

Assignment. No. 1	e-Mail Application
Date: 18-07-2017	

Aim

To develop a Android application to develop send mail by using Java Mail API.

Requirements

1. JDK
2. Android SDK
3. [jar files] : additional.jar , activation.jar, mail.jar

Youtube Link

<https://youtu.be/jAljGKGw0Ps>

Source Code

Java code:

1.SenderMail.java

```
package com.example.floura.mailsender;
```

```
/**
```

```
* Created by floura on 4/11/17.
```

```
*/
```

```
import android.app.ProgressDialog;
```

```
import android.content.Context;
```

```
import android.os.AsyncTask;
```

```
import android.widget.Toast;
```

```
import java.util.Properties;
```

```
import javax.mail.Message;
```

```
import javax.mail.MessagingException;
```

```
import javax.mail.PasswordAuthentication;
```

```
import javax.mail.Session;
```

```
import javax.mail.Transport;
```

```
import javax.mail.internet.InternetAddress;
```

```
import javax.mail.internet.MimeMessage;
```

```
//Class is extending AsyncTask because this class is going to perform a networking operation
```

```
/**
```

```
* Android defines AsyncTask as "a class that extends the Object class to allow short operations  
to
```

```
* run asynchronously in the background." With "doInBackground" and "onPostExecute,"
```

```
* Async can run tasks asynchronously on new threads.
```

```
* Asynchronous tasks use: Params, parameters that are sent to the task upon execution.
```

```
*/
```

```
public class SendMail extends AsyncTask<Void,Void,Void> {
```

```
//Declaring Variables
```

```
private Context context;
```

```
private Session session;
```

```
//Information to send email
```

```
private String email;
```

```
private String subject;
```

```
private String message;
```

```
//Progress dialog to show while sending email
```

```
private ProgressDialog progressDialog;
```

//Class Constructor

```
public SendMail(Context context, String email, String subject, String message){
```

//Initializing variables

```
this.context = context;
```

```
this.email = email;
```

```
this.subject = subject;
```

```
this.message = message;
```

```
}
```

*/***

** onPreExecute() , invoked on the UI thread before the task is executed. This step is normally used to setup the task*

**/*

@Override

```
protected void onPreExecute() {
```

```
super.onPreExecute();
```

//Showing progress dialog while sending email

```
progressDialog = ProgressDialog.show(context, "Sending message", "Please wait...", false, false);
```

```
}
```

*/***

** onPostExecute(Result), invoked on the UI thread after the background computation finishes.*

** The result of the background computation is passed to this step as a parameter.*

** @param aVoid = result of background*

**/*

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

//Dismissing the progress dialog

progressDialog.dismiss();

//Showing a success message

 Toast.makeText(**context**, "Message Sent", Toast.**LENGTH_LONG**).show();

}

*/***

** doInBackground(Params...), invoked on the background thread immediately after
onPostExecute()*

** finishes executing. This step is used to perform background computation that can take a long
time.*

** The parameters of the asynchronous task are passed to this step.*

** The result of the computation must be returned by this step and will be passed back to the
last step.*

** @param params = parameters of the asynchronous task are passed to this step*

** @return = result of the computation must be returned by this step*

**/*

@Override

protected Void doInBackground(Void... params) {

//Creating properties

 Properties props = **new** Properties();

//Configuring properties for gmail

//If you are not using gmail you may need to change the values

```
props.put("mail.smtp.host", "smtp.gmail.com");  
props.put("mail.smtp.socketFactory.port", "465");  
props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");  
props.put("mail.smtp.auth", "true");  
props.put("mail.smtp.port", "465");
```

//Creating a new session

```
session = Session.getDefaultInstance(props,  
    new javax.mail.Authenticator() {  
        //Authenticating the password  
        protected PasswordAuthentication getPasswordAuthentication() {  
            return new PasswordAuthentication(Config.EMAIL, Config.PASSWORD);  
        }  
    });
```

try {

//Creating MimeMessage object

```
MimeMessage mm = new MimeMessage(session);
```

//Setting sender address

```
mm.setFrom(new InternetAddress(Config.EMAIL));
```

//Adding receiver

```
mm.addRecipient(Message.RecipientType.TO, new InternetAddress(email));
```

//Adding subject

```
mm.setSubject(subject);
```

//Adding message

mm.setText(**message**);

//Sending email

Transport.*send*(mm);

} **catch** (MessagingException e) {

 e.printStackTrace();

}

return null;

}

}

2. Config.java

package com.example.floura.mailsender;

