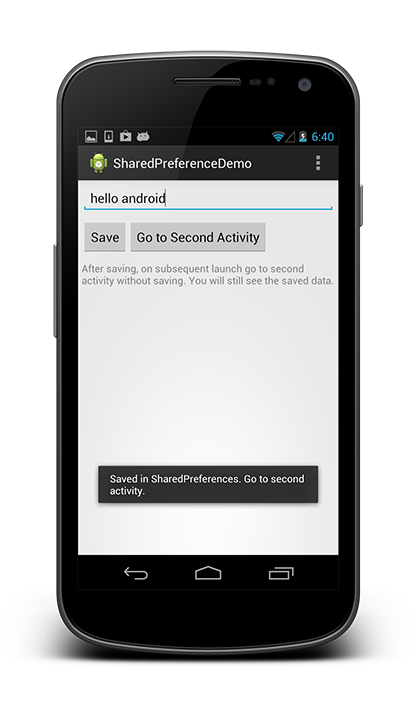
Android SharedPreferences

n this Android tutorial we are going to see how to use Android SharedPreferences class to store and retrieve application specific persistent data.

[](http://androidopentutorials.com/wp/wp-content/uploads/2014/03/1.android-sharedpreferences-example-output.png?1a8226)

## **Android SharedPreferences Tutorial**

* Android SharedPreferences allows us to store private primitive application data in the form of key-value pair.
* Android stores shared preference settings as XML file in shared\_prefs folder under DATA/data/[application package] directory. The DATA folder can be obtained by calling Environment.getDataDirectory() (usually it is /data).
* SharedPreferences is application specific, i.e.) the data is lost when you perform one of the options,
  + once you uninstall the application
  + once you clear application data (through Settings)

### **How to get SharedPreferences instance?**

To use shared preferences, you can use one of the following methods,

#### **Method 1:**

Use **SharedPreferences getSharedPreferences (String name, int mode)**. This method gets shared preferences from a specified file.



|  |  |
| --- | --- |
| 1  2 | public static final String PREFS\_NAME = "AOP\_PREFS";  SharedPreferences  settings = getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE); |

Where,

* PREFS\_NAME is the name of the file.
* Context.MODE\_PRIVATE is the operating mode.

Other modes are,

|  |  |  |
| --- | --- | --- |
| Operating Mode | Constant value | Description |
| MODE\_PRIVATE | 0 | File creation mode: the default mode, where the created file can only be accessed by the calling application. |
| MODE\_WORLD\_READABLE | 1 | This constant was deprecated in API level 17. Creating world-readable files is very dangerous, and likely to cause security holes in applications. |
| MODE\_WORLD\_WRITEABLE | 2 | This constant was deprecated in API level 17. Creating world-writable files is very dangerous, and likely to cause security holes in applications. |
| MODE\_MULTI\_PROCESS | 4 | This method will check for modification of preferences even if the sharedpreference instance has already been loaded |
| MODE\_APPEND | 32768 | This will append the new preferences with the already exisiting preferences |
| MODE\_ENABLE\_WRITE\_AHEAD\_LOGGING | 8 | Database open flag. When it is set , it would enable write ahead logging by default |

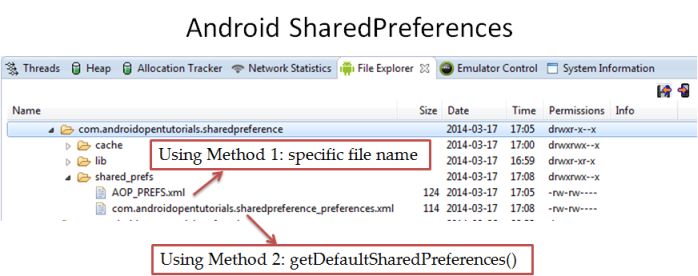
#### **Method 2:**



|  |  |
| --- | --- |
| 1 | SharedPreferences  settings = PreferenceManager.getDefaultSharedPreferences(context) |

Gets a SharedPreferences instance that points to the default file that is used by the preference framework in the given context. Here the file is stored as

DATA/data/[application package name]/shared\_prefs/[application package name]\_preferences.xml

[](http://androidopentutorials.com/wp/wp-content/uploads/2014/03/android-sharedpreferences-files.jpg?1a8226)

