App Shortcuts

The Android 7.1.1 release allows developers to define action-specific shortcuts in their apps that can be displayed in a launcher. These app shortcuts (https://developer.android.com/guide/topics/ui/shortcuts.html) let users quickly start common or recommended tasks within an app.

Overview

Each shortcut references an intent that launches a specific action in the app when users select the shortcut. Examples of actions you can express as app shortcuts include:

- Navigating users to a particular location in a mapping app
- Sending messages to a friend in a communication app
- Playing the next episode of a TV show in a media app
- Loading the last save point in a gaming app

Examples and source

You can find the primary implementation of this feature in the following files:

```
frameworks/base/services/core/java/com/android/server/policy/ShortcutManager.java
frameworks/base/services/core/java/com/android/server/pm/ShortcutPackage.java
frameworks/base/services/core/java/com/android/server/pm/ShortcutUser.java
frameworks/base/services/core/java/com/android/server/pm/ShortcutPackageInfo.java
frameworks/base/services/core/java/com/android/server/pm/ShortcutLauncher.java
frameworks/base/services/core/java/com/android/server/pm/ShortcutParser.java
frameworks/base/services/core/java/com/android/server/pm/ShortcutService.java
frameworks/base/services/core/java/com/android/server/pm/LauncherAppsService.java
frameworks/base/services/core/java/com/android/server/pm/ShortcutPackageItem.java
frameworks/base/core/java/com/android/server/backup/ShortcutBackupHelper.java
frameworks/base/core/java/android/content/pm/ShortcutManager.java
frameworks/base/core/java/android/content/pm/ShortcutServiceInternal.java
frameworks/base/core/java/android/content/pm/ShortcutInfo.java
frameworks/base/core/java/android/content/pm/LauncherApps.java
```

With the following files providing supporting features (called hidden APIs in `ShortcutManager.java`):

```
packages/apps/Settings/src/com/android/settings/DevelopmentSettings.java
frameworks/base/packages/SystemUI/src/com/android/systemui/statusbar/policy/RemoteInputView.java
```

And, for an example, the Android Open Source Project Launcher version 3 supports shortcuts:

```
packages/apps/Launcher3/
```

Finally, see the following files for public Javadoc.

```
frameworks/base/core/java/android/content/pm/ShortcutManager.java
frameworks/base/core/java/android/content/pm/ShortcutInfo.java
frameworks/base/core/java/android/content/pm/LauncherApps.java
```

Implementation

The AOSP Launcher3 supports shortcuts already. In cases where a partner has its own launcher, that launcher should support shortcuts too.

- When the user performs a certain gesture (e.g. long press) on an app icon, the launcher should show the dynamic and manifest shortcuts associated with each launcher activity icon.
  The shortcut sort order is defined in the ShorctutManager Javadoc within the "Shortcut Display Order" section. For example, show the manifest shortcuts first, then the dynamic shortcuts. The shortcuts are sorted by rank in ascending order within each group.
- The user should be able to drag each dynamic/manifest shortcut and "pin" it to the home screen.

- Pinned shortcuts should be backed up and restored. (See ShortcutManager's javadoc for details)
- Doing an "Inline reply" on Notification should internally call ShortcutManager.onApplicationActive.

In addition, some Google Mobile Services (GMS) apps have shortcuts. The OEM launcher should show shortcuts for them and ideally support "pinning (https://support.google.com/nexus/answer/6118421)" (or creating a shortcut icon) too.

See the Launcher3 source for details on how to interact with the framework for the above operations.

Validation

Use the following Android Compatibility Test Suite (CTS) tests to ensure your version of the feature (ShortcutManager and LauncherApps) works as intended:

```
cts/tests/tests/shortcutmanager/
cts/hostsidetests/shortcuts/
```

And find the unit tests for the AOSP implementation here:

```
frameworks/base/services/tests/servicestests/
```

Which includes:

```
src/com/android/server/pm/ShortcutManagerTest*.java
```

You may also employ the CTS Verifier test for shortcut manager:

```
cts/apps/CtsVerifier/src/com/android/cts/verifier/notifications/ShortcutThrottlingResetActivity.java
```

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 3.0 License (http://creativecommons.org/licenses/by/3.0/)*, and code samples are licensed under the Apache 2.0 License* (http://www.apache.org/licenses/LICENSE-2.0)*. For details, see our Site Policies* (https://developers.google.com/terms/site-policies)*. Java is a registered trademark of Oracle and/or its affiliates.*

Last updated March 27, 2017.