

App Shortcuts in Android



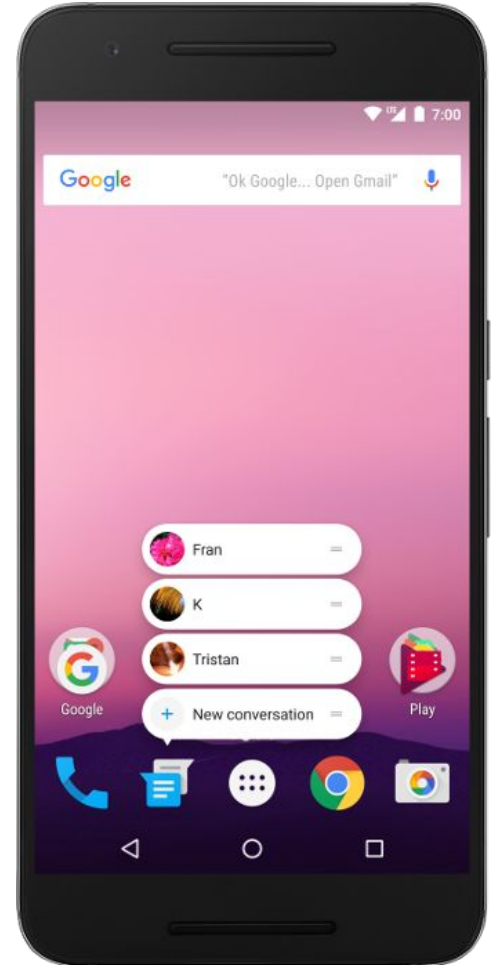
Caren Chang

“People will forget what you said and what you did, but they will never forget how you made them feel.”

- Maya Angelou

What are app shortcuts?

- “Bookmarks” for apps
- But smarter!



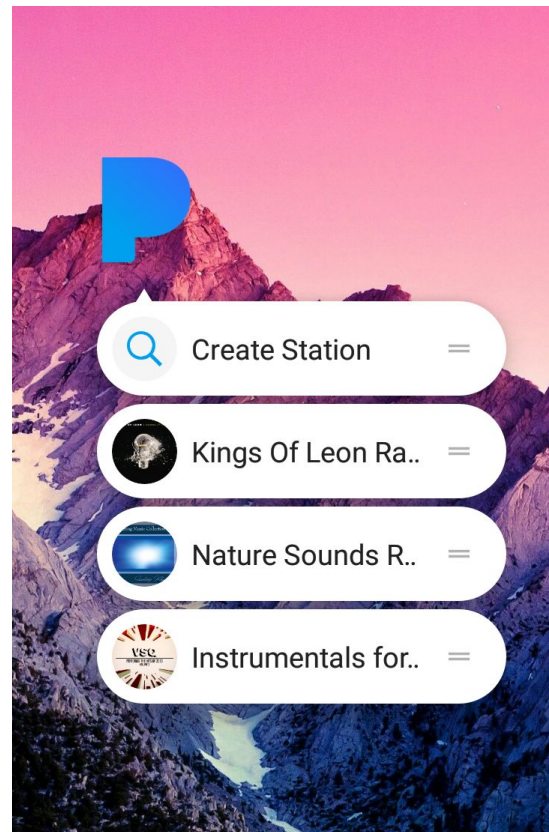
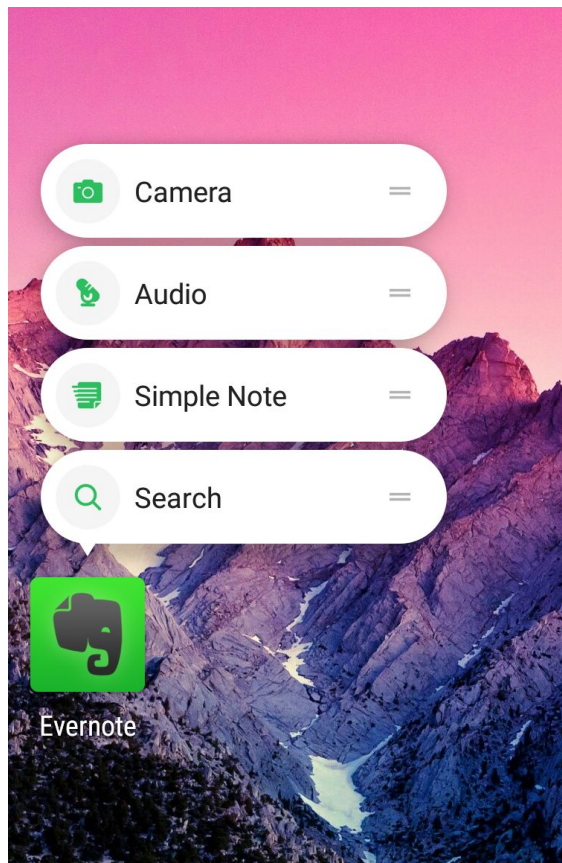
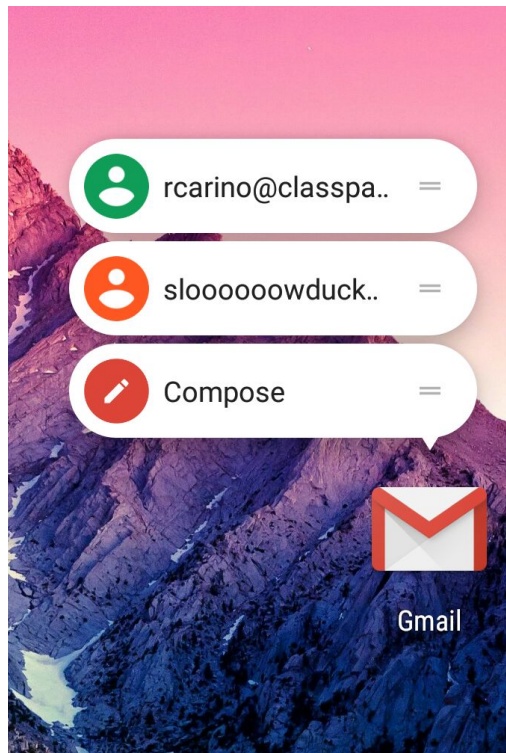
Motivations and Overview

