# Chương 1. Cơ sở lý thuyết

## 1.1. Lý thuyết về android

### 1.1.1. Lịch sử về android



Ban đầu, Android là hệ điều hành cho các thiết bị cầm tay dựa trên lõi Linux do công ty Android Inc. (California, Mỹ) thiết kế. Công ty này sau đó được Google mua lại vào năm 2005 và bắt đầu xây dựng Android Platform. Các thành viên chủ chốt ở Android Inc. gồm có: Andy Rubin, Rich Miner, Nick Sears, and Chris White..

Và sau tiếp, vào cuối năm 2007, thuộc về Liên minh Thiết bị Cầm tay Mã Nguồn mở (Open Handset Alliance) gồm các thành viên nổi bật trong ngành viễn thông và thiết bị cầm tay như: Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile, ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, and Vodafone Group,…



Mục tiêu của Liên minh này là nhanh chóng đổi mới để đáp ứng tốt hơn cho nhu cầu người tiêu dùng và kết quả đầu tiên của nó chính là nền tảng Android. Android được thiết kế để phục vụ nhu cầu của các nhà sản xuất thiết, các nhà khai thác và các lập trình viên thiết bị cầm tay. Phiên bản SDK lần đầu tiên phát hành vào tháng 11 năm 2007, hãng TMobile cũng công bố chiếc điện thoại Android đầu tiên đó là chiếc T-Mobile G1, chiếc smartphone đầu tiên dựa trên nền tảng Android. Một vài ngày sau đó, Google lại tiếp tục công bố sự ra mắt phiên bản Android SDK release Candidate 1.0.

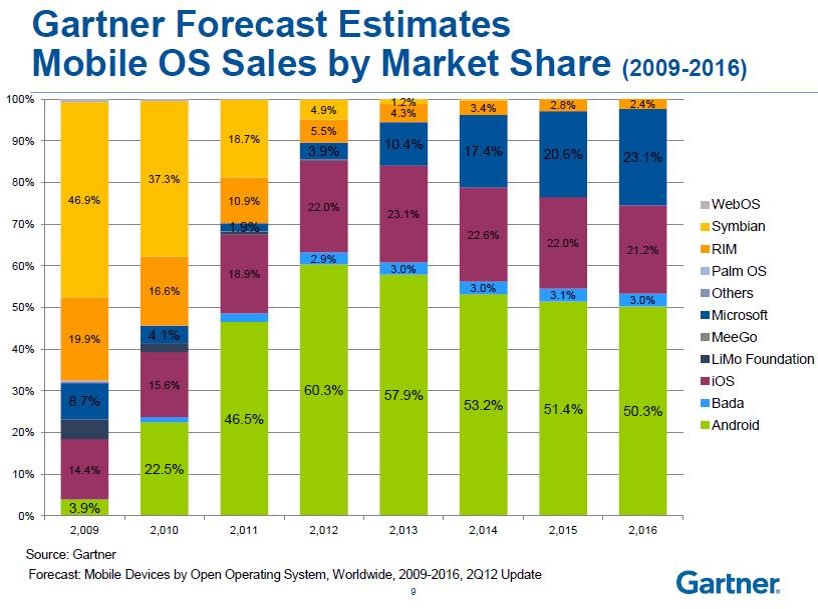
Trong tháng 10 năm 2008, Google được cấp giấy phép mã nguồn mở cho Android Platform. Khi Android được phát hành thì một trong số các mục tiêu trong kiến trúc của nó là cho phép các ứng dụng có thể tương tác được với nhau và có thể sử dụng lại các thành phần từ những ứng dụng khác. Việc tái sử dụng không chỉ được áp dụng cho các dịch vụ mà nó còn được áp dụng cho cả các thành phần dữ liệu và giao diện người dùng. Vào cuối năm 2008, Google cho phát hành một thiết bị cầm tay được gọi là Android Dev Phone 1 có thể chạy được các ứng dụng Android mà không bị ràng buộc vào các nhà cung cấp mạng điện thoại di động. Mục tiêu của thiết bị này là cho phép các nhà phát triển thực hiện các cuộc thí nghiệm trên một thiết bị thực có thể chạy hệ điều hành Android mà không phải ký một bản hợp đồng nào. Vào khoảng cùng thời gian đó thì Google cũng cho phát hành một phiên vản vá lỗi 1.1 của hệ điều hành này. Ở cả hai phiên bản 1.0 và 1.1 Android chưa hỗ trợ soft-keyboard mà đòi hỏi các thiết bị phải sử dụng bàn phím vật lý. Android cố định vấn đề này bằng cách phát hành SDK 1.5 vào tháng Tư năm 2009, cùng với một số tính năng khác. Chẳng hạn như nâng cao khả năng ghi âm truyền thông, vật dụng, và các live folder..

