Handling Features Not Supported on TV

THIS LESSON TEACHES YOU TO

TVs are much different from other Android-powered devices:

- They're not mobile.
- Out of habit, people use them for watching media with little or no interaction.
- People interact with them from a distance.

Because TVs have a different purpose from other devices, they usually don't have hardware features that other Android-powered devices often have. For this reason, the Android system does not support the following features for a TV device:

Hardware	Android feature descriptor
Camera	android.hardware.camera
GPS	android.hardware.location.gps
Microphone	android.hardware.microphone
Near Field Communications (NFC)	android.hardware.nfc
Telephony	android.hardware.telephony
Touchscreen	android.hardware.touchscreen

This lesson shows you how to work around features that are not available on TV by:

- Providing work arounds for some non-supported features.
- Checking for available features at runtime and conditionally activating/deactivating certain code paths based on availability of those features.

Work Around Features Not Supported on TV

Android doesn't support touchscreen interaction for TV devices, most TVs don't have touch screens, and interacting with a TV using a touchscreen is not consistent with the 10 foot environment. For these reasons, users interact with Android-powered TVs using a remote. In consideration of this, ensure that every control in your app can be accessed with the D-pad. Refer back to the previous two lessons Optimizing Layouts for TV (/training/tv/optimizing-layouts-tv.html) and Optimize Navigation for TV (/training/tv/optimizing-navigation-tv.html) for more details on this topic. The Android system assumes that a device has a touchscreen, so if you want your application to run on a TV, you must **explicitly** disable the touchscreen requirement in your manifest file:

```
<uses-feature android:name="android.hardware.touchscreen" android:required="false"/>
```

Although a TV doesn't have a camera, you can still provide a photography-related application on a TV. For example, if you have an app that takes, views and edits photos, you can disable its picture-taking functionality for TVs and still allow users to view and even edit photos. The next section talks about how to deactivate or activate specific functions in the application based on runtime device type detection.

Because TVs are stationary, indoor devices, they don't have built-in GPS. If your application uses location information, allow users to search for a location or use a "static" location provider to get a location from the zip code configured during the TV setup.

```
LocationManager locationManager = (LocationManager) this.getSystemService(Context.LOCA
Location location = locationManager.getLastKnownLocation("static");
Geocoder geocoder = new Geocoder(this);
Address address = null;

try {
  address = geocoder.getFromLocation(location.getLatitude(), location.getLongitude(),
  Log.d("Zip code", address.getPostalCode());

} catch (IOException e) {
  Log.e(TAG, "Geocoder error", e);
}
```

TVs usually don't support microphones, but if you have an application that uses voice control, you can create a mobile device app that takes voice input and then acts as a remote control for a TV.

Check for Available Features at Runtime

To check if a feature is available at runtime, call `hasSystemFeature(String)` `(/reference/android/content/pm/PackageManager.html#hasSystemFeature(java.lang.String))`. This method takes a single argument : a string corresponding to the feature you want to check. For example, to check for touchscreen, use `hasSystemFeature(String)` `(/reference/android/content/pm/PackageManager.html#hasSystemFeature(java.lang.String))` with the argument `FEATURE_TOUCHSCREEN` `(/reference/android/content/pm/PackageManager.html#FEATURE_TOUCHSCREEN)`.

The following code snippet demonstrates how to detect device type at runtime based on supported features:

```
// Check if android.hardware.telephony feature is available.
if (getPackageManager().hasSystemFeature("android.hardware.telephony")) {
   Log.d("Mobile Test", "Running on phone");
// Check if android.hardware.touchscreen feature is available.
} else if (getPackageManager().hasSystemFeature("android.hardware.touchscreen")) {
   Log.d("Tablet Test", "Running on devices that don't support telphony but have a tou
} else {
    Log.d("TV Test", "Running on a TV!");
}
```

This is just one example of using runtime checks to deactivate app functionality that depends on features that aren't available on TVs.