

# Tổng quan về dự án

Một *dự án* trong Android Studio chứa mọi thông tin giúp xác định không gian làm việc cho ứng dụng, từ mã nguồn và tài sản, cho đến mã thử nghiệm và cấu hình bản dựng.

Khi bạn bắt đầu một dự án mới, Android Studio sẽ tạo cấu trúc cần thiết cho tất cả các tệp của bạn và hiển thị các cấu hình đó trong cửa sổ **Dự án** trong Android Studio. Để mở cửa sổ, hãy chọn **View > Tool Windows > Project** (Xem > Cửa sổ công cụ > Dự án).

Trang này cung cấp thông tin tổng quan về các thành phần chính trong dự án của bạn.

## Mô-đun

*Mô-đun* là một tập hợp gồm các tệp nguồn và tùy chọn cài đặt bản dựng cho phép bạn chia dự án của mình thành các đơn vị chức năng riêng biệt. Dự án của bạn có thể có một hoặc nhiều mô-đun và một mô-đun có thể sử dụng một mô-đun khác làm phần phụ thuộc. Bạn có thể xây dựng, kiểm tra và gỡ lỗi độc lập từng mô-đun.

Mô-đun bổ sung hữu ích khi tạo các thư viện mã trong dự án của riêng bạn hoặc khi bạn muốn tạo các bộ mã và tài nguyên khác nhau cho các loại thiết bị khác nhau, chẳng hạn như điện thoại và thiết bị đeo, nhưng giữ tất cả các tệp trong cùng một dự án và chia sẻ một số mã.

Để thêm mô-đun mới vào dự án của bạn, hãy nhấp vào **File > New > New Module** (Tệp > Mới > Mô-đun mới).

Android Studio cung cấp một số loại mô-đun riêng biệt:

### Mô-đun ứng dụng Android

Cung cấp một vùng chứa cho mã nguồn, tệp tài nguyên và các chế độ cài đặt cấp ứng dụng của bạn, chẳng hạn như tệp bản dựng ở cấp mô-đun và tệp kê khai Android. Khi bạn tạo một dự án mới, mô-đun ứng dụng mặc định có tên là "app" (ứng dụng).

Android Studio cung cấp các loại mô-đun ứng dụng sau đây:

- Điện thoại và máy tính bảng
- Automotive
- Wear OS
- Truyền hình
- Trình tạo hồ sơ cơ sở
- Điểm chuẩn

Mỗi mô-đun đều cung cấp các tệp thiết yếu và một số mẫu mã phù hợp với loại ứng dụng hoặc thiết bị tương ứng.

Để biết thêm thông tin về cách thêm mô-đun, hãy đọc phần [Thêm mô-đun cho thiết bị mới](https://developer.android.com/studio/projects/add-app-module?hl=vi) (<https://developer.android.com/studio/projects/add-app-module?hl=vi>).

### Mô-đun tính năng

Đại diện cho một tính năng được mô-đun hóa của ứng dụng có thể tận dụng *Play Feature Delivery*. Ví dụ: với các mô-đun tính năng, bạn có thể cung cấp cho người dùng một số tính năng nhất định theo yêu cầu của bạn hoặc dưới dạng trải nghiệm tức thì thông qua [Google Play Instant](https://developer.android.com/topic/google-play-instant/overview?hl=vi) (<https://developer.android.com/topic/google-play-instant/overview?hl=vi>).

Android Studio cung cấp các loại mô-đun tính năng sau:

- Mô-đun tính năng động
- Mô-đun thư viện tính năng động tức thì

Để tìm hiểu thêm, hãy đọc bài viết về [Play Feature Delivery](https://developer.android.com/studio/projects/dynamic-delivery?hl=vi) (<https://developer.android.com/studio/projects/dynamic-delivery?hl=vi>).