### 1.1.2. Các phiên bản của android

Tính đến nay, Android đã có các phiên bản (kèm tên mã) lần lượt là:

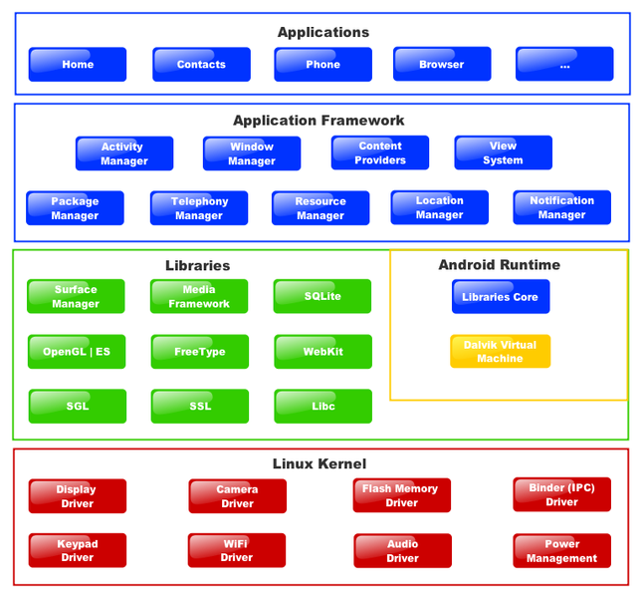
* Android 1.5 Cupcake
* Android 1.6 Donut
* Android 2.1 Eclair
* Android 2.2 Froyo
* Android 2.3 Gingerbread
* Android 3.0 Honeycomb
* Android 3.2 Honeycomb – Phiên bản Android đầu tiên được thiết kế cho máy tính bảng.
* Android 4.0 Ice Cream Sandwich
* Android 4.0.3 Ice Cream Sandwich
* Android 4.1 Jelly Bean
* Android 4.2 Jelly Bean
* Android 4.3 Jelly Bean
* Android 4.4 KitKat
* Android 4.4W KiKat Wear
* Android 5.0 Lollipop
* Android 5.1 Lollipop
* Android 6.0 Marshmallow
* Android 7.0 Nougat
* Android 7.1 Nougat
* Android 8.0 Oreo
* Android 8.1 Oreo
* Android 9.0 Pie
* Android 10.0 Q

### 1.1.3. **So sánh** **các** **nền** **tảng mobile platform**



Android là nền tảng mobile được sử dụng nhiều nhất.

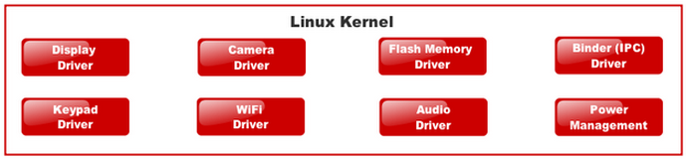
### 1.1.4 Kiến trúc android



**Nhân Linux**

Android dựa trên Linux version 2.6 gồm các dịch vụ hệ thống cơ bản như bảo mật, quản lý bộ nhớ, quản lý tiến trình, network và driver.

Nhân linux là tầng trừu tượng giữa phần cứng và software stack.

[](http://giasutinhoc.vn/wp-content/uploads/2016/03/gioi-thieu-ve-android-6.png)

**Thư** **viện**

Android bao gồm tập hợp các thư viện C/C++ sử dụng bởi các thành phần khác nhau của Android System.

Nhà phát triển khai thác thư viện này thông qua Android application framework. Một số thư viện cơ bản được liệt kê ở dưới.

[](http://giasutinhoc.vn/wp-content/uploads/2016/03/gioi-thieu-ve-android-7.png)

**Application framework**

Bằng cách cung cấp một nền tảng phát triển mở, Android cung cấp cho các nhà phát triển khả năng xây dựng các ứng dụng cực kỳ phong phú và sáng tạo. Nhà phát triển được tự do tận dụng các thiết bị phần cứng, thông tin địa điểm truy cập, các dịch vụ chạy nền, thiết lập hệ thống báo động, thêm các thông báo để các thanh trạng thái, và nhiều, nhiều hơn nữa. Nhà phát triển có thể truy cập vào các API cùng một khuôn khổ được sử dụng bởi các ứng dụng lõi. Các kiến trúc ứng dụng được thiết kế để đơn giản hóa việc sử dụng lại các thành phần; bất kỳ ứng dụng có thể xuất bản khả năng của mình và ứng dụng nào khác sau đó có thể sử dụng những khả năng (có thể hạn chế bảo mật được thực thi bởi khuôn khổ). Cơ chế này cho phép các thành phần tương tự sẽ được thay thế bởi người sử dụng. Cơ bản tất cả các ứng dụng là một bộ các dịch vụ và các hệ thống, bao gồm:

- Một tập hợp rất nhiều các View có khả năng kế thừa lẫn nhau dùng để thiết kế phần giao diện ứng dụng như: gridview, tableview, linearlayout,…

- Một “Content Provider” cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác (chẳng hạn như Contacts) hoặc là chia sẻ dữ liệu giữa các ứng dụng đó.

