**CONTENTS PAGE NO**

1. INTRODUCTION

1.1 Synopsis 1

2. SYSTEM STUDY

2.1 Feasibility Analysis 2

3. TECHNOLOGY REQUIREMENTS

3.1 Hardware Requirements 3

3.2 Software Requirements 3

3.3 Technical Concepts 4

3.4 Software Description 5

4. SYSTEM ANALYSIS

4.1 Existing System 7 4.2 Proposed System 7

5. SYSTEM DESIGN

5.1 Data Flow Diagrams 8

5.2 Use Case Diagram 9

5.3 Screen Shots 10

6. IMPLMENTATION

6.1 Code Explanation 23

6.2 Sample Coding 28

6.3 API Documentation 32

7. TESTING

7.1 Unit Testing 73

7.2 Performance Testing 74

7.3 Compatibility Testing 74

8. FUTURE ENHANCEMENT 75

9. CONCLUSION 76

10. BIBLIOGRAPHY 77

**1. INTRODUCTION**

**1.1 Synopsis:**

**SPY DROID** is an Android Application which is used to monitor the users by logging their activities like Incoming and Outgoing Calls, Incoming and Outgoing SMS Conversations and also their Location information time to time which helps organizations, parents and guardians to take control or safeguard their employees or children.

**PURPOSE OF THE PROJECT:**

The main purpose of this project is to provide security for Companies and Family members by safeguarding their data or children. Companies can safeguard their confidential data that may be leaked through any of their employee by telecom means and also parents can monitor their children and track their activities on their mobile phone and also get their exact location time to time and feel secured.

**SCOPE OF THE PROJECT:**

This application can be used in many ways, for instance, a father can use this application by installing it in his son’s mobile phone and then track his locations and SMS conversation and come to know whether he is going or doing anything wrong. In the same way a Company can spy or log its employees by installing this application in the employee mobile phones and then tracking them for any suspicious behaviour or any theft of confidential company information.

**2. SYSTEM STUDY**

**2.1 Feasibility Analysis**

Android Smartphones are expanding in exponential way since last 4 years and has become the leader and most used Mobile Operating System in the world. Android is so vast and has many advantages compared to the other Mobile Operating Systems in the market. So, an Application like SPY DROID, if programmed for Android Smartphones then the reach would be higher and also the users can take advantage of the application for any of their personal reasons.

This project SPY DROID will be highly useful and helpful in many sectors in the current growing situation of the world. Analysing the number of information leakage from that companies and children growing or taking a wrong route and habiting bad stuff have increased a lot since last 10 years. A Parent can of course directly ask a child about his activities and stuff but we cannot assume that the children are not lying. So, an application like SPY DROID is of most use in these kinds of situations.

And the applications or usage of this SPY DROID is just not limited to Family members or Companies but can also be used within friends who want to share their location information or for couples i.e. husband and wife who want to live happily without doubting one another and this application also has a very good implementation in Government sectors. The Government can track its own employees i.e. Government officials to check whether they are in time to office and can also track the social links or connections of the government officials and check if they are into any bribery or any other illegal activity.

**3. TECHNOLOGY REQUIREMENTS  
3.1 Requirements to develop the application**

**HARDWARE SPECIFICATIONS**

Processor : Pentium IV or Higher

Main Memory : 4 GB or Above

Secondary Memory : 40 GB or Above

**SOFTWARE SPECIFICATIONS**

Operating System : Windows 7 / Linux 2.6.x or higher

IDE : Eclipse (Classic or Galileo)

Plugins : Android SDK Plugin for Eclipse

Software : Android SDK Tools, Java 1.6 or Higher

Technologies : Java, Android, SQLite, XML

Debugger : Android Dalvik Debug Monitor Service

**3.2 Requirements to deploy the application**

**HARDWARE SPECIFICATIONS**

Device : Android Smartphone

Memory : 4MB or Above

**SOFTWARE SPECIFICATIONS**

Operating System : Android 2.2 or Higher

Network Connectivity : Wi-Fi Internet or Cellular Network

**FUNCTIONAL SPECIFICATIONS**

* GUI with Screen touch functionality
* Wi-Fi Internet or Cellular Network Access

**3.3 TECHNICAL CONCEPTS**

The main reasons for picking up Android are :

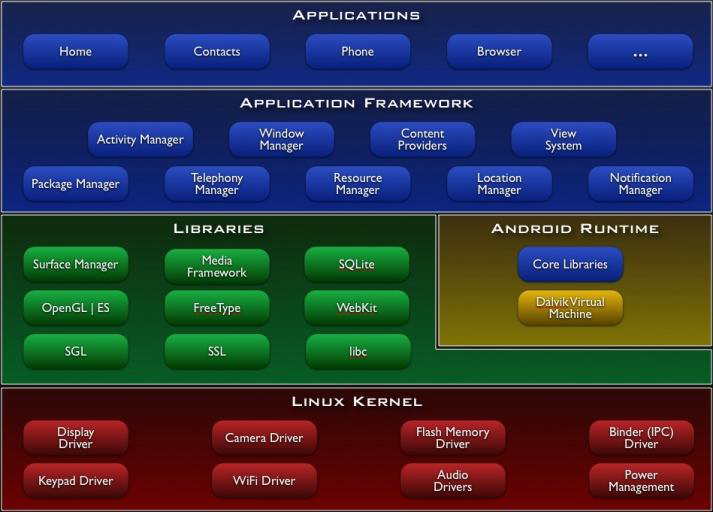
* **GOOGLE**  
   Android is owned by Google and that is one of the biggest criteria to select Android as the base for the development. Google provides rich sets of its products to its own product i.e. Android in a very good and an efficient manner.
* **AVAILABILITY**  
   Android is one of the most widely used Mobile phone Operating System these days. And this availability will help the application to reach more and more people.
* **ELASTIC GUI**  
   With XML, Android makes it possible to design the application pages or windows very easily and are highly flexible and customizable for all kind of different screen resolutions and screen sizes.
* **BUILT-IN LIBRARY FUNCTIONS** Android SDK comes with rich set of APIs which makes it very easy to program and Android application compared to any iPhone application or Blackberry OS based application.

**3.4 SOFTWARE DESCRIPTION**

**ANDROID**

Operating Systems have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android.

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Android allows background processing, provides a rich user interface library, supports 2-D and 3-D graphics using the OpenGL libraries, access to the file system and provides an embedded SQLite database. Android Inc. was founded by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later Android Inc. was acquired by Google in 2005. After the original release there have been number of updates in the original version of Android.



**ANDROID SDK**

The Android Software Development Kit (Android SDK) provides all necessary tools to develop Android applications. This includes a compiler, debugger and a device emulator, as well as its own virtual machine to run Android programs.

**ANDROID VIRTUAL MACHINE**

The Android tools include an *Emulator*, a piece of software that pretends to be an

Android device. This is very useful for development on Android without a device, but the emulator can help test all other device configurations. The Android emulator can emulate one or several Android devices.



**4. SYSTEM ANALYSIS**

**4.1 Existing System**

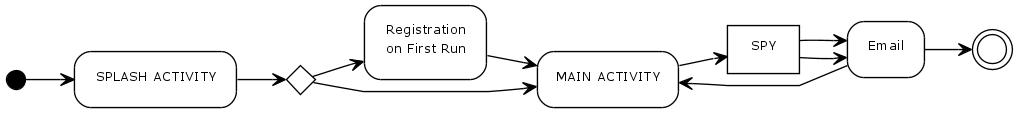
* Application Functionality visible to the user and is vain
* 2 Applications must be installed and heavy to use in both mobile phones
* Application does not provide any alternative if Internet Connection is not available

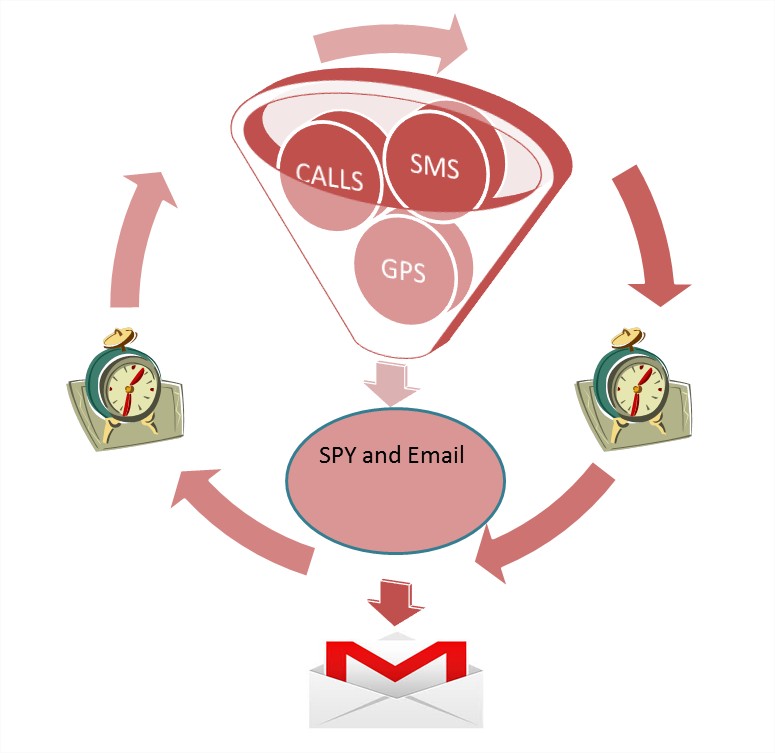
**4.2 Proposed System**

* Application Functionality made completely invisible
* One application installation is enough and very less and optimized use of the memory
* Alternative actions like sending an SMS and other are present if there is no Internet
* Application doesn’t need any user interaction or initiation again and again for every restart of the mobile phone and can start automatically after the first instance.

