

API Integration Notes

1. What is an API?

API (Application Programming Interface) allows two applications to communicate and exchange data.

In Android, APIs are used to connect apps with web servers, databases, or third-party services.

2. Types of APIs

- **REST API (Representational State Transfer)** – Most common; uses HTTP methods (GET, POST, PUT, DELETE).
 - **SOAP API** – XML-based, used in enterprise systems.
 - **GraphQL API** – Flexible query-based API.
-

3. Common HTTP Methods

Method Purpose

GET Retrieve data

POST Send new data

PUT Update existing data

DELETE Remove data

4. API Integration Steps (Android)

1. Add Internet Permission in `AndroidManifest.xml`
2. `<uses-permission android:name="android.permission.INTERNET"/>`
3. Choose a Networking Library
 - **HttpURLConnection (basic)**
 - **Retrofit (recommended)**
 - **Volley (simpler for JSON)**
 - **OkHttp (low-level HTTP client)**

4. Define the Base URL and Endpoints

```
5. const val BASE_URL = "https://api.example.com/"
```

6. Make Network Requests

- Use Retrofit Interface:
- `interface ApiService {`
- `@GET("users")`
- `suspend fun getUsers(): Response<List<User>>`
- `}`
- Create Retrofit instance:
- `val retrofit = Retrofit.Builder()`
- `.baseUrl(BASE_URL)`
- `.addConverterFactory(GsonConverterFactory.create())`
- `.build()`
- `val api = retrofit.create(ApiService::class.java)`

7. Handle Response

```
8. lifecycleScope.launch {  
9.     val response = api.getUsers()  
10.    if (response.isSuccessful) {  
11.        val users = response.body()  
12.    } else {  
13.        Log.e("API", "Error: ${response.code()}")  
14.    }  
15. }
```

5. JSON Parsing

- Use Gson or Moshi libraries to convert JSON to Kotlin objects.
 - `data class User(val id: Int, val name: String, val email: String)`
-

6. Error Handling

- Always check `response.isSuccessful`.
 - Handle exceptions with try-catch.
 - Display user-friendly error messages.
-

7. Best Practices

- Use Coroutines for async network calls.
 - Implement ViewModel + LiveData (MVVM pattern).
 - Cache data when offline (e.g., Room DB).
 - Keep API keys secure using `.env` or `local.properties`.
 - Avoid network calls on the main thread.
-

Summary

API integration connects Android apps to external data and services. Using Retrofit with Coroutines is the most efficient approach for modern Android development, ensuring clean, scalable, and maintainable code.