- Một “Resource Manager” cung cấp truy xuất tới các tài nguyên không phải là mã nguồn, chẳng hạn như: localized strings, graphics, and layout files.

- Một “Notifycation Manager” cho phép tất cả các ứng dụng hiển thị các custom alerts trong status bar. Activity Manager được dùng để quản lý chu trình sống của ứng dụng và điều hướng các activity.

**Application**

Các phiên bản Android bao gồm các ứng dụng cơ bản như email, sms, calendar, map, contact,… Tất cả ứng dụng được viết bằng ngôn ngữ lập trình Java.

[](http://giasutinhoc.vn/wp-content/uploads/2016/03/gioi-thieu-ve-android-8.png)

### 1.1.5. Các công cụ lập trình android

Có rất nhiều công cụ để phục vụ cho các nhà phát triển lập trình android. Một số công cụ tốt nhất:

**Studio Android**

Không có danh sách các công cụ phát triển Android nào hoàn chỉnh nếu không có [Android Studio](https://quantrimang.com/url?q=aHR0cHM6Ly9kb3dubG9hZC5jb20udm4vYW5kcm9pZC1zdHVkaW8vZG93bmxvYWQ%3D). Đây là phần mềm IDE chính thức (Môi trường phát triển tích hợp - Integrated Development Environment) cho Android, làm cho nó trở thành lựa chọn số một của phần lớn các nhà phát triển để tạo các ứng dụng cơ bản phù hợp với Material Design của Google và truy cập vào tất cả các tính năng nâng cao của nền tảng này.

Phần mềm này hoạt động như một trình biên tập cho ngôn ngữ lập trình đã chọn (Android Studio hỗ trợ [Java](https://quantrimang.com/java-la-gi-tai-sao-nen-chon-java-143130), C++ và bây giờ là Kotlin, mặc dù Java là ngôn ngữ chính thức của Android), một trình biên dịch có thể tạo các [file APK](https://quantrimang.com/file-apk-la-gi-lam-the-nao-de-tai-va-cai-dat-file-apk-124538) và hệ thống file để sắp xếp dự án của bạn. Nó cũng bao gồm một trình biên tập XML và "chế độ xem thiết kế" để sắp xếp các phần tử trên màn hình. Android Studio cung cấp toàn bộ bộ công cụ bổ sung và phần lớn các công cụ này sẽ được gói lại để có thể tải một lần. Trong thực tế nó cũng đi kèm với Android SDK, mặc dù bạn vẫn sẽ cần phải tải về và cài đặt JDK Java riêng biệt.

**Unity 3D**

[Unity 3D](https://quantrimang.com/url?q=aHR0cHM6Ly91bml0eTNkLmNvbS9mci8%3D) là công cụ được đề xuất đầu tiên không được cài đặt sẵn với Android Studio. Đây là một game engine và IDE cho phát triển trò chơi đa nền tảng. Unity rất dễ học và có rất nhiều tính năng cho việc phát triển game. Mặc dù bạn có thể tạo một trò chơi với Android Studio, nhưng Unity khiến công việc này trở nên dễ dàng hơn và bạn sẽ tiết kiệm được kha khá thời gian. Bạn thậm chí có thể sử dụng nó để tạo ra các ứng dụng thực tế ảo cho Daydream, Cardboard hoặc Gear VR.

**Eclipse**

Trước khi có sự xuất hiện của Android Studio, [Eclipse](https://quantrimang.com/url?q=aHR0cHM6Ly9lY2xpcHNlLm9yZy9pZGUv) là công cụ phát triển Android chính mà các nhà phát triển sử dụng. Công cụ này là một IDE chung hỗ trợ nhiều ngôn ngữ lập trình khác nhau, bao gồm cả Java với SDK Android. Không giống với Android Studio, Eclipse không cung cấp sẵn những hỗ trợ này và yêu cầu thiết lập một chút để có thể sử dụng chúng.

**GitHub**

GitHub là một công cụ trực tuyến mà bạn có thể sử dụng để chia sẻ các dự án và theo dõi nhiều phiên bản của những dự án đó khi bạn làm việc trong một nhóm. Nó tiện dụng cho việc sao lưu công việc, cộng tác và tìm mẫu code, hướng dẫn. Đối với người mới bắt đầu, khi sử dụng GitHub bạn có thể sẽ thấy bị hạn chế khi tải xuống các dự án mẫu có thể sử dụng kỹ thuật đảo ngược.