**5. SYSTEM DESIGN**

**5.1 DATA FLOW DIAGRAM**

****

****

**5.2 USE CASE DIAGRAM**

System

**Users**

**Register**

**View Location Map**

**Configure Anti-Theft Settings**

**SPY Services (Background)**

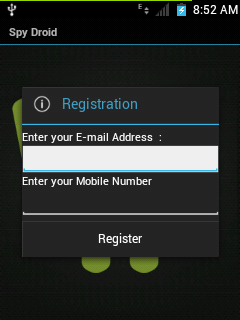
**Set Alert, Email or SMS**

**5.3 SCREET SHOTS**



**SPLASH SCREEN**

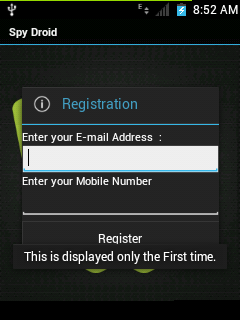
First Screen that is loaded or shown after opening the application.



REGISTRATION DIALOG

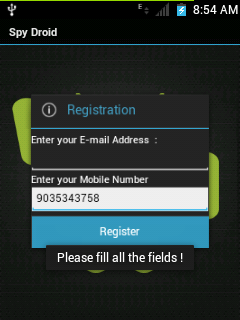
The Alert Dialog that is opened only for once to complete the Registration.

Here the alert email address and alert phone number are taken and saved.

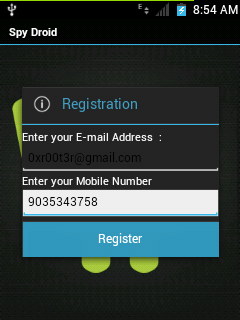


**SPLASH ACTIVITY**

Notification Showing that Registration is only for the first time.

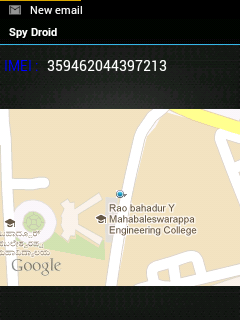


**SPLASH ACTIVITY**  
The Registration Dialog will warn the user to fill all the fields if left empty.

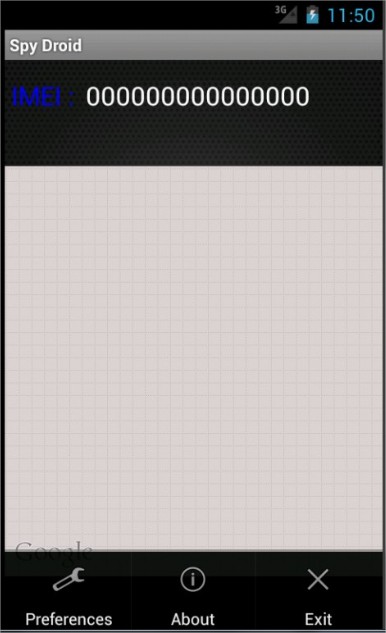


**SPLASH ACTIVITY**

After completing the Registration

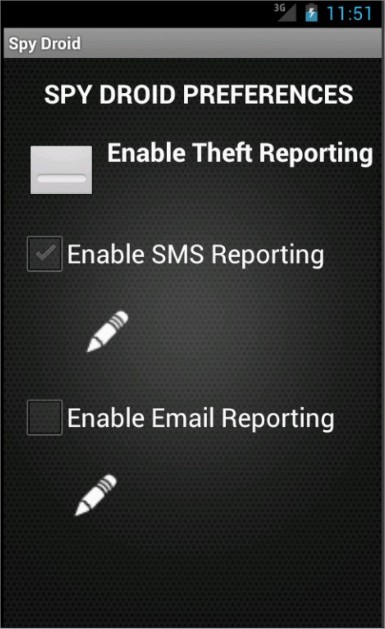


**ANTI-THEFT MAIN WINDOW**  
The Faker Module main screen to fake the victim showing the app as some Anti-theft genuine application and not a Spying app.



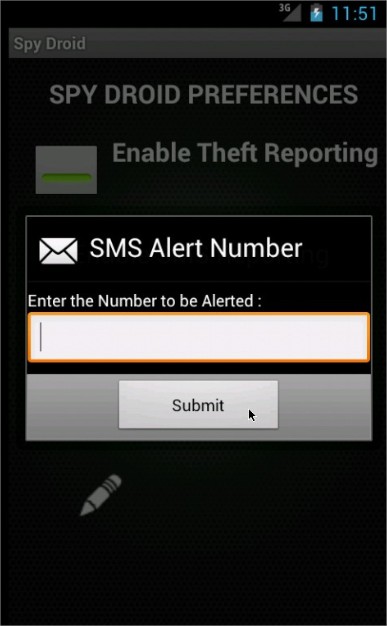
**ANTI-THEFT OPTIONS MENU**

Anti-Theft Module’s Options Menu to show the list of Options



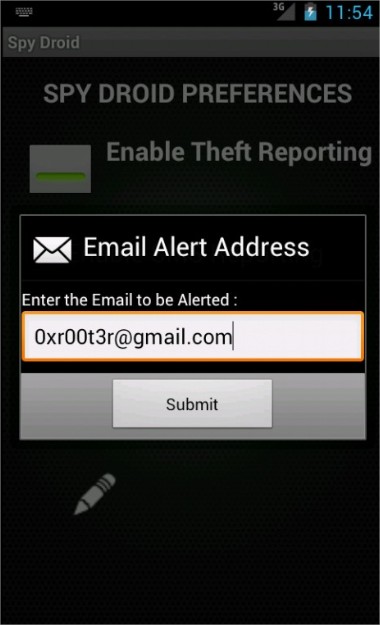
**ANTI-THEFT PREFERENCES WINDOW**

Preference Activity Window for the Anti-theft module of the SPY DROID Application in which user can configure the THEFT PREFERENCES.



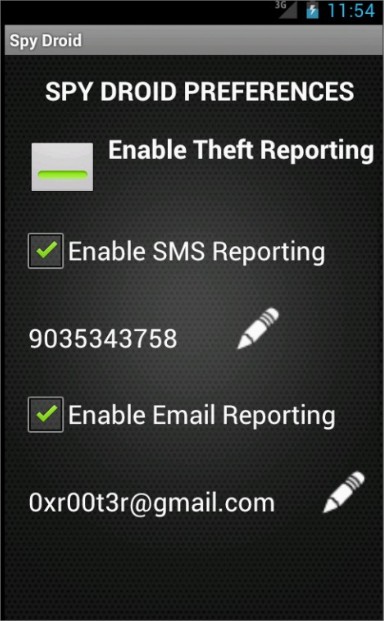
**SMS ALERT NUMBER INPUT DIALOG**

Alert Dialog Window to accept the SMS Alert Number that will be saved to report the SIM Change Activity in the application.



**EMAIL ALERT ADDRESS INPUT DIALOG**

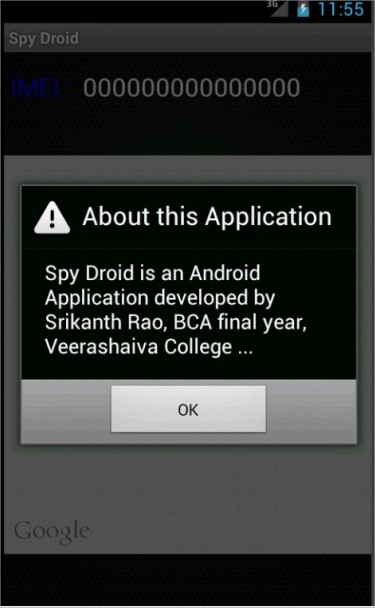
Alert Dialog Window to accept the Email Alert address that will be saved to report the SIM Change Activity in the application.



**ANTI-THEFT PREFERENCES WINDOW**

Anti-Theft Preferences Page of the Anti-Theft Module in the Application.

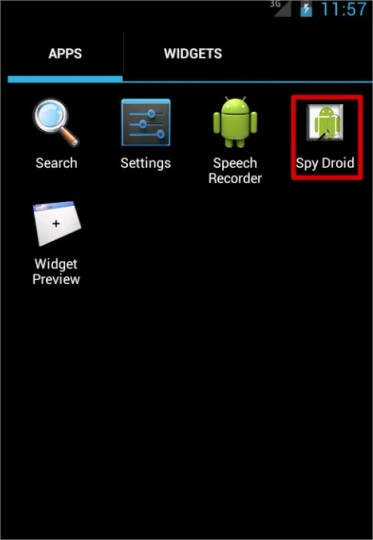
This is the view of the Window after entering the user input.



**ABOUT ALERT DIALOG**

Alert Dialog to show the information about the application to the user.

The alert dialog will be revoked when About Menu item is clicked from the Anti-Theft Module Options Menu item.



**APPLICATION ICON IN THE LAUNCHER MENU OF THE MOBILE**

This is the Menu of the Mobile phone and the spotted icon is application icon for SPY DROID which will initialize the application.

