

LẬP TRÌNH ANDROID CĂN BẢN

Bài 5: Advanced widgets (p2)

Ths. Trần Xuân Thanh Phúc | Trường Đại học Công Nghiệp Thực Phẩm

Nội dung

- ActionBar & Toolbar
- Navigation Drawer
- Fragment
- Bottom Navigation
- ViewPager

1.1 ActionBar

- Action Bar là một trong những view quan trọng có thể triển khai tới các activity trong ứng dụng.
- Các chức năng chính:
 - Một không gian dành riêng cho ứng dụng, nhận dạng và chỉ ra vị trí người sử dụng đang ở đâu trong ứng dụng.
 - Truy cập tới các hành động (Actions) quan trọng (chẳng hạn như action tìm kiếm).
 - Hỗ trợ điều hướng và chuyển đổi giữa các.

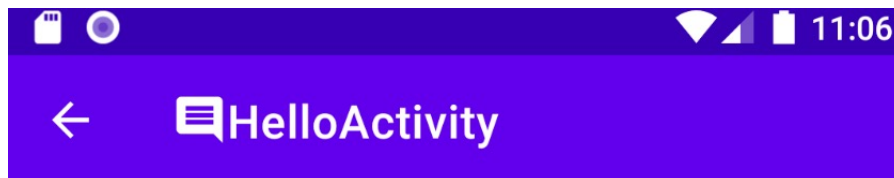
A blue rectangular button with the text "SimpleApp" in white, sans-serif font.

1.1 ActionBar

- Đổi tiêu đề: set **android:label** trong Manifest file, hoặc **getSupportActionBar().setTitle(...)** trong Activity class
- Hiển thị icon:

```
actionBar.setDisplayHomeAsUpEnabled(true);  
actionBar.setLogo(R.mipmap.ic_launcher); //Icon muốn hiện thị  
actionBar.setDisplayUseLogoEnabled(true);
```
- Thêm biểu tượng bấm quay trở lại

```
getSupportActionBar()  
    .setDisplayHomeAsUpEnabled(true);
```



1.1 ActionBar

- Thêm menu vào ActionBar

```
// Menu File
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item android:title="item 1" android:icon="@drawable/google_fit_button" />
    <item android:title="item 2" android:icon="@drawable/ic_baseline_qr_code_scanner_24"
        app:showAsAction="always" />
</menu>
```

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.demo_menu, menu);
    return true;
}
```



1.1 ActionBar

- Bắt sự kiện các action

```
@Override public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case android.R.id.home:  
            onBackPressed();  
            return true;  
        default:break;  
    }  
    return super.onOptionsItemSelected(item);  
}
```

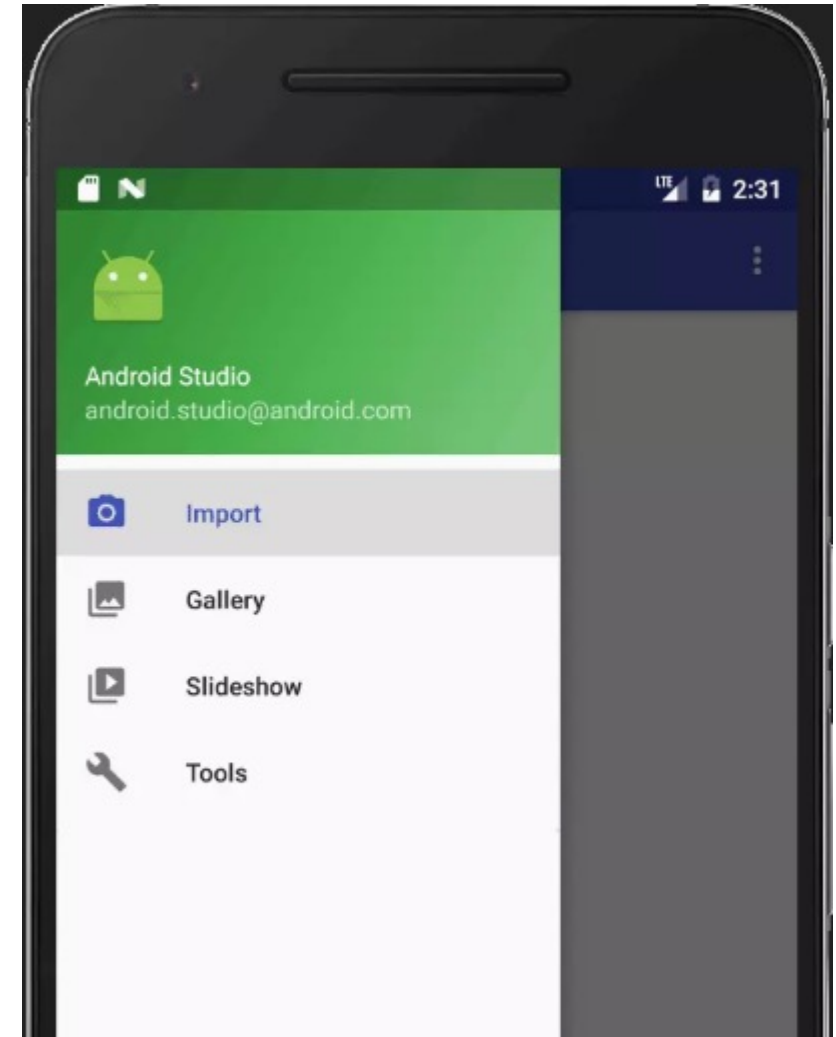
1.2 Toolbar

- Toolbar (`android.support.v7.widget.Toolbar`) được giới thiệu từ API 21, nó như là mở rộng của ActionBar, giúp cho việc tùy biến dễ dàng hơn ActionBar.
- Hủy ActionBar bằng cách thay đổi thông số Theme có `<item name="windowActionBar">false</item>` hoặc sử dụng theme `<prefix>.NoActionBar`
- Chèn Toolbar vào trong file layout, sử dụng `android:fitsSystemWindows="true"` ở layout gốc để Android tính toán chính xác chiều cao khi đã hủy ActionBar

