

Contents

Class CategoryList	2
○ Nested Class Summary.....	3
▪ Nested classes/interfaces inherited from class android.support.v4.app.Fragment.....	3
• Field Detail	3
• Constructor Detail.....	3
• Method Detail	3
Class CustomActivity.....	4
• Field Detail	5
• Constructor Detail.....	5
• Method Detail	5
Class CustomFragment	6
• Constructor Detail.....	7
• Method Detail	7
Class Data	8
• Field Detail	8
• Constructor Detail.....	9
• Method Detail	9
Class DetailActivity.....	12
• Constructor Detail.....	12
• Method Detail	12
Class LeftNavAdapter.....	13
• Field Detail	13
• Constructor Detail.....	14
• Method Detail	14
Class Login.....	15
• Constructor Detail.....	15
• Method Detail	15
Class MainActivity	16
• Field Detail	16

- Constructor Detail 17
- Method Detail 17
- Class RecipeDetail 19
 - Constructor Detail 20
 - Method Detail 20
- Class RecipeList 20
 - Nested Class Summary 21
 - Nested classes/interfaces inherited from class android.support.v4.app.Fragment 21
 - Field Detail 21
 - Constructor Detail 21
 - Method Detail 21
- Class Search 22
 - Constructor Detail 23
 - Method Detail 23
- Class SplashScreen 24
 - Field Detail 24
 - Constructor Detail 24
 - Method Detail 24
- Class TouchEffect 25
 - Constructor Detail 25
 - Method Detail 26

com.food.ui

Class CategoryList

```
public class CategoryList
```

extends [CustomFragment](#)

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
	<u><code>CategoryList.CategoryAdapter</code></u>
<code>private class</code>	The Class <code>CategoryAdapter</code> 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> 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](#), [Login](#), [MainActivity](#), [Search](#)

---

```
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 TouchEvent 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](#), [RecipeDetail](#), [RecipeList](#)

---

```
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
```

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](#) in class [CustomActivity](#)

- *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> 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> 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 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
```

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](#) in class [CustomActivity](#)

- *onClick*

```
public void onClick(android.view.View v)
```

**Specified by:**

[onClick](#) in interface [android.view.View.OnClickListener](#)

**Overrides:**

[onClick](#) in class [CustomActivity](#)

com.food

## Class MainActivity

---

```
public class MainActivity
extends CustomActivity
```

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 adapter
```

The adapter for left navigation ListView.



- **Constructor Detail**

- *MainActivity*

```
public MainActivity()
```

- **Method Detail**

- *onCreate*

```
protected void onCreate(android.os.Bundle savedInstanceState)
```

- Overrides:**

[onCreate](#) in class [CustomActivity](#)

- *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> 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

- o **onKeyDown**
- o 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

- o **onClick**
- public void onClick(android.view.View v)

**Specified by:**

onClick in interface android.view.View.OnClickListener

**Overrides:**

[onClick](#) in class [CustomActivity](#)

com.food.ui

## Class RecipeDetail

---

```
public class RecipeDetail
extends CustomFragment
```

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
```

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  
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> 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
```

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](#) in class [CustomActivity](#)

- ***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 be launched at the start of the application. It will be displayed for 3 seconds and then 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`