Contents

com.events.ui

Class About

```
public class About
```

```
extends CustomFragment
```

The Class About is the Fragment class that is launched when the user clicks on About option in Left navigation drawer and it simply shows a dummy text for About. You can customize this to display actual contents.

- **Constructor Detail**
  - ***About***

    ```
    public About()
    ```

- **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`

com.events.calendar

Class CalendarAdapter

```
public class CalendarAdapter
extends android.widget.BaseAdapter
```

The Class CalendarAdapter is the Adapter class for Calendar view to display dates in Grid format.

- **Field Detail**
```

- ***mContext***

  ```
  private android.content.Context mContext
  ```

  The context.

- ***month***

  ```
  private java.util.Calendar month
  ```

  The month.

- ***pmonth***

  ```
  public java.util.GregorianCalendar pmonth
  ```

  The calendar instance for previous month.

- ***pmonthmaxset***

  ```
  public java.util.GregorianCalendar pmonthmaxset
  ```

  Calendar instance for previous month for getting complete view.

- ***selectedDate***

  ```
  private java.util.GregorianCalendar selectedDate
  ```

  The selected date.

- ***firstDay***

  ```
  int firstDay
  ```

  The first day.

- ***maxWeeknumber***

  ```
  int maxWeeknumber
  ```

  The max week number.

- ***maxP***

  ```
  int maxP
  ```

The previous month maximum day.

- ***calMaxP***

  ```
  int calMaxP
  ```

  The Calendar max off days.

- ***lastWeekDay***

  ```
  int lastWeekDay
  ```

  The last week day.

- ***leftDays***

  ```
  int leftDays
  ```

  The left days.

- ***mnthlength***

  ```
  int mnthlength
  ```

  The month length.

- ***itemvalue***

  ```
  java.lang.String itemvalue
  ```

  The current date string.

- ***curentDateString***

  ```
  java.lang.String curentDateString
  ```

  The current date string.

- ***df***

  ```
  java.text.DateFormat df
  ```

  The DateFormat.

- ***items***

```
private java.util.ArrayList<java.lang.String> items
```

The items.

- ***dayString***

```
public static java.util.List<java.lang.String> dayString
```

The day string.

- ***previousView***

```
private android.view.View previousView
```

The previous view.

- **Constructor Detail**
  - ***CalendarAdapter***
  - `public CalendarAdapter(android.content.Context c, java.util.GregorianCalendar monthCalendar)`

Instantiates a new calendar adapter.

Parameters:

`c` - the c

`monthCalendar` - the month calendar

- **Method Detail**
  - ***setItems***

```
public void setItems(java.util.ArrayList<java.lang.String> items)
```

Sets the items.

Parameters:

`items` - the new items

- ***getCount***

```
public int getCount()
```

- ***getItem***

```
public java.lang.Object getItem(int position)
```

- ***getItemId***

```
public long getItemId(int position)
```

- ***getView***
- `public android.view.View getView(int position,`
- `                        android.view.View convertView,`
  `                        android.view.ViewGroup parent)`

- ***setSelected***

```
public android.view.View setSelected(android.view.View view)
```

Sets the selected date.

Parameters:

`view` - the view

Returns:

the view

- ***refreshDays***

```
public void refreshDays()
```

Refresh days.

- ***getMaxP***

```
private int getMaxP()
```

Gets the previous month maximum day.

Returns:

the max day

com.events.calendar

Class CalendarView

```
public class CalendarView
extends CustomFragment
```

The Class CalendarView is Fragment class to hold the Calendar view.

-
  - ## Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
private class	**CalendarView.EventAdapter** The Class EventAdapter is the adapter class that is used show list of Events for a selected date in the ListView.

- ## Field Detail
  - ***month***

    ```
    public java.util.GregorianCalendar month
    ```

    The item month.

  - ***itemmonth***

    ```
    public java.util.GregorianCalendar itemmonth
    ```

    The item month.

- ***adapter***

  ```
  public CalendarAdapter adapter
  ```

  The adapter.

- ***handler***

  ```
  public android.os.Handler handler
  ```

  The handler.

- ***items***

  ```
  public java.util.ArrayList<java.lang.String> items
  ```

  The items.

- ***events***