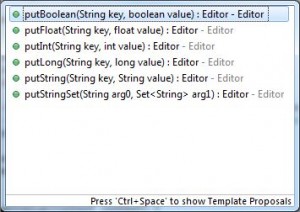
### **Store Data in SharedPreferences**

To save a value in SharedPreferences, you can use **SharedPreferences.Editor** class.  
**Steps:**

1. Get SharedPreferences instance using one of the methods explained above.
2. Get SharedPreferences.Editor instance by calling edit() method in SharedPreferences instance.
3. Store values by calling one of the putXXXX() methods.
4. Commit the editor object.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | ...  public static final String PREFS\_NAME = "AOP\_PREFS";  public static final String PREFS\_KEY = "AOP\_PREFS\_String";  ...  public void save(Context context, String text) {      SharedPreferences settings;      Editor editor;      settings = context.getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE); //1      editor = settings.edit(); //2        editor.putString(PREFS\_KEY, text); //3      editor.commit(); //4  } |

Only primitive data types can be stored in SharedPreferences. Other methods are,  
[](http://androidopentutorials.com/wp/wp-content/uploads/2014/03/android-sharedpreferences-editor-put-methods.jpg?1a8226)

### **Retrieve Data from SharedPreferences**

To get a value from shared preferences, you can use the **SharedPreferences class’ getXXXX methods** without the Editor object.  
**Steps:**

* Get SharedPreferences instance using one of the methods explained above.
* Call one of the getXXXX() methods using SharedPreferences instance.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | ...  public static final String PREFS\_NAME = "AOP\_PREFS";  public static final String PREFS\_KEY = "AOP\_PREFS\_String";  ...  public String getValue(Context context) {      SharedPreferences settings;      String text;  settings = context.getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE); //1      text = settings.getString(PREFS\_KEY, null); //2      return text;  } |

### **Clear SharedPreferences Data**

**To remove all values** from preferences use **editor.clear()** method as shown below.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | ...  public static final String PREFS\_NAME = "AOP\_PREFS";  public static final String PREFS\_KEY = "AOP\_PREFS\_String";  ...    public void clearSharedPreference(Context context) {      SharedPreferences settings;      Editor editor;        settings = context.getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE);      editor = settings.edit();        editor.clear();      editor.commit();  } |

**To remove a specific key-value pair**use **editor.remove(KEY)** method as shown below.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | ...  public static final String PREFS\_NAME = "AOP\_PREFS";  public static final String PREFS\_KEY = "AOP\_PREFS\_String";  ...    public void removeValue(Context context) {      SharedPreferences settings;      Editor editor;      settings = context.getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE);      editor = settings.edit();        editor.remove(PREFS\_KEY);      editor.commit();  } |

## **Android SharedPreferences Example**

### **Project Description**

In this Android Example, we will see how to use SharedPreferences to share data from one activity to another.

* We create a separate SharedPreference utility class with methods to save, get, clear, remove from SharedPreferences.
* We create two activities,
  + MainActivity – to save the value entered in EditText.
  + SecondActivity – to get the value from SharedPreferences and display it in TextView.

[Download “Android SharedPreferences Demo”SharedPreferenceDemo.zip – Downloaded 4582 times – 2 MB](http://androidopentutorials.com/download/android-sharedpreferences-demo/)

### **Android Project**

Create a new Android project and name it as **SharedPreferenceDemo**.

#### **Resources**

##### **strings.xml**

Open **res/values/strings.xml** and edit to have the content as shown below.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <?xml version="1.0" encoding="utf-8"?>  <resources>        <string name="app\_name">SharedPreferenceDemo</string>      <string name="action\_settings">Settings</string>      <string name="hint">Enter some text here</string>      <string name="save">Save</string>      <string name="go\_to\_second\_activity">Go to Second Activity</string>      <string name="second\_activity">Second Activity</string>      <string name="saved">Saved in SharedPreferences. Go to second activity.</string>      <string name="info">After saving, on subsequent launch go to second activity without saving. You will still see the saved data.</string>    </resources> |