**6. IMPLEMENTATION**

**6.1 CODE EXPLANATION**

SPY DROID Application’s API Code Documentation contains:

* **Package**

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain six categories:

* + Interfaces (italic)
  + Classes
  + Enums
  + Exceptions
  + Errors
  + Annotation Types
* **Class/Interface**

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

* + Class inheritance diagram
  + Direct Subclasses
  + All Known Subinterfaces
  + All Known Implementing Classes
  + Class/interface declaration
  + Class/interface description
  + Nested Class Summary
  + Field Summary
  + Constructor Summary
  + Method Summary
  + Field Detail
  + Constructor Detail
  + Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

* **Annotation Type**

Each annotation type has its own separate page with the following sections:

* + Annotation Type declaration
  + Annotation Type description
  + Required Element Summary
  + Optional Element Summary
  + Element Detail
* **Use**

Each documented package, class and interface has its own Use page. This page describes what packages, classes, methods, constructors and fields use any part of the given class or package. Given a class or interface A, its Use page includes subclasses of A, fields declared as A, methods that return A, and methods and constructors with parameters of type A. You can access this page by first going to the package, class or interface, then clicking on the "Use" link in the navigation bar.

* **Tree (Class Hierarchy)**

There is a [Class Hierarchy](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\overview-tree.html) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with java.lang.Object. The interfaces do not inherit from java.lang.Object.

* + When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
  + When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.
* **Deprecated API**

The [Deprecated API](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\deprecated-list.html) page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

* **Index**

The [Index](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\index-files\index-1.html) contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

* **Constant Field Values**

The [Constant Field Values](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\constant-values.html) page lists the static final fields and their values.

*SPY DROID Android Application contains the below classes,*

|  |  |
| --- | --- |
| **Class** | **Description** |
| [AntiTheftMailSenderService](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftMailSenderService.html) | AntiTheft Mail Sender Service is a Service that sends the After Theft details to the user either by e-mail or SMS. |
| [AntiTheftPreferences](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html) | Activity to show the Anti-Theft Preferences. |
| [AntiTheftService](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html) | Anti Theft Service is a service to check for the theft of the device and return the result |
| [BootReceiver](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\BootReceiver.html) | Boot Receiver is a class of Android Component type Broadcast Receiver which initiates on Receiving some Intent. |
| [CallListeners](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\CallListeners.html) | Call Listener Service is a service used to SPY the Call logs from the device. |
| [CurrentLocationOverlay](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\CurrentLocationOverlay.html) | Current Location Overlay is a class extending Itemized Overlay from Google Maps API which helps to add a custom icon or custom image on the Current location Marker on the Google Map |
| [DataBase](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html) | DataBase is a generic class for the SQLite Database creation in Android and is used to access the SQLite Database features with parameters and making it as Generic for all Database access types in the Application |
| [LocationsListener](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\LocationsListener.html) | Location Listener is a class used to listen or track the locations of the user and storing them to the database |
| [Mail](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\Mail.html) | Mail is a predefined class provided by Google to create a socket connection and send the mail using SMTP |
| [MainActivity](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html) | Activity to display the main page of the application. |
| [MainService](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainService.html) | Main Service is a Android service to start all other Services of the application |
| [RegistrationDialog](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\RegistrationDialog.html) | An alert dialog for user registration purpose. |
| [RootService](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\RootService.html) | Root Service is the main Service in Spy Droid Application which starts the Main Service which inturn starts all other Services |
| [SimChangeChecker](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SimChangeChecker.html) | Sim Changer Service is a class which has methods which checks for the Sim Change Activity and initiates the Anti Theft Service |
| [SMSListeners](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html) | SMS Listener Service is a service used to SPY the SMS logs from the device. |
| [SplashActvity](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html) | Activity used to show the Splash Screen. |
| [SpyDroidMailSenderService](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SpyDroidMailSenderService.html) | Spy Droid Mail Sender Service is a Service that sends the SPY data to the registrar either by e-mail or SMS. |
| [WriteDataBasetoXML](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\WriteDataBasetoXML.html) | Write DataBase to XML is a class which acts as a Service in the Application and helps in writing the spy data from the database to an XML file. |
| [XmlBuilder](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html) | Xml Builder is a class which creates XML files of the SPY data by getting the logs from the Database in which the SPY data is stored. |

|  |
| --- |
| **Generic Classes in** [**com.innolabs.spydroid**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\package-summary.html) **used by SPY DROID** |
| [RegistrationDialog](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\class-use\RegistrationDialog.html#com.innolabs.spydroid) An alert dialog for user registration purpose. |
| [DataBase](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\class-use\DataBase.html#com.innolabs.spydroid) DataBase is a generic class for the SQLite Database creation in Android and is used to access the SQLite Database features with parameters and making it as Generic for all Database access types in the Application |
| [XmlBuilder](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\class-use\XmlBuilder.html#com.innolabs.spydroid) Xml Builder is a class which creates XML files of the SPY data by getting the logs from the Database in which the SPY data is stored. |

**6.2 SAMPLE CODE**

**package** com.innolabs.spydroid;

**import** java.io.IOException;

**import** java.util.List;

**import** android.app.Service;

**import** android.content.ContentValues;

**import** android.content.Context;

**import** android.content.Intent;

**import** android.location.Address;

**import** android.location.Criteria;

**import** android.location.Geocoder;

**import** android.location.Location;

**import** android.location.LocationManager;

**import** android.os.IBinder;

**import** android.util.Log;

**public** **class** LocationsListener **extends** Service {

/\*\* Static variable dataBase of type DataBase \*/

**static** DataBase *dataBase*;

/\*\* List<Address> variable which stores the location of the device \*/

List<Address> locationdetails;

/\*\* String value to store the address of the device \*/

String address = **null**;

/\*\* String variable for Database table name \*/

String locationsTable = "LOCATIONS";

/\*\* String value to show the TAG for the activity in the LOG CAT \*/

**public** **static** String *GPS\_TAG* = "SPY DROID - LOCATION LISTENER SERVICE";

@Override

**public** **int** onStartCommand(Intent intent, **int** flags, **int** startId) {

Log.*v*(*GPS\_TAG*, "Location Listener Service Started");

// Creating or Opening the database and getting Write Permissions

*dataBase* = **new** DataBase(**this**, "SPY\_DROID");

*dataBase*.getWritableDatabase();

Log.*v*(*GPS\_TAG*, "Database Opened or Created");

// Creating or Opening table

*dataBase*.createTable(locationsTable, getTableValues());

Log.*v*(*GPS\_TAG*, "Opening or Creating Table : "+locationsTable);

// Deleting the old table rows if exists

*dataBase*.deleteTableElements("LOCATIONS");

// Getting Location

getLocation(LocationsListener.**this**);

Log.*v*(*GPS\_TAG*, "Getting Location");

// Closing the database

*dataBase*.close();

**return** **super**.onStartCommand(intent, flags, startId);

}

**private** **void** getLocation(Context context) {

Geocoder geoCoder;

String bestProvider;

**double** lattitude = 0, longitude = 0 ;

// Accessing Location Manger

LocationManager lm =

(LocationManager) context.getSystemService(*LOCATION\_SERVICE*);

Log.*v*(*GPS\_TAG*, "Accessing Location Manager");

// Getting Best Provider

Criteria cr = **new** Criteria();

bestProvider = lm.getBestProvider(cr, **false**);

Location location = lm.getLastKnownLocation(bestProvider);

// Checking if location is not available

**if**(location == **null**)

{

**return**;

}

**else**

{

geoCoder = **new** Geocoder(context);

**try**

{

// Getting Location details

locationdetails = geoCoder.getFromLocation(location.getLatitude(), location.getLongitude(), 1);

// Saving the retrieved location details

lattitude = locationdetails.get(0).getLatitude();

longitude = locationdetails.get(0).getLongitude();

address = locationdetails.get(0).getAddressLine(0).toString()+" , "+locationdetails.get(0).getAddressLine(1).toString()+" , "+locationdetails.get(0).getAddressLine(2).toString();

Log.*v*(*GPS\_TAG*, "Locatin Details : "+address);

// Inserting the location values into database table

ContentValues gps\_contents = **new** ContentValues();

gps\_contents.put("time", System.*currentTimeMillis*());

gps\_contents.put("lattitude", lattitude);

gps\_contents.put("longitude", longitude);

gps\_contents.put("address", address);

*dataBase*.insertValues(locationsTable, gps\_contents);

Log.*v*(*GPS\_TAG*, "Location details stored to database");

} **catch** (IOException e) {

// Printing the Exceptions

e.printStackTrace();

}

}

}

**private** String getTableValues() {

String tablevalues = "SlNo INTEGER PRIMARY KEY,time TEXT, lattitude TEXT,longitude TEXT, address TEXT";

**return** tablevalues;

}

}

**6.3 API DOCUMENTATION**

**6.3.1 IMPORTANT DEFINATIONS**

**ACTIVITY**: An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window in which one can place your UI. While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating or embedded inside of another activity (using [ActivityGroup](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\android\app\ActivityGroup.html)).

**SERVICE**: A Service is an application component representing either an application's desire to perform a longer-running operation while not interacting with the user or to supply functionality for other applications to use.

**6.3.2 FLOW AND API DOCUMENTATION**

As soon as the app is started SPLASH ACTIVITY class will be executed. SPLASH Activity is an Android Activity which has a Splash Screen and waits for the user touch input and checks if the app is configured at its first run or not. If it is First Run after Installation then REGISTRATION DIALOG class will be opened else MAIN ACTIVITY class will be started.