### Mô-đun thư viện

Cung cấp một vùng chứa cho mã có thể sử dụng lại mà bạn có thể dùng làm phần phụ thuộc trong các mô-đun ứng dụng khác hoặc nhập vào các dự án khác. Về cấu trúc, một mô-đun thư viện giống như một mô-đun ứng dụng, nhưng khi được tạo, mô-đun này tạo tệp lưu trữ mã thay vì APK, vì vậy, bạn không thể cài đặt mô-đun này trên thiết bị.

Trong cửa sổ **Create New Module** (Tạo mô-đun mới), Android Studio cung cấp các loại mô-đun thư viện sau đây:

- **Android Library** (Thư viện Android): Chứa mọi loại tệp được hỗ trợ trong một dự án Android, ngoại trừ mã C++ gốc, bao gồm cả mã nguồn Java và Kotlin, tài nguyên và tệp kê khai. Kết quả là một tệp Android Archive (AAR) mà bạn có thể thêm dưới dạng phần phụ thuộc cho các mô-đun ứng dụng Android của mình.
- **Android Native Library** (Thư viện gốc Android) Chứa tất cả các loại tệp được hỗ trợ trong một dự án Android, tương tự như Thư viện Android. Tuy nhiên, Thư viện gốc Android cũng có thể chứa mã nguồn C++ gốc. Kết quả là một tệp Android Archive (AAR) mà bạn có thể thêm dưới dạng phần phụ thuộc cho các mô-đun ứng dụng Android của mình.
- **Java or Kotlin Library** (Thư viện Java hoặc Kotlin): Chỉ chứa các tệp nguồn Kotlin hoặc Java. Kết quả là một tệp Java Archive (JAR) mà bạn có thể thêm vào dưới dạng phần phụ thuộc cho mô-đun ứng dụng Android hoặc các dự án Kotlin hoặc Java khác.

Đôi khi, mô-đun được gọi là *dự án phụ*, vì Gradle cũng gọi các mô-đun là các dự án.

Khi tạo một mô-đun thư viện và muốn thêm mô-đun đó làm phần phụ thuộc cho mô-đun ứng dụng Android, bạn phải khai báo mô-đun đó như sau:

**Groovy**Kotlin (#kotlin)  
(#groovy)

```
dependencies {  
    implementation project(':my-library-module')  
}
```

## Tệp dự án

Theo mặc định, Android Studio sẽ hiển thị các tệp dự án của bạn ở chế độ xem **Android**. Chế độ xem này không phản ánh hệ phân cấp tệp thực tế trên đĩa. Thay vào đó, tệp này được sắp xếp theo các mô-đun và loại tệp để đơn giản hóa việc di chuyển giữa các tệp nguồn chính của dự án, ẩn một số tệp nhất định hoặc thư mục không thường được sử dụng.

Một số điểm khác biệt về cấu trúc giữa chế độ xem **Android** và cấu trúc trên đĩa là chế độ xem **Android**:

- Hiển thị tất cả các tệp cấu hình liên quan đến bản dựng của dự án trong một nhóm **Gradle Script** cấp cao nhất.
- Hiển thị tất cả các tệp kê khai cho mỗi mô-đun trong một nhóm cấp mô-đun (khi bạn có nhiều tệp kê khai cho phiên bản sản phẩm và các loại bản dựng khác nhau).
- Hiển thị tất cả các tệp tài nguyên thay thế trong một nhóm, thay vì trong các thư mục riêng trên mỗi bộ hạn định tài nguyên. Ví dụ: tất cả các phiên bản mật độ của biểu tượng trình chạy đều hiển thị cạnh nhau.