```
Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
setSupportActionBar(toolbar);
```

2. Navigation Drawer

- Navigation Drawer là một menu kéo hiện thị như là một ngăn xếp ở cách cạnh bên của màn hình. Nó được ẩn đi khi không sử dụng, nhưng sẽ xuất hiện khi người sử dụng vuốt ngón tay của họ từ mép trái của màn hình hoặc người sử dụng vuốt từ phía mép trên của màn hình ứng dụng , người sử dụng chạm vào icon trên thanh công cụ.



2. Navigation Drawer

- Để thêm Navigation Drawer , bạn cần khai báo DrawerLayout như là một view gốc. Bên trong thêm view có chứa nội dung chính của layout và một view khác (**NavigationView**) nó chứa nội dung của Navigation Drawer.
- NavigationView phải chỉnh thanh ngang với thuộc tính **android:layout_gravity**
- Thiết lập thuộc tính **android:fitsSystemWindows="true"** để đảm bảo nội dung của Navigation Drawer không được phủ lên thanh trạng thái và cách hệ thống khác.

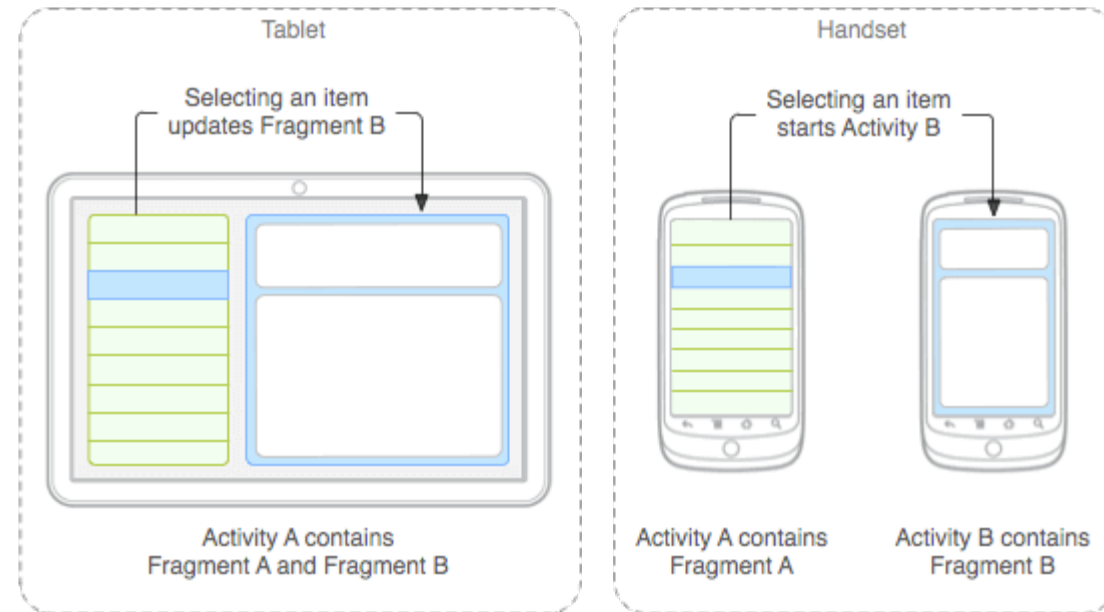
```
<android.support.design.widget.NavigationView  
    android:id="@+id/nav_view"  
    android:layout_width="wrap_content"  
    android:layout_height="match_parent"  
    android:layout_gravity="start"  
    android:fitsSystemWindows="true"  
    app:menu="@menu/drawer_view" />
```

2. Navigation Drawer

- Để nhận sự kiện khi người sử dụng chạm vào của danh sách item trong Drawer , gắn **OnNavigationItemSelectedListener** nó tới NavigationView bằng cách gọi **setNavigationItemSelectedListener()**
- Lắng nghe sự kiện đóng hay mở Drawer bằng cách gọi **addDrawerListener()**

