishi shart:

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

3.3. Ma'lumotlar modeli (POJO) yaratish. Gson kutubxonasi JSON obyektining har bir maydonini mos Java klass maydoniga avtomatik moslashtirishi uchun @SerializedName annotatsiyasidan foydalaniladi. Bu, ayniqsa, JSON maydon nomi Java o'zgaruvchisi nomlash qoidalariga mos kelmagan hollarda foydalidir:

```
public class Foydalanuvchi {
```

```
@SerializedName("id")
private int id;

@SerializedName("full_name")
private String toliqIsmi;

@SerializedName("email")
private String email;

public int getId() { return id; }
public String getToliqIsmi() { return toliqIsmi; }
public String getEmail() { return email; }
}
```

3.4. Retrofit interfeysini loyihalash. Retrofit'ning markaziy g'oyasi — har bir REST endpoint oddiy Java interfeysi metodi sifatida e'lon qilinishi va HTTP so'rov turi hamda manzili annotatsiyalar orqali ko'rsatilishidir:

```
public interface ApiXizmati {

    @GET("users")
    Call<List<Foydalanuvchi>> foydalanuvchilarniOlish();

    @GET("users/{id}")
    Call<Foydalanuvchi> foydalanuvchiniOlish(@Path("id") int id);

    @POST("users")
    Call<Foydalanuvchi> foydalanuvchiQoshish(@Body Foydalanuvchi
```

```
yangi);  
}
```

3.5. Retrofit obyektini sozlash (singleton naqshi). Ilova davomida faqat bitta Retrofit obyektini yaratilishi tavsiya etiladi, chunki uning ichki OkHttpClient komponenti resursta lab bo'lib, qayta-qayta yaratilishi ilovaning unumdorligiga salbiy ta'sir ko'rsatadi. Shu sababli singleton loyihalash naqshidan foydalaniladi:

```
public class RetrofitKlient {  
  
    private static final String BASE_URL = "https://api.example.com/";  
    private static Retrofit retrofit;  
  
    public static Retrofit getInstance() {  
        if (retrofit == null) {  
            retrofit = new Retrofit.Builder()  
                .baseUrl(BASE_URL)  
                .addConverterFactory(GsonConverterFactory.create())  
                .build();  
        }  
        return retrofit;  
    }  
}
```

3.6. So'rovni asinxron tarzda yuborish. Android'da tarmoq operatsiyalarini asosiy (UI) oqimda bajarish taqiqlanadi, shu sababli Retrofit so'rovlari enqueue() metodi orqali asinxron tarzda yuboriladi, natija esa Callback interfeysining onResponse() va onFailure() metodlari orqali qayta ishlanadi:

```
ApiXizmati xizmat =  
RetrofitKlient.getInstance().create(ApiXizmati.class);
```

```
xizmat.foydalanuvchilarniOlish().enqueue(new
Callback<List<Foydalanuvchi>>() {
    @Override
    public void onResponse(Call<List<Foydalanuvchi>> call,
        Response<List<Foydalanuvchi>> response) {
        if (response.isSuccessful() && response.body() != null) {
            List<Foydalanuvchi> royxat = response.body();
            // ro'yxatni RecyclerView adapteriga uzatish
        } else {
            Log.e("API", "Server xatosi: " + response.code());
        }
    }
});

@Override
public void onFailure(Call<List<Foydalanuvchi>> call, Throwable t)
{
    Log.e("API", "Tarmoq xatosi: " + t.getMessage());
}
});
```

Keltirilgan tuzilma xatoliklarni ikki darajada ajratish imkonini beradi: onResponse() ichida serverdan qaytgan, ammo muvaffaqiyatsiz javob kodlari (masalan, 404, 500) tekshiriladi, onFailure() esa internet aloqasi yo'qligi yoki so'rov vaqti tugashi kabi tarmoq darajasidagi xatoliklarni qamrab oladi. Ushbu ikki bosqichli yondashuv talabalarga xatoliklarni boshqarishning yaxlit tizimini tushuntirish uchun qulay metodik asos yaratadi.

3.7. An'anaviy yondashuvlar bilan solishtirma tahlil. HttpURLConnection, Volley va Retrofit + Gson yondashuvlarining asosiy mezonlar bo'yicha solishtirma tahlili 1-jadvalda keltirilgan.

Mezon	HttpURLConnection	Volley	Retrofit + Gson
Tur xavfsizligi (type safety)	Yo'q, qo'lda parsing	Qisman, JSON obyekt darajasida	To'liq, interfeys va POJO orqali
Boilerplate kod hajmi	Yuqori	O'rtacha	Minimal
JSON serializatsiya/deserializatsiya	Qo'lda (JSONObject)	Qo'lda yoki Gson bilan qo'shimcha sozlash	Avtomatik, GsonConverterFactory orqali
Asinxron so'rovlarni boshqarish	AsyncTask/Thread qo'lda yaratiladi	Ichki RequestQueue orqali	enqueue()/Call orqali tayyor mexanizm
Xatoliklarni boshqarish	try-catch, IOException qo'lda	ErrorListener orqali	onFailure()/onResponse() orqali strukturaviy
O'quv jarayoni uchun qulaylik	Past — talaba past darajadagi tafsilotlarga chalg'iydi	O'rtacha	Yuqori — arxitektura aniq, kod o'qilishi oson