Trong mỗi mô-đun ứng dụng Android, tệp sẽ được hiển thị ở các nhóm sau:

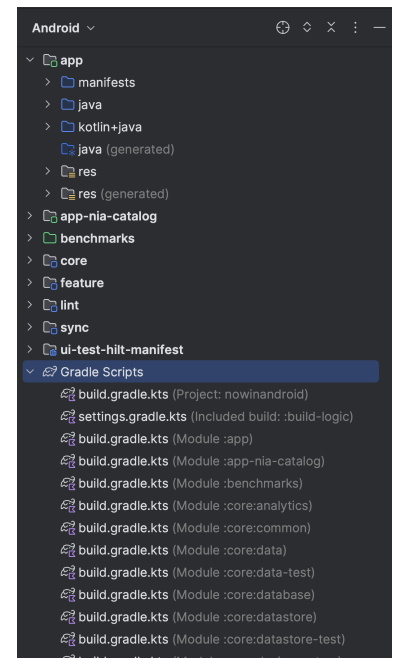
### tệp kê khai

Chứa **AndroidManifest.xml**

(<https://developer.android.com/guide/topics/manifest/manifest-intro?hl=vi>) tệp.

### java

Chứa các tệp mã nguồn Kotlin và Java, phân tách bằng tên gói, bao gồm cả mã thử nghiệm JUnit.



## res

Chứa mọi tài nguyên không phải mã, chẳng hạn như chuỗi giao diện người dùng và hình ảnh bitmap, được chia thành các thư mục con tương ứng. Để biết thêm thông tin về các loại tài nguyên có thể có, hãy xem nội dung [Tổng quan về tài nguyên ứng dụng](https://developer.android.com/guide/topics/resources/providing-resources?hl=vi) (<https://developer.android.com/guide/topics/resources/providing-resources?hl=vi>).

**Lưu ý:** Các loại mô-đun khác nhau có cấu trúc nhóm khác nhau. Ví dụ: một mô-đun gốc cũng chứa thư mục `cpp/` bên trong nhóm `main`, và thư viện Kotlin hoặc Java không chứa `manifests` hoặc `res` nhóm.

## Chế độ xem Dự án

Để xem thực tế cấu trúc tệp của dự án, bao gồm tất cả các tệp bị ẩn khỏi giao diện **Android**, hãy xem **Dự án** từ trình đơn ở đầu cửa sổ **Dự án**.

Khi chọn chế độ xem **Dự án**, bạn có thể xem nhiều tệp và thư mục hơn, bao gồm:

`module-name/`

`build/`

Bao gồm dữ liệu đầu ra của bản dựng.

`libs/`

Chứa thư viện riêng tư.

`src/`

Chứa tất cả mã và tệp tài nguyên cho mô-đun trong thư mục con sau:

`androidTest/`

Chứa mã để kiểm tra khả năng đo lường chạy trên một thiết bị Android. Để biết thêm thông tin chi tiết, vui lòng xem bài viết [Kiểm thử trong Android Studio](https://developer.android.com/studio/test/test-in-android-studio?hl=vi) (<https://developer.android.com/studio/test/test-in-android-studio?hl=vi>).

`cpp/`

Chứa mã gốc C hoặc C++ bằng Giao diện gốc Java (JNI). Để biết thêm thông tin, hãy xem [tài liệu về Android NDK](https://developer.android.com/ndk?hl=vi) (<https://developer.android.com/ndk?hl=vi>).

`main/`

Chứa các tệp nguồn "chính": mã Android và các tài nguyên dùng chung cho tất cả các biến thể bản dựng (các tệp cho các biến thể bản dựng khác nằm trong các thư mục khác nhau, chẳng hạn như `src/debug/` cho loại bản dựng gỡ lỗi):

`AndroidManifest.xml`

Mô tả tính chất của ứng dụng và từng thành phần của ứng dụng. Để biết thêm thông tin, hãy xem nội dung tổng quan về [tệp kê khai ứng dụng](https://developer.android.com/guide/topics/manifest/manifest-intro?hl=vi) (<https://developer.android.com/guide/topics/manifest/manifest-intro?hl=vi>).

