## ­­Exercise 4.1: Recipe Book: Add a Menu to Start a new Activity

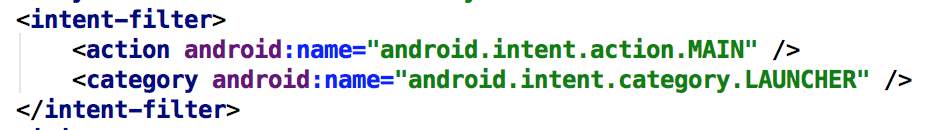
On clicking a menu entry on the *App Bar* in the *RecipeListActvity*, the *SaveRecipeActivity* should be open.

Proceed as follows:

1. Create a menu and a menu entry as shown during the lecture (chapter 4).
2. In the method *RecipeListActivity.onOptionsItemSelected()* create and invoke an *Intent* to start the new *Activity*.

Remark:

* Remember to only set your main activity (*RecipeListActivity*) as  
    
  in *AndroidManifest.xml* if you changed this before.



## Exercise 4.2: Open the Recipe Details activity on item click from the Recipe List

Extend the Recipe List activity by adding an *OnItemClickListener* to the *ListView*. When you click on an item from the list, load the corresponding recipe inside the *RecipeDetailsActivity*. You can do that by passing the recipe’s id (for now the index inside the *List*) as an *Extra* and use it to fetch the *Recipe* object. Remove the Next Recipe *Button,* as well as all code related to it. We won’t be needing it anymore.

## Exercise 4.3: Show Country on Map (Recipe Details)

Extend the *RecipeDetailsActivity*. On click on the country text there should appear a map (in Google Maps or any other maps app) with the country shown.

## Exercise 4.4: Destroy the SaveRecipeActivty after successfully creating a Recipe

After successfully creating a *Recipe*, use the *finish()* method to destroy the *SaveRecipeActivty* and return back to the *RecipeListActivity*.

## Exercise 4.5: Include Share-Button (Recipe Details)

In this exercise you will extend the *RecipeDetailsActivity* with a Share *Button*. The *Recipe* information can be shared with other apps.

For the *App Bar* there is a standard approach to include this functionality:

1. Create a new *Menu* and add the fallowing item:

<**item android:id="@+id/action\_share" android:title="Share"  
 app:showAsAction="ifRoom"  
 app:actionProviderClass="androidx.appcompat.widget.ShareActionProvider"** />

1. **Extend the method** RecipeDetailsActivity.onCreateOptionsMenu() **like what is shown below:**

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.recipe\_details\_menu, menu);

shareItem = menu.findItem(R.id.action\_share);

shareActionProvider = (ShareActionProvider)

MenuItemCompat.getActionProvider(shareItem);

setShareText();

return true;

}

private void setShareText() {

Intent shareIntent = new Intent(Intent.ACTION\_SEND);

shareIntent.setType("text/plain");

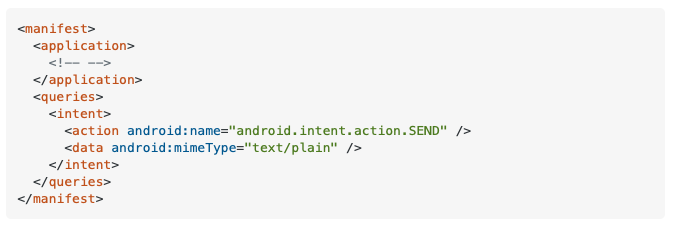
shareActionProvider.setShareIntent(shareIntent);

shareIntent.putExtra(Intent.EXTRA\_TEXT, recipe.toString());

}

Overwrite the *toString()* method of class *Recipe* and choose an appropriate format.

1. Starting with Android 11 you must add the following to AndroidManifest.xml to really see all share targets[[1]](#footnote-1):



1. See <https://stackoverflow.com/a/64777763> [↑](#footnote-ref-1)