  ```
  private java.util.ArrayList<Event> events
  ```

  The events.

- ***list***

  ```
  private android.widget.ListView list
  ```

  The list.

- ***eventSel***

  ```
  private java.util.ArrayList<Event> eventSel
  ```

  The events for selected date.

- ***calendarUpdater***

  ```
  public java.lang.Runnable calendarUpdater
  ```

  The calendar updater to update the Calendar grids and data

- **Constructor Detail**
  - ***CalendarView***

```
public CalendarView()
```

- **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`

  - ***initCalendarView***

    `private void initCalendarView(android.view.View v)`

    Initialize the calendar view.

    Parameters:

`v` - the v

  - ***setupEventList***

    `private void setupEventList(android.view.View v)`

    Set the up event list.

    Parameters:

`v` - the root view

  - ***setNextMonth***

    `protected void setNextMonth()`

    Sets the next month.

  - ***setPreviousMonth***

    `protected void setPreviousMonth()`

    Sets the previous month.

- ***showToast***

  ```
  protected void showToast(java.lang.String string)
  ```

  Show toast.

  Parameters:

`string` - the string message

- ***refreshCalendar***

  ```
  public void refreshCalendar()
  ```

  Refresh calendar.

- ***onCreateOptionsMenu***
- ```
  public void onCreateOptionsMenu(android.view.Menu menu,
                                  android.view.MenuInflater inflater)
  ```

  Overrides:

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

com.events.custom

Class CustomActivity

Direct Known Subclasses:

EventDetailActivity, MainActivity, MapViewActivity

```
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 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.events.custom

Class CustomFragment

Direct Known Subclasses:

About, CalendarView, EventDetail, FeedList, MapViewer, More, MyTickets, Programs

```
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.events.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 title2,`
  - `java.lang.String desc,`
    `int image1)`

    Instantiates a new data.

    Parameters:

`title1` - the title1

`title2` - the title2

`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.events.model

Class Event

```
public class Event
extends java.lang.Object
```

The Class Event is a simple Java Bean that is used to hold data related to an Event like title, date etc.

- **Field Detail**
  - ***title***

    ```
    private java.lang.String title
    ```

    The title.

  - ***date***

    ```
    private java.lang.String date
    ```

    The date.

  - ***time***

    ```
    private java.lang.String time
    ```

    The time.

- ***location***

  ```
  private java.lang.String location
  ```

  The location.

- ***image***

  ```
  private int image
  ```

  The image.

- **Constructor Detail**
  - ***Event***
  - `public Event(java.lang.String title,`
  - `java.lang.String date,`
  - `java.lang.String time,`
  - `java.lang.String location,`
    `int image)`

    Instantiates a new event.

    Parameters:

`title` - the title

`date` - the date

`time` - the time

`location` - the location

`image` - the image

- **Method Detail**
  - ***getTitle***

    ```
    public java.lang.String getTitle()
    ```

    Gets the title.

    Returns:

the title

- ***setTitle***

  ```
  public void setTitle(java.lang.String title)
  ```

  Sets the title.

  Parameters:

`title` - the new title

- ***getDate***

  ```
  public java.lang.String getDate()
  ```

  Gets the date.

  Returns:

the date

- ***setDate***

  ```
  public void setDate(java.lang.String date)
  ```

  Sets the date.

  Parameters:

`date` - the new date

- ***getTime***

  ```
  public java.lang.String getTime()
  ```

  Gets the time.

  Returns:

the time

- ***setTime***

  ```
  public void setTime(java.lang.String time)
  ```

Sets the time.

Parameters:

`time` - the new time

- ***getLocation***

  `public java.lang.String getLocation()`

  Gets the location.

  Returns:

the location

- ***setLocation***

  `public void setLocation(java.lang.String location)`

  Sets the location.

  Parameters:

`location` - the new location

- ***getImage***

  `public int getImage()`

  Gets the image.

  Returns:

the image

- ***setImage***

  `public void setImage(int image)`

  Sets the image.

  Parameters:

`image` - the new image

com.events.ui

Class EventDetail

- java.lang.Object
  -
    - android.support.v4.app.Fragment
      -
        - com.events.custom.CustomFragment
          -
            - com.events.ui.EventDetail

- All Implemented Interfaces:
android.content.ComponentCallbacks, android.view.View.OnClickListener, android.view.View.OnCreateContextMenuListener

---

