



Lập trình Android

Bài 29: *App shortcuts*

Phòng LT & Mạng

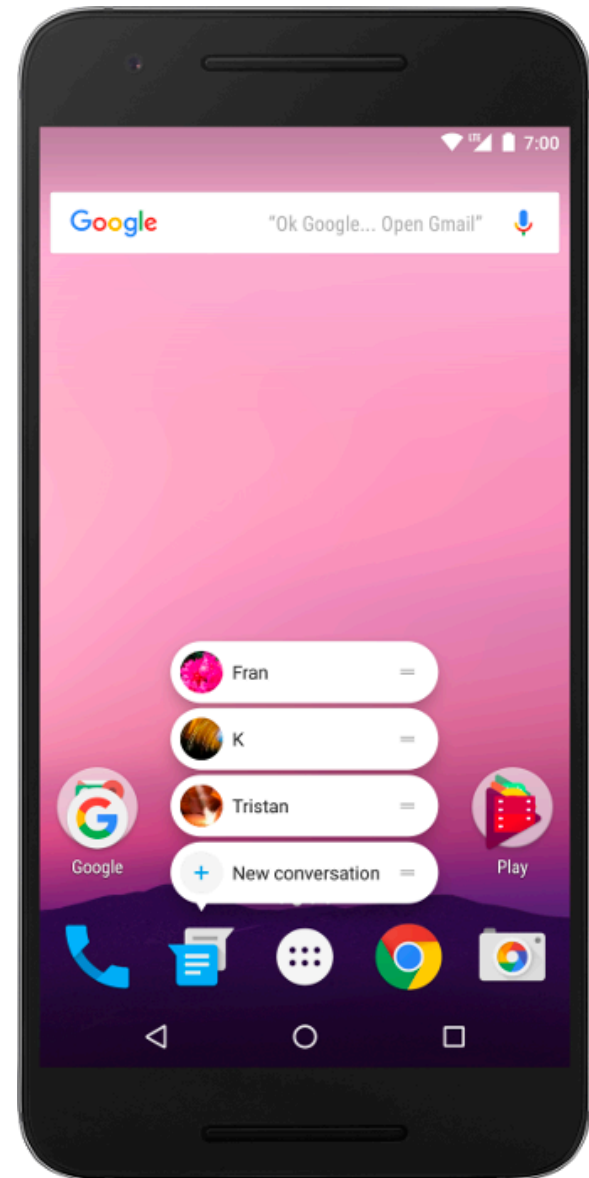
<http://csc.edu.vn/lap-trinh-va-csdl>





Nội dung

1. App shortcuts
2. Static shortcuts
3. Dynamic shortcuts
4. Pinned shortcuts





App shortcuts

- Từ Android 7.1, có thể tạo app **shortcuts** (lối tắt) để người dùng có thể truy cập nhanh đến các thành phần đặc biệt của ứng dụng
- **Shortcuts** hiển thị khi người dùng **long click** lên launcher icon
- Ví dụ shortcut trong một số ứng dụng:
 - Soạn email mới trong email app (Gmail)
 - Mở một vị trí đặc biệt trên bản đồ (Google Maps)
 - Gửi tin nhắn tới bạn bè trong ứng dụng nhắn tin
- Có **3 loại** shortcuts:
 - **Static shortcuts**: giúp link đến các nội dung cố định trong ứng dụng; áp dụng cho các ứng dụng có cấu trúc cố định; không thay đổi được lúc runtime
 - **Dynamic shortcuts**: tùy vào ngữ cảnh mà app cung cấp các shortcuts khác nhau
 - **Pinned shortcuts**: do người dùng thiết lập lúc runtime, hiển thị như launcher icon (chỉ có từ Android 8)