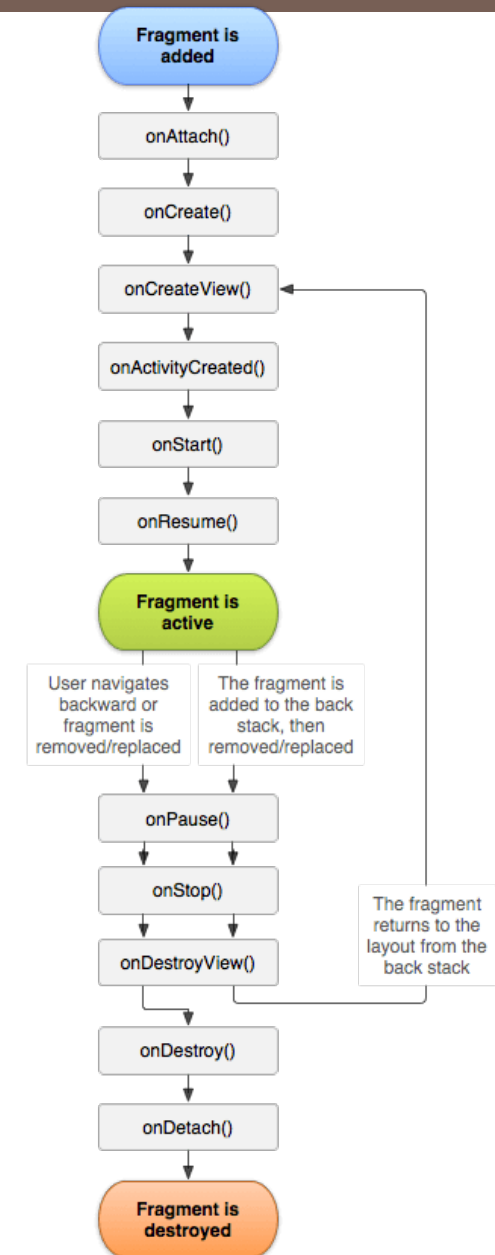
3. Fragment

- Fragment là phần module của thiết kế activity, được sử dụng để thể hiện giao diện người dùng trong một activity. Bằng cách sử dụng các fragment, có thể tạo ra một thiết kế UI linh hoạt, có thể được điều chỉnh dựa trên kích thước màn hình của thiết bị
- Fragment có riêng các phương thức callback trong vòng đời
- Có thể thêm hoặc xóa các fragment trong một activity trong khi activity đang hoạt động



3. Fragment

onAttach ()	Được gọi khi fragment đã được liên kết với một activity.
onCreate ()	Được sử dụng để khởi tạo fragment.
onCreateView()	Được sử dụng để tạo fragment view
onActivityCreated ()	Được gọi khi fragment và view của fragment được khởi tạo, dùng để hoàn thành nốt công đoạn khởi tạo fragment và activity
onStart()	Được gọi để hiển thị fragment.
onResume ()	Fragment hoạt động hoàn toàn trong activity.
onPause ()	Được gọi khi fragment không còn hiển thị và người dùng đang rời khỏi fragment.
onStop ()	Được gọi để dừng fragment.
onDestoryView ()	Giao diện view fragment bị xóa sau khi thực hiện.
onDestroy ()	Được gọi khi hủy fragment.
onDetach ()	Được gọi ngay sau khi fragment bị tách ra khỏi activity.



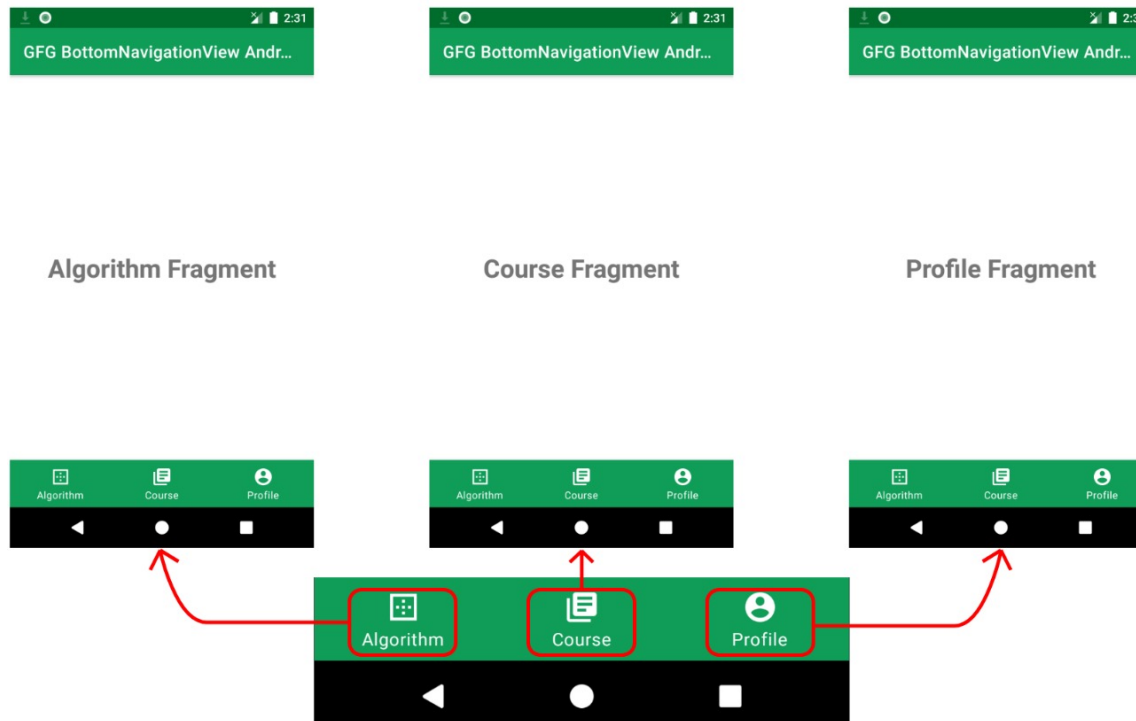
3. Fragment

- FragmentManager: Đây là một lớp dùng để quản lý các Fragment, được tích hợp vào trong mỗi Activity để giúp các Activity có thể dễ dàng để thêm, xóa hoặc thay thế các Fragment ra khỏi một vùng không gian một cách linh động. Lấy đối tượng FragmentManager bằng hàm `getSupportFragmentManager()`