### 1.1.6. Máy ảo android( Android Emulator)

Android SDK và Plugin Eclipse được gọi là một Android Deverloper Tool (ADT). Các Android coder sẽ cần phải sử dụng công cụ IDE (Integrated Development Enveronment) này để phát triển, debugging và testing cho ứng dụng. Tuy nhiên, các coder cũng có thể không cần phải sử dụng IDE mà thay vào đó là sử dụng command line để biên dịch và tất nhiên là vẫn có Emulator như thường.

Android Emulator được trang bị đầy đủ hầu hết các tính năng của một thiết bị thật. Tuy nhiên, một số đã bị giới hạn như là kết nối qua cổng USB, camera và video, nghe phone, nguồn điện giả lập và bluetooth.

Android Emulator thực hiện các công việc thông qua một bộ xử lý mã nguồn mở, công nghệ này được gọi là QEMU (http://bellard.org/qemu/) được phát triển bởi Fabrice Bellard

Các nhà phát triển android thường hay sử dụng 2 Android Emulator phổ biến nhất:

**AVD Manager**

Công cụ [ADV Manager](https://quantrimang.com/url?q=aHR0cHM6Ly9kZXZlbG9wZXIuYW5kcm9pZC5jb20vc3R1ZGlvL3J1bi9tYW5hZ2luZy1hdmRzLmh0bWw%3D) được tích hợp với Android Studio. AVD viết tắt của Android Virtual Device – thiết bị ảo mô phỏng, vì vậy về cơ bản đây là trình giả lập để chạy các ứng dụng Android trên PC. Điều này rất hữu ích vì bạn có thể nhanh chóng kiểm tra ứng dụng của mình mà không phải cài đặt chúng liên tục trên các thiết bị thực. Quan trọng hơn, AVD Manager cho phép bạn tạo ra nhiều trình giả lập khác nhau với các kích thước màn hình, thông số kỹ thuật và phiên bản Android khác nhau. Điều này có nghĩa là bạn có thể thấy được thiết kế của mình trên các thiết bị cụ thể và do đó đảm bảo hỗ trợ qua các tiện ích phổ biến nhất. Hiệu suất trở nên tốt hơn, đặc biệt là với chế độ fast virt, chạy một phiên bản Android của Intel trên PC.

[**Genymotion**](https://quantrimang.com/url?q=aHR0cHM6Ly9kb3dubG9hZC5jb20udm4vZG93bmxvYWQvZ2VueW1vdGlvbi83NDMwNQ%3D%3D)

Trình giả lập này hướng đến các nhà phát triển muốn thử nghiệm ứng dụng, trò chơi của mình trên một loạt thiết bị Android mà không cần thực sự sở hữu những chiếc điện thoại đó. Có thể cấu hình trình giả lập cho nhiều thiết bị với phiên bản Android khác nhau để phù hợp với nhu cầu kiểm thử. Ví dụ, chạy Nexus One với Android 4.2 hoặc Nexus 6 với Android 6.0. Sự lựa chọn tùy thuộc ở bạn và có thể dễ dàng chuyển đổi giữa các thiết bị. Vì quá trình cài đặt và cấu hình hơi phức tạp hơn so với các trình giả lập khác nên Genymotion không phù hợp với những người dùng thông thường, sử dụng trình giả lập chỉ để kiểm tra email, chơi game. Nhưng Genymotion cung cấp dịch vụ miễn phí cho mục đích sử dụng cá nhân, do đó nếu muốn bạn vẫn có thể trải nghiệm phần mềm này.

### 1.1.7. Đặc điểm nổi bật của android

**Dùng Ngôn Ngữ Lập Trình Java**



Java có thể xem là ngôn ngữ lập trình phổ biến nhất hiện nay, được sử dụng rộng rãi trên nhiều thiết bị và hệ điều hành. Và dĩ nhiên, ứng dụng Android cũng được viết dựa trên ngôn ngữ phổ biến nhất này.

Tuy từ giữa năm 2017 Google có nói rằng họ sẽ thay thế Java bằng ngôn ngữ mới toanh có tên [Kotlin](https://yellowcodebooks.com/category/lap-trinh-android-bang-kotlin/bai-hoc-theo-chuong-trinh-lap-trinh-android-bang-kotlin/), nhưng , Java, cho đến giờ phút này, vẫn đang là ngôn ngữ thống trị trong các ứng dụng Android.

**Dùng Công Cụ Lập Trình Android Studio**



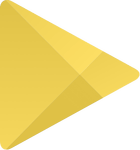
Nếu bạn nào từng sử dụng Eclipse cho lập trình ứng dụng Android trước đây thì dễ dàng so sánh được tính tiện dụng khi chuyển sang Android Studio. Nhìn chung Android Studio là một công cụ lập trình xuất sắc

**Tính Tương Thích**



Các ứng dụng Android gốc đều được viết bằng ngôn ngữ lập trình Java, nên ngoài việc thực thi trên hệ điều hành Android, nó rất dễ dàng để port sang các hệ điều hành khác như Backberry, Symbian, Ubuntu hay giờ đây có thêm Chrome OS. Có nguồn tin chưa chính xác là nhiều khả năng cả Windows 10 cũng có thể chạy các ứng dụng Android gốc.