## CLASS SPLASHACTVITY

* [java.lang.Object](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\Object.html?is-external=true)
  + Activity
    - com.innolabs.spydroid.SplashActvity
* All Implemented Interfaces: [Serializable](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\io\Serializable.html?is-external=true)

public class SplashActvity extends Activity implements [Serializable](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\io\Serializable.html?is-external=true)

Activity used to show the Splash Screen. This is an activity which shows up firstwhen a user opens the application.

### Field Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Field and Description** |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**email**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html#email) Variable to store email value. |
| (package private) boolean | [**isFirstInstallation**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html#isFirstInstallation) Boolean to check whether the runtime is the first time after the installation or not. |
| (package private) ImageView | [**iv**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html#iv) ImageView element used to add the background which is used to listening the touch interactions. |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**mobile**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html#mobile) Variable to store the mobile phone number value. |
| (package private) [RegistrationDialog](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\RegistrationDialog.html) | [**registrationDialog**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html#registrationDialog) Variable to type Registration Dialog initiated to null. |
| (package private) SharedPreferences | [**sd\_sharedprefs**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html#sd_sharedprefs) Shared Preference used to store the isFirstInstallation value for the next run. |
| (package private) SharedPreferences.Editor | [**sd\_sharedprefseditor**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html#sd_sharedprefseditor) Shared Preference editor to edit the values in Shared Preferneces. |
| (package private) SharedPreferences | [**sd\_sharedprefsMail**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html#sd_sharedprefsMail) Shared Preference used to store the email, password, and mobile number to which the spy data will be sent. |
| private static long | [**serialVersionUID**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html#serialVersionUID) Auto-generated constant value. |
| static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**SPLASH\_TAG**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html#SPLASH_TAG) String value to show the TAG for the activity in the LOG CAT |

### 

### Method Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Method and Description** |
| protected void | [**onCreate**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SplashActvity.html#onCreate%28Bundle%29)(Bundle savedInstanceState)  Method to initialize the view and set listeners for touch interactions. |

### Field Detail

### MAINACTIVITY\_TAG

public static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) MAINACTIVITY\_TAG

String value to show the TAG for the activity in the LOG CAT

#### tv\_imei

TextView tv\_imei

Text View to display the IMEI number of the device.

#### telephoneManager

TelephonyManager telephoneManager

Telephony Manager to access the phone details which are stored to check for the theft activity.

#### mapView

MapView mapView

MapView displays the map on the Activity screen.

#### geoCoder

Geocoder geoCoder

GeoCoder variable is used in the process of transforming a (latitude, longitude) coordinate into a (partial) address.

#### bestProvider

[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) bestProvider

String value of the bestProvider.

#### lattitude

double lattitude

Double value to store the latitude value.

#### longitude

double longitude

Double value to store the longitude value.

### Method Detail

#### onCreate

protected void onCreate(Bundle savedInstanceState)

Method to initialize and add some view to the user interface.

Parameters:

savedInstanceState - State which is saved and retrieved when an application is paused and re-opened.

#### onLocationChanged

public void onLocationChanged(Location location)

Method which is called when the user location has been changed.

Parameters:

location - The new value of location to where the user has been moved from old location.

#### onProviderDisabled

public void onProviderDisabled([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) provider)

Method initiated when the provider is disabled

Parameters:

provider - String value of the provider.

#### onProviderEnabled

public void onProviderEnabled([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) provider)

Method initiated when the provider is enabled

Parameters:

provider - String value of the provider.

#### onStatusChanged

public void onStatusChanged([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) provider, int status, Bundle extras)

Method to check for the status change

Parameters:

provider - String value of the provider.

status - The status value for the map.

extras - Bundle value which is not of any specific type and are incoming.

#### onPause

protected void onPause()

Method which is called when the application window i.e activity window is closed or minimized

#### onCreateOptionsMenu

public boolean onCreateOptionsMenu(Menu menu)

Method to add menu for the Activity. This method is used to add the menu items to the activity page i.e Settings option for the Activity.

Parameters:

menu - The menu varaible that is used to add options for the Menu table.

returns - Menu Items

#### onOptionsItemSelected

public boolean onOptionsItemSelected(MenuItem item)

Method to add listener to the menu item selection panel

Parameters:

item - The menu item which is clicked.

returns - Option Item which was selected

#### isRouteDisplayed

protected boolean isRouteDisplayed()

Predefined method to check whether the route in the map is displayed or not.

## CLASS REGISTRATIONDIALOG

* [java.lang.Object](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\Object.html?is-external=true)
  + - com.innolabs.spydroid.RegistrationDialog

public class RegistrationDialog extends [Object](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\Object.html?is-external=true)

An alert dialog for user registration purpose.

### Field Summary

|  |  |
| --- | --- |
| **Modifier and Type** | **Field and Description** |
| static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**REGDIALOG\_TAG**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\RegistrationDialog.html#REGDIALOG_TAG)  String value to show the TAG for the activity in the LOG CAT |
| (package private) SharedPreferences | [**sd\_sharedprefs**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\RegistrationDialog.html#sd_sharedprefs)  Shared preferences to save the Registration values |
| (package private) SharedPreferences.Editor | [**sd\_sharedprefseditor**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\RegistrationDialog.html#sd_sharedprefseditor)  Shared preferences editor variable to edit the Shared Preferences. |

### Method Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Method and Description** |
| void | [**Registration\_Dialog**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\RegistrationDialog.html#Registration_Dialog%28Context%29)(Context context)  Registration Dialog method is an Alert Dialog to get the registration values from the user only for the first time after installation. |

### Field Detail

#### REGDIALOG\_TAG

public static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) REGDIALOG\_TAG

String value to show the TAG for the activity in the LOG CAT

#### sd\_sharedprefs

SharedPreferences sd\_sharedprefs

Shared preferences to save the Registration values

#### sd\_sharedprefseditor

SharedPreferences.Editor sd\_sharedprefseditor

Shared preferences editor variable to edit the Shared Preferences.

### Method Detail

#### Registration\_Dialog

public void Registration\_Dialog(Context context)

Registration Dialog method is an Alert Dialog to get the registration values from the user only for the first time after installation.

Parameters:

context - The context of the Class i.e. Here the Splash Activity's context.

## CLASS MAINACTIVITY

* [java.lang.Object](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\Object.html?is-external=true)
  + MapActivity
    - com.innolabs.spydroid.MainActivity

public class MainActivity extends MapActivity

Activity to display the main page of the application. This is the activity which is displayed to the user after the Splash Screen Activity.

[**MapActivity**](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\add-ons\addon-google_apis-google-8\docs\reference\com\google\android\maps\MapActivity.html?is-external=true): Base class with code to manage the boring necessities of any activity that displays a [MapView](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\add-ons\addon-google_apis-google-8\docs\reference\com\google\android\maps\MapView.html). Activity responsibilities include:

* Activity lifecycle management
* Setup and teardown of services behind a [MapView](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\add-ons\addon-google_apis-google-8\docs\reference\com\google\android\maps\MapView.html)

This is not a standard package in the Android library. In order to use it, following XML element is to be added, as a child of the application element, in AndroidManifest.xml file:

**<uses-library android:name="com.google.android.maps" />**

Only one MapActivity is supported per process. Multiple MapActivities running simultaneously are likely to interfere in unexpected and undesired ways.

[**LocationListener**](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\android\location\LocationListener.html?is-external=true): Used for receiving notifications from the LocationManager when the location has changed. These methods are called if the LocationListener has been registered with the location manager service using the [requestLocationUpdates(String, long, float, LocationListener)](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\android\location\LocationManager.html#requestLocationUpdates%28java.lang.String,%20long,%20float,%20android.location.LocationListener%29) method.

### Field Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Field and Description** |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**bestProvider**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#bestProvider)  String value of the bestProvider. |
| (package private) Geocoder | [**geoCoder**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#geoCoder)  GeoCoder variable is used in the process of transforming a (latitude, longitude) coordinate into a (partial) address. |
| (package private) double | [**lattitude**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#lattitude)  Double value to store the latitude value. |
| (package private) double | [**longitude**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#longitude)  Double value to store the longitude value. |
| static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**MAINACTIVITY\_TAG**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#MAINACTIVITY_TAG)  String value to show the TAG for the activity in the LOG CAT |
| (package private) MapView | [**mapView**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#mapView)  MapView displays the map on the Activity screen. |
| (package private) TelephonyManager | [**telephoneManager**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#telephoneManager)  Telephony Manager to access the phone details which are stored to check for the theft activity. |
| (package private) TextView | [**tv\_imei**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#tv_imei)  Text View to display the IMEI number of the device. |

### Method Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Method and Description** |
| protected boolean | [**isRouteDisplayed**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#isRouteDisplayed%28%29)()  Predefined method to check whether the route in the map is displayed or not. |
| protected void | [**onCreate**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#onCreate%28Bundle%29)(Bundle savedInstanceState)  Method to initialize and add some view to the user interface. |
| boolean | [**onCreateOptionsMenu**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#onCreateOptionsMenu%28Menu%29)(Menu menu)  Method to add menu for the Activity. |
| void | [**onLocationChanged**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#onLocationChanged%28Location%29)(Location location)  Method which is called when the user location has been changed. |
| boolean | [**onOptionsItemSelected**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#onOptionsItemSelected%28MenuItem%29)(MenuItem item)  Method to add listener to the menu item selection panel |
| protected void | [**onPause**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#onPause%28%29)()  Method which is called when the application window i.e activity window is closed or minimized |
| void | [**onProviderDisabled**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#onProviderDisabled%28java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) provider)  Method initiated when the provider is disabled |
| void | [**onProviderEnabled**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#onProviderEnabled%28java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) provider)  Method initiated when the provider is enabled |
| void | [**onStatusChanged**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\MainActivity.html#onStatusChanged%28java.lang.String,%20int,%20Bundle%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) provider, int status, Bundle extras)  Method to check for the status change |

### Field Detail

#### MAINACTIVITY\_TAG

public static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) MAINACTIVITY\_TAG

String value to show the TAG for the activity in the LOG CAT

#### tv\_imei

TextView tv\_imei

Text View to display the IMEI number of the device.