- Easy way to bring user into a specific part of the app
- Simplifies user's interaction and improves experience
- Gives developers more control over the user's attention

In the beginning...







Overview

- Different types of shortcuts
- Examples of how to create shortcuts
- How to make users think your app is magical and amazing



Imagine if we were the Netflix developers..

Imagine if we were the Netflix developers..

- We implemented downloadable content!

Imagine if we were the Netflix developers..

- We implemented downloadable content!
- ... but now user has to launch the Netflix app, scroll around and click random buttons trying to find their downloaded content

Imagine if we were the Netflix developers..

- We implemented downloadable content!
- ... but now user has to launch the Netflix app, scroll around and click random buttons trying to find their downloaded content
- Great! They finally found how to get to their downloaded content

Imagine if we were the Netflix developers..

- We implemented downloadable content!
- ... but now user has to launch the Netflix app, scroll around and click random buttons trying to find their downloaded content
- Great! They finally found how to get to their downloaded content
- One day later, repeat the same sad process

Getting Started

App Shortcuts - Intents

- Shortcuts are mostly programmed through Android Intents
- Android Intents - used to start actions / pass around messages

Android Intents

- Often used to launch activities with the ability to pass around data

```
Intent myIntent = new Intent(Home.this, DetailActivity.class);  
myIntent.putExtra("key", 123); //Optional parameters  
Home.this.startActivity(myIntent);
```

