Islamic University of Gaza
Faculty of Engineering
Department of Computer Engineering

Graduation Project
"Data sharing based on Wi-Fi"

API, Android API

Prof. Aiman Abu Samra

BY: ***Aaya M. El-Sharief*** ***Ghadeer A. ALMoqayad***
ID: ***220171534*** ***220171315***

August 2021

Contents:

1. Introduction into API

API is the acronym for Application Programming Interface, essentially a gateway that allows software to talk to other software, and is a software intermediary that allows two applications to talk to each other, each time we use the mobile or use an app like messenger to send or receive message, here we use API.

APIs have become so valuable that they comprise a large part of many business' revenue. Major companies like Google, eBay, Salesforce.com, Amazon

In other words, An API is a type of program that contains code for you to use in your own applications. This code normally allows you to add certain functionalities to your application.

2. What happens via API?

When use an application on mobile phone, the application connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to the phone. The application then interprets that data and presents it with the information that wanted in a readable way.

3. API Management

API management is the process of distributing, controlling, designing, publishing, documenting and analyzing APIs in a secure environment, Through an API management we can ensure the needs of the developers and applications using the API are being met.

The majority of API management solutions allow users to perform the following tasks:

- **API design:** API management solutions provide users – from developers to partners – the ability to design, publish and deploy APIs.

- **API gateway:** API management solutions also serve as an API gateway, which acts as a gatekeeper for all APIs by enforcing relevant API security policies and requests and also guarantees authorization and security.

- **API store:** API management solutions provide users with the ability to keep their APIs in a store or catalog where they can expose them to internal and/or external stakeholders. This API "store" then serves as a marketplace for APIs, where users can subscribe to APIs, obtain support from users and the community and so on.

- **API analytics:** API management allow users to monitor API usage, load, transaction logs, historical data and other metrics that better inform the status as well as the success of the APIs available.

4. API security

API security is the protection of the integrity of APIs, each client (mobile phone) and server communicates with small packets of data, sharing only that which is necessary, when phone send data is never fully exposed to the server and also server is never exposed to the phone one of API management Solutions is API gateway that provide security.

5. APIs and Socket Programming

Traditional network programming followed a client-server model. The primary APIs used for client-server networking were implemented in socket libraries built into operating systems. For example, Berkeley sockets and Windows Sockets (Winsock) APIs were the two primary standards for socket programming for many years.

6. Types of APIs by purpose

different types of APIs use for different purposes:

- **System APIs:** System APIs unlock data from core systems of record within an organization. Examples of critical systems that APIs could unlock data from include ERP, customer and billing systems, and proprietary databases.

- **Process APIs:** Process APIs interact with and shape data within a single system or across systems — breaking down data silos.

- **Experience APIs:** Experience APIs provide a business context for the data and processes that were unlocked and established with System and Process APIs. Experience APIs expose the data to be consumed by its intended audience — such as mobile applications, internal portals for customer data, etc.

7. Types of API architectural styles

There are various styles of architecture for APIs, as well as varying data formats within these styles, below the most common:

- **REST:** (Representational State Transfer) is an architectural style that separates the concerns of the API consumer from the API provider by relying on commands that are built-into the underlying networking protocol. HTML is the best-known example of this style.

- **RPC:** Remote Procedure Calls — or RPCs — typically require developers to execute specific blocks of code on another system. RPC-style remote invocation of procedures other systems usually requires developers to call those procedures by name. RPC architectural patterns can be observed in popular API technologies such as SOAP, GraphQL, and gRPC.

- **Event-driven/Streaming:** Sometimes referred to as evented, real-time, streaming, asynchronous, or push architectures, event-driven APIs don't wait for an API consumer to call them before delivering a response. Instead, a response is triggered by the occurrence of an event.

8. Real world API usage:

APIs are transforming the world. Businesses are using APIs to achieve digital transformation; that might mean moving faster, here we show some of how world use api.

- **Sharing flight information between airlines and travel sites**
  Discount travel sites use APIs to aggregate real-time information from airlines, hotels, and excursions. Without a streamlined way of accessing this information, they would either need to manually collect this data or cease to offer competitive travel deals altogether.

- **To search, collect, and share data** An examples of APIs that share real-time information include The New York Times, which allows you to analyze their database of thousands of articles, Even NASA has an open API chock-full of satellite imagery and constellation data for public use.