#### telephoneManager

TelephonyManager telephoneManager

Telephony Manager to access the phone details which are stored to check for the theft activity.

#### mapView

MapView mapView

MapView displays the map on the Activity screen.

#### geoCoder

Geocoder geoCoder

GeoCoder variable is used in the process of transforming a (latitude, longitude) coordinate into a (partial) address.

#### bestProvider

[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) bestProvider

String value of the bestProvider.

#### lattitude

double lattitude

Double value to store the latitude value.

#### longitude

double longitude

Double value to store the longitude value.

### Method Detail

#### onCreate

protected void onCreate(Bundle savedInstanceState)

Method to initialize and add some view to the user interface.

Parameters:

savedInstanceState - State which is saved and retrieved when an application is paused and re-opened.

#### onLocationChanged

public void onLocationChanged(Location location)

Method which is called when the user location has been changed.

Parameters:

location - The new value of location to where the user has been moved from old location.

#### onProviderDisabled

public void onProviderDisabled([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) provider)

Method initiated when the provider is disabled

Parameters:

provider - String value of the provider.

#### onProviderEnabled

public void onProviderEnabled([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) provider)

Method initiated when the provider is enabled

Parameters:

provider - String value of the provider.

#### onStatusChanged

public void onStatusChanged([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) provider, int status, Bundle extras)

Method to check for the status change

Parameters:

provider - String value of the provider.

status - The status value for the map.

extras - Bundle value which is not of any specific type and are incoming.

#### onPause

protected void onPause()

Method which is called when the application window i.e activity window is closed or minimized

#### onCreateOptionsMenu

public boolean onCreateOptionsMenu(Menu menu)

Method to add menu for the Activity. This method is used to add the menu items to the activity page i.e Settings option for the Activity.

Parameters:

menu - The menu varaible that is used to add options for the Menu table.

returns: Menu Items

#### onOptionsItemSelected

public boolean onOptionsItemSelected(MenuItem item)

Method to add listener to the menu item selection panel

Parameters:

item - The menu item which is clicked.

returns: Option Item which was selected

#### isRouteDisplayed

protected boolean isRouteDisplayed()

Predefined method to check whether the route in the map is displayed or not.

As the Map View is displayed once the registration is completed and Main Activity is displayed, the application will be having an Options Menu through which one can configure the anti-theft properties for the application. The first item in the Options Menu of the Main Activity is Anti-Theft Preferences.

## CLASS ANTITHEFTPREFERENCES

* [java.lang.Object](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\Object.html?is-external=true)
  + Activity
    - com.innolabs.spydroid.AntiTheftPreferences

public class AntiTheftPreferences extends Activity

Activity to show the Anti-Theft Preferences. This window displays the Settings or Preferences for the Anti-Theft module which is installed on the device.

### Field Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Field and Description** |
| (package private) SharedPreferences | [**antiTheftSharedPrefs**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#antiTheftSharedPrefs)  Shared Preference to store the Preference values |
| (package private) SharedPreferences.Editor | [**antiTheftSharedPrefsEditor**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#antiTheftSharedPrefsEditor)  Shared Preference editor variable to edit the shared preferences |
| (package private) Button | [**btn\_email\_alert\_change**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#btn_email_alert_change)  Button to change the Email alert address |
| (package private) Button | [**btn\_sms\_alert\_change**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#btn_sms_alert_change)  Button to change the SMS alert number |
| static boolean | [**check\_btn\_email**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#check_btn_email)  Boolean value to store the check value of the Emial Reporting Check box |
| static boolean | [**check\_btn\_sms**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#check_btn_sms)  Boolean value to store the check value of the SMS Reporting Check box |
| (package private) CheckBox | [**chk\_box\_email\_reporting**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#chk_box_email_reporting)  Check box for enabling or disabling the Email Reporting |
| (package private) CheckBox | [**chk\_box\_sms\_reporting**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#chk_box_sms_reporting)  Check box for enabling or disabling the SMS Reporting |
| static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**PREFERENCE\_TAG**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#PREFERENCE_TAG)  String value to show the TAG for the activity in the LOG CAT |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**reporting\_email**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#reporting_email)  String value to store the reporting email address |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**reporting\_number**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#reporting_number)  String value to store the reporting phone number |
| (package private) ToggleButton | [**tgl\_btn\_reporting**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#tgl_btn_reporting)  Toggle button for enabling or disabling the Anti-Theft Management |
| static boolean | [**toggle\_btn\_value**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#toggle_btn_value)  Boolean value to store the check value of the Toggle Button |
| (package private) TextView | [**tv\_email\_alert\_address**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#tv_email_alert_address)  Text View for showing the Email alert address |
| (package private) TextView | [**tv\_sms\_alert\_number**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#tv_sms_alert_number)  Text View for showing the SMS alert number |
|  |

### Method Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Method and Description** |
| private void | [**intializeViews**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#intializeViews%28%29)()  Method to register the views inside the Activity for which actions will be given |
| void | [**onClick**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#onClick%28View%29)(View v)  Method to do some action on Click activity of the element in the Activity window |
| protected void | [**onCreate**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#onCreate%28Bundle%29)(Bundle savedInstanceState)  Method to initialize and add some view to the user interface. |
| protected void | [**onPause**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftPreferences.html#onPause%28%29)()  Method which is called when the application window i.e activity window is closed or minimized |

### Field Detail

#### tgl\_btn\_reporting

ToggleButton tgl\_btn\_reporting

Toggle button for enabling or disabling the Anti-Theft Management

#### chk\_box\_sms\_reporting

CheckBox chk\_box\_sms\_reporting

Check box for enabling or disabling the SMS Reporting

#### chk\_box\_email\_reporting

CheckBox chk\_box\_email\_reporting

Check box for enabling or disabling the Email Reporting

#### tv\_sms\_alert\_number

TextView tv\_sms\_alert\_number

Text View for showing the SMS alert number

#### tv\_email\_alert\_address

TextView tv\_email\_alert\_address

Text View for showing the Email alert address

#### btn\_sms\_alert\_change

Button btn\_sms\_alert\_change

Button to change the SMS alert number

#### btn\_email\_alert\_change

Button btn\_email\_alert\_change

Button to change the Email alert address

#### reporting\_number

[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) reporting\_number

String value to store the reporting phone number

#### reporting\_email

[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) reporting\_email

String value to store the reporting email address

#### antiTheftSharedPrefs

SharedPreferences antiTheftSharedPrefs

Shared Preference to store the Preference values

#### antiTheftSharedPrefsEditor

SharedPreferences.Editor antiTheftSharedPrefsEditor

Shared Preference editor variable to edit the shared preferences

#### toggle\_btn\_value

public static boolean toggle\_btn\_value

Boolean value to store the check value of the Toggle Button

#### check\_btn\_sms

public static boolean check\_btn\_sms

Boolean value to store the check value of the SMS Reporting Check box

#### check\_btn\_email

public static boolean check\_btn\_email

Boolean value to store the check value of the Emial Reporting Check box

#### PREFERENCE\_TAG

public static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) PREFERENCE\_TAG

String value to show the TAG for the activity in the LOG CAT

### Method Detail

#### onCreate

protected void onCreate(Bundle savedInstanceState)

Method to initialize and add some view to the user interface.

Parameters:

Bundle - Which is saved and retrieved when an application is paused and re-opened.

#### onPause

protected void onPause()

Method which is called when the application window i.e activity window is closed or minimized

#### intializeViews

private void intializeViews()

Method to register the views inside the Activity for which actions will be given

#### onClick

public void onClick(View v)

Method to do some action on Click activity of the element in the Activity window

Parameters:

v - The view which is clicked

Once the Preference section is opened, and the MAIN ACTIVITY Window is closed the Services for the Application will be started. Firstly the ROOT SERVICE will be started.

**CLASS ROOTSERVICE**

Root Service is the main Service in Spy Droid Application which starts the Main Service which in turn starts all other Services. This Root Service is the service which is started even after the System Restart and also this Service has the capability to start itself even if the Android System shuts the app down due to memory issues. This Service makes itself available in the memory and starts automatically time to time.

**CLASS MAINSERVICE**

* [java.lang.Object](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\Object.html?is-external=true)
  + Service
    - com.innolabs.spydroid.MainService

public class MainService extends Service

Main Service is the Android service to start all other Services of the application. Once the MAIN SERVICE is started it initiates Services like **CALLSLISTENER**, **SMSLISTENERS**, **LOCATIONLISTENERS**, **WRITEFILESTOXML** and **SPYDROIDMAILSENDER.**

**CLASS CALLSLISTENER**

Call Listener Service is a service used to SPY the Call logs from the device. All INCOMING, OUTGOING and MISSED calls are logged and store to a database in this service.

