Contents

[Class CategoryList 2](#_Toc382268991)

[o Nested Class Summary 3](#_Toc382268992)

[ Nested classes/interfaces inherited from class android.support.v4.app.Fragment 3](#_Toc382268993)

[ Field Detail 3](#_Toc382268994)

[ Constructor Detail 3](#_Toc382268995)

[ Method Detail 3](#_Toc382268996)

[Class CustomActivity 4](#_Toc382268997)

[ Field Detail 5](#_Toc382268998)

[ Constructor Detail 5](#_Toc382268999)

[ Method Detail 5](#_Toc382269000)

[Class CustomFragment 6](#_Toc382269001)

[ Constructor Detail 7](#_Toc382269002)

[ Method Detail 7](#_Toc382269003)

[Class Data 8](#_Toc382269004)

[ Field Detail 8](#_Toc382269005)

[ Constructor Detail 9](#_Toc382269006)

[ Method Detail 9](#_Toc382269007)

[Class DetailActivity 12](#_Toc382269008)

[ Constructor Detail 12](#_Toc382269009)

[ Method Detail 12](#_Toc382269010)

[Class LeftNavAdapter 13](#_Toc382269011)

[ Field Detail 13](#_Toc382269012)

[ Constructor Detail 14](#_Toc382269013)

[ Method Detail 14](#_Toc382269014)

[Class Login 15](#_Toc382269015)

[ Constructor Detail 15](#_Toc382269016)

[ Method Detail 15](#_Toc382269017)

[Class MainActivity 16](#_Toc382269018)

[ Field Detail 16](#_Toc382269019)

[ Constructor Detail 17](#_Toc382269020)

[ Method Detail 17](#_Toc382269021)

[Class RecipeDetail 19](#_Toc382269022)

[ Constructor Detail 20](#_Toc382269023)

[ Method Detail 20](#_Toc382269024)

[Class RecipeList 20](#_Toc382269025)

[o Nested Class Summary 21](#_Toc382269026)

[ Nested classes/interfaces inherited from class android.support.v4.app.Fragment 21](#_Toc382269027)

[ Field Detail 21](#_Toc382269028)

[ Constructor Detail 21](#_Toc382269029)

[ Method Detail 21](#_Toc382269030)

[Class Search 22](#_Toc382269031)

[ Constructor Detail 23](#_Toc382269032)

[ Method Detail 23](#_Toc382269033)

[Class SplashScreen 24](#_Toc382269034)

[ Field Detail 24](#_Toc382269035)

[ Constructor Detail 24](#_Toc382269036)

[ Method Detail 24](#_Toc382269037)

[Class TouchEffect 25](#_Toc382269038)

[ Constructor Detail 25](#_Toc382269039)

[ Method Detail 26](#_Toc382269040)

com.food.ui

## Class CategoryList

public class **CategoryList**

extends [CustomFragment](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomFragment.html)

The Class CategoryList is the Fragment class that is launched when the user clicks on Categories option in Left navigation drawer. It simply display a dummy list of Categories. You need to write actual implementation for loading and displaying Categories.

### Nested Class Summary

|  |  |
| --- | --- |
| Nested Classes | |
| **Modifier and Type** | **Class and Description** |
| private class | [**CategoryList.CategoryAdapter**](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\ui\CategoryList.CategoryAdapter.html)  The Class CategoryAdapter is the adapter class for Categories ListView. |

### Nested classes/interfaces inherited from class android.support.v4.app.Fragment

android.support.v4.app.Fragment.InstantiationException, android.support.v4.app.Fragment.SavedState

### Field Detail

#### catList

private java.util.ArrayList<[Data](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\model\Data.html)> catList

The Category list.

### Constructor Detail

#### CategoryList

public CategoryList()

### Method Detail

#### onCreateView

* + public android.view.View onCreateView(android.view.LayoutInflater inflater,
  + android.view.ViewGroup container,

android.os.Bundle savedInstanceState)

**Overrides:**

onCreateView in class android.support.v4.app.Fragment

#### loadCategoryList

private void loadCategoryList()

This method currently loads a dummy list of Categories. You can write the actual implementation of loading Categories.