*/***

** Created by floura on 4/11/17.*

**/*

*/***

** this class is used for authentication*

**/*

public class Config {

public static final String *EMAIL* = "floura71196@gmail.com";

```
public static final String PASSWORD = "71196@angel";  
}
```

3. MainActivity.java

```
package com.example.floura.mailsender;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
```

```
//Declaring EditText
```

```
private EditText editTextEmail;
```

```
private EditText editTextSubject;
```

```
private EditText editTextMessage;
```

```
//Send button
```

```
private Button buttonSend;
```

```
/**
```

```
* onCreate(Bundle) is where you initialize your activity.
```

```
* Most importantly, here you will usually call setContentView(int) with a layout resource  
defining your UI,
```

** and using findViewById(int) to retrieve the widgets in that UI that you need to interact with programmatically.*

** @param savedInstanceState = allow to save away any dynamic instance state in your activity into the given Bundle*

**/*

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.**activity_main**);

//Initializing the views

editTextEmail = (EditText) findViewById(R.id.**editTextEmail**);

editTextSubject = (EditText) findViewById(R.id.**editTextSubject**);

editTextMessage = (EditText) findViewById(R.id.**editTextMessage**);

buttonSend = (Button) findViewById(R.id.**buttonSend**);

//Adding click listener

buttonSend.setOnClickListener(**this**);

}

private void sendEmail() {

//Getting content for email

String email = **editTextEmail**.getText().toString().trim();

String subject = **editTextSubject**.getText().toString().trim();


```
String message = editTextMessage.getText().toString().trim();
```

```
//Creating SendMail object
```

```
SendMail sm = new SendMail(this, email, subject, message);
```

```
//Executing sendmail to send email
```

```
sm.execute();
```

```
}
```

```
@Override
```

```
public void onClick(View v) {
```

```
    sendEmail();
```

```
}
```

```
}
```

Resource File:

1.activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:orientation="vertical"
```

```
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
```

```
        android:layout_height="match_parent"
```

```
    android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
    android:paddingRight="@dimen/activity_horizontal_margin"
```

```
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
        android:paddingBottom="@dimen/activity_vertical_margin"  
tools:context=".MainActivity">
```

```
<TextView  
    android:text="Recipient Email"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />  
<EditText  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:id="@+id/editTextEmail" />  
<TextView  
    android:text="Subject"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />  
<EditText  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:id="@+id/editTextSubject" />  
<TextView  
    android:text="Message"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />  
<EditText  
    android:lines="4"  
    android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:id="@+id/editTextMessage" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/buttonSend"
    android:text="Send"/>
</LinearLayout>

Manifest File:
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.floura.mailsender">