```
public class EventDetail
extends CustomFragment
```

The Class EventDetail is the Fragment class that shows the details about an Event. This Fragment is used inside the EventDetailActivity class. It also show a Map with a marker on map for showing the location of that event. You need to write your own logic for loading actual contents related to Events and also need to show actual location for Event.

- **Field Detail**
  - ***mMapView***

```
private com.google.android.gms.maps.MapView mMapView
```

The map view.

- ***mMap***

```
private com.google.android.gms.maps.GoogleMap mMap
```

The Google map.

- **Constructor Detail**
  - ***EventDetail***

```
public EventDetail()
```

- **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`

- ***onPause***

```
public void onPause()
```

Overrides:

`onPause` in class `android.support.v4.app.Fragment`

- ***onDestroy***

```
public void onDestroy()
```

Overrides:

`onDestroy` in class `android.support.v4.app.Fragment`

- ***onResume***

```
public void onResume()
```

Overrides:

`onResume` in class `android.support.v4.app.Fragment`

- ***setupMap***
- `private void setupMap(android.view.View v, android.os.Bundle savedInstanceState)`

  Setup and initialize the Google map view.

  Parameters:

`v` - the root view

`savedInstanceState` - the saved instance state

- ***setupMarker***

  `private void setupMarker()`

  This method simply place a few dummy location markers on Map View. You can write your own logic for loading the locations and placing the marker for each location as per your need.

- ***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.events

Class EventDetailActivity

```
public class EventDetailActivity
extends CustomActivity
```

The EventDetailActivity is the activity class that shows the details about an Event. This is launched when ever user select an Event from the Event listing or from Events on Map. It also show a Map with a marker on map for showing the location of that event. You need to write your own logic for loading actual contents related to Events and also need to show actual location for Event.

- **Constructor Detail**
  - ***EventDetailActivity***

    ```
    public EventDetailActivity()
    ```

- **Method Detail**
  - ***onCreate***

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

`onCreate` in class `CustomActivity`

  - ***addFragment***

    ```
    private void addFragment()
    ```

    Attach the appropriate fragment with this activity.

  - ***onOptionsItemSelected***

    ```
    public boolean onOptionsItemSelected(android.view.MenuItem item)
    ```
    Overrides:

`onOptionsItemSelected` in class `android.app.Activity`

com.events.model

Class Feed

```
public class Feed
extends java.lang.Object
```

The Class Feed is a simple Java Bean that is used to hold the data of a particular Feed item like title, date etc.

- **Field Detail**
  - ***title***

    ```
    private java.lang.String title
    ```

    The title.

  - ***name***

    ```
    private java.lang.String name
    ```

    The name.

  - ***msg***

    ```
    private java.lang.String msg
    ```

    The msg.

  - ***comment***

    ```
    private java.lang.String comment
    ```

    The comment.

  - ***images***

```
private int[] images
```

The images.

- ***image***

```
private int image
```

The image.

- **Constructor Detail**
  - ***Feed***
  - `public Feed(java.lang.String title,`
  - `java.lang.String name,`
  - `java.lang.String msg,`
  - `java.lang.String comment,`
  - `int[] images,`
    `int image)`

Instantiates a new feed.

Parameters:

`title` - the title

`name` - the name

`msg` - the msg

`comment` - the comment

`images` - the images

`image` - the image

- **Method Detail**
  - ***getTitle***

```
public java.lang.String getTitle()
```

Gets the title.

Returns:

the title

- ***setTitle***

  ```
  public void setTitle(java.lang.String title)
  ```

  Sets the title.

  Parameters:

`title` - the new title

- ***getName***

  ```
  public java.lang.String getName()
  ```

  Gets the name.

  Returns:

the name

- ***setName***

  ```
  public void setName(java.lang.String name)
  ```

  Sets the name.

  Parameters:

`name` - the new name

- ***getMsg***

  ```
  public java.lang.String getMsg()
  ```

  Gets the msg.

  Returns:

the msg

- ***setMsg***