1-jadval. Android'da REST API bilan ishlash yondashuvlarining qiyosiy tahlili

4. Muhokama

Olingan natijalar shuni ko'rsatadiki, Retrofit + Gson arxitekturasi nafaqat

kodni qisqartiradi, balki dasturiy ta'minotning arxitekturaviy sifatini ham oshiradi. Xususan, API endpoint'larining interfeys ko'rinishida ifodalanishi mas'uliyatlarni aniq ajratish (separation of concerns) tamoyilini amalda ko'rsatish imkonini beradi bu tushuncha ko'plab talabalar uchun mavhum bo'lib qoladigan tamoyillardan biridir. Shu bilan birga, POJO va Gson annotatsiyalari orqali JSON strukturasi bilan Java obyektlari o'rtasidagi bog'liqlikni ko'rgazmali tarzda tushuntirish mumkin. Pedagogik nuqtai nazardan, Retrofit'ni o'qitishda talabalar ko'pincha uchta xato bilan duch kelishadi: birinchidan, Retrofit obyektini har bir so'rov uchun qayta yaratish, bu esa singleton naqshi mohiyatini tushunmaslikdan kelib chiqadi; ikkinchidan, tarmoq so'rovini asosiy oqimda sinxron tarzda bajarishga urinish, natijada `NetworkOnMainThreadException` xatoligi yuzaga keladi; uchinchidan, `onFailure()` va muvaffaqiyatsiz `onResponse()` holatlarini farqlamaslik, buning oqibatida xatoliklar to'g'ri diagnostika qilinmaydi. Shu sababli o'quv dasturida ushbu uchta holatni maxsus amaliy mashqlar orqali alohida ko'rib chiqish tavsiya etiladi. Tadqiqotning cheklovi shundaki, u faqat Java tilidagi klassik Callback-asosli Retrofit yondashuvini qamrab oladi. Zamonaviy Android dasturlashda Kotlin tilidagi coroutine va suspend funksiyalar asosidagi Retrofit integratsiyasi, shuningdek Retrofit'ni Jetpack (ViewModel, LiveData, Room) komponentlari bilan birgalikda qo'llash alohida tadqiqot mavzusi bo'lishi mumkin. Kelgusi tadqiqotlarda ushbu yo'nalishlarni o'quv dasturlariga bosqichma-bosqich kiritish metodikasini ishlab chiqish maqsadga muvofiqdir.

5. Xulosa

Ushbu maqolada Java tilida Android ilovalarini REST API bilan bog'lash uchun Retrofit va Gson kutubxonalaridan foydalanish arxitekturasi tizimli tarzda tavsiflandi, amalga oshirish bosqichlari kod namunalari bilan ko'rsatildi va an'anaviy yondashuvlar bilan qiyosiy tahlil o'tkazildi. Tahlil natijalari Retrofit + Gson kombinatsiyasining tur xavfsizligi, kod hajmining qisqarishi, xatoliklarni boshqarishning tuzilganligi va o'quv jarayonida tushunarligi jihatidan `HttpURLConnection` va `Volley`'dan ustunligini tasdiqladi. Shu sababli ushbu yondashuvni oliy ta'lim muassasalarining "Mobil ilovalar yaratish", "Java

dasturlash" va "Dasturiy ta'minot arxitekturasini" fanlari o'quv dasturlariga alohida mavzu sifatida kiritish tavsiya etiladi. Bunday amaliyot talabalarda nafaqat texnik ko'nikmalarni, balki zamonaviy dasturiy ta'minot ishlab chiqishning asosiy arxitekturaviy tamoyillarini shakllantirishga xizmat qiladi.

Foydalanilgan adabiyotlar

1. Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures [Doctoral dissertation, University of California, Irvine].
2. Hafidz, M. F., & Suhendar, A. (2023). Implementation of REST API in Android-based Student Services Application. International Journal of Computer Applications, 185(40), 11–15. <https://doi.org/10.5120/ijca2023923201>
3. Baeldung. (2025). Introduction to Retrofit. <https://www.baeldung.com/retrofit>
4. GeeksforGeeks. (2025). Introduction to Retrofit in Android. <https://www.geeksforgeeks.org/android/introduction-retrofit-2-android-set-1/>
5. Vogella GmbH & Co. KG. (n.d.). Using Retrofit 2.x as REST client – Tutorial. <https://www.vogella.com/tutorials/Retrofit/article.html>
6. DigitalOcean. (2022). Retrofit Android example tutorial. <https://www.digitalocean.com/community/tutorials/retrofit-android-example-tutorial>
7. Reintech. (2023). Kotlin and REST APIs: Building a network layer with Retrofit and Gson. <https://reintech.io/blog/kotlin-and-rest-apis-building-a-network-layer-with-retrofit-and-gson>
8. Square, Inc. (n.d.). Retrofit — a type-safe HTTP client for Android and Java. <https://square.github.io/retrofit/>
9. Google. (n.d.). Gson — a Java serialization/deserialization library. <https://github.com/google/gson>