- Intent filters

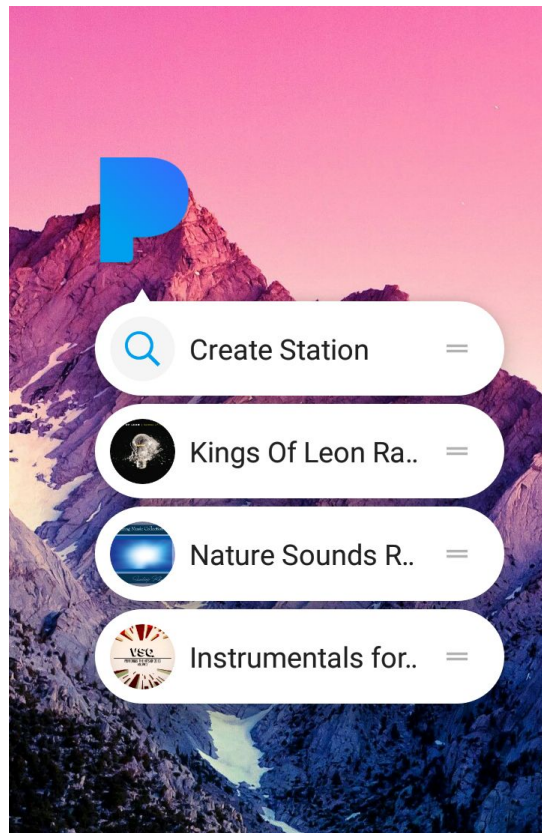
```
<activity android:name=".DetailActivity">  
    <intent-filter>  
        <action android:name="com.example.caren.sushi.DetailShortcut" />  
        <category android:name="android.intent.category.DEFAULT" />  
    </intent-filter>  
</activity>
```

build.gradle

```
android {  
    compileSdkVersion 25  
    buildToolsVersion "25.0.1"  
    defaultConfig {  
        applicationId "com.example.caren.sushi"  
        targetSdkVersion 25  
    }  
}
```

Creating Shortcuts

- Icon
- Long (25 char) / short (10 char) label
- Intent



Static and Dynamic Shortcuts

- **Static:** Defined in a resource file (created at compile time when the APK is made)
- **Dynamic:** published at runtime (can be changed / updated at runtime!)

Static Shortcuts

- Create a new file res/xml/shortcuts.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shortcuts xmlns:android="http://schemas.android.com/apk/res/android">
  <shortcut
    android:enabled="true"
    android:shortcutId="static"
    android:shortcutLongLabel="@string/static_shortcut_long_label"
    android:shortcutShortLabel="@string/static_shortcut_short_label">
    <intent
      android:action="android.intent.action.VIEW"
      android:targetClass="com.example.caren.sushi.HomeActivity"
      android:targetPackage="com.example.caren.sushi" />
    </shortcut>
  </shortcuts>
```

Static Shortcuts

- Create a new file res/xml/shortcuts.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shortcuts xmlns:android="http://schemas.android.com/apk/res/android">
  <shortcut
    android:enabled="true"
    android:shortcutId="static"
    android:shortcutLongLabel="@string/static_shortcut_long_label"
    android:shortcutShortLabel="@string/static_shortcut_short_label">
    <intent
      android:action="android.intent.action.VIEW"
      android:targetClass="com.example.caren.sushi.HomeActivity"
      android:targetPackage="com.example.caren.sushi" />
    </shortcut>
  </shortcuts>
```

Static Shortcuts

- Create a new file res/xml/shortcuts.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shortcuts xmlns:android="http://schemas.android.com/apk/res/android">
  <shortcut
    android:enabled="true"
    android:shortcutId="static"
    android:shortcutLongLabel="@string/static_shortcut_long_label"
    android:shortcutShortLabel="@string/static_shortcut_short_label">
    <intent
      android:action="android.intent.action.VIEW"
      android:targetClass="com.example.caren.sushi.HomeActivity"
      android:targetPackage="com.example.caren.sushi" />
    </shortcut>
  </shortcuts>
```


Defining Static Shortcuts

- set shortcuts file in AndroidManifest.xml

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



Dynamic Shortcuts

Dynamic Shortcuts

```
// creates one dynamic shortcut that opens the home activity
private void createDynamicShortcut() {
    ShortcutManager shortcutManager = getSystemService(ShortcutManager.class);

    ShortcutInfo firstShortcut = new ShortcutInfo.Builder(this, "firstShortcut")
        .setShortLabel(getResources().getString(R.string.dy_shortcut_short))
        .setLongLabel(getResources().getString(R.string.dy_shotcut_long))
        .setIcon(Icon.createWithResource(this, R.drawable.sushi))
        .setIntent(new Intent(HomeActivity.INTENT_FILTER_SHORTCUT))
        .build();

    shortcutManager.setDynamicShortcuts(Arrays.asList(firstShortcut));
}
```

Dynamic Shortcuts

```
// creates one dynamic shortcut that opens the home activity
private void createDynamicShortcut() {
    ShortcutManager shortcutManager = getSystemService(ShortcutManager.class);

