Module: Mobile Application development (Android)

Session 33: Android Preferences and External resources assignments (Practice)

#### In this practice session, you will create Simple Android SharedPreferences application. You will learn about Android SharedPreferences. That means you will learn about how to store persistent user data in our own Android application and get it back when required. Also you will learn how to add, edit, retrieve (fetch), load, store, delete and update data in Android SharedPreferences. Android provides different options to save persistent application data. It is completely your choices to which option you want to use for your application. Android provides below different data storage options.

#### ****SharedPreferences****

Store your application’s private primitive data (working as key-value pairs).  
You can use SharedPreferences to save any primitive data of your application like: boolean, float, int, long, and string etc. This data will persist across user sessions (even if your application is closed or forcible killed by Android OS).

#### ****SQLite Databases****

Store your application’s structured data in your application’s private database. It will store in the application’s context. I have already written a very nice article about how to save application’s data in your own Android SQLite database.

#### ****Network Connection****