    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
```

```
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>

</manifest>
```

Documentation

Config Class

com.example.floura.mailsender

Class Config

- java.lang.Object
 - com.example.floura.mailsender.Config
- _____
- public class **Config**
extends java.lang.Object
- this class is used for authentication

• *Field Summary*

• **Modifier and Type**

- static
java.la

• **Field and Description**

- **EMAIL**

ng.Stri
ng

- static
- **PASSWORD**
- java.la
- ng.Stri
- ng

- **Constructor Summary**

- **Constructor and Description**

- **Config()**

- **Method Summary**

- **Methods inherited from class java.lang.Object**

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- **Field Detail**

- **EMAIL**

- public static final java.lang.String EMAIL
- **See Also:**
- **Constant Field Values**

- **PASSWORD**

- public static final java.lang.String PASSWORD
- **See Also:**
- [Constant Field Values](#)

- ***Constructor Detail***

- **Config**

- public Config()

SendMail Class

com.example.floura.mailsender

Class SendMail

- java.lang.Object
 - android.os.AsyncTask<java.lang.Void,java.lang.Void,java.lang.Void>
 - com.example.floura.mailsender.SendMail
- _____
- public class **SendMail**
extends android.os.AsyncTask<java.lang.Void,java.lang.Void,java.lang.Void>
- Android defines AsyncTask as “a class that extends the Object class to allow short operations to run asynchronously in the background.” With “doInBackground” and

“onPostExecute,” Async can run tasks asynchronously on new threads. Asynchronous tasks use: Params, parameters that are sent to the task upon execution.

- ***Nested Class Summary***

- **Nested classes/interfaces inherited from class android.os.AsyncTask**

- android.os.AsyncTask.Status

- ***Field Summary***

- **Fields inherited from class android.os.AsyncTask**

- SERIAL_EXECUTOR, THREAD_POOL_EXECUTOR

- ***Constructor Summary***

- **Constructor and Description**

- **SendMail**(android.content.Context context, java.lang.String email, java.lang.String subject, java.lang.String message)

- ***Method Summary***

- **Modifier and Type**

- **Method and Description**

- protected
java.lang.Void
- **doInBackground**(java.lang.Void... params)
- doInBackground(Params...), invoked on the background thread immediately after onPreExecute() finishes executing.
- protected void
- **onPostExecute**(java.lang.Void aVoid)
- onPostExecute(Result), invoked on the UI thread after the background computation finishes.
- protected void
- **onPreExecute**()
- onPreExecute() , invoked on the UI thread before the task is executed.

• **Methods inherited from class android.os.AsyncTask**

- cancel, execute, execute, executeOnExecutor, get, get, getStatus, isCancelled, onCancelled, onCancelled, onProgressUpdate, publishProgress

- **Methods inherited from class java.lang.Object**

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- ***Constructor Detail***

- **SendMail**