### Field Detail

#### dataBase

static [DataBase](file:///C:\\Users\\Mr.Rao\\Documents\\Droid%20Space\\SpyDroid\\docs\\com\\innolabs\\spydroid\\DataBase.html" \o "class in com.innolabs.spydroid) dataBase

Static variable dataBase of type DataBase

#### incomingCallsTable

[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) incomingCallsTable

String variables to store the Incoming Calls table name

#### outgoingCallsTable

[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) outgoingCallsTable

String variable to store the Outgoing Calls table name

#### missedCallsTable

[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) missedCallsTable

String variable to store the Missed Calls table name

#### CALLS\_TAG

public static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) CALLS\_TAG

String value to show the TAG for the activity in the LOG CAT

#### al\_incoming\_number

public static [ArrayList](file:///C:\\dev\\adt-bundle-windows-x86_64-20130219\\sdk\\docs\\reference\\java\\util\\ArrayList.html?is-external=true" \o "class or interface in java.util)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> al\_incoming\_number

Arraylist of String to store the Incoming call Phone Number

#### al\_incoming\_datetime

public static [ArrayList](file:///C:\\dev\\adt-bundle-windows-x86_64-20130219\\sdk\\docs\\reference\\java\\util\\ArrayList.html?is-external=true" \o "class or interface in java.util)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> al\_incoming\_datetime

Arraylist of String to store the Incoming call Date and Time

#### al\_incoming\_duration

public static [ArrayList](file:///C:\\dev\\adt-bundle-windows-x86_64-20130219\\sdk\\docs\\reference\\java\\util\\ArrayList.html?is-external=true" \o "class or interface in java.util)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> al\_incoming\_duration

Arraylist of String to store the Incoming call Duration

#### al\_outgoing\_number

public static [ArrayList](file:///C:\\dev\\adt-bundle-windows-x86_64-20130219\\sdk\\docs\\reference\\java\\util\\ArrayList.html?is-external=true" \o "class or interface in java.util)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> al\_outgoing\_number

Arraylist of String to store the Outgoing call Phone Number

#### al\_outgoing\_datetime

public static [ArrayList](file:///C:\\dev\\adt-bundle-windows-x86_64-20130219\\sdk\\docs\\reference\\java\\util\\ArrayList.html?is-external=true" \o "class or interface in java.util)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> al\_outgoing\_datetime

Arraylist of String to store the Outgoing call Date and Time

#### al\_outgoing\_duration

public static [ArrayList](file:///C:\\dev\\adt-bundle-windows-x86_64-20130219\\sdk\\docs\\reference\\java\\util\\ArrayList.html?is-external=true" \o "class or interface in java.util)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> al\_outgoing\_duration

Arraylist of String to store the Outgoing call Duration

#### al\_missed\_number

public static [ArrayList](file:///C:\\dev\\adt-bundle-windows-x86_64-20130219\\sdk\\docs\\reference\\java\\util\\ArrayList.html?is-external=true" \o "class or interface in java.util)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> al\_missed\_number

Arraylist of String to store the Missed call Phone Number

#### al\_missed\_datetime

public static [ArrayList](file:///C:\\dev\\adt-bundle-windows-x86_64-20130219\\sdk\\docs\\reference\\java\\util\\ArrayList.html?is-external=true" \o "class or interface in java.util)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> al\_missed\_datetime

Arraylist of String to store the Missed call Date and Time

#### al\_missed\_duration

public static [ArrayList](file:///C:\\dev\\adt-bundle-windows-x86_64-20130219\\sdk\\docs\\reference\\java\\util\\ArrayList.html?is-external=true" \o "class or interface in java.util)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> al\_missed\_duration

Arraylist of String to store the Missed call Duration

### Method Detail

#### onBind

public IBinder onBind(Intent intent)

Method called when BindService is called from other instance

Parameters:

intent - The Intent from which the method was called

returns: null Noting is sent back to the bindService method

#### onStartCommand

public int onStartCommand(Intent intent, int flags, int startId)

Method that starts as soon as the Calls Listener service is started. This method is used to log or spy or track all the Incoming, Outgoing and Missed calls from the device and save them to a database.

returns: The Intent, Flags and startId which was sent as arguments

#### spyCalls

private void spyCalls()

Spy Calls is a method in which the logs are tracked and stored to database

#### getTableValues

private [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) getTableValues()

Get Table values is a method to return the Column names for the Database Table

**CLASS SMSLISTENERS**

SMS Listener Service is a service used to SPY the SMS logs from the device. All INCOMING, OUTGOING SMS messages are logged and store to a database in this service.

### Field Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Field and Description** |
| static [ArrayList](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\util\ArrayList.html?is-external=true)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> | [**al\_incoming\_sms\_body**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#al_incoming_sms_body)  Arraylist of String to store the Incoming SMS Message Body |
| static [ArrayList](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\util\ArrayList.html?is-external=true)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> | [**al\_incoming\_sms\_datetime**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#al_incoming_sms_datetime)  Arraylist of String to store the Incoming SMS Date Time |
| static [ArrayList](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\util\ArrayList.html?is-external=true)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> | [**al\_incoming\_sms\_from**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#al_incoming_sms_from)  Arraylist of String to store the Incoming SMS Phone Number |
| static [ArrayList](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\util\ArrayList.html?is-external=true)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> | [**al\_outgoing\_sms\_datetime**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#al_outgoing_sms_datetime)  Arraylist of String to store the Outgoing SMS Date Time |
| static [ArrayList](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\util\ArrayList.html?is-external=true)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> | [**al\_outgoing\_sms\_to**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#al_outgoing_sms_to)  Arraylist of String to store the Outgoing SMS Phone Number |
| static [ArrayList](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\util\ArrayList.html?is-external=true)<[String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true)> | [**al\_outoing\_sms\_body**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#al_outoing_sms_body)  Arraylist of String to store the Outgoing SMS Message Body |
| (package private) static [DataBase](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html) | [**dataBase**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#dataBase)  Static variable of type DataBase |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**incomingSMSTable**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#incomingSMSTable)  String value to store Incoming SMS database table name |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**outgoingSMSTable**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#outgoingSMSTable)  String value to store Outgoing SMS database table name |
| static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**SMS\_TAG**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#SMS_TAG)  String value to show the TAG for the activity in the LOG CAT |
| (package private) Uri | [**uriIncomingSMS**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#uriIncomingSMS)  Uri value for Incoming SMS which is to be parsed |
| (package private) Uri | [**uriOutgoingSMS**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#uriOutgoingSMS)  Uri value for Outgoing SMS which is to be parsed |

### Method Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Method and Description** |
| private [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**getTableValues**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#getTableValues%28%29)()  Get Table values is a method to return the Column names for the Database Table |
| IBinder | [**onBind**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#onBind%28Intent%29)(Intent intent)  Method called when BindService is called from other instance |
| int | [**onStartCommand**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#onStartCommand%28Intent,%20int,%20int%29)(Intent intent, int flags, int startId)  Method that starts as soon as the SMS Listener service is started. |
| private void | [**spySMS**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\SMSListeners.html#spySMS%28%29)()  Spy SMS is a method in which the logs are tracked and stored to database |

**CLASS LOCATIONLISTENER**

Location Listener is a class used to listen or track the locations of the user and storing them to the database. Just like the previous CALLS LISTENER and SMS LISTENER Services, LOCATION LISTENER is also an Android Service which logs or tracks the locations of the user or the device and stores them into a database.

**CLASS WRITEDATABASETOXML**

public class WriteDataBasetoXML extends Service

Write DataBase to XML is a class which acts as a Service in the Application and helps in writing the spy data from the database to an XML file.

### Field Summary

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Modifier and Type** | | **Field and Description** | |
| private static [DataBase](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html) | | [**dataBase**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\WriteDataBasetoXML.html#dataBase)  Static variable of type DataBase to access the database | |
| (package private) static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**DATASUBDIRECTORY**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\WriteDataBasetoXML.html#DATASUBDIRECTORY)  Static string value of the folder which will be created in the application directory to store the XML files | |
| static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**WRITEDATATOXML\_TAG**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\WriteDataBasetoXML.html#WRITEDATATOXML_TAG)  String value to show the TAG for the activity in the LOG CAT | |
| private [XmlBuilder](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html) | | [**xmlBuilder**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\WriteDataBasetoXML.html#xmlBuilder)  Private variable of type XmlBuilder to build an XML from a String list with data from the database | |
|  | |

### Method Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Method and Description** |
| void | [**export**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\WriteDataBasetoXML.html#export%28java.lang.String,%20java.lang.String,%20java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) dbName, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) exportFileNamePrefix, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) tableName)  Export is a method to send the Database, Exporting file name and the table to be exported to the XML builder class which in turn saves the data to an XML document |
| private void | [**exportTable**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\WriteDataBasetoXML.html#exportTable%28java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) tableName)  Export table is a sub-method in Export main method through which the table elements are exported to a String |
| private boolean | [**InternetConnectionAvailable**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\WriteDataBasetoXML.html#InternetConnectionAvailable%28%29)()  Method to check for Internet Connectivity to send the e-mail |
| IBinder | [**onBind**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\WriteDataBasetoXML.html#onBind%28Intent%29)(Intent intent)  Method called when BindService is called from other instance |
| int | [**onStartCommand**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\WriteDataBasetoXML.html#onStartCommand%28Intent,%20int,%20int%29)(Intent intent, int flags, int startId)  Method that starts as soon as the Calls Listener service is started. |
| private void | [**writeToFile**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\WriteDataBasetoXML.html#writeToFile%28java.lang.String,%20java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) xmlString, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) exportFileName)  Write to File is a sub-method in Export method through which the exported database file is stored in an XML file |