    ShortcutInfo firstShortcut = new ShortcutInfo.Builder(this, "firstShortcut")
        .setShortLabel(getResources().getString(R.string.dy_shortcut_short))
        .setLongLabel(getResources().getString(R.string.dy_shortcut_long))
        .setIcon(Icon.createWithResource(this, R.drawable.sushi))
        .setIntent(new Intent(HomeActivity.INTENT_FILTER_SHORTCUT))
        .build();

    shortcutManager.setDynamicShortcuts(Arrays.asList(firstShortcut));
}
```

Dynamic Shortcuts

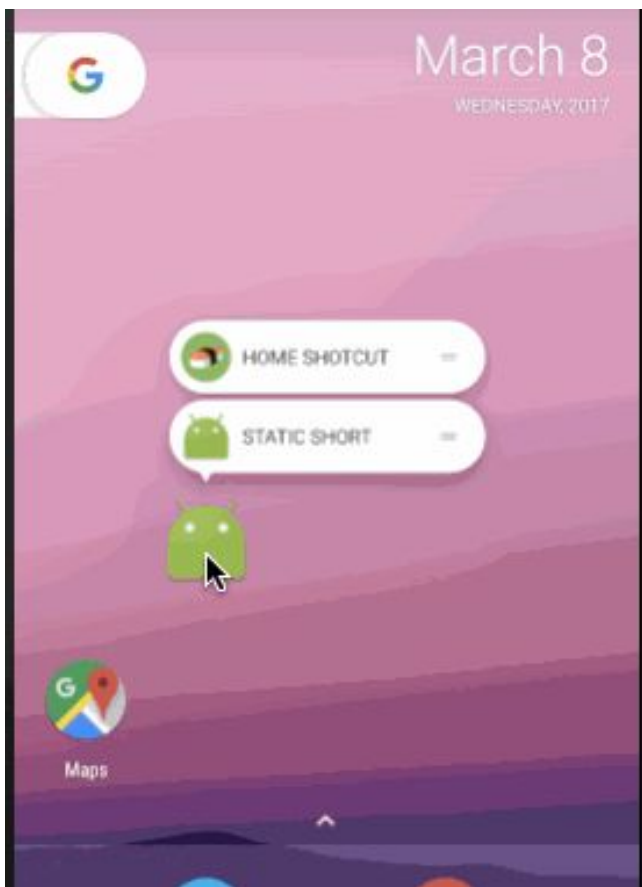
```
// creates one dynamic shortcut that opens the home activity
private void createDynamicShortcut() {
    ShortcutManager shortcutManager = getSystemService(ShortcutManager.class);

    ShortcutInfo firstShortcut = new ShortcutInfo.Builder(this, "firstShortcut")
        .setShortLabel(getResources().getString(R.string.dy_shortcut_short))
        .setLongLabel(getResources().getString(R.string.dy_shotcut_long))
        .setIcon(Icon.createWithResource(this, R.drawable.sushi))
        .setIntent(new Intent(HomeActivity.INTENT_FILTER_SHORTCUT))
        .build();

    shortcutManager.setDynamicShortcuts(Arrays.asList(firstShortcut));
}
```

Dynamic Shortcuts

```
public class HomeActivity extends Activity {  
  
    ...  
  
    @Override  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.activity_home);  
  
        createDynamicShortcuts();  
  
    }
```



But this isn't very exciting..

Changing Shortcuts - Simple Example

- We have a shortcut to start HomeActivity
- If the shortcut was utilized, update the shortcut to open DetailActivity

Changing Shortcuts - Simple Example

```
private void updateDynamicShortcut() {  
    // if we launched this activity from the app shortcut, update the dynamic shortcut  
    if (getIntent().getBooleanExtra(HomeActivity.SHORTCUT_REFERRER_INTENT_KEY, false)) {  
        ShortcutManager shortcutManager = getSystemService(ShortcutManager.class);  
  
        Intent intent = new Intent(SecondDetailActivity.INTENT_FILTER_SHORTCUT);  
  