#### onCreateOptionsMenu

* + public void onCreateOptionsMenu(android.view.Menu menu,

android.view.MenuInflater inflater)

**Overrides:**

onCreateOptionsMenu in class android.support.v4.app.Fragment

#### onOptionsItemSelected

public boolean onOptionsItemSelected(android.view.MenuItem item)

**Overrides:**

onOptionsItemSelected in class android.support.v4.app.Fragment

com.food.custom

## Class CustomActivity

Direct Known Subclasses:

[DetailActivity](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\DetailActivity.html), [Login](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\Login.html), [MainActivity](file:///D:\\Android\\Work%20Docs\\Template%20Apps\\Food%20Android\\Source%20Code\\Food\\doc\\com\\food\\MainActivity.html" \o "class in com.food), [Search](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\Search.html)

public class **CustomActivity**

extends android.support.v4.app.FragmentActivity

implements android.view.View.OnClickListener

This is a common activity that all other activities of the app can extend to inherit the common behaviors like implementing a common interface that can be used in all child activities.

### Field Detail

#### TOUCH

public static final [TouchEffect](file:///D:\\Android\\Work%20Docs\\Template%20Apps\\Food%20Android\\Source%20Code\\Food\\doc\\com\\food\\utils\\TouchEffect.html" \o "class in com.food.utils) TOUCH

Apply this Constant as touch listener for views to provide alpha touch effect. The view must have a Non-Transparent background.

### Constructor Detail

#### CustomActivity

public CustomActivity()

### Method Detail

#### onCreate

protected void onCreate(android.os.Bundle savedInstanceState)

**Overrides:**

onCreate in class android.support.v4.app.FragmentActivity

#### setupActionBar

protected void setupActionBar()

This method will setup the top title bar (Action bar) content and display values. It will also setup the custom background theme for ActionBar. You can override this method to change the behavior of ActionBar for particular Activity

#### setTouchNClick

public android.view.View setTouchNClick(int id)

Sets the touch and click listener for a view with given id.

**Parameters:**

id - the id

**Returns:**

the view on which listeners applied

#### setClick

public android.view.View setClick(int id)

Sets the click listener for a view with given id.

**Parameters:**

id - the id

**Returns:**

the view on which listener is applied

#### onClick

public void onClick(android.view.View v)

**Specified by:**

onClick in interface android.view.View.OnClickListener

com.food.custom

## Class CustomFragment

Direct Known Subclasses:

[CategoryList](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\ui\CategoryList.html), [RecipeDetail](file:///D:\\Android\\Work%20Docs\\Template%20Apps\\Food%20Android\\Source%20Code\\Food\\doc\\com\\food\\ui\\RecipeDetail.html" \o "class in com.food.ui), [RecipeList](file:///D:\\Android\\Work%20Docs\\Template%20Apps\\Food%20Android\\Source%20Code\\Food\\doc\\com\\food\\ui\\RecipeList.html" \o "class in com.food.ui)

public class **CustomFragment**

extends android.support.v4.app.Fragment

implements android.view.View.OnClickListener

The Class CustomFragment is the base Fragment class. You can extend your Fragment classes with this class in case you want to apply common set of rules for those Fragments.

### Constructor Detail

#### CustomFragment

public CustomFragment()

### Method Detail

#### setTouchNClick

public android.view.View setTouchNClick(android.view.View v)

Set the touch and click listener for a View.

**Parameters:**

v - the view

**Returns:**

the same view

#### onClick

public void onClick(android.view.View v)

**Specified by:**

onClick in interface android.view.View.OnClickListener

com.food.model

**Class Data**

public class Data

extends java.lang.Object

The Class Data is a simple Java Bean that is used to hold Name, Detail and image pairs.

### Field Detail

#### title1

private java.lang.String title1

The title1.

#### title2

private java.lang.String title2

The title2.

#### desc

private java.lang.String desc

The description.

#### image1

private int image1

The image resource id.

#### image2

private int image2

The image2.

### Constructor Detail

#### Data

* + public Data(java.lang.String title1,
  + java.lang.String desc,

int image1)

Instantiates a new feed class.

**Parameters:**

title1 - the title1

desc - the desc

image1 - the image1

#### Data

* + public Data(java.lang.String title1,
  + int image1,

int image2)

Instantiates a new data.

**Parameters:**

title1 - the title1

image1 - the image1

image2 - the image2

### Method Detail

#### getTitle1

public java.lang.String getTitle1()

Gets the title1.

**Returns:**

the title1

#### setTitle1

public void setTitle1(java.lang.String title1)

Sets the title1.

**Parameters:**

title1 - the new title1

#### getTitle2

public java.lang.String getTitle2()

Gets the title2.

**Returns:**

the title2

#### setTitle2

public void setTitle2(java.lang.String title2)

Sets the title2.