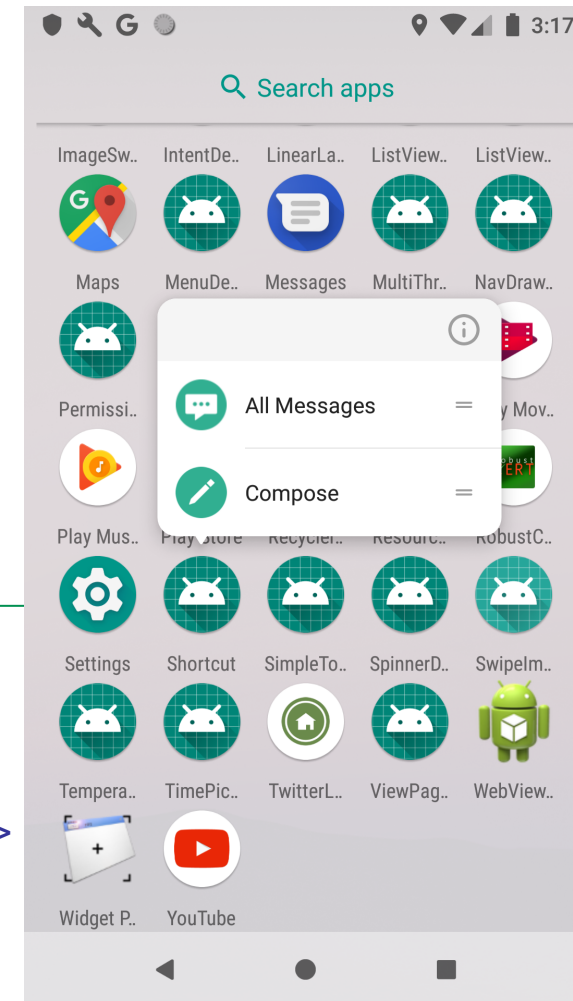


Static shortcuts

- Giúp link đến các action cố định trong app
- Các bước tạo **static shortcuts**:
 - (1) Trong file **AndroidManifest.xml**, tìm activity có intent filters là **android.intent.action.MAIN** action và **android.intent.category.LAUNCHER** category.
 - (2) Thêm thẻ **metadata** khai báo sử dụng shortcuts

```
...  
<activity android:name=".MainActivity">  
  <intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
  </intent-filter>  
  <meta-data android:name="android.app.shortcuts"  
    android:resource="@xml/shortcuts" />  
</activity>  
...
```

AndroidManifest.xml





Static shortcuts (2)

- (3) Tạo resource file: **res/xml/shortcuts.xml**
- (4) Khai báo các **shortcuts** trong file trên:

```
<?xml version="1.0" encoding="utf-8"?>
<shortcuts xmlns:android="http://schemas.android.com/apk/res/android">
  <shortcut
    android:shortcutId="compose"
    android:enabled="true"
    android:icon="@drawable/compose_icon"
    android:shortcutShortLabel="@string/compose_shortcut_short_label"
    android:shortcutLongLabel="@string/compose_shortcut_long_label"
    android:shortcutDisabledMessage="@string/compose_disabled_message">
    <intent
      android:action="android.intent.action.VIEW"
      android:targetPackage="edu.csc.shortcut"
      android:targetClass="edu.csc.shortcut.ComposeActivity" />
    <categories android:name="android.shortcut.conversation" />
  </shortcut>

  <!-- Specify more shortcuts here. -->
</shortcuts>
```

res/xml/shortcuts.xml





Static shortcuts (3)

- Các thuộc tính cần biết của **shortcut**:
 - **android:shortcutId** – String: id của shortcut, bắt buộc khai báo, không dùng @string/xyz
 - **android:shortcutShortLabel** – String: tên viết tắt của shortcut, hạn chế 10 ký tự, bắt buộc sử dụng chuỗi trong strings.xml (@string/xyz)
 - **android:shortcutLongLabel** – String: tên đầy đủ của shortcut, hạn chế 25 ký tự, bắt buộc sử dụng chuỗi trong strings.xml (@string/xyz). Nếu tên đầy đủ quá dài, tên viết tắt sẽ được hiển thị
 - **android:shortcutDisabledMessage** – string: thông báo sẽ hiện để giải thích nếu người dùng chọn shortcut đã disabled, bắt buộc sử dụng chuỗi trong strings.xml (@string/xyz)
 - **android:enabled** – Boolean: cho biết shortcut có được enabled hay disabled
 - **android:icon** – Drawable: icon của shortcut



Static short (4)

- Các thẻ con cần biết của **shortcut**:
 - **intent**: bắt buộc phải có, trong thẻ này bắt buộc phải có thuộc tính android:action
 - **categories**: nhóm các shortcut theo loại



Dynamic shortcuts

- Giúp link đến các action trong app, thay đổi theo ngữ cảnh. Ví dụ:
 - Gọi điện thoại đến 1 người cụ thể
 - Mở bản đồ với một vị trí xác định
 - Load game đã chơi ở tương ứng với lần save game trước đó
- API **ShortcutManager** cung cấp các phương thức sau:
 - **setDynamicShortcuts()**: định nghĩa lại toàn bộ danh sách shortcuts, hoặc **addDynamicShortcuts()** để thêm shortcuts vào danh sách hiện tại
 - **updateShortcuts()**: cập nhật danh sách shortcuts
 - **removeDynamicShortcuts()**: để xóa các shortcuts, hoặc **removeAllDynamicShortcuts()** để xóa toàn bộ shortcuts đang có