- public SendMail(android.content.Context context,
java.lang.String email,
java.lang.String subject,
java.lang.String message)

- ***Method Detail***

- **onPreExecute**

- protected void onPreExecute()
- onPreExecute() , invoked on the UI thread before the task is executed. This step is normally used to setup the task
- **Overrides:**
- onPreExecute in class
android.os.AsyncTask<java.lang.Void,java.lang.Void,java.lang.Void>

- **onPostExecute**

- protected void onPostExecute(java.lang.Void aVoid)

- `onPostExecute(Result)`, invoked on the UI thread after the background computation finishes. The result of the background computation is passed to this step as a parameter.
- **Overrides:**
- `onPostExecute` in class `android.os.AsyncTask<java.lang.Void,java.lang.Void,java.lang.Void>`
- **Parameters:**
- `aVoid` - = result of background

- **doInBackground**

- `protected java.lang.Void doInBackground(java.lang.Void... params)`
- `doInBackground(Params...)`, invoked on the background thread immediately after `onPreExecute()` finishes executing. This step is used to perform background computation that can take a long time. The parameters of the asynchronous task are passed to this step. The result of the computation must be returned by this step and will be passed back to the last step.
- **Specified by:**
- `doInBackground` in class `android.os.AsyncTask<java.lang.Void,java.lang.Void,java.lang.Void>`
- **Parameters:**
- `params` - = parameters of the asynchronous task are passed to this step
- **Returns:**
- = result of the computation must be returned by this step

3.MainActivity Class

com.example.floura.mailsender

Class MainActivity

- java.lang.Object
 - android.content.Context
 - android.content.ContextWrapper
 - android.view.ContextThemeWrapper
 - android.app.Activity
 - android.support.v4.app.SupportActivity
 - android.support.v4.app.FragmentActivity
 - android.support.v7.app.AppCompatActivity
 - com.example.floura.mailsender.MainActivity
- **All Implemented Interfaces:**
- android.content.ComponentCallbacks, android.content.ComponentCallbacks2, android.support.v4.app.ActivityCompat.OnRequestPermissionsResultCallback,

android.support.v4.app.ActivityCompatApi23.RequestPermissionsRequestCodeValidator
, android.support.v4.app.TaskStackBuilder.SupportParentable,
android.support.v7.app.ActionBarDrawerToggle.DelegateProvider,
android.support.v7.app.AppCompatActivity.Callback, android.view.KeyEvent.Callback,
android.view.LayoutInflater.Factory, android.view.LayoutInflater.Factory2,
android.view.View.OnClickListener, android.view.View.OnCreateContextMenuListener,
android.view.Window.Callback

- _____
- public class **MainActivity**
extends android.support.v7.app.AppCompatActivity
implements android.view.View.OnClickListener

- ***Nested Class Summary***

- **Nested classes/interfaces inherited from class
android.support.v4.app.SupportActivity**

- android.support.v4.app.SupportActivity.ExtraData

- ***Field Summary***

- **Fields inherited from class android.app.Activity**