  ```
  public void setMsg(java.lang.String msg)
  ```

Sets the msg.

Parameters:

`msg` - the new msg

- ***getComment***

  `public java.lang.String getComment()`

  Gets the comment.

  Returns:

the comment

- ***setComment***

  `public void setComment(java.lang.String comment)`

  Sets the comment.

  Parameters:

`comment` - the new comment

- ***getImages***

  `public int[] getImages()`

  Gets the images.

  Returns:

the images

- ***setImages***

  `public void setImages(int[] images)`

  Sets the images.

  Parameters:

`images` - the new images

- ***getImage***

  `public int getImage()`

  Gets the image.

  Returns:

the image

- ***setImage***

  `public void setImage(int image)`

  Sets the image.

  Parameters:

`image` - the new image

com.events.ui

Class FeedList

`public class FeedList`
`extends CustomFragment`

The Class FeedList is the Fragment class that is launched when the user clicks on Feed option in Left navigation drawer . It simply shows a dummy list of Social media Feeds. You can customize this class to display actual Feed listing.

-
  - ## Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
private class	**FeedList.ProgramAdapter** The Class FeedAdapter is the adapter class for Feed ListView.

- ## Field Detail
  - ***fList***

    ```
    private java.util.ArrayList<Feed> fList
    ```

    The feed list.

- ## Constructor Detail
  - ***FeedList***

    ```
    public FeedList()
    ```

- ## 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`

  - ***setFeedList***

    ```
    private void setFeedList(android.view.View v)
    ```

    Setup and initialize the feed list view.

Parameters:

`v` - the root view

- ***loadDummyFeeds***

  ```
  private void loadDummyFeeds()
  ```

  Load a dummy list of feeds. You need to write your own logic to load actual list of feeds.

- ***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.events.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 of items to be displayed on ListView

- **Method Detail**
  - ***setSelection***

    ```
    public void setSelection(int position)
    ```

    Setup the current selected position of adapter.

    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.events

Class MainActivity

```
public class MainActivity
extends CustomActivity
```

The Class MainActivity is the base activity class of the application. This activity is launched after the Splash 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.widget.ListView drawerLeft
  ```

  ListView for left side drawer.

- ***drawerToggle***

  ```
  private android.support.v4.app.ActionBarDrawerToggle drawerToggle
  ```

  The drawer toggle.

- ***adapter***

  ```
  private LeftNavAdapter adapter
  ```

  The left navigation list adapter.

- ***tab***

  ```
  private android.view.View tab
  ```

  The tab.

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***

```
public 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 0. 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.Configurat
  ion 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 in class CustomActivity

com.events

Class MapViewActivity

```
public class MapViewActivity
extends CustomActivity
```

The MapViewActivity is the activity class that shows Map fragment. This activity is only created to show Back button on ActionBar.

- **Constructor Detail**
  - ***MapViewActivity***

    ```
    public MapViewActivity()
    ```

- **Method Detail**
  - ***onCreate***

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

    Overrides:

    onCreate in class CustomActivity

  - ***addFragment***

    ```
    private void addFragment()
    ```

    Attach the appropriate MapViewer fragment with activity.

- ***onOptionsItemSelected***

  ```
  public boolean onOptionsItemSelected(android.view.MenuItem item)
  ```

  Overrides:

`onOptionsItemSelected` in class `android.app.Activity`

com.events.ui

Class MapViewer

```
public class MapViewer
extends CustomFragment
```

The Class MapViewer is the Fragment class that is launched when the user clicks on Map option in Left navigation drawer or when user tap on the Map icon in action bar. It simply shows a Map View with a few dummy location markers on map. You can customize this class to load and display actual locations on map.

- **Field Detail**
  - ***mMapView***

    ```
    private com.google.android.gms.maps.MapView mMapView
    ```

    The map view.

  - ***mMap***

    ```
    private com.google.android.gms.maps.GoogleMap mMap
    ```

The Google map.

- **Constructor Detail**
  - ***MapViewer***

    ```
    public MapViewer()
    ```

- **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`

  - ***onPause***

    ```
    public void onPause()
    ```
    Overrides:

`onPause` in class `android.support.v4.app.Fragment`

  - ***onDestroy***

    ```
    public void onDestroy()
    ```
    Overrides:

`onDestroy` in class `android.support.v4.app.Fragment`

  - ***onResume***

    ```
    public void onResume()
    ```
    Overrides:

`onResume` in class `android.support.v4.app.Fragment`

  - ***setupMap***
  - ```
    private void setupMap(android.view.View v,
                  android.os.Bundle savedInstanceState)
    ```

    Setup and initialize the Google map view.