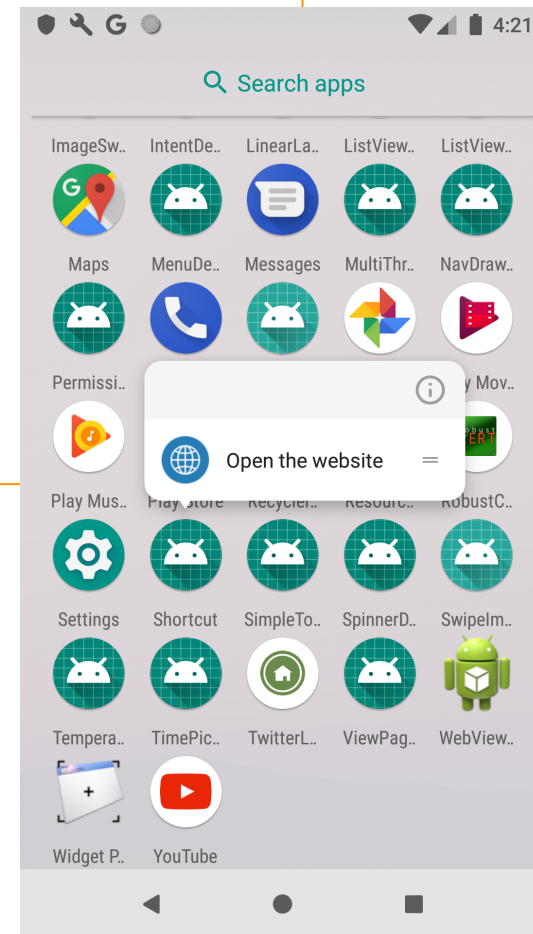
Dynamic shortcuts (2)

- Minh họa sau tạo 1 dynamic shortcut, giúp mở một website và đưa vào danh sách shortcuts của ứng dụng

```
ShortcutManager shortcutManager = context.getSystemService(ShortcutManager.class);
```

```
ShortcutInfo shortcut = new ShortcutInfo.Builder(context, "openWebsite")  
    .setShortLabel("Website")  
    .setLongLabel("Open the website")  
    .setIcon(Icon.createWithResource(context, R.drawable.icon_website))  
    .setIntent(new Intent(Intent.ACTION_VIEW,  
        Uri.parse("https://csc.edu.vn")))  
    .build();
```

```
shortcutManager.setDynamicShortcuts(Arrays.asList(shortcut));
```

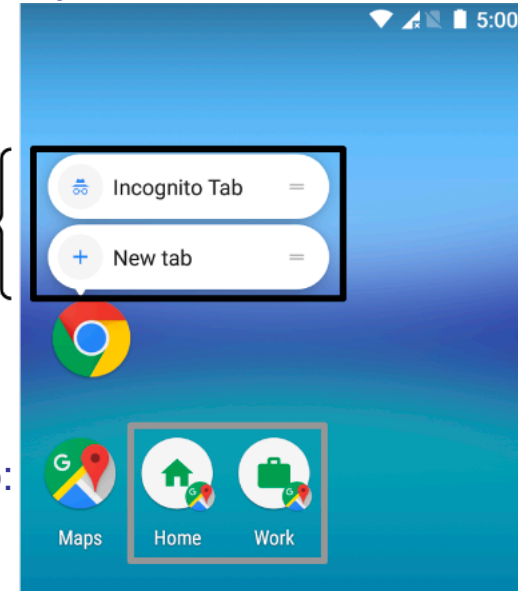




Pinned shortcuts

- Từ Android 8 (API level 26 hoặc cao hơn), có thể tạo được pinned shortcuts
- Pinned shortcuts hiển thị thành các launcher icon
- Các bước để pin một shortcut như sau:
 - (1) Dùng **isRequestPinShortcutSupported()** để kiểm tra thiết bị có hỗ trợ pin shortcuts hay không?
 - (2) Tạo đối tượng **shortcut info**. Có 2 cách tạo shortcut info:
 - (a) Nếu shortcut đã tồn tại → tạo shortcut info tương ứng với id của shortcut đó
 - (b) Nếu shortcut chưa tồn tại → tạo shortcut info tương ứng với các thông tin: id, intent, short label, long label, icon...
 - (3) Pin shortcut bằng phương thức **requestPinShortcut()**. Cần truyền đối tượng PendingIntent để thông báo cho ứng dụng sau khi shortcut được pin thành công

App shortcuts



Pinned shortcuts



Pinned shortcuts (2)

- Ví dụ: pin "openWebsite" shortcut ở mục dynamic shortcuts trên

```
ShortcutManager shortcutManager =  
    context.getSystemService(ShortcutManager.class);  
  
if (shortcutManager.isRequestPinShortcutSupported()) {  
    ShortcutInfo pinShortcutInfo =  
        new ShortcutInfo.Builder(context, "openWebsite").build();  
  
    Intent pinnedShortcutCallbackIntent =  
        shortcutManager.createShortcutResultIntent(pinShortcutInfo);  
  
    PendingIntent successCallback  
        = PendingIntent.getBroadcast(context, REQUEST_CODE,  
                                     pinnedShortcutCallbackIntent, FLAG);  
  
    shortcutManager.requestPinShortcut(pinShortcutInfo,  
        successCallback.getIntentSender());  
}
```