- DEFAULT_KEYS_DIALER, DEFAULT_KEYS_DISABLE,
DEFAULT_KEYS_SEARCH_GLOBAL,
DEFAULT_KEYS_SEARCH_LOCAL,
DEFAULT_KEYS_SHORTCUT, FOCUSED_STATE_SET,
RESULT_CANCELED, RESULT_FIRST_USER, RESULT_OK

- **Fields inherited from class android.content.Context**

- ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE,
ACTIVITY_SERVICE, ALARM_SERVICE, APP_OPS_SERVICE,

APPWIDGET_SERVICE, AUDIO_SERVICE, BATTERY_SERVICE,
 BIND_ABOVE_CLIENT, BIND_ADJUST_WITH_ACTIVITY,
 BIND_ALLOW_OOM_MANAGEMENT, BIND_AUTO_CREATE,
 BIND_DEBUG_UNBIND, BIND_EXTERNAL_SERVICE,
 BIND_IMPORTANT, BIND_NOT_FOREGROUND,
 BIND_WAIVE_PRIORITY, BLUETOOTH_SERVICE,
 CAMERA_SERVICE, CAPTIONING_SERVICE,
 CARRIER_CONFIG_SERVICE, CLIPBOARD_SERVICE,
 CONNECTIVITY_SERVICE, CONSUMER_IR_SERVICE,
 CONTEXT_IGNORE_SECURITY, CONTEXT_INCLUDE_CODE,
 CONTEXT_RESTRICTED, DEVICE_POLICY_SERVICE,
 DISPLAY_SERVICE, DOWNLOAD_SERVICE,
 DROPBOX_SERVICE, FINGERPRINT_SERVICE,
 HARDWARE_PROPERTIES_SERVICE, INPUT_METHOD_SERVICE,
 INPUT_SERVICE, JOB_SCHEDULER_SERVICE,
 KEYGUARD_SERVICE, LAUNCHER_APPS_SERVICE,
 LAYOUT_INFLATER_SERVICE, LOCATION_SERVICE,
 MEDIA_PROJECTION_SERVICE, MEDIA_ROUTER_SERVICE,
 MEDIA_SESSION_SERVICE, MIDI_SERVICE, MODE_APPEND,
 MODE_ENABLE_WRITE_AHEAD_LOGGING,
 MODE_MULTI_PROCESS, MODE_NO_LOCALIZED_COLLATORS,
 MODE_PRIVATE, MODE_WORLD_READABLE,
 MODE_WORLD_WRITEABLE, NETWORK_STATS_SERVICE,
 NFC_SERVICE, NOTIFICATION_SERVICE, NSD_SERVICE,
 POWER_SERVICE, PRINT_SERVICE, RESTRICTIONS_SERVICE,
 SEARCH_SERVICE, SENSOR_SERVICE, SHORTCUT_SERVICE,
 STORAGE_SERVICE, SYSTEM_HEALTH_SERVICE,
 TELECOM_SERVICE, TELEPHONY_SERVICE,

TELEPHONY_SUBSCRIPTION_SERVICE,
TEXT_SERVICES_MANAGER_SERVICE, TV_INPUT_SERVICE,
UI_MODE_SERVICE, USAGE_STATS_SERVICE, USB_SERVICE,
USER_SERVICE, VIBRATOR_SERVICE, WALLPAPER_SERVICE,
WIFI_P2P_SERVICE, WIFI_SERVICE, WINDOW_SERVICE

- **Fields inherited from interface android.content.ComponentCallbacks2**

- TRIM_MEMORY_BACKGROUND, TRIM_MEMORY_COMPLETE,
TRIM_MEMORY_MODERATE,
TRIM_MEMORY_RUNNING_CRITICAL,
TRIM_MEMORY_RUNNING_LOW,
TRIM_MEMORY_RUNNING_MODERATE,
TRIM_MEMORY_UI_HIDDEN

- *Constructor Summary*

- **Constructor and Description**

- **MainActivity()**

- *Method Summary*

- **Modifier and Type**

- **Method and Description**

- void
- **onClick**(android.view.View v)
- protect
ed void
- **onCreate**(android.os.Bundle savedInstanceState)
- onCreate(Bundle) is where you initialize your activity.

• Methods	inherited	from	class
android.support.v7.app.AppCompatActivity			
• addContentView, dispatchKeyEvent, findViewById, getDelegate, getDrawerToggleDelegate, getMenuInflater, getResources, getSupportActionBar, getSupportParentActivityIntent, invalidateOptionsMenu, onConfigurationChanged, onContentChanged, onCreateSupportNavigateUpTaskStack, onDestroy, onMenuItemSelected, onMenuOpened, onPanelClosed, onPostCreate, onPostResume, onPrepareSupportNavigateUpTaskStack, onSaveInstanceState, onStart, onStop, onSupportActionModeFinished, onSupportActionModeStarted, onSupportContentChanged, onSupportNavigateUp, onTitleChanged, onWindowStartingSupportActionMode, setContentView, setContentView, setContentView, setSupportActionBar, setSupportProgress, setSupportProgressBarIndeterminate, setSupportProgressBarIndeterminateVisibility, setSupportProgressBarVisibility, setTheme, startSupportActionMode,			