**Parameters:**

title2 - the new title2

#### getDesc

public java.lang.String getDesc()

Gets the desc.

**Returns:**

the desc

#### setDesc

public void setDesc(java.lang.String desc)

Sets the desc.

**Parameters:**

desc - the new desc

#### getImage1

public int getImage1()

Gets the image1.

**Returns:**

the image1

#### setImage1

public void setImage1(int image1)

Sets the image1.

**Parameters:**

image1 - the new image1

#### getImage2

public int getImage2()

Gets the image2.

**Returns:**

the image2

#### setImage2

public void setImage2(int image2)

Sets the image2.

**Parameters:**

image2 - the new image2

com.food

## Class DetailActivity

public class **DetailActivity**

extends [CustomActivity](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomActivity.html)

The DetailActivity is the activity class that shows either the Recipe detail fragment or the recipe list fragment based on the parameter passed in intent. This activity is only created to show Back button on ActionBar.

### Constructor Detail

#### DetailActivity

public DetailActivity()

### Method Detail

#### onCreate

protected void onCreate(android.os.Bundle savedInstanceState)

**Overrides:**

[onCreate](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomActivity.html#onCreate(android.os.Bundle)) in class [CustomActivity](file:///D:\\Android\\Work%20Docs\\Template%20Apps\\Food%20Android\\Source%20Code\\Food\\doc\\com\\food\\custom\\CustomActivity.html" \o "class in com.food.custom)

#### addFragment

private void addFragment()

Attach the appropriate fragment with activity based on the 'detail' parameter in Intent.

#### onOptionsItemSelected

public boolean onOptionsItemSelected(android.view.MenuItem item)

**Overrides:**

onOptionsItemSelected in class android.app.Activity

com.food.ui

**Class LeftNavAdapter**

public class LeftNavAdapter

extends android.widget.BaseAdapter

The Adapter class for the ListView displayed in the left navigation drawer.

### Field Detail

#### items

private java.util.ArrayList<[Data](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\model\Data.html)> items

The items.

#### context

private android.content.Context context

The context.

#### selected

private int selected

The selected.

### Constructor Detail

#### LeftNavAdapter

* + public LeftNavAdapter(android.content.Context context,

java.util.ArrayList<[Data](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\model\Data.html)> items)

Instantiates a new left navigation adapter.

**Parameters:**

context - the context of activity

items - the array list of items to be displayed on ListView

### Method Detail

#### setSelection

public void setSelection(int position)

Sets the current selected position for adapter. This position will appear as selected.

**Parameters:**

position - the new selection

#### getCount

public int getCount()

#### getItem

public [Data](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\model\Data.html) getItem(int arg0)

#### getItemId

public long getItemId(int position)

#### getView

* + public android.view.View getView(int position,
  + android.view.View convertView,

android.view.ViewGroup parent)

com.food

## Class Login

public class **Login**

extends [CustomActivity](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomActivity.html)

The Class Login is an Activity class that shows the login screen to users. The current implementation simply start the MainActivity. You can write your own logic for actual login and for login using Facebook and Twitter.

### Constructor Detail

#### Login

public Login()

### Method Detail

#### onCreate

protected void onCreate(android.os.Bundle savedInstanceState)

**Overrides:**

[onCreate](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomActivity.html#onCreate(android.os.Bundle)) in class [CustomActivity](file:///D:\\Android\\Work%20Docs\\Template%20Apps\\Food%20Android\\Source%20Code\\Food\\doc\\com\\food\\custom\\CustomActivity.html" \o "class in com.food.custom)

#### onClick

public void onClick(android.view.View v)

**Specified by:**

onClick in interface android.view.View.OnClickListener

**Overrides:**

[onClick](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomActivity.html#onClick(android.view.View)) in class [CustomActivity](file:///D:\\Android\\Work%20Docs\\Template%20Apps\\Food%20Android\\Source%20Code\\Food\\doc\\com\\food\\custom\\CustomActivity.html" \o "class in com.food.custom)

com.food

## Class MainActivity

public class **MainActivity**

extends [CustomActivity](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomActivity.html)

The Class MainActivity is the base activity class of the application. This activity is launched after the Login and it holds all the Fragments used in the app. It also creates the Navigation Drawer on left side.

### Field Detail

#### drawerLayout

private android.support.v4.widget.DrawerLayout drawerLayout

The drawer layout.

#### drawerLeft

private android.view.View drawerLeft

ListView for left side drawer.

#### drawerToggle

private android.support.v4.app.ActionBarDrawerToggle drawerToggle

The drawer toggle.

