**PHẦN 1: LIBRARIES USED**

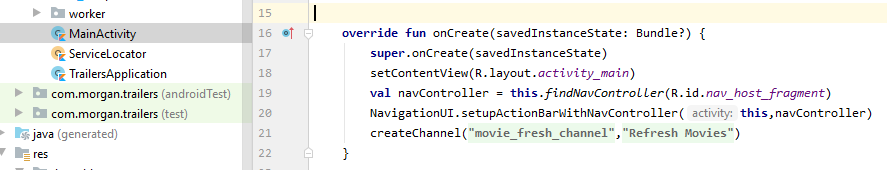
**I. FOUNDATION**

**1. AppCompat**

Cho phép truy cập vào các API mới trên các phiên bản API cũ hơn của platforms ( nhiều phiên bản sử dụng Material Design)

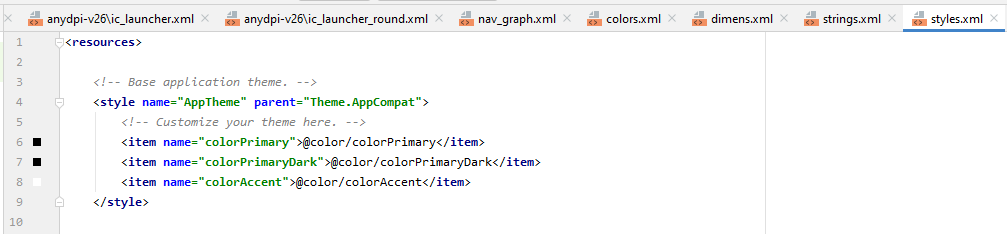
Khai báo phụ thuộc trong Build.grade.



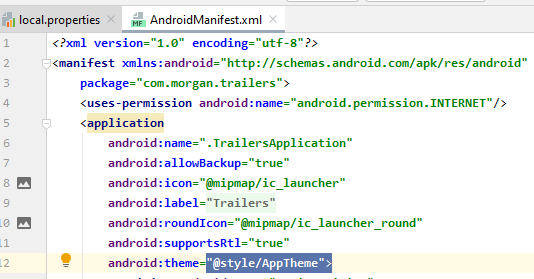


Ứng dụng định nghĩa sẵn một them tên là AppTheme ( trong styles.xml), theme này được gán vào application trong manifests ( có nghĩa là các Activity nằm trong applicatuon sẽ kế thừa theme này.

Theme AppTheme kế thừa từ Theme.AppCompat, sau đó nó điều chỉnh lại các thuộc tính colorPrimary, colorDark, colorAccent.



Để áp dụng Theme này, cần thay thuộc tính theme của application trong manifests để tất cả Activity kế thừa từ theme này



**2. Android KTX ( Android Kotlin Extensions)**

Là tiện ích mở rộng của Kotlin để giúp xây dựng ứng dụng Android. Tiện ích này là các thành phần được xây dựng sẵn, được tổng hợp lại thành các gói thư viện riêng lẻ để mở rộng ,để bổ sung cho các thư viện sẵn có,giúp chúng ta xây dựng chức năng, code Kotlin sẽ trở nên đẹp đẽ, dễ đọc hơn.

Để sử dụng KTX Android trong project, bạn phải thêm một số phụ thuộc vào dự án. Thêm phầndency vào tệp build.grade



Core KTX chứa tất cả các thư viện có trong Android Framework.

Sử dụng mô-đun Fragment KTX



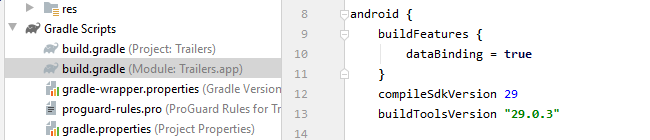
Fragment KTX được sử dụng để đơn giản hóa API phân mảnh