### Field Detail

#### dataBase

private static [DataBase](file:///C:\\Users\\Mr.Rao\\Documents\\Droid%20Space\\SpyDroid\\docs\\com\\innolabs\\spydroid\\DataBase.html" \o "class in com.innolabs.spydroid) dataBase

Static variable of type DataBase to access the database

#### xmlBuilder

private [XmlBuilder](file:///C:\\Users\\Mr.Rao\\Documents\\Droid%20Space\\SpyDroid\\docs\\com\\innolabs\\spydroid\\XmlBuilder.html" \o "class in com.innolabs.spydroid) xmlBuilder

Private variable of type XmlBuilder to build an XML from a String list with data from the database

#### DATASUBDIRECTORY

static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) DATASUBDIRECTORY

Static string value of the folder which will be created in the application directory to store the XML files

#### WRITEDATATOXML\_TAG

public static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) WRITEDATATOXML\_TAG

String value to show the TAG for the activity in the LOG CAT

### Method Detail

#### onStartCommand

public int onStartCommand(Intent intent, int flags, int startId)

Method that starts as soon as the Calls Listener service is started. This method is used to write the log files from the database into an XML document

returns: The Intent, Flags and startId which was sent as arguments

#### export

public void export([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) dbName, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) exportFileNamePrefix, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) tableName) throws [IOException](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\io\IOException.html?is-external=true)

Export is a method to send the Database, Exporting file name and the table to be exported to the XML builder class which in turn saves the data to an XML document

Parameters:

dbName - String value of Database name from which the data is accessed

exportFileNamePrefix - String value for the file name with which the exported file will be named and stored

tableName - Table name of the database from which the data must be retrieved to write it in an XML file

Throws:

[IOException](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\io\IOException.html?is-external=true) - Exception which may occur due to run time error while pulling or saving the files from the Android System

#### exportTable

private void exportTable([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) tableName)throws [IOException](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\io\IOException.html?is-external=true)

Export table is a sub-method in Export main method through which the table elements are exported to a String

Parameters:

tableName - Table name of the database from which the data is extracted

Throws:

[IOException](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\io\IOException.html?is-external=true) - Input Output Exception i.e run time error if the File is not found or not accessible

#### writeToFile

private void writeToFile([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) xmlString, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) exportFileName)throws [IOException](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\io\IOException.html?is-external=true)

Write to File is a sub-method in Export method through which the exported database file is stored in an XML file

Parameters:

xmlString - String value which has the single row string from the database table

exportFileName - File name for the exported XML file

Throws:

[IOException](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\io\IOException.html?is-external=true) - Input Output Exception that may occur during accessing the Files i.e either writing into a file or storing the file

#### InternetConnectionAvailable

private boolean InternetConnectionAvailable()

Method to check for Internet Connectivity to send the e-mail

Returns: Boolean value whether Internet connection is available or not

#### onBind

public IBinder onBind(Intent intent)

Method called when BindService is called from other instance

Parameters:

intent - The Intent from which the method was called

returns: null Noting is sent back to the bindService method

For the creation of XML files in the above class, XML Builder class is used. XML Builder class acts as a Generic class which has the methods to convert a SQL output into a String and then build an XML out of it.

**CLASS XMLBUILDER**

public class XmlBuilder extends [Object](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\Object.html?is-external=true)

Xml Builder is a class which creates XML files of the SPY data by getting the logs from the Database in which the SPY data is stored.

### Field Summary

|  |  |  |
| --- | --- | --- |
|  | | |
| **Modifier and Type** | | **Field and Description** |
| private static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**CLOSE\_WITH\_TICK**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#CLOSE_WITH_TICK)  Static string for adding the closing tick for ending the tag in XML file |
| private static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**COL\_CLOSE**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#COL_CLOSE)  Static string for closing the column item tag in the XML file |
| private static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**COL\_OPEN**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#COL_OPEN)  Static string for opening the column item tag in the XML file |
| private static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**DB\_CLOSE**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#DB_CLOSE)  Static string for closing database tag in the XML file |
| private static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**DB\_OPEN**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#DB_OPEN)  Static string for opening database tag in the XML file |
| private static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**OPEN\_XML\_STANZA**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#OPEN_XML_STANZA)  Static string for adding the XML Stanza in the XML file |
| private static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**ROW\_CLOSE**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#ROW_CLOSE)  Static string for closing the database table row tag in the XML file |
| private static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**ROW\_OPEN**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#ROW_OPEN)  Static string for opening a database table row tag in the XML file |
| private [StringBuilder](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\StringBuilder.html?is-external=true) | | [**sb**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#sb)  String Builder file to get the String from Database and append them into the XML file |
| private static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**TABLE\_CLOSE**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#TABLE_CLOSE)  Static string for closing the database table name tag in the XMl file |
| private static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | | [**TABLE\_OPEN**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#TABLE_OPEN)  Static string for opening database table name tag in the XML file |
|  |

### Method Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Method and Description** |
| (package private) void | [**addColumn**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#addColumn%28java.lang.String,%20java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) name, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) val)  Method to add Column tag file to the XML file |
| (package private) void | [**closeRow**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#closeRow%28%29)()  Method to add Table name closing tag into the XML file |
| (package private) void | [**closeTable**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#closeTable%28%29)()  Method to add Table name closing tag into the XML file |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**end**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#end%28%29)()  Method that ends the Database Tag in the XML file |
| (package private) void | [**openRow**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#openRow%28%29)()  Method to add row name tag into the XML file |
| (package private) void | [**openTable**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#openTable%28java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) tableName)  Method to add Table name tag into the XML file |
| (package private) void | [**start**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\XmlBuilder.html#start%28java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) dbName)  Method to Start creation of XML file by adding STANZA file and Main Database TAG into the XML file |

The application has a feature or functionality to wait for other services or processes to complete, i.e. The Mail Sending Service waits for the completion of Write Database to XML Service and will start only after the completion of its previous services.

**CLASS SPYDROIDMAILSENDERSERVICE**

public class SpyDroidMailSenderService extends Service

Spy Droid Mail Sender Service is a Service that sends the SPY data to the registrar either by e-mail or SMS.

### Method Detail

#### onStartCommand

public int onStartCommand(Intent intent, int flags, int startId)

Method that is called as soon as the service is started. This method is used to send the spy data either through Email or SMS

Returns:

The Intent, Flags and startId which was sent as arguments

#### sendEmail

private void sendEmail()

Send Email Method to send Email with all the SPY data if Internet Connection is available

#### sendSMS

private void sendSMS([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) PhoneNumber, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) Message)

Send SMS method to send SMS if Internet Not available

Parameters:

PhoneNumber - Phone Number to which SMS is sent

Message - Message body that is sent in the SMS

#### getLatestOutgoingCalls

private [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) getLatestOutgoingCalls([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) outGoingCalls)

Getting Latest Outgoing Calls Method to get the latest 3 Outgoing Calls

Parameters:

outGoingCalls - String variable for Outgoing Calls Table name

Returns:

String value of Message built from String Builder class

#### getLatestIncomingCalls

private [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) getLatestIncomingCalls([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) inComingCalls)

#### InternetConnectionAvailable

private boolean InternetConnectionAvailable()

Method to check for Internet Connectivity to send the e-mail

Returns:

Boolean value whether Internet connection is available or not

To save the data into a database a generic database class is created and through that class Saving and Retrieving of data throughout the database is achieved.

**CLASS DATABASE**

* [java.lang.Object](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\Object.html?is-external=true)
  + SQLiteOpenHelper
    - com.innolabs.spydroid.DataBase

public class DataBase extends SQLiteOpenHelper

DataBase is a generic class for the SQLite Database creation in Android and is used to access the SQLite Database features with parameters and making it as Generic for all Database access types in the Application

[SQLiteOpenHelper](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\android\database\sqlite\SQLiteOpenHelper.html?is-external=true) : A helper class to manage database creation and version management.