#### adapter

private [LeftNavAdapter](file:///D:\\Android\\Work%20Docs\\Template%20Apps\\Food%20Android\\Source%20Code\\Food\\doc\\com\\food\\ui\\LeftNavAdapter.html" \o "class in com.food.ui) adapter

The adapter for left navigation ListView.

### Constructor Detail

#### MainActivity

public MainActivity()

### Method Detail

#### onCreate

protected void onCreate(android.os.Bundle savedInstanceState)

**Overrides:**

[onCreate](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomActivity.html#onCreate(android.os.Bundle)) in class [CustomActivity](file:///D:\\Android\\Work%20Docs\\Template%20Apps\\Food%20Android\\Source%20Code\\Food\\doc\\com\\food\\custom\\CustomActivity.html" \o "class in com.food.custom)

#### setupDrawer

private void setupDrawer()

Setup the drawer layout. This method also includes the method calls for setting up the Left side drawer.

#### setupLeftNavDrawer

private void setupLeftNavDrawer()

Setup the left navigation drawer/slider. You can add your logic to load the contents to be displayed on the left side drawer. You can also setup the Header and Footer contents of left drawer if you need them.

#### getDummyLeftNavItems

private java.util.ArrayList<[Data](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\model\Data.html)> getDummyLeftNavItems()

This method returns a list of dummy items for left navigation slider. You can write or replace this method with the actual implementation for list items.

**Returns:**

the dummy items

#### launchFragment

private void launchFragment(int pos)

This method can be used to attach Fragment on activity view for a particular tab position. You can customize this method as per your need.

**Parameters:**

pos - the position of tab selected.

#### setupContainer

private void setupContainer()

Setup the container fragment for drawer layout. The current implementation of this method simply calls launchFragment method for tab position 2 . You can customize this method as per your need to display specific content.

#### setActionBarTitle

private void setActionBarTitle()

Set the action bar title text.

#### onPostCreate

protected void onPostCreate(android.os.Bundle savedInstanceState)

**Overrides:**

onPostCreate in class android.app.Activity

#### onConfigurationChanged

public void onConfigurationChanged(android.content.res.Configuration newConfig)

**Specified by:**

onConfigurationChanged in interface android.content.ComponentCallbacks

**Overrides:**

onConfigurationChanged in class android.support.v4.app.FragmentActivity

#### onOptionsItemSelected

public boolean onOptionsItemSelected(android.view.MenuItem item)

**Overrides:**

onOptionsItemSelected in class android.app.Activity

#### onKeyDown

* + public boolean onKeyDown(int keyCode,

android.view.KeyEvent event)

**Specified by:**

onKeyDown in interface android.view.KeyEvent.Callback

**Overrides:**

onKeyDown in class android.support.v4.app.FragmentActivity

#### onClick

public void onClick(android.view.View v)

**Specified by:**

onClick in interface android.view.View.OnClickListener

**Overrides:**

[onClick](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomActivity.html#onClick(android.view.View)) in class [CustomActivity](file:///D:\\Android\\Work%20Docs\\Template%20Apps\\Food%20Android\\Source%20Code\\Food\\doc\\com\\food\\custom\\CustomActivity.html" \o "class in com.food.custom)

com.food.ui

## Class RecipeDetail

public class **RecipeDetail**

extends [CustomFragment](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomFragment.html)

The Class RecipeDetail is the Fragment class that is launched when the user select a recipe from Recipe List and it simply shows dummy recipe detail text and images. You can customize this to display actual images and text.

### Constructor Detail

#### RecipeDetail

public RecipeDetail()

### Method Detail

#### onCreateView

* + public android.view.View onCreateView(android.view.LayoutInflater inflater,
  + android.view.ViewGroup container,

android.os.Bundle savedInstanceState)

**Overrides:**

onCreateView in class android.support.v4.app.Fragment

#### onCreateOptionsMenu

* + public void onCreateOptionsMenu(android.view.Menu menu,

android.view.MenuInflater inflater)

**Overrides:**

onCreateOptionsMenu in class android.support.v4.app.Fragment

com.food.ui

## Class RecipeList

public class **RecipeList**

extends [CustomFragment](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomFragment.html)

The Class RecipeList is the Fragment class that is launched when the user clicks on Recipe option in Left navigation drawer or when user select a Category from Category list. It simply display a dummy list of Recipes. You need to write actual implementation for loading and displaying Recipes.