`java/`

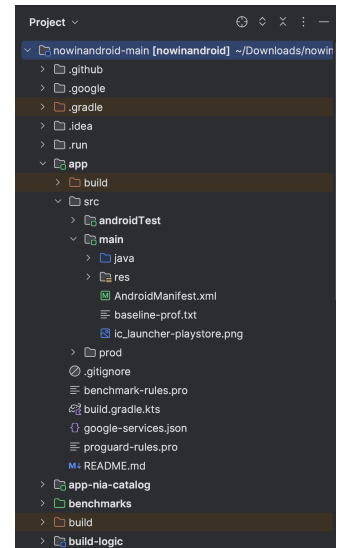
Chứa các nguồn mã Kotlin hoặc Java hoặc cả hai, nếu ứng dụng của bạn có cả mã nguồn Kotlin và Java.

`kotlin/`

Chỉ chứa các nguồn mã Kotlin.

`res/`

Chứa các tài nguyên ứng dụng, chẳng hạn như tệp có thể vẽ và tệp chuỗi giao diện người dùng. Để biết thêm thông tin, hãy xem nội dung tổng quan về [tài nguyên ứng dụng](https://developer.android.com/guide/topics/resources?hl=vi) (<https://developer.android.com/guide/topics/resources?hl=vi>).



`assets/`

Chứa các tệp cần được biên dịch thành tệp APK nguyên trạng. Ví dụ: đây là vị trí phù hợp để lưu trữ các họa tiết và dữ liệu trò chơi. Bạn có thể điều hướng thư mục này theo cách tương tự như một hệ thống tệp thông thường, sử dụng URI và đọc tệp dưới dạng luồng byte bằng [AssetManager](https://developer.android.com/reference/android/content/res/AssetManager?hl=vi) (<https://developer.android.com/reference/android/content/res/AssetManager?hl=vi>).

`test/`

Chứa mã cho các thử nghiệm cục bộ chạy trên Máy chủ ảo của bạn.

#### `build.gradle` hoặc `build.gradle.kts` (mô-đun)

Quy trình này xác định các cấu hình bản dựng dành riêng cho mô-đun. `build.gradle` là tên tệp chính xác nếu bạn sử dụng Groovy làm ngôn ngữ cho tập lệnh bản dựng và `build.gradle.kts` nếu bạn đang sử dụng tập lệnh Kotlin.

#### `build.gradle` hoặc `build.gradle.kts` (dự án)

Quy trình này xác định cấu hình bản dựng áp dụng cho mọi mô-đun. `build.gradle` là tên tệp chính xác nếu bạn sử dụng Groovy làm ngôn ngữ cho tập lệnh bản dựng và `build.gradle.kts` nếu bạn đang sử dụng tập lệnh Kotlin. Tệp này là phần không thể thiếu trong dự án, vì vậy, hãy duy trì tệp này trong tùy chọn quản lý phiên bản với tất cả mã nguồn khác.

Để biết thông tin về các tệp bản dựng khác, hãy xem phần [Định cấu hình bản dựng của bạn](https://developer.android.com/studio/build?hl=vi) (<https://developer.android.com/studio/build?hl=vi>).

**Lưu ý:** Để thay đổi chế độ xem mặc định từ **Android** sang **Dự án**, hãy chọn **Help (Trợ giúp) > Edit Custom Properties (Chỉnh sửa thuộc tính tùy chỉnh)** rồi thêm `studio.projectview=true`.

## Chế độ cài đặt cấu trúc dự án

Để thay đổi các tùy chọn cài đặt khác nhau cho dự án Android Studio của bạn, hãy mở hộp thoại **Project Structure** (Cấu trúc dự án) bằng cách nhấp vào **File > Project Structure** (Tệp > Cấu trúc dự án). Phương thức này bao gồm các mục sau:

- **Dự án:** Đặt phiên bản cho [Gradle và trình hỗ trợ Android cho Gradle](https://developer.android.com/studio/build?hl=vi#build-files) (<https://developer.android.com/studio/build?hl=vi#build-files>) và tên vị trí lưu trữ.
- **Vị trí SDK:** Đặt vị trí của JDK, SDK Android và NDK Android mà dự án của bạn sử dụng.
- **Biến:** Cho phép bạn chỉnh sửa các biến được dùng trong tập lệnh bản dựng.
- **Mô-đun:** Cho phép bạn chỉnh sửa các cấu hình bản dựng dành riêng cho mô-đun, bao gồm SDK mục tiêu và tối thiểu, chữ ký ứng dụng và các phần phụ thuộc thư viện. Trang cài đặt của mỗi mô-đun được chia thành các thẻ sau:
  - **Thuộc tính:** Chỉ định các phiên bản của SDK và xây dựng các công cụ cần sử dụng để biên dịch mô-đun.
  - **Ký:** Chỉ định chứng chỉ cần sử dụng để [ký ứng dụng](https://developer.android.com/tools/publishing/app-signing?hl=vi#sign-auto) (<https://developer.android.com/tools/publishing/app-signing?hl=vi#sign-auto>).
- **Phần phụ thuộc:** Liệt kê các phần phụ thuộc trong thư viện, tệp và mô-đun đối với mô-đun này. Bạn có thể thêm, sửa đổi và xóa các phần phụ thuộc khỏi ngăn này. Để biết thêm thông tin về các phần phụ thuộc, hãy xem bài viết [Định cấu hình các biến thể bản dựng](https://developer.android.com/tools/building/configuring-gradle?hl=vi#declareDeps) (<https://developer.android.com/tools/building/configuring-gradle?hl=vi#declareDeps>).



**Xem trước:** Nếu bạn đang sử dụng danh mục phiên bản, hãy lưu ý một vấn đề là phần phụ thuộc đã thêm qua hộp thoại **Project Structure** (Cấu trúc dự án) sẽ được thêm vào tệp `build.gradle` của mô-đun, chứ không phải danh mục. Để tìm hiểu thêm về tính năng hỗ trợ Danh mục phiên bản Gradle (Gradle Version Catalogs) trong Android Studio, hãy xem phần [ghi chú về bản thử nghiệm](https://developer.android.com/studio/preview/features?hl=vi#gradle-version-catalogs) (<https://developer.android.com/studio/preview/features?hl=vi#gradle-version-catalogs>).

- **Biến thể bản dựng:** Cho phép bạn định cấu hình các phiên bản và loại bản dựng khác nhau cho dự án.

- **Phiên bản:** Cho phép bạn tạo nhiều *phiên bản* bản dựng, trong đó mỗi phiên bản chỉ định một tập hợp cài đặt cấu hình, chẳng hạn như phiên bản SDK mục tiêu và tối thiểu của mô-đun, và mã phiên bản và tên phiên bản (<https://developer.android.com/studio/publish/versioning?hl=vi#versioningsettings.html>).

Ví dụ: bạn có thể xác định một phiên bản có SDK tối thiểu là 21 và SDK mục tiêu là 29, còn một phiên bản khác có SDK tối thiểu là 24 và SDK mục tiêu là 33.

- **Loại bản dựng:** Cho phép bạn tạo và sửa đổi cấu hình của bản dựng, như mô tả trong bài viết Định cấu hình các biến thể bản dựng (<https://developer.android.com/tools/building/configuring-gradle?hl=vi>). Theo mặc định, mọi mô-đun đều có các loại bản dựng *gỡ lỗi* và *phát hành*, và bạn có thể xác định thêm các loại bản dựng nếu cần.

Nội dung và mã mẫu trên trang này phải tuân thủ các giấy phép như mô tả trong phần Giấy phép nội dung (<https://developer.android.com/license?hl=vi>). Java và OpenJDK là nhãn hiệu hoặc nhãn hiệu đã đăng ký của Oracle và/hoặc đơn vị liên kết của Oracle.

Cập nhật lần gần đây nhất: 2023-08-10 UTC.