##### **Layout files**

###### **activity\_main.xml**

This XML layout file (activity\_main.xml) defines an EditText and a Button which is ued by MainActivity.java. Open **activity\_main.xml file in res/layout** and copy the following content.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"      xmlns:tools="http://schemas.android.com/tools"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:padding="5dp"      tools:context=".MainActivity" >        <EditText          android:id="@+id/etxt\_text"          android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content"          android:layout\_centerHorizontal="true"          android:hint="@string/hint"          android:inputType="text" />        <Button          android:id="@+id/button\_save"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_below="@+id/etxt\_text"          android:layout\_marginTop="10dp"          android:text="@string/save" />        <Button          android:id="@+id/button\_second\_activity"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_below="@+id/etxt\_text"          android:layout\_marginTop="10dp"          android:layout\_toRightOf="@+id/button\_save"          android:text="@string/go\_to\_second\_activity" />        <TextView          android:id="@+id/txt\_info"          android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content"          android:layout\_below="@+id/button\_second\_activity"          android:layout\_marginTop="10dp"          android:hint="@string/info" />    </RelativeLayout> |

activity\_second.xml

This XML layout file (activity\_second.xml) defines a TextView which is used by SecondActivity.java. Create a new **activity\_second.xml file in res/layout** and copy the following content

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

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:padding="5dp"

    tools:context=".SecondActivity" >

    <TextView

        android:id="@+id/txt\_text"

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_centerHorizontal="true" />

</RelativeLayout>

#### **Source files**

##### **SharedPreference class**

Create a new class **SharedPreference** in the package **com.androidopentutorials.sharedpreference.utils**. This class defines methods to save, get and remove shared preferences values.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62 | package com.androidopentutorials.sharedpreference.utils;    import android.content.Context;  import android.content.SharedPreferences;  import android.content.SharedPreferences.Editor;  import android.preference.PreferenceManager;    public class SharedPreference {        public static final String PREFS\_NAME = "AOP\_PREFS";      public static final String PREFS\_KEY = "AOP\_PREFS\_String";        public SharedPreference() {          super();      }        public void save(Context context, String text) {          SharedPreferences settings;          Editor editor;            //settings = PreferenceManager.getDefaultSharedPreferences(context);          settings = context.getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE); //1          editor = settings.edit(); //2            editor.putString(PREFS\_KEY, text); //3            editor.commit(); //4      }        public String getValue(Context context) {          SharedPreferences settings;          String text;            //settings = PreferenceManager.getDefaultSharedPreferences(context);          settings = context.getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE);          text = settings.getString(PREFS\_KEY, null);          return text;      }        public void clearSharedPreference(Context context) {          SharedPreferences settings;          Editor editor;            //settings = PreferenceManager.getDefaultSharedPreferences(context);          settings = context.getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE);          editor = settings.edit();            editor.clear();          editor.commit();      }        public void removeValue(Context context) {          SharedPreferences settings;          Editor editor;            settings = context.getSharedPreferences(PREFS\_NAME, Context.MODE\_PRIVATE);          editor = settings.edit();            editor.remove(PREFS\_KEY);          editor.commit();      }  } |

##### **MainActivity class**

Open MainActivity.java class and copy the following code. This class gets value from EditText, stores it in SharedPreferences and starts the second activity.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79 | package com.androidopentutorials.sharedpreference;    import android.app.Activity;  import android.content.Context;  import android.content.Intent;  import android.os.Bundle;  import android.view.Menu;  import android.view.View;  import android.view.View.OnClickListener;  import android.view.inputmethod.InputMethodManager;  import android.widget.Button;  import android.widget.EditText;  import android.widget.Toast;    import com.androidopentutorials.sharedpreference.utils.SharedPreference;    public class MainActivity extends Activity {        // UI References      private EditText textEtxt;      private Button saveButton;      private Button activity2Button;        private String text;        private SharedPreference sharedPreference;        Activity context = this;        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            sharedPreference = new SharedPreference();            findViewsById();            saveButton.setOnClickListener(new OnClickListener() {                @Override              public void onClick(View v) {                  text = textEtxt.getText().toString();                    // Hides the soft keyboard                  InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT\_METHOD\_SERVICE);                  imm.hideSoftInputFromWindow(textEtxt.getWindowToken(), 0);                    // Save the text in SharedPreference                  sharedPreference.save(context, text);                  Toast.makeText(context,                          getResources().getString(R.string.saved),                          Toast.LENGTH\_LONG).show();              }          });            activity2Button.setOnClickListener(new OnClickListener() {                @Override              public void onClick(View v) {                  Intent intent = new Intent(context, SecondActivity.class);                  // Start next activity                  startActivity(intent);              }          });      }        private void findViewsById() {          textEtxt = (EditText) findViewById(R.id.etxt\_text);          saveButton = (Button) findViewById(R.id.button\_save);          activity2Button = (Button) findViewById(R.id.button\_second\_activity);      }        @Override      public boolean onCreateOptionsMenu(Menu menu) {          getMenuInflater().inflate(R.menu.main, menu);          return true;      }  } |

