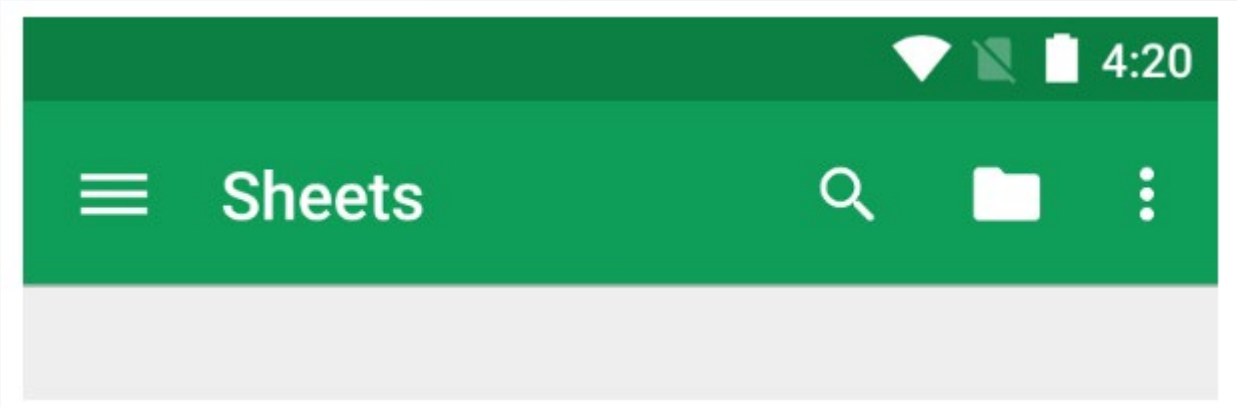
App Bar

App Bar

- The app bar, also known as the action bar, is one of the most important design elements in your app's activities:
  - Provides a visual structure and interactive elements that are familiar to users
  - Makes your app consistent with other Android apps
  - Letting users quickly understand how to operate your app and have a great experience

App Bar

- The key functions of the app bar are as follows:
  - Dedicated space for giving your app an identity and indicating the user's location in the app
  - Predictable access to important actions, such as search
  - Support for navigation and view switching, using tabs or menus



Setting up the App Bar

- Step 1: Add a Toolbar to the activity's layout

```
<androidx.appcompat.widget.Toolbar
  android:id="@+id/my_toolbar"
  android:layout_width="match_parent"
  android:layout_height="?attr/actionBarSize"
  android:background="?attr/colorPrimary"
  android:elevation="4dp"
  android:theme="@style/ThemeOverlay.AppCompat.ActionBar"
  app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>
```

Setting up the App Bar

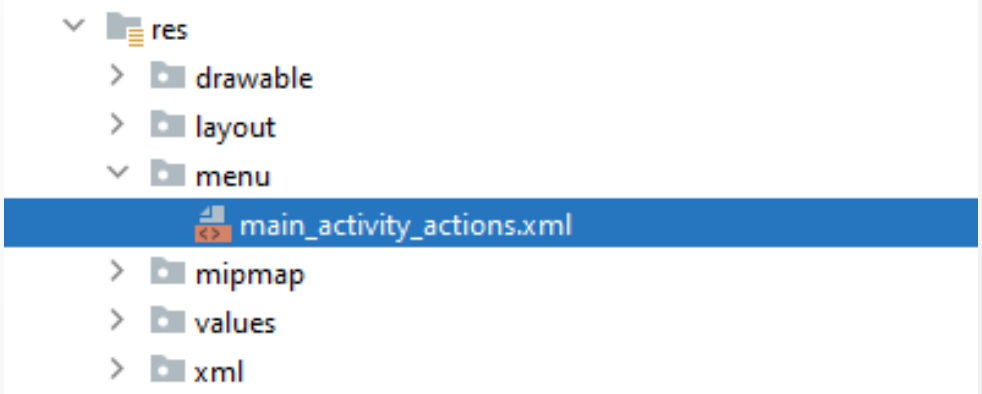
- Step 2: Set the toolbar as the app bar for the activity.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_my);

    Toolbar myToolbar = (Toolbar) findViewById(R.id.my_toolbar);
    setSupportActionBar(myToolbar);
}
```

Setting up the App Bar

- Step 3: Create a menu xml file in res/menu



Setting up the App Bar

- Step 4: Add actions to the menu

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto">
   <item
      android:id="@+id/action_call"
      android:icon="@android:drawable/ic_menu_call"
      android:title="Call"
      app:showAsAction="“always"/>
   <item
      android:id="@+id/action_search"
      android:icon="@android:drawable/ic_menu_search"
      android:title="Search"
      app:showAsAction="ifRoom"/>
   <item android:id="@+id/action_settings"
      android:title="Settings"
      app:showAsAction="never"/>
</menu>
```

Setting up the App Bar

- Step 5: Inflate menu in the toolbar

```
@Override
   public boolean onCreateOptionsMenu(Menu menu) {
      getMenuInflater().inflate(R.menu.main_activity_actions, menu);
      return super.onCreateOptionsMenu(menu);
   }
```

Setting up the App Bar

- Step 6: Respond to actions

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if(id == R.id.action_settings){
        // User chooses the "Settings" item. Show the app settings UI.
        return true;
    }
    if(id == R.id.action_call){
        // User chooses the "call" item. Perform the call.
        return true;
    }
    if(id == R.id.action_search){
        // User chooses the "search" item. Perform the search.
        return true;
    }
    return super.onOptionsItemSelected(item);
}
```

Add an action view

- Step 1: Add a SearchView widget to the app bar

```
<item
    android:id="@+id/action_search"
    android:icon="@android:drawable/ic_menu_search"
    android:title="Search"
    app:showAsAction="ifRoom|collapseActionView"
    app:actionViewClass="androidx.appcompat.widget.SearchView" />
```

Add an action view

- Step 2: Respond to action

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main_activity_actions, menu);

    MenuItem searchItem = menu.findItem(R.id.action_search);
    SearchView searchView =
            (SearchView) searchItem.getActionView();
    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            return false;
        }
        @Override
        public boolean onQueryTextChange(String newText) {
            return false;
        }
    });
    return super.onCreateOptionsMenu(menu);
}
```