supportInvalidateOptionsMenu, supportNavigateUpTo,
supportRequestWindowFeature, supportShouldUpRecreateTask

• **Methods inherited from class android.support.v4.app.FragmentActivity**

- dump, getLastCustomNonConfigurationInstance, getSupportFragmentManager, getSupportLoaderManager, getSupportMediaController, onActivityResult, onAttachFragment, onBackPressed, onCreatePanelMenu, onLowMemory, onMultiWindowModeChanged, onNewIntent, onPause, onPictureInPictureModeChanged, onPrepareOptionsPanel, onPreparePanel, onRequestPermissionsResult, onResume, onResumeFragments, onRetainCustomNonConfigurationInstance, onRetainNonConfigurationInstance, onStateNotSaved, setEnterSharedElementCallback, setExitSharedElementCallback, setSupportMediaController, startActivityForResult, startActivityFromFragment, startActivityFromFragment, startActivitySenderFromFragment, supportFinishAfterTransition, supportPostponeEnterTransition, supportStartPostponedEnterTransition, validateRequestPermissionsRequestCode

• **Methods inherited from class android.support.v4.app.SupportActivity**

- getExtraData, putExtraData

• **Methods inherited from class android.app.Activity**

- closeContextMenu, closeOptionsMenu, createPendingResult, dismissDialog, dismissKeyboardShortcutsHelper, dispatchGenericMotionEvent, dispatchKeyShortcutEvent,

dispatchPopulateAccessibilityEvent, dispatchTouchEvent,
dispatchTrackballEvent, enterPictureInPictureMode, finish, finishActivity,
finishActivityFromChild, finishAffinity, finishAfterTransition,
finishAndRemoveTask, finishFromChild, getActionBar, getApplication,
getCallingActivity, getCallingPackage, getChangingConfigurations,
getComponentName, getContentScene, getContentTransitionManager,
getCurrentFocus, getFragmentManager, getIntent,
getLastNonConfigurationInstance, getLayoutInflater, getLoaderManager,
getLocalClassName, getMediaController, getParent,
getParentActivityIntent, getPreferences, getReferrer,
getRequestedOrientation, getSearchEvent, getSystemService, getTaskId,
getTitle, getTitleColor, getVoiceInteractor, getVolumeControlStream,
getWindow, getWindowManager, hasWindowFocus,
isChangingConfigurations, isChild, isDestroyed, isFinishing, isImmersive,
isInMultiWindowMode, isInPictureInPictureMode,
isLocalVoiceInteractionSupported, isTaskRoot, isVoiceInteraction,
isVoiceInteractionRoot, managedQuery, moveTaskToBack,
navigateUpTo, navigateUpToFromChild, onActionModeFinished,
onActionModeStarted, onActivityReenter, onApplyThemeResource,
onAttachedToWindow, onAttachFragment, onChildTitleChanged,
onContextItemSelected, onContextMenuClosed, onCreate,
onCreateContextMenu, onCreateDescription, onCreateDialog,
onCreateDialog, onCreateNavigateUpTaskStack, onCreateOptionsMenu,