**Ứng Dụng Của Bạn Sẽ Được Đưa Lên Google Play**



Một điều tuyệt vời nhất với Android là khi bạn quyết định đưa ứng dụng lên Google Playstore thì chỉ phải mất vài giờ để ứng dụng của bạn đến với tất cả người dùng trên toàn thế giới (con số này so với Apple App Store là vài ngày, thậm chí có khi đến tuần). Điều này có nghĩa là bạn hoàn toàn có thể đưa ra nhiều bản cập nhật lên store chỉ trong một ngày, dùng cho các trường hợp khẩn cấp khi cần cập nhật các tính năng quan trọng hoặc sửa các lỗi mà phiên bản trước vô tình chúng ta không kiểm tra kỹ (dù sao cũng đừng lạm dụng đặc tính này của lập trình Android mà không test kỹ ứng dụng trước khi đưa lên Google Play nhé bạn).

Một điều nữa chắc chắn bạn sẽ vui hơn, đó là để đăng ký cho việc đưa ứng dụng lên store, thì với Google Play bạn chỉ phải trả $25 duy nhất một lần, sau đó bạn sẽ tha hồ tung ra nhiều ứng dụng khác nhau. Con số này so với App Store là $99 cho mỗi năm.

**Khả Năng Kiếm Tiền**



Nói đến kiếm tiền dựa trên việc viết ứng dụng mobile người ta thường thiên về iOS, điều đó cho thấy các ứng dụng trên iPhone & iPad kiếm được nhiều tiền hơn, về cơ bản quan điểm này không sai, nhưng nó dần dần thay đổi khi mà người ta đang nhận ra rằng kiếm tiền trên Android lại ngày càng dễ dàng hơn. Ngoài việc số lượng người dùng trên Play store nhiều hơn thì còn có một lý do nữa, đó là người dùng Android đã quen với các baner quảng cáo nhỏ xuất hiện trong app, do đó sẽ dễ dàng cho bạn đăng ký đặt các baner quảng cáo vào ứng dụng của mình và nhận tiền quảng cáo đó. Trong khi người dùng iOS khắt khe hơn về mặt này và họ hầu như khó chấp nhận một ứng dụng miễn phí mà có chứa quảng cáo.

### 1.1.8. Giới thiệu về Android Studio và Android SDK

Tháng 5 năm 2013, Google công bố Android Studio, một môi trường phát triển ứng dụng tích hợp (IDE) dành riêng cho Android, mã nguồn mở, dựa trên IDE **Java IntelliJ** của hãng **JetBrains**(đối thủ với Eclipse và Netbeans, vốn khá quen thuộc với dân lập trình Java).

**Android Studio** chạy trên Windows, Mac và Linux, nhằm thay thế cho **Eclipse Android Development Tool**(ADT) vốn được sử dụng làm IDE chính trong các năm trước đó

**Một số tính năng nổi bật:**

* Bộ công cụ build ứng dụng dựa trên Gradle (thay vì Maven).
* Chức năng dò và sửa lỗi nhanh, hướng Android.
* Công cụ chỉnh sửa màn hình dạng kéo thả tiện lợi.
* Các wizard tích hợp nhằm giúp lập trình viên tạo ứng dụng từ mẫu có sẵn.
* Tích hợp Google Cloud Platform, dễ dàng tích hợp với Google Cloud Messaging và App Engine của Google.

Và đi kèm với Android Studio là **Android SDK**. Nếu Android Studio là trình soạn thảo code (IDE) thì Android SDK là bộ tổng hợp các công cụ để build app, các bản mẫu máy ảo Android (sử dụng để test app) cần thiết để làm ra một ứng dụng Android hoàn chỉnh.

## 1.2. Lý thuyết về retrofit

### 1.2.1. REST API

Đối với một trang web thông thường việc tạo ra (creat), sửa (edit), cập nhật (update) và xóa (delete) taì nguyên (resource) là việc thường xuyên và phổ biến với hầu hết các trang Web nên người ta đã chế ra một một quy luật thống nhất để tất cả các lập trình viên cứ thế mà làm theo mỗi lần phải lập trình một trang có 4 chức năng trên:

* Khi tạo ra mộtresource thì sử dụng phương thức **POST** để gửi dữ liệu lên máy chủ
* Khi tạo cập nhật mộtresource thì sử dụng phương thức **PUT**
* Khi xoá mộtresource thì sử dụng phương thức **DELETE** .
* Và cuối cùng là khi hiển thị một bài viết cho độc giả họ đọc thì sử dụng phương thức **GET**.