**II. ARCHITECTURE**

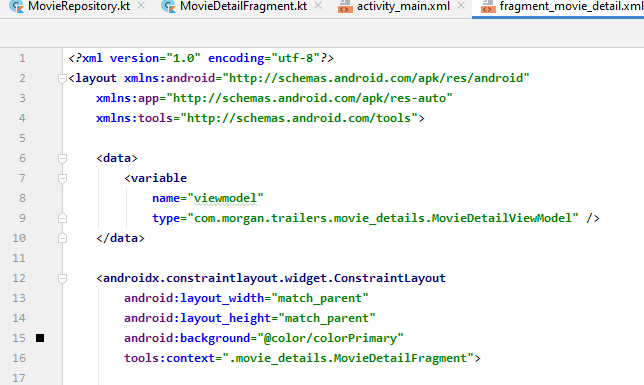
**1. Data Binding**

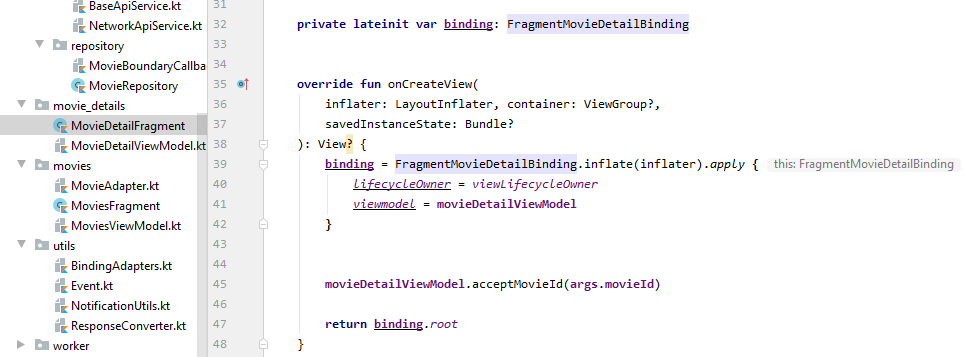
Data binding là một trong những thành phần của Android Jetpack cho phép người lập trình liên kết dữ liệu tới UI.Nó xóa bỏ nhiều UI framework được gọi trong Activity của bạn,làm thêm dễ đọc và dễ bảo trì.

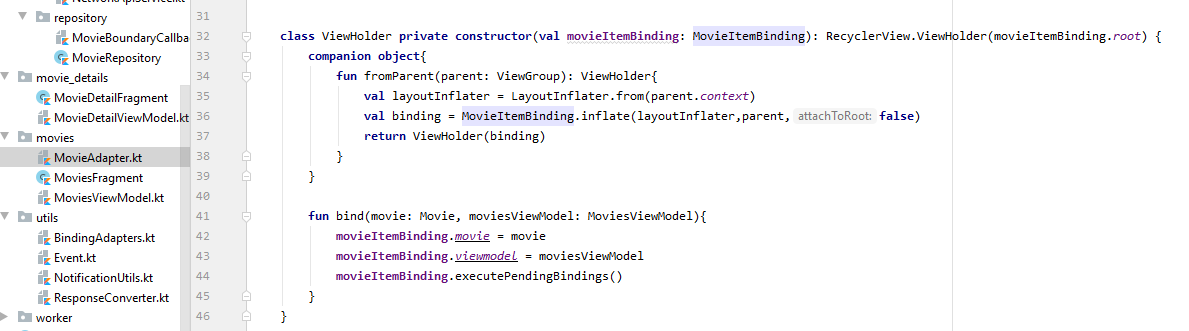
đoạn config dưới vào build.gradle để kích hoạt databinding:

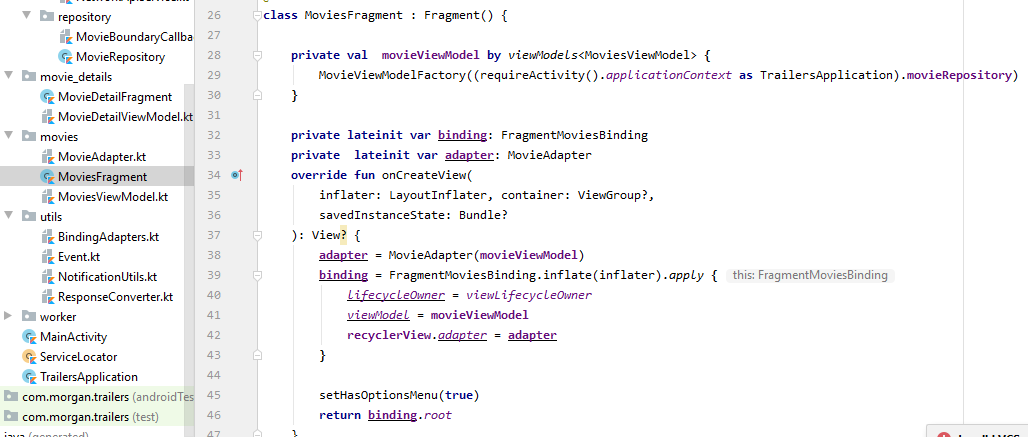


Thư viện Data Binding tự động tạo ra các lớp cần thiết để liên kết đến view trong layout với các dữ liệu trong priject.Đối với các layout binding file chúng ta thêm các thẻ root Layout theo sau là một thẻ data nằm trong view root như sau:









**2. LiveData**

Là một lớp , nó dùng để truyền tải các thông điệp về dữ liệu, dựa trên mô hình của Observer. Sử dụng LiveData để cập nhật dữ liệu theo thời gian thực cho UI