        ShortcutInfo secondDetailShortcut = new ShortcutInfo.Builder(this, "s2")  
            .setShortLabel(getResources().getString(R.string.detail2_short))  
            .setLongLabel(getResources().getString(R.string.detail2_long))  
            .setIcon(Icon.createWithResource(this, R.drawable.hamburger))  
            .setIntent(intent)  
            .build();  
  
        shortcutManager.removeAllDynamicShortcuts();  
        shortcutManager.addDynamicShortcuts(Arrays.asList(secondDetailShortcut));  
    }  
}
```

Changing Shortcuts - Simple Example

```
private void updateDynamicShortcut() {  
    // if we launched this activity from the app shortcut, update the dynamic shortcut  
    if (getIntent().getBooleanExtra(HomeActivity.SHORTCUT_REFERRER_INTENT_KEY, false)) {  
        ShortcutManager shortcutManager = getSystemService(ShortcutManager.class);  
  
        Intent intent = new Intent(SecondDetailActivity.INTENT_FILTER_SHORTCUT);  
  
        ShortcutInfo secondDetailShortcut = new ShortcutInfo.Builder(this, "s2")  
            .setShortLabel(getResources().getString(R.string.detail2_short))  
            .setLongLabel(getResources().getString(R.string.detail2_long))  
            .setIcon(Icon.createWithResource(this, R.drawable.hamburger))  
            .setIntent(intent)  
            .build();  
  
        shortcutManager.removeAllDynamicShortcuts();  
        shortcutManager.addDynamicShortcuts(Arrays.asList(secondDetailShortcut));  
    }  
}
```

Changing Shortcuts - Simple Example

```
private void updateDynamicShortcut() {  
    // if we launched this activity from the app shortcut, update the dynamic shortcut  
    if (getIntent().getBooleanExtra(HomeActivity.SHORTCUT_REFERRER_INTENT_KEY, false)) {  
        ShortcutManager shortcutManager = getSystemService(ShortcutManager.class);  
  
        Intent intent = new Intent(SecondDetailActivity.INTENT_FILTER_SHORTCUT);  
  
        ShortcutInfo secondDetailShortcut = new ShortcutInfo.Builder(this, "s2")  
            .setShortLabel(getResources().getString(R.string.detail2_short))  
            .setLongLabel(getResources().getString(R.string.detail2_long))  
            .setIcon(Icon.createWithResource(this, R.drawable.hamburger))  
            .setIntent(intent)  
            .build();  
  
        shortcutManager.removeAllDynamicShortcuts();  
        shortcutManager.addDynamicShortcuts(Arrays.asList(secondDetailShortcut));  
    }  
}
```

Changing Shortcuts - Simple Example

```
private void updateDynamicShortcut() {  
    // if we launched this activity from the app shortcut, update the dynamic shortcut  
    if (getIntent().getBooleanExtra(HomeActivity.SHORTCUT_REFERRER_INTENT_KEY, false)) {  
        ShortcutManager shortcutManager = getSystemService(ShortcutManager.class);  
  
        Intent intent = new Intent(SecondDetailActivity.INTENT_FILTER_SHORTCUT);  
  
        ShortcutInfo secondDetailShortcut = new ShortcutInfo.Builder(this, "s2")  
            .setShortLabel(getResources().getString(R.string.detail2_short))  
            .setLongLabel(getResources().getString(R.string.detail2_long))  
            .setIcon(Icon.createWithResource(this, R.drawable.hamburger))  
            .setIntent(intent)  
            .build();  
  
        shortcutManager.removeAllDynamicShortcuts();  
        shortcutManager.addDynamicShortcuts(Arrays.asList(secondDetailShortcut));  
    }  
}
```



Creating a good user experience

- App shortcuts is much like deep linking
- Brings users into a specific part of the app

Imagine we're back to being Netflix developers

- We created an app shortcut to launch the user directly to their downloaded content!

Imagine we're back to being Netflix developers

- We created an app shortcut to launch the user directly to their downloaded content!
- But users are finicky, they don't always know what they want

Imagine we're back to being Netflix developers

- We created an app shortcut to launch the user directly to their downloaded content!
- But users are finicky, they don't always know what they want
- So a user goes to their downloaded content through the app shortcut

Imagine we're back to being Netflix developers

- We created an app shortcut to launch the user directly to their downloaded content!
- But users are finicky, they don't always know what they want
- So a user goes to their downloaded content through the app shortcut
 - **Decides, nope, I actually want to see what else is available**

Imagine we're back to being Netflix developers

- We created an app shortcut to launch the user directly to their downloaded content!
- But users are finicky, they don't always know what they want
- So a user goes to their downloaded content through the app shortcut
 - Decides, nope, I actually want to see what else is available
 - **Presses the back button**

Imagine we're back to being Netflix developers

- We created an app shortcut to launch the user directly to their downloaded content!
- But users are finicky, they don't always know what they want
- So a user goes to their downloaded content through the app shortcut
 - Decides, nope, I actually want to see what else is available
 - Presses the back button
 - **Ends up exiting the app**

Imagine we're back to being Netflix developers

- We created an app shortcut to launch the user directly to their downloaded content!
- But users are finicky, they don't always know what they want
- So a user goes to their downloaded content through the app shortcut
 - Decides, nope, I actually want to see what else is available
 - Presses the back button
 - Ends up exiting the app
 - **The app has now made the user feel sad**

Enter backstacks

Creating back stack (static)