```
public void replaceFragment(Fragment f)
{
    FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
    transaction.replace(R.id.fragment_demo, f); // ID layout muốn chèn fragment
    // transaction.addToBackStack(null);
    transaction.commit();
}
```

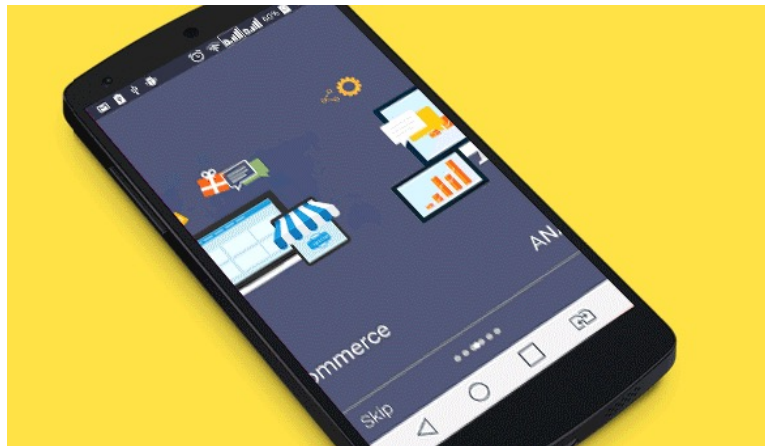
4. Bottom Navigation

- Bottom Navigation thường ở cuối màn hình cung cấp điều hướng giữa các chế độ views ở top-level trong ứng dụng.



5. ViewPager

- Là một layout manager cho phép người dùng duyệt qua từng trang nội dung (fragment) bằng thao tác vuốt
- ViewPager sử dụng **FragmentPagerAdapter** để quản lý dữ liệu, **TabLayout** để triển khai giao diện thông qua hàm **setupWithViewPager(viewPager)**



LẬP TRÌNH ANDROID CĂN BẢN

Kết thúc 🤗

Ths. Trần Xuân Thanh Phúc | Trường Đại học Công Nghiệp Thực Phẩm