- **To reduce redundant work**
  Suppose you are a developer, and you work hard to create new something that already been made, APIs can provide teams with premade tools and features that save them from creating these tools from scratch for example YouTube's API enables developers to embed video players into their site, reproduce reports, and access other helpful resources.

- **Using Google Maps**
  Automating workflows between B2B software tools, and many uses

9. API for android

The Android API refers to the collection of various software modules which make up the complete Android SDK. In simpler words the Android API or Android SDK or just plain simple Android basically refers to the same thing. Since the software you write yourself interacts with the Android software to do various things, so the Android part is like an API.

The other API you are referring to is where the term API is most commonly used, especially when developing websites. It's the set of commands which you send to a server and then get some reply back from it. Sending commands to the server is called 'request' and the reply back from it is called 'response'. This request /response between the web browser and the server is the most common use of an API and its programming is on the rise, since nowadays most of the online services we use are

dependent on various API calls to different servers. By sending a request to the server, the sender asks the server to do something, like check weather for example and the response could be the forecast for today for your city. This way you don't have to write your own program for weather forecasting and can simply use the API for this weather forecasting service to get the weather information to use in your website or any other application.

In case of Android, you use their Maps API as an example to get the location of your device, or any other coordinates you provide to it.

To program a web service API, you need to know some server-side programming languages like PHP, .Net, Java, C#. This is called the backend programming. Then you need some front-end programming knowledge for which the language used is JavaScript. To program for the Android APIs, you need to know Java.

10. Wi-Fi Direct

Wi-Fi Direct is a connection that allows for peer-to-peer communication, linking devices together without a nearby centralized network, one device acts as an access point, and the other device connects to it using Wi-Fi Protected Setup (WPS) and Wi-Fi Protected Access (WPA/WPA2) security protocols.

11. What is API Level?

API Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform.

The Android platform provides a framework API that applications can use to interact with the underlying Android system. The framework API consists of:

- A core set of packages and classes
- A set of XML elements and attributes for declaring a manifest file
- A set of XML elements and attributes for declaring and accessing resources
- A set of Intents
- A set of permissions that applications can request, as well as permission enforcements included in the system.

APIs levels with Android versions:

Android Version	APIs Level
Android 2.3	API level 9
Android 2.3.3	API level 10
Android 3.0	API level 11
Android 3.1	API level 12
Android 3.2	API level 13
Android 4.0	API level 14
Android 4.0.3	API level 15
Android 4.1	API level 16
Android 4.2	API level 17
Android 4.3	API level 18
Android 4.4	API level 19
Android 4.4W	API level 20
Android 5.0	API level 21
Android 5.1	API level 22
Android 6.0	API level 23
Android 7.0	API level 24
Android 7.1	API level 25
Android 8.0	API level 26
Android 8.1	API level 27
Android 9	API level 28
Android 10	API level 29
Android 11	API level 30

Wi-Fi direct is used in android 4.0 Api 14

How many API are there in Android?

Each Android device runs at exactly one API level, this API level is guaranteed to be unique per Android platform version

12. APIs and libraries for Android

Let's review some of the most APIs and libraries that every Android developer should know about.

- **Cloud Storage API**
  The Cloud Storage interface from CloudRail, combines several common functions of several cloud storage systems. Through abstraction of methods and references to files and folders, it manages to greatly simplify integration with Dropbox, Google Drive, Microsoft OneDrive, Box and more. Also, it has all the basic methods, like uploading files, downloading files, creating directories, copying, moving and deleting.

- **Retrofit**
  Retrofit from square, is a REST Client for Java and Android. It makes it relatively easy to retrieve and upload JSON (or other structured data) via a REST based webservice. In Retrofit you configure which converter is used for the data serialization.

- **GSON**
  Retrofit, and so many others, use GSON. GSON is a library created by Google that optimizes and facilitates converting Java objects into JSON and vice versa.

- **EventBus**
  As application grows, it will be important to maintain a simplified system of communication between services, threads, fragments and activities. EventBus does just that by creating listeners in all elements: background services, activities, fragments, and helper classes. A similar library, from the creators of Retrofit, is Otto. Both allow events to be registered and subscribe to with scarcely any source code.

- **Android Pay**
  Android Pay from Google, simple and fast native method for creating secure payments in your application's payment infrastructure. This will allow your users to make payments in a consistent way.