##### **SecondActivity class**

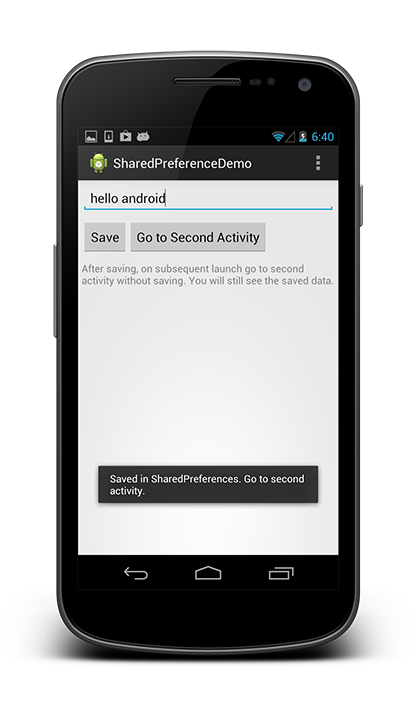
Create a new SecondActivity.java class and copy the following code. This class gets the value from SharedPreferences and displays it in TextView.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | package com.androidopentutorials.sharedpreference;    import android.app.Activity;  import android.os.Bundle;  import android.widget.TextView;  import com.actionbarsherlock.app.SherlockActivity;  import com.actionbarsherlock.view.Menu;  import com.androidopentutorials.sharedpreference.utils.SharedPreference;    public class SecondActivity extends SherlockActivity {        // UI References      private TextView textTxt;        private String text;        private SharedPreference sharedPreference;        Activity context = this;        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_second);            sharedPreference = new SharedPreference();            findViewsById();            //Retrieve a value from SharedPreference          text = sharedPreference.getValue(context);          textTxt.setText(text);        }        private void findViewsById() {          textTxt = (TextView) findViewById(R.id.txt\_text);      }        @Override      public boolean onCreateOptionsMenu(Menu menu) {          getSupportMenuInflater().inflate(R.menu.main, menu);          return true;      }  } |

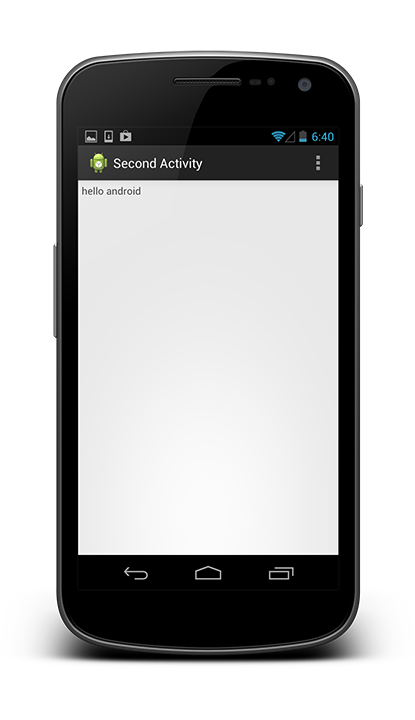
### **Output**

#### **MainActivity**

[](http://androidopentutorials.com/wp/wp-content/uploads/2014/03/1.android-sharedpreferences-example-output.png?1a8226)

When we go directly to second activity, no value is displayed as we have not yet stored the value in shared preference. Once the submit button is pressed, the value is saved in shared preference. Now this value will be shown on subsequent app launches by directly going to the second activity.

#### **SecondActivity**

[](http://androidopentutorials.com/wp/wp-content/uploads/2014/03/2.android-sharedpreferences-example-output.png?1a8226)