**FRAGMENT – DOCUMENTATION**

**Giới thiệu**

- Fragment là một thành phần UI linh hoạt trong Android, giúp chia nhỏ giao diện và logic của ứng dụng thành các phần độc lập, tái sử dụng được trong nhiều Activity hoặc Fragment khác.

- Fragment có vòng đời riêng nhưng luôn nằm trong một Activity, giúp quản lý UI hiệu quả hơn, đặc biệt trong các ứng dụng có giao diện phức tạp.

**0. Thiết lập môi trường của bạn:**

- Tìm đến thư mục gradle/ trong dự án của bạn và mở file libs.versions.toml.

- Thêm Fragment vào versions và libraries.

|  |
| --- |
| [versions]  fragment = "1.6.2" # Phiên bản mới nhất của Fragment  [libraries]  androidx-fragment = { module = "androidx.fragment:fragment", version.ref = "fragment" } |

- Mở file **build.gradle.kts (Module: app)**, thêm Fragment vào dependencies:

|  |
| --- |
| dependencies {  implementation(libs.androidx.fragment)  } |

**1. Cách Khởi Tạo Fragment**

**- Tạo Fragment cơ bản:**

**+** Mỗi Fragment là một class mở rộng từ Fragment.

+ Cách tạo một Fragment đơn giản:

|  |
| --- |
| public class ExampleFragment extends Fragment {  @Nullable  @Override  public View onCreateView(@NonNull LayoutInflater inflater,  @Nullable ViewGroup container,  @Nullable Bundle savedInstanceState) {  return inflater.inflate(R.layout.fragment\_example, container, false);  }  } |

- Giải thích:

+ onCreateView(): Tạo giao diện của Fragment từ một file XML.

**2. Thêm Fragment vào Activity**

- Có hai cách để thêm Fragment vào một Activity:

+ Thêm Fragment bằng XML (Fragment tĩnh)

|  |
| --- |
| <fragment  android:id="@+id/fragmentExample"  android:name="com.example.ExampleFragment"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content" /> |

+ Thêm Fragment bằng Code (Fragment động)

|  |
| --- |
| FragmentManager fragmentManager = getSupportFragmentManager();  FragmentTransaction transaction = fragmentManager.beginTransaction();  transaction.add(R.id.fragment\_container, new ExampleFragment());  transaction.commit(); |

- Giải thích:

+ FragmentManager: Quản lý các Fragment trong Activity.

+ FragmentTransaction: Xử lý thêm, xóa, thay thế Fragment.

**3. Vòng Đời Fragment**

- Fragment có vòng đời độc lập nhưng phụ thuộc vào Activity.

- Vòng đời chính:

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| onAttach() | Khi Fragment được gắn vào Activity |
| onCreate() | Khi Fragment được tạo lần đầu |
| onCreateView() | Tạo giao diện UI |
| onViewCreated() | Khởi tạo các thành phần bên trong View |
| onStart() | Khi Fragment hiển thị trên màn hình |
| onResume() | Khi Fragment có thể tương tác với người dùng |
| onPause() | Khi Fragment bị tạm dừng |
| onStop() | Khi Fragment không còn hiển thị |
| onDestroyView() | Hủy View nhưng Fragment vẫn tồn tại |
| onDestroy() | Hủy Fragment |
| onDetach() | Khi Fragment tách khỏi Activity |

- Ví dụ về vòng đời

|  |
| --- |
| @Override  public void onStart() {  super.onStart();  Log.d("FragmentLifecycle", "Fragment is visible");  } |

- Ứng dụng: Giúp kiểm soát trạng thái của Fragment trong ứng dụng.

**4. Giao Tiếp Giữa Fragment và Activity**

- Fragment có thể giao tiếp với Activity thông qua Interface.

- Ví dụ:

+ Khai báo Interface trong Fragment

|  |
| --- |
| public interface OnFragmentInteractionListener {  void onMessageSent(String message);  }  public class ExampleFragment extends Fragment {  private OnFragmentInteractionListener listener;  @Override  public void onAttach(@NonNull Context context) {  super.onAttach(context);  if (context instanceof OnFragmentInteractionListener) {  listener = (OnFragmentInteractionListener) context;  } else {  throw new RuntimeException(context.toString() + " must implement OnFragmentInteractionListener");  }  }  public void sendMessage() {  listener.onMessageSent("Hello from Fragment!");  }  } |

+ Cài đặt Interface trong Activity

|  |
| --- |
| public class MainActivity extends AppCompatActivity implements OnFragmentInteractionListener {  @Override  public void onMessageSent(String message) {  Toast.makeText(this, "Received: " + message, Toast.LENGTH\_SHORT).show();  }  } |

- Ứng dụng: Truyền dữ liệu từ Fragment sang Activity.

**5. Giao Tiếp Giữa Các Fragment**

- Sử dụng ViewModel để chia sẻ dữ liệu giữa các Fragment.

- Tạo ViewModel

|  |
| --- |
| public class SharedViewModel extends ViewModel {  private final MutableLiveData<String> message = new MutableLiveData<>();  public void sendMessage(String msg) {  message.setValue(msg);  }  public LiveData<String> getMessage() {  return message;  }  } |

- Gửi dữ liệu từ Fragment A

|  |
| --- |
| sharedViewModel.sendMessage("Hello from Fragment A"); |

- Nhận dữ liệu ở Fragment B

|  |
| --- |
| sharedViewModel.getMessage().observe(getViewLifecycleOwner(), message -> textView.setText(message)); |

- Ứng dụng: Giúp Fragment giao tiếp mà không phụ thuộc trực tiếp vào nhau.

**6. Thay Thế Fragment**

- Dùng replace() để thay thế Fragment trong Activity.

- Ví dụ

|  |
| --- |
| FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();  transaction.replace(R.id.fragment\_container, new NewFragment());  transaction.addToBackStack(null);  transaction.commit(); |

- Ứng dụng: Cho phép người dùng quay lại Fragment trước đó bằng nút "Back".

**7. Sử Dụng Fragment trong ViewPager**

- Fragment thường dùng trong ViewPager2 để tạo giao diện vuốt ngang.

- Tạo Adapter cho ViewPager2

|  |
| --- |
| public class ViewPagerAdapter extends FragmentStateAdapter {  public ViewPagerAdapter(FragmentActivity activity) {  super(activity);  }  @NonNull  @Override  public Fragment createFragment(int position) {  return new ExampleFragment();  }  @Override  public int getItemCount() {  return 3; // Số lượng Fragment  }  } |

- Sử dụng trong Activity

|  |
| --- |
| ViewPager2 viewPager = findViewById(R.id.viewPager);  viewPager.setAdapter(new ViewPagerAdapter(this)); |

- Ứng dụng: Dùng cho màn hình TabLayout hoặc Carousel.

**8. Hiển Thị Fragment dưới dạng Dialog**

- Dùng DialogFragment thay thế AlertDialog.

- Ví dụ

|  |
| --- |
| public class MyDialogFragment extends DialogFragment {  @NonNull  @Override  public Dialog onCreateDialog(Bundle savedInstanceState) {  return new AlertDialog.Builder(getActivity())  .setTitle("Thông báo")  .setMessage("Bạn có muốn tiếp tục?")  .setPositiveButton("OK", (dialog, which) -> dismiss())  .setNegativeButton("Hủy", (dialog, which) -> dismiss())  .create();  }  } |

- Ứng dụng: Hiển thị hộp thoại xác nhận hoặc nhập dữ liệu.

Mọi thắc mắc có thể lên tham khảo tại: https://developer.android.com/guide/fragments