onCreatePanelView, onCreateThumbnail, onDetachedFromWindow,
onEnterAnimationComplete, onGenericMotionEvent, onKeyDown,
onKeyLongPress, onKeyMultiple, onKeyShortcut, onKeyUp,
onLocalVoiceInteractionStarted, onLocalVoiceInteractionStopped,
onNavigateUp, onNavigateUpFromChild, onOptionsItemSelected,

onOptionsMenuClosed, onCreate, onPrepareDialog, onPrepareDialog,
onPrepareNavigateUpTaskStack, onPrepareOptionsMenu,
onProvideAssistContent, onProvideAssistData,
onProvideKeyboardShortcuts, onProvideReferrer, onRestart,
onRestoreInstanceState, onRestoreInstanceState, onSaveInstanceState,
onSearchRequested, onSearchRequested, onTouchEvent,
onTrackballEvent, onTrimMemory, onUserInteraction, onUserLeaveHint,
onVisibleBehindCanceled, onWindowAttributesChanged,
onWindowFocusChanged, onWindowStartingActionMode,
onWindowStartingActionMode, openContextMenu, openOptionsMenu,
overridePendingTransition, postponeEnterTransition, recreate,
registerForContextMenu, releaseInstance, removeDialog,
reportFullyDrawn, requestDragAndDropPermissions, requestPermissions,
requestShowKeyboardShortcuts, requestVisibleBehind,
requestWindowFeature, runOnUiThread, setActionBar,
setContentTransitionManager, setDefaultKeyMode,
setEnterSharedElementCallback, setExitSharedElementCallback,
setFeatureDrawable, setFeatureDrawableAlpha,
setFeatureDrawableResource, setFeatureDrawableUri,
setFinishOnTouchOutside, setImmersive, setIntent, setMediaController,
setProgress, setProgressBarIndeterminate,
setProgressBarIndeterminateVisibility, setProgressBarVisibility,
setRequestedOrientation, setResult, setResult, setSecondaryProgress,
setTaskDescription, setTitle, setTitle, setTitleColor, setVisible,
setVolumeControlStream, setVrModeEnabled,
shouldShowRequestPermissionRationale, setUpRecreateTask,
showAssist, showDialog, showDialog, showLockTaskEscapeMessage,
startActionMode, startActionMode, startActivities, startActivities,

startActivity, startActivity, startActivityFromChild,
startActivityFromChild, startActivityFromFragment,
startActivityFromFragment, startActivityIfNeeded, startActivityIfNeeded,
startIntentSender, startIntentSender, startIntentSenderFromChild,
startIntentSenderFromChild, startLocalVoiceInteraction, startLockTask,
startManagingCursor, startNextMatchingActivity,
startNextMatchingActivity, startPostponedEnterTransition, startSearch,
stopLocalVoiceInteraction, stopLockTask, stopManagingCursor,
takeKeyEvents, triggerSearch, unregisterForContextMenu

- **Methods inherited from class android.view.ContextThemeWrapper**

- applyOverrideConfiguration, attachBaseContext, getAssets, getTheme

- **Methods inherited from class android.content.ContextWrapper**

- bindService, checkCallingOrSelfPermission,
checkCallingOrSelfPermission, checkCallingPermission,
checkCallingUriPermission, checkPermission, checkSelfPermission,
checkUriPermission, checkUriPermission, clearWallpaper,
createConfigurationContext, createDeviceProtectedStorageContext,
createDisplayContext, createPackageContext, databaseList,