### Nested Class Summary

|  |  |
| --- | --- |
| Nested Classes | |
| **Modifier and Type** | **Class and Description** |
| private class | [**RecipeList.RecipeAdapter**](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\ui\RecipeList.RecipeAdapter.html)  The Class RecipeAdapter is the adapter class for Recipes ListView. |

### Nested classes/interfaces inherited from class android.support.v4.app.Fragment

android.support.v4.app.Fragment.InstantiationException, android.support.v4.app.Fragment.SavedState

### Field Detail

#### recipeList

private java.util.ArrayList<[Data](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\model\Data.html)> recipeList

The Activity list.

### Constructor Detail

#### RecipeList

public RecipeList()

### Method Detail

#### onCreateView

* + public android.view.View onCreateView(android.view.LayoutInflater inflater,
  + android.view.ViewGroup container,

android.os.Bundle savedInstanceState)

**Overrides:**

onCreateView in class android.support.v4.app.Fragment

#### loadRecipeList

private void loadRecipeList()

This method currently loads a dummy list of Recipes. You can write the actual implementation of loading Recipes.

#### onCreateOptionsMenu

* + public void onCreateOptionsMenu(android.view.Menu menu,

android.view.MenuInflater inflater)

**Overrides:**

onCreateOptionsMenu in class android.support.v4.app.Fragment

#### onOptionsItemSelected

public boolean onOptionsItemSelected(android.view.MenuItem item)

**Overrides:**

onOptionsItemSelected in class android.support.v4.app.Fragment

com.food

## Class Search

public class **Search**

extends [CustomActivity](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomActivity.html)

The main Search screen, launched when user click on Search button on Top Action bar. Write your code inside onQueryTextSubmit/onQueryTextChange method(s) to perform required work. The current implementation simply search for Applications installed in the phone.

### Constructor Detail

#### Search

public Search()

### Method Detail

#### onCreate

protected void onCreate(android.os.Bundle savedInstanceState)

**Overrides:**

[onCreate](file:///D:\Android\Work%20Docs\Template%20Apps\Food%20Android\Source%20Code\Food\doc\com\food\custom\CustomActivity.html#onCreate(android.os.Bundle)) in class [CustomActivity](file:///D:\\Android\\Work%20Docs\\Template%20Apps\\Food%20Android\\Source%20Code\\Food\\doc\\com\\food\\custom\\CustomActivity.html" \o "class in com.food.custom)

#### onCreateOptionsMenu

public boolean onCreateOptionsMenu(android.view.Menu menu)

**Overrides:**

onCreateOptionsMenu in class android.app.Activity

#### setupSearchView

private void setupSearchView(android.view.Menu menu)

Setup the up search view for ActionBar search. The current implementation simply search for Applications installed in the phone.

**Parameters:**

menu - the ActionBar Menu

#### onOptionsItemSelected

public boolean onOptionsItemSelected(android.view.MenuItem item)

**Overrides:**

onOptionsItemSelected in class android.app.Activity

com.food

**Class SplashScreen**

public class SplashScreen

extends android.app.Activity

The Class SplashScreen will launched at the start of the application. It will be displayed for 3 seconds and than finished automatically and it will also start the next activity of app.

### Field Detail

#### isRunning

private boolean isRunning

Check if the app is running.

### Constructor Detail

#### SplashScreen

public SplashScreen()

### Method Detail

#### onCreate

public void onCreate(android.os.Bundle savedInstanceState)

**Overrides:**

onCreate in class android.app.Activity

#### startSplash

private void startSplash()

Starts the count down timer for 3-seconds. It simply sleeps the thread for 3-seconds.

#### doFinish

private void doFinish()

If the app is still running than this method will start the Login activity and finish the Splash.

#### onKeyDown

* + public boolean onKeyDown(int keyCode,

android.view.KeyEvent event)

**Specified by:**

onKeyDown in interface android.view.KeyEvent.Callback

**Overrides:**

onKeyDown in class android.app.Activity

om.food.utils

**Class TouchEffect**

public class TouchEffect

extends java.lang.Object

implements android.view.View.OnTouchListener

The Class TouchEffect is the implementation of OnTouchListener interface. You can apply this to views mostly Buttons to provide Touch effect and that view must have a valid background. The current implementation simply set Alpha value of View background.

### Constructor Detail

#### TouchEffect

public TouchEffect()

### Method Detail

#### onTouch

* + public boolean onTouch(android.view.View v,

android.view.MotionEvent event)

**Specified by:**

onTouch in interface android.view.View.OnTouchListener