

## Intents in Android

Intents are messaging objects used to request an action from another app component. They are a fundamental aspect of Android development, allowing for communication between different components of an application or between different applications.

### Implicit Intent

- **Definition:** An implicit intent does not specify the component to receive the intent. Instead, it declares a general action to perform, which allows a component from another app to handle it.
- **Usage:** The Android OS determines which component(s) can handle the intent based on the action specified and any data associated with it.

**Example:** If you want to open a webpage, you create an intent with the action ***Intent.ACTION\_VIEW*** and the URL. The OS will then decide which activity (e.g., a web browser) can handle this intent.

**Example xml:**

```
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.example.com"));  
startActivity(intent);
```

### Explicit Intent

- **Definition:** An explicit intent specifies the exact component (activity, service, or broadcast receiver) to handle the intent. This is done by providing the component's class name.
- **Usage:** It is used within the same application when you know the exact activity or service that should respond to the intent.
- **Example:** Starting an activity within the same app by explicitly mentioning its class.

**Example xml:**

```
Intent intent = new Intent(this, TargetActivity.class);  
startActivity(intent);
```

## Intent Filters

- **Definition:** An intent filter is an expression in an app's manifest file that specifies the types of intents the component can respond to.
- **Usage:** By declaring intent filters, an application can specify the intents it can handle. This is used for components like BroadcastReceivers and activities that are intended to be launchable by certain types of intents.
- **Example:** An activity declared as a launcher activity (the main entry point of the application).

### Example Xml:

```
<activity android:name=".MainActivity">  
  <intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
  </intent-filter>  
</activity>
```

## Shared Preferences

- **Definition:** Shared Preferences is a mechanism for storing and retrieving small amounts of primitive data in the form of key-value pairs.
- **Storage:** The data is stored in an XML file in the application's private directory.
- **Use Cases:** It is typically used to store simple data such as user preferences, settings, and configuration values.
- **Security:** Shared Preferences is not suitable for storing sensitive information because it can be easily accessed by rooted devices or by users with physical access to the device.