deleteDatabase, deleteFile, deleteSharedPreferences,
enforceCallingOrSelfPermission, enforceCallingOrSelfPermission,
enforceCallingPermission, enforceCallingUriPermission,
enforcePermission, enforceUriPermission, enforceUriPermission, fileList,
getApplicationContext, getApplicationInfo, getBaseContext, getCacheDir,
getClassLoader, getCodeCacheDir, getContentResolver, getDatabasePath,
getDataDir, getDir, getExternalCacheDir, getExternalCacheDirs,
getExternalFilesDir, getExternalFilesDirs, getExternalMediaDirs,

getFilesDir, getFilePath, getMainLooper, getNoBackupFilesDir,
getObbDir, getObbDirs, getPackageCodePath, getPackageManager,
getPackageName, getPackageResourcePath, getSharedPreferences,
getSystemServiceName, getWallpaper,
getWallpaperDesiredMinimumHeight,
getWallpaperDesiredMinimumWidth, grantUriPermission,
isDeviceProtectedStorage, isRestricted, moveDatabaseFrom,
moveSharedPreferencesFrom, openFileInput, openFileOutput,
openOrCreateDatabase, openOrCreateDatabase, peekWallpaper,
registerReceiver, registerReceiver, removeStickyBroadcast,
removeStickyBroadcastAsUser, revokeUriPermission, sendBroadcast,
sendBroadcast, sendBroadcastAsUser, sendBroadcastAsUser,
sendOrderedBroadcast, sendOrderedBroadcast,
sendOrderedBroadcastAsUser, sendStickyBroadcast,
sendStickyBroadcastAsUser, sendStickyOrderedBroadcast,
sendStickyOrderedBroadcastAsUser, setWallpaper, setWallpaper,
startInstrumentation, startService, stopService, unbindService,
unregisterReceiver

- **Methods inherited from class android.content.Context**

- getColor, getColorStateList, getDrawable, getString, getString,
getSystemService, getText, obtainStyledAttributes,
obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes,
registerComponentCallbacks, unregisterComponentCallbacks

- **Methods inherited from class java.lang.Object**

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString,
wait, wait, wait

- **Constructor Detail**

- **MainActivity**

- `public MainActivity()`

- **Method Detail**

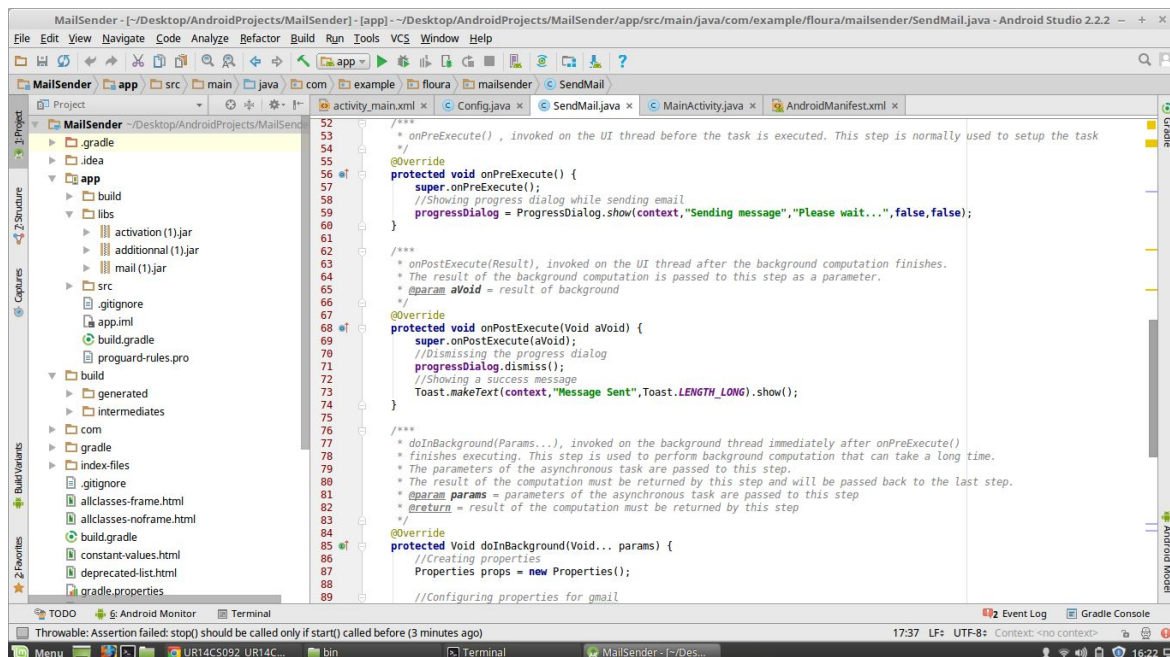
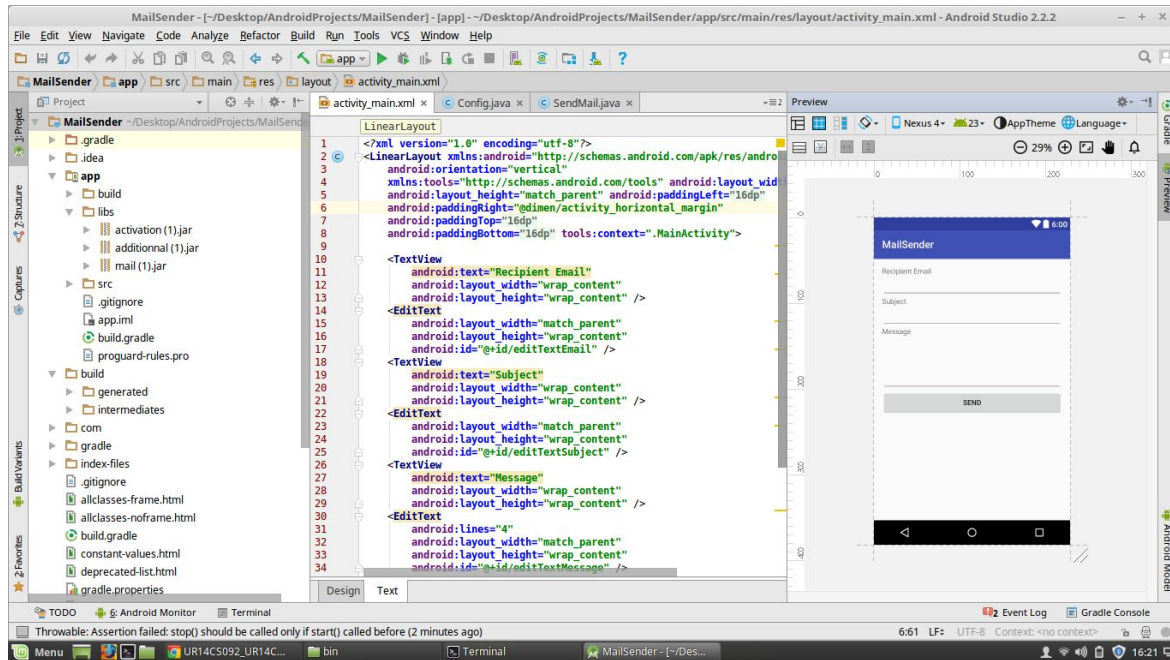
- **onCreate**

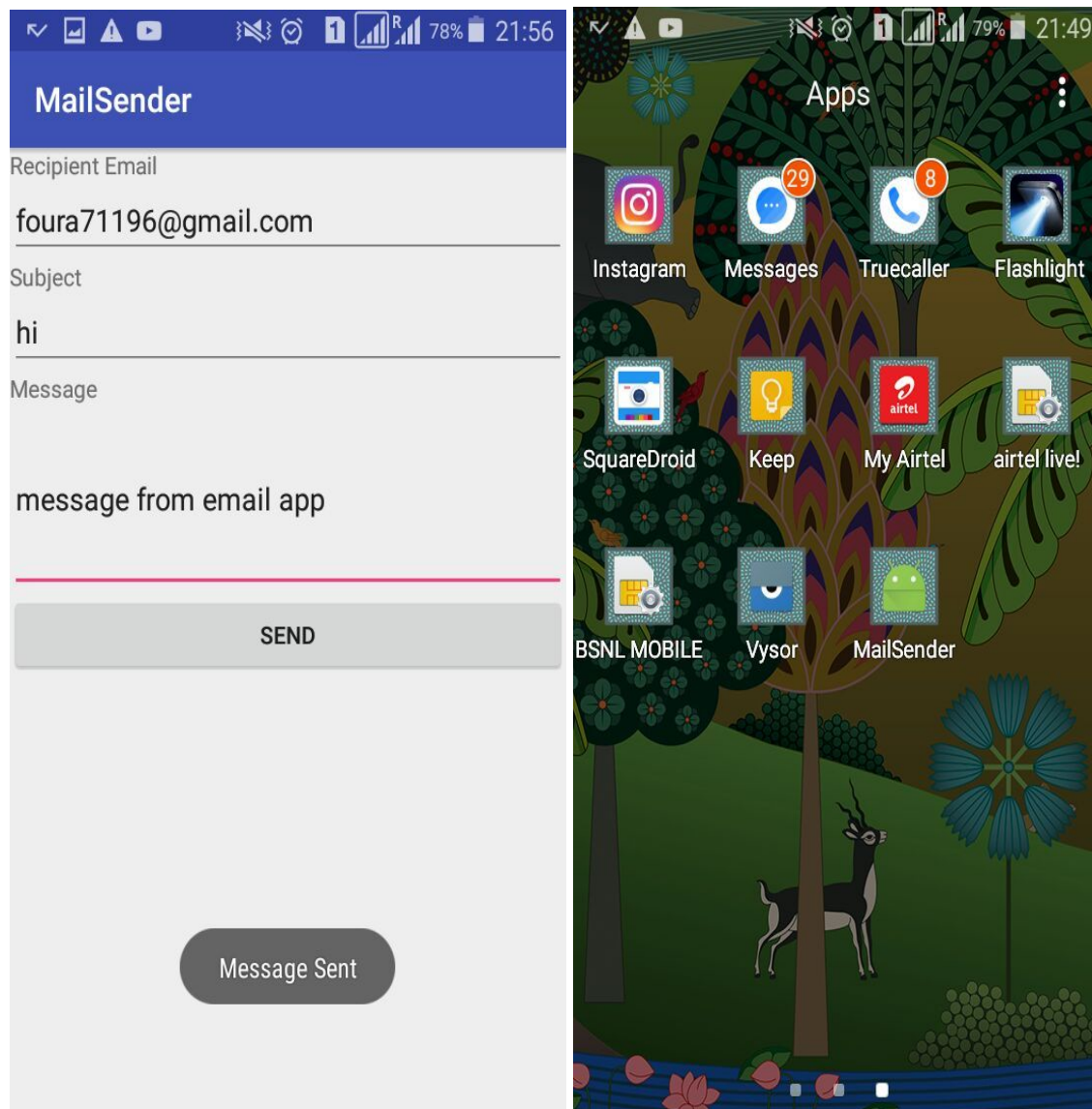
- `protected void onCreate(android.os.Bundle savedInstanceState)`
- `onCreate(Bundle)` is where you initialize your activity. Most importantly, here you will usually call `setContentView(int)` with a layout resource defining your UI, and using `findViewById(int)` to retrieve the widgets in that UI that you need to interact with programmatically.
- **Overrides:**
- `onCreate` in class `android.support.v7.app.AppCompatActivity`
- **Parameters:**
- `savedInstanceState` - = allow to save away any dynamic instance state in your activity into the given Bundle

- **onClick**

- `public void onClick(android.view.View v)`
- **Specified by:**
- `onClick` in interface `android.view.View.OnClickListener`

Output





Result

The application for sending email using JavaMail API was implemented successfully.

Submitted by:

Floura Angel (UR14CS290)

Priya Shaji (UR14CS092)