Và như vậy REST API ra đời, một tiêu chuẩn dùng trong việc thết kế các thiết kế API cho các ứng dụng web để quản lý các resource. REST là một trong những kiểu thiết kế API được sử dụng phổ biến nhất ngày nay.

Trọng tâm của REST quy định cách sử dụng các HTTP method (như GET,POST,PUT,DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource

### 1.2.2. Retrofit



Nói một cách dễ hiểu Retrofit là một thư viện giúp phân tích cú pháp phản hồi API dễ dàng và được xử lý tốt hơn để sử dụng trong ứng dụng.

Theo định nghĩa của Square( nhà phát triển Retrofit):

A type-safe HTTP client for Android and Java

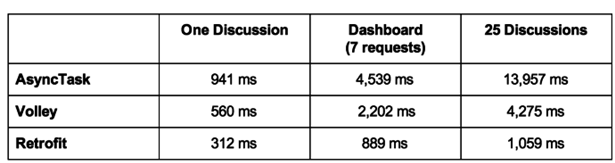
Retrofit là một type-safe HTTP client cho Java và Android được phát triển bởi Square. Retrofit giúp dễ dàng kết nối đến một dịch vụ REST trên web bằng cách chyển đổi API thành Java Interface.

**Type-safe** một ví dụ đơn giản là trình biên dịch sẽ xác nhận hợp lệ các kiểu dữ liệu trong khi biên dịch và ném một lỗi nếu bạn cố gán kiểu sai cho một biến.

Thư viện mạnh mẽ này giúp bạn dễ dàng xử lý dữ liệu JSON hoặc XML sau đó phân tích cú pháp thành Plain Old Java Objects (POJOs). Tất cả các yêu cầu **GET**, **POST**, **PUT**, **PATCH**, và **DELETE** đều có thể được thực thi.

Giống như hầu hết các phần mềm mã nguồn mở khác, Retrofit được xây dựng dựa trên một số thư viện mạnh mẽ và công cụ khác. Đằng sau nó, Retrofit làm cho việc sử dụng **OkHttp** để xử lý các yêu cầu trên mạng. Ngoài ra, từ Retrofit2 không tích hợp bất kỳ một bộ chuyển đổi JSON nào để phân tích từ JSON thành các đối tượng Java. Thay vào đó nó đi kèm với các thư viện chuyển đổi JSON sau đây để xử lý điều đó:

* Gson: com.squareup.retrofit:converter-gson
* Jackson: com.squareup.retrofit:converter-jackson
* Moshi: com.squareup.retrofit:converter-moshi

Dưới đây là bảng so sánh thời gian thực hiện Task của Retrofit so với Volley và AsyncTask (hình ảnh chỉ mang tính tham khảo):

Để sử dụng Retrofit2 ta thêm dependency vào trong file build.gradle :

implementation 'com.squareup.retrofit2:retrofit:2.4.0'

implementation 'com.squareup.retrofit2:converter-gson:2.3.0'

### **1.2.3. Retrofit Annotations**

Sử dụng Annotations để mô tả yêu cầu HTTP:

* Hỗ trợ tham số URL và tham số truy vấn
* Chuyển đổi đối tượng để yêu cầu nội dung
* Multipart request body và file upload

public interface StackOverflowApi {

@GET("/2.2/questions?order=desc&sort=creation&site=stackoverflow&tagged={tag}")

Observable<StackOverflowQuestion> loadQuestion(@Path("tag") String tag);

}

Annotation trên các phương thức của interface và các tham số của nó cho biết cách xử lý yêu cầu

**REQUEST METHOD**

Mỗi phương thức phải có Annotation HTTP cung cấp request method và URL. Có năm Annotation được tích hợp sẵn: [@GET](http://square.github.io/retrofit/), [@POST](http://square.github.io/retrofit/), [@PUT](http://square.github.io/retrofit/), [@DELETE](http://square.github.io/retrofit/) và [@HEAD](http://square.github.io/retrofit/) URL tương đối của tài nguyên được chỉ định trong Annotation.

@GET("users/list")

Bạn cũng có thể chỉ định tham số truy vấn trong URL.

@GET("users/list?sort=desc")

**URL MANIPULATION**

URL request có thể được cập nhật tự động bằng cách sử dụng các khối thay thế và tham số trên phương thức.

Chúng ta có thể sử dụng URL 1 cách động dựa vào biến truyền vào, bằng cách sử dụng anotation [@Path](http://square.github.io/retrofit/)

@GET("group/{id}/users")

Call<List<User>> groupList(@Path("id") int groupId);

Chúng ta có thể nối thêm paramater vào sau URL bằng cách sử dụng [@Query](http://square.github.io/retrofit/)

@GET("group/{id}/users")

Call<List<User>> groupList(@Path("id") int groupId, @Query("sort") String sort);

Đối với các kết hợp tham số truy vấn phức tạp, có thể sử dụng [@QueryMap](http://square.github.io/retrofit/).