```
<shortcut
  android:enabled="true"
  android:shortcutId="static"
  android:shortcutLongLabel="@string/static_shortcut_long_label"
  android:shortcutShortLabel="@string/static_shortcut_short_label">
  <intent
    android:action="android.intent.action.VIEW"
    android:targetClass="com.example.caren.sushi.HomeActivity"
    android:targetPackage="com.example.caren.sushi" />
  <intent
    android:action="android.intent.action.VIEW"
    android:targetClass="com.example.caren.sushi.DetailActivity"
    android:targetPackage="com.example.caren.sushi" />
</shortcut>
```

Creating back stack (dynamic)

```
ShortcutInfo secondShortcut = new ShortcutInfo.Builder(this, "secondShortcut")
    .setShortLabel("Detail")
    .setLongLabel("Detail Shortcut")
    .setIcon(Icon.createWithResource(this, R.drawable.hamburger))
    .setIntents(
        new Intent[]{
            new Intent(HomeActivity.INTENT_FILTER_SHORTCUT).
                setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK),
            new Intent(DetailActivity.INTENT_FILTER_SHORTCUT)
        })
    .build();
```

Creating back stack (dynamic)

```
ShortcutInfo secondShortcut = new ShortcutInfo.Builder(this, "secondShortcut")
    .setShortLabel("Detail")
    .setLongLabel("Detail Shortcut")
    .setIcon(Icon.createWithResource(this, R.drawable.hamburger))
    .setIntents(
        new Intent[]{
            new Intent(HomeActivity.INTENT_FILTER_SHORTCUT).
                setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK),
            new Intent(DetailActivity.INTENT_FILTER_SHORTCUT)
        })
    .build();
```



The real value of dynamic shortcuts

Updating dynamic shortcuts

- Dynamic shortcuts can be updated at any time!
 - As a result of an user action
 - Result of many user actions
 - In the background
 - From an alarm

When to use Static vs Dynamic?

- Static:
 - Performs specific action that remains the same
 - Should allow the user to access important parts of your app (feed, publish)
 - Examples: setting alarm, view feed
- Dynamic
 - Performs action based on context
 - Context-sensitive actions
 - Example : redial last called person, navigation, viewing ongoing sport scores

If we were Netflix

- Static
 - Downloaded content
- Dynamic
 - Resume episode of last watched show
 - Resume episode of last last watched show
 - Mystery episode of a show!

Bits and Extras

Ordering shortcuts

- App shortcuts can be ordered manually
- Static shortcuts are **ordered by their definition** in shortcuts.xml
- Dynamic shortcuts can be ordered by **setRank()**

Pinning Shortcuts

- Users have the ability to pin shortcuts
 - Shortcut of shortcut!



Deprecating shortcuts

- No control over pinned shortcuts
- Pinned shortcuts cannot be deleted programmatically
 - `setDisabledMessage()`

Informing Users

- Many users won't know about Android's new capabilities
- Mention it in your app release notes
- Create an 'onboarding' banner
- Tell the users when they're in the specific part of the app

Summary

- App shortcuts can help drive engagement and simplify user experience
- There are static and dynamic shortcuts
- Create a good user experience by populating backstacks
- Inform the user of your app's new abilities
- Make users happy

Thanks Boston!



github.com/calren/appShortcuts

carenslaboratory.com/speaking.html



[@calren24](https://twitter.com/calren24)