### Field Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Field and Description** |
| static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**DATABASE\_TAG**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#DATABASE_TAG)  String value to show the TAG for the activity in the LOG CAT |
| (package private) static Context | [**dbcontext**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#dbcontext)  Static variable for the Context of the Database |
| [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**SlNo**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#SlNo)  String value used as a primary key for all the database tables created using @link DataBase class |

### Constructor Summary

|  |
| --- |
|  |
| **Constructor and Description** |
| [**DataBase**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#DataBase%28Context,%20java.lang.String%29)(Context context, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) name)  Simple Constructor for the DataBase class |

### Method Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Method and Description** |
| void | [**createTable**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#createTable%28java.lang.String,%20java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) table, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) Values)  Method used to create a new table if it does not exist in a Database |
| void | [**deleteRecord**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#deleteRecord%28java.lang.String,%20java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) table, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) id)  Delete Record method is called if it is needed delete a particular row from a database table |
| void | [**deleteTableElements**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#deleteTableElements%28java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) tableName)  Method used to delete all the rows from a database table |
| void | [**deleteValues**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#deleteValues%28java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) tableName)  Delete values method called to delete the database tables |
| Cursor | [**execSQL**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#execSQL%28java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) sql)  Method used to execute raw queries to the database |
| Cursor | [**getValues**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#getValues%28java.lang.String%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) table)  Method to retreive the rows from a database table |
| void | [**insertValues**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#insertValues%28java.lang.String,%20ContentValues%29)([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) table, ContentValues contentval)  Method called to insert values into a database table |
| void | [**onCreate**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#onCreate%28SQLiteDatabase%29)(SQLiteDatabase db)  onCreate is a method in DataBase used to execSQL commands as rawQueries |
| void | [**onUpgrade**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\DataBase.html#onUpgrade%28SQLiteDatabase,%20int,%20int%29)(SQLiteDatabase db, int oldVersion, int newVersion)  onUpgrade method is called when the database is upgraded and need to be update in our application |

### Field Detail

#### SlNo

public [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) SlNo

String value used as a primary key for all the database tables created using DataBase class

#### dbcontext

static Context dbcontext

Static variable for the Context of the Database

#### DATABASE\_TAG

public static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) DATABASE\_TAG

String value to show the TAG for the activity in the LOG CAT

### Constructor Detail

#### DataBase

public DataBase(Context context, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) name)

Simple Constructor for the DataBase class

Parameters:

context - The context through which the database is accessed

name - The name of the Database that is to be accessed

### Method Detail

#### onCreate

public void onCreate(SQLiteDatabase db)

onCreate is a method in DataBase used to execSQL commands as rawQueries

#### createTable

public void createTable([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) table, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) Values)

Method used to create a new table if it does not exist in a Database

Parameters:

table - String value of the Table name

Values - String value for the Table Column names

#### onUpgrade

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)

onUpgrade method is called when the database is upgraded and need to be update in our application

#### insertValues

public void insertValues([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) table, ContentValues contentval)

Method called to insert values into a database table

Parameters:

table - String value of the table name in which the data must be inserted

contentval - Content Value object i.e data that is to be inserted into the database table

#### deleteValues

public void deleteValues([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) tableName)

Delete values method called to delete the database tables

#### getValues

public Cursor getValues([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) table)

Method to retreive the rows from a database table

Parameters:

table - Table name from which the rows must be returned

returns: Return the rows of the database table which was requested

#### deleteRecord

public void deleteRecord([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) table, [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) id)

Delete Record method is called if it is needed delete a particular row from a database table

Parameters:

table - table name from which the row must be deleted

id - the key i.e primary key through which the row is accessed

#### execSQL

public Cursor execSQL([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) sql)

Method used to execute raw queries to the database

Parameters:

sql - SQL command

returns: null

#### deleteTableElements

public void deleteTableElements([String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) tableName)

Method used to delete all the rows from a database table

Parameters:

tableName - String value of table name from which all the rows must be deleted

And here ends the SPY DROID Main application’s documentation. SPY DROID App for Android comes with another adavantage. It also has an Anti-Theft Module embedded in the application which is used as FAKER MODULE and also helps the user to secure his mobile phone.

Anti-Theft Service Module is a Service inside Spy Droid Application which frequently checks for the SIM card properties. On Every restart the application will check for the SIM properties and if anything is wrong i.e. the old SIM properties doesn’t match with the new SIM properties then it starts Anti-Theft Service which collects the new SIM data and then invokes Anti-Theft Mail Sender Service, which mails or sends an SMS to the subscriber informing the new SIM details and the location information of the device.

**CLASS ANTI-THEFTSERVICE**

public class AntiTheftService extends Service

Anti Theft Service is a service to check for the theft of the device and return the result

### Field Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Field and Description** |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**address**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#address)  String the stores the address i.e GPS values of the device's location |
| (package private) SharedPreferences | [**afterTheftSharedDetails**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#afterTheftSharedDetails)  Shared Preference to save the After Theft Preferences to the device |
| (package private) SharedPreferences.Editor | [**afterTheftSharedDetailsEditor**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#afterTheftSharedDetailsEditor)  Shared Preferences Editor variable to edit and save the After Theft data to Shared Preference |
| static [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**ANTITHEFT\_TAG**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#ANTITHEFT_TAG)  String value to show the TAG for the activity in the LOG CAT |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**imeiNumber**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#imeiNumber)  String to save the IMEI number of the device |
| (package private) [List](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\util\List.html?is-external=true)<Address> | [**locationdetails**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#locationdetails)  List  type variable to save the location details of the user/device |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**mobileLocation**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#mobileLocation)  String value to store the mobile location |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**operatorName**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#operatorName)  String to save the Operator name |
| (package private) SharedPreferences.Editor | [**simChangedPreferencesEditor**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#simChangedPreferencesEditor)  Shared Preferences Editor variable to edit and save the SIM card properties data to Shared Preference |
| (package private) SharedPreferences | [**simChangePreferences**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#simChangePreferences)  Shared Preference to store the SIM card Change settings |
| (package private) [String](file:///C:\dev\adt-bundle-windows-x86_64-20130219\sdk\docs\reference\java\lang\String.html?is-external=true) | [**simNumber**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#simNumber)  String to save the unique IMSI number of the SIM card |
| (package private) TelephonyManager | [**telephoneManager**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#telephoneManager)  Telephony Manager to access the Services from the device and provider |

### Method Summary

|  |  |
| --- | --- |
|  | |
| **Modifier and Type** | **Method and Description** |
| private void | [**getLocation**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#getLocation%28Context%29)(Context context)  Method used to get Location of the device |
| private boolean | [**InternetConnectionAvailable**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#InternetConnectionAvailable%28%29)()  Method to check for Internet Connectivity to send the e-mail |
| IBinder | [**onBind**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#onBind%28Intent%29)(Intent intent)  Method called when BindService is called from other instance |
| int | [**onStartCommand**](file:///C:\Users\Mr.Rao\Documents\Droid%20Space\SpyDroid\docs\com\innolabs\spydroid\AntiTheftService.html#onStartCommand%28Intent,%20int,%20int%29)(Intent intent, int flags, int startId)  Method that starts as soon as the AntiTheft service is started. |

**CLASS SIMCHANGECHECKERSERVICE**

Sim Changer Service is a class which has methods which checks for the Sim Change Activity and initiates the Anti Theft Service.

**CLASS ANTITHEFTMAILSENDERSERVICE**

AntiTheft Mail Sender Service is a Service that sends the After Theft details to the user either by e-mail or SMS.

**7. TESTING**

**7.1 UNIT TESTING**

In unit testing, various modules have been tested individually. This has been done manually to test if the expected result is actually seen on the screen. The following are test cases with the help of which the application has been tested.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Test Case Description | Expected Result | Actual Result |
| 1 | On load of splash screen | Run until home load | Pass |
| 2 | On load of home screen | Loading home Screen | Pass |
| 3 | On clicking the Splash screen to open up Registration Dialog | Showing Registration Dialog for first time | Pass |
| 4 | On clicking Splash screen second time after Registration | Loading Main Activity | Pass |
| 5 | Map View of current location | Showing current location on the Map | Pass |
| 6 | Start Services | Services started in background | Pass |
| 7 | Sending Mail or SMS | Mail is being sent | Pass |

**7.2 PERFORMANCE TESTING**

Performance testing has been done to measure the responsiveness of the application to the workload such as increasing users’ requests. The parameters were chosen randomly till the application performed consistently. The test has been done on an office Wi-Fi network with a speed of 5Mbps

**System Configuration:**

Operating System: Windows 7 64 bit and Linux 3.2.x kernel

Processor: AMD Quad core x64 Processor

RAM: 6 GB

*Splash Screen:* It is clear from the graph that the application performs consistently for 500 users as it has an average response time of 0.4 sec.

*Main Screen:* The home page was tested by analysing the performance of the Main activity page which is responsible for returning the map with current location of the user through Wi-Fi, Cellular network or GPS. It was really important to have a good response time for this screen as it is the home page. The application performs consistently for 1000 users as the average response time is 0.9 sec but it might not respond quickly enough for 2000 or more users as the average response time is as high as 1.4 sec.

*Services:* The Services was tested by analysing the performance which made the application to move smoother even while the background services are running. The application gave satisfactory results. As the application had many services, one more root service was added to start all other services. Thus the application doesn’t crash because of the extra service add.

**7.3 COMPATIBILITY TESTING**

This application was mainly designed for android phones as it helps the users to secure or monitor their children or concerned stuff. Different android phones have different screen sizes and resolution. The application has been tested for its compatibility with different screen sizes on the emulator.

**8. FUTURE ENHANCEMENTS**

As the application has its own advantages and disadvantages, the changes and future scope is very trophic for this application.

This application can be associated with different other features such as:

* Removal of FAKER module and creating 2 different application for both Master and Slave
* Adding Web Interface for the Master to check for the data even from the Website with their own Master account
* Record Calls and then include them and send as SPY Data to the Master.

**9. CONCLUSION**

This is my first attempt in developing a mobile application which gave me a basic understanding of development and challenges of mobile application development. The main aim of the project SPY DROID will be highly useful and helpful in many sectors in the current growing situation of the world. Analysing the number of information leakage from that companies and children growing or taking a wrong route and habiting bad stuff have increased a lot since last 10 years. A Parent can of course directly ask a child about his activities and stuff but we cannot assume that the children are not lying. So, an application like SPY DROID is of most use in these kinds of situations.

**10. BIBLIOGRAPHY**

Android Services and Broadcast Receiver Concepts : Google Developer Blog

User Interface from Droid Development Guide : Mark. L Murphy

Basic Concept of Listeners : **thenewboston** Youtube Channel