    Parameters:

`v` - the root view

`savedInstanceState` - the saved instance state

- ***setupMarker***

  ```
  private void setupMarker()
  ```

  This method simply place a few dummy location markers on Map View. You can write your own logic for loading the locations and placing the marker for each location as per your need.

- ***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.events.ui

Class More

```
public class More
```
```

```
extends CustomFragment
```

The Class More is the Fragment class that is launched when the user clicks on More option in Left navigation drawer and it simply shows a few options for like Help, Privacy, Account, About etc. You can customize this to display actual contents.

- **Constructor Detail**
  - ***More***

    ```
    public More()
    ```

- **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`

com.events.ui

Class MyTickets

```
public class MyTickets
extends CustomFragment
```
```

The Class MyTickets is the Fragment class that is launched when the user clicks on My Tickets tab in MyProgram section and It simply shows a dummy list of user's tickets . You can customize this class to display actual ticket listing.

-
  - **Nested Class Summary**

Nested Classes

Modifier and Type	Class and Description
private class	**MyTickets.TicketAdapter** The Class TicketAdapter is the adapter class for ticket ListView.

- **Field Detail**
  - ***tList***

    ```
    private java.util.ArrayList<Data> tList
    ```

    The ticket list.

- **Constructor Detail**
  - ***MyTickets***

    ```
    public MyTickets()
    ```

- **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`

  - ***setTicketList***

    ```
    private void setTicketList(android.view.View v)
    ```

    Setup and initialize the ticket list view.

Parameters:

`v` - the root view

- ***loadDummyTickets***

  ```
  private void loadDummyTickets()
  ```

  Load a dummy list of tickets. You need to write your own logic to load actual list of tickets.

- ***onCreateOptionsMenu***
- ```
  public void onCreateOptionsMenu(android.view.Menu menu,
                                  android.view.MenuInflater inflater)
  ```

  Overrides:

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

com.events.ui

Class Programs

```
public class Programs
extends CustomFragment
```

The Class Programs is the Fragment class that is launched when the user clicks on Programs or on MyPrograms option in Left navigation drawer and this is also used as a default fragment for MainActivity. It simply shows a dummy list of Events/Programs. . You can customize this class to display actual Feed listing.
```

-
  - ## Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
private class	**Programs.ProgramAdapter** The Class ProgramAdapter is the adapter class for Feed ListView.

- ## Field Detail
  - ***pList***

    ```
    private java.util.ArrayList<Data> pList
    ```

    The Programs list.

- ## Constructor Detail
  - ***Programs***

    ```
    public Programs()
    ```

- ## 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`

  - ***setProgramList***

    ```
    private void setProgramList(android.view.View v)
    ```

    Setup and initialize the Program list view.

    Parameters:

`v` - the root view

- ***loadDummyPrograms***

  ```
  private void loadDummyPrograms()
  ```

  Load a dummy list of Programs. You need to write your own logic to load actual list of Programs.

- ***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.events

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 MainActivity 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`
    ```

com.events.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`
    ```

com.events.calendar

Class Utility

```
public class Utility
extends java.lang.Object
```

The Class Utility is the Utility class for Calendar section. You can write all of your utility method related to Calendar section like loading events from device calendar.

- **Constructor Detail**
  - ***Utility***

    ```
    public Utility()
    ```

- **Method Detail**
  - ***readCalendarEvent***

    ```
    public
    static java.util.ArrayList<Event> readCalendarEvent(android.conte
    nt.Context context)
    ```

    Read calendar events.

    Parameters:

`context` - the context

    Returns:

the array list

  - ***getDate***

    ```
    public static java.lang.String getDate(long milliSeconds)
    ```

    Gets the date.

Parameters:

`milliSeconds` - the milli seconds

Returns:

the date