@GET("group/{id}/users")

Call<List<User>> groupList(@Path("id") int groupId, @QueryMap Map<String, String> options);

**REQUEST BODY**

Một đối tượng có thể được chỉ định để sử dụng làm phần thân yêu cầu HTTP với Annotation [@Body.](http://square.github.io/retrofit/)

@POST("users/new")

Call<User> createUser(@Body User user);

Đối tượng cũng sẽ được chuyển đổi bằng cách sử dụng **Converter** được chỉ định trên instance của Retrofit. Nếu không có Converter nào được thêm vào, chỉ có thể sử dụng **RequestBody**.

**FORM ENCODED AND MULTIPART**

Các phương thức cũng có thể được khai báo để gửi dữ liệu được mã hóa và dữ liệu multipart(nhiều phần). Dữ liệu được mã hóa theo form được gửi khi [@FormUrlEncoded](http://square.github.io/retrofit/) được chỉ định trên phương thức. Mỗi cặp key-value được chú thích bằng [@Field](http://square.github.io/retrofit/) chứa tên và đối tượng cung cấp giá trị.

@FormUrlEncoded

@POST("user/edit")

Call<User> updateUser(@Field("first\_name") String first, @Field("last\_name") String last);

Các yêu cầu multipart được sử dụng khi [@Multipart](https://viblo.asia/u/Multipart) xuất hiện trên phương thức. Các phần được khai báo bằng cách sử dụng [@Part](http://square.github.io/retrofit/" \t "_blank)

@Multipart

@PUT("user/photo")

Call<User> updateUser(@Part("photo") RequestBody photo, @Part("description") RequestBody description);

Các phần của multiparts sử dụng một trong các bộ chuyển đổi của Retrofit hoặc chúng có thể implement RequestBody để xử lý serialization của riêng chúng.

**Chú thích:**

Serialization là một khái niệm giúp chúng ta có thể chuyển đổi trạng thái của một Java object thành một định dạng nào đó để Java object này có thể được lưu trữ ở đâu đó và sau đó, nó sẽ được sử dụng bởi một tiến trình khác.

Thông thường, khi sử dụng Serialization, Java object sẽ được chuyển đổi qua một dãy byte liên tục và chúng ta có thể lưu trong bộ nhớ, trên ổ đĩa, truyền qua mạng đến một server nào đó hoặc cũng có thể lưu chúng vào database.

Và khi một tiến trình khác sử dụng một Java object đã được Serialization, nó sẽ chuyển đổi định dạng đã Serialization về trạng thái của Java object ban đầu. Nhờ vậy, tiến trình đó có thể sử dụng lại Java object đã Serialization.

### **1.2.4. CONVERTERS**

Mặc định Retrofit chỉ có thể deserialize phần thân bản tin HTTP thành kiểu OkHttp's ResponseBody  
Và nó chỉ chấp nhận kiểu RequetsBody cho Annotation [@Body](https://viblo.asia/u/Body).

Converter có thể được thêm vào để hỗ trợ các loại khác:

**GSON**: com.squareup.retrofit2:converter-gson

Gson là một thư viện Java có thể được sử dụng để chuyển đổi các đối tượng Java thành biểu diễn JSON của chúng. Nó cũng có thể được sử dụng để chuyển đổi một chuỗi JSON thành một đối tượng Java tương đương.

public class Question {

@SerializedName("title")

@Expose

private String mTitle;

@SerializedName("link")

@Expose

private String mLink;

public String toString() {

return mTitle;

}

}

[@SerializedName](https://viblo.asia/u/SerializedName) cần thiết cho Gson để ánh xạ các khoá JSON với các trường dữ liệu. Để phù hợp với quy ước đặt tên kiểu camelCase của Java cho các thuộc tính thành viên của lớp, bạn không nên sử dụng dấu gạch dưới để tách các từ ngữ trong một biến.

[@Expose](https://viblo.asia/u/Expose) chỉ ra rằng trường này nên được định nghĩa với JSON serialization hoặc deserialization.

# Tạo instance Retrofit

Ví dụ: Tạo interface cho API từ [stackoverflow.com](http://stackoverflow.com/) lấy danh sách bài đăng thuộc tag "Android" :

public interface StackOverflowApi {

@GET("/2.2/questions?order=desc&sort=creation&site=stackoverflow&tagged={tag}")

Call<StackOverflowQuestion> loadQuestion(@Path("tag") String tag);

}

Tạo instance của Retrofit2 :

Retrofit retrofit = new Retrofit.Builder()

.baseUrl("https://api.stackexchange.com")

.addConverterFactory(GsonConverterFactory.create(gson))

.build();

Thực hiện Request đến API và xử lý kết quả trả về :

StackOverflowApi stackOverflowApi = retrofit.create(StackOverflowApi.class);

stackOverflowApi.loadQuestion("android")

.enqueue(new Callback<StackOverflowQuestion>() {

@Override

public void onResponse(Call<StackOverflowQuestion> call, Response<StackOverflowQuestion> response) {

//TODO Xử ký dữ liệu trả về

}

@Override

public void onFailure(Call<StackOverflowQuestion> call, Throwable t) {

//TODO Xử lý lỗi

}

});

## 1.3. Lý thuyết về hệ quản trị cơ sở dữ liệu MySQL

### 1.3.1.Tổng quan về MySQL

* MySQL là một phần mềm quản trị CSDL dạng server-based (gần tương đương với SQL Server của Microsoft).
* MySQL quản lý dữ liệu thông qua các CSDL, mỗi CSDL có thể có nhiều bảng quan hệ chứa dữ liệu.

MySQL có cơ chế phân quyền người sử dụng riêng, mỗi người dùng có thể được quản lý một hoặc nhiều CSDL khác nhau, mỗi người dùng có một tên truy cập (user name) và mật khẩu tương ứng để truy xuất đến CSDL.

* Khi ta truy vấn tới CSDL MySQL, ta phải cung cấp tên truy cập và mật khẩu của tài khoản có quyền sử dụng CSDL đó. Nếu không, chúng ta sẽ không làm được gì cả giống như quyền chứng thực người dung trong SQL Server vậy.

### 1.3.2. KHÁI NIỆM VỀ DATABASE, TABLE VÀ FIELD.

* ***Database:*** là một CSDL (cơ sở dữ liệu), trong CSDL bao gồm các table
* ***Table:*** là bảng dữ liệu, trong table bao gồm các field
* ***Field:*** là trường dữ liệu.

Cấu trúc của MySQL là: **Database -> Table -> Field**.

### 1.3.3.Các lệnh trong MySQL

Lệnh Insert

Lệnh Insert là câu lệnh SQL dùng để chèn một bản ghi vào table

**Cú pháp:**

INSERT INTO table\_name (field1, field2) VALUES (‘value1’, ‘value2’);

* ***table\_name:*** là tên của bảng
* ***field:*** là trường dữ liệu, các field được tách nhau bởi dấu phẩy
* ***value:*** là giá trị tương ứng với các field cũng được tách nhau bởi dấu phẩy

Lệnh Update

Lệnh Update là câu lệnh SQL dùng để cập nhật giá trị cho một hoặc nhiều bản ghi đang có trong bảng (tùy vào điều kiện giới hạn số bản ghi thực hiện của bạn).

***Cú pháp:***

UPDATE table\_name SET field1 = ‘value1’, field2 = ‘value2’ WHERE điều\_kiện

* ***table\_name:*** là bảng dữ liệu cần cập nhật
* ***field = ‘value’***: các trường và giá trị tương ứng cần cập nhật
* ***WHERE điều\_kiện***: Điều kiện giới hạn số bản ghi cần thực hiện (nếu không có điều kiện WHERE) MySQL sẽ xử lý toàn bộ các bản ghi trên bảng.

**Lệnh Delete**

Lệnh Delete là câu lệnh SQL dùng để xóa một hay nhiều bản ghi

***Cú pháp:***

DELETE FROM table\_name WHERE điều\_kiện

* ***table\_name:*** là tên bảng dữ liệu cần xóa
* ***WHERE điều\_kiện:*** là các điều kiện để giới hạn số bản ghi bị tác động (nếu bỏ trống toàn bộ các bản ghi trong bảng này sẽ bị xóa bỏ)

**Lệnh Select**

Câu lệnh SELECT là câu lệnh SQL dùng để truy vấn các bản ghi trong cơ sở dữ liệu của bạn. Đây là câu lệnh SQL được sử dụng nhiều nhất, tùy biến nhất để giải quyết mọi yêu cầu khai thác dữ liệu của bài toán.

***Cú pháp:***

SELECT field1, field2 FROM table\_name WHERE điều\_kiện ORDER BY field ASC|field DESC|rand() LIMIT Số bản ghi|Giá trị đầu, giá trị cuối

* ***Field1, Field2*** là các trường dữ liệu cần truy vấn
* ***table\_name*** là bảng dữ liệu cần truy suất
* ***WHERE điều\_kiện*** là các điều kiện để có được dữ liệu mong muốn
* ***ORDER BY field ASC*** lệnh sắp xếp dữ liệu tăng dần
* ***ORDER BY field DESC*** lệnh sắp xếp dữ liệu giảm dần
* ***ORDER BY rand()*** lệnh lấy ngẫu nhiên
* ***LIMIT số\_bản\_ghi*** lấy n bản ghi
* ***LIMIT o,n***: lấy n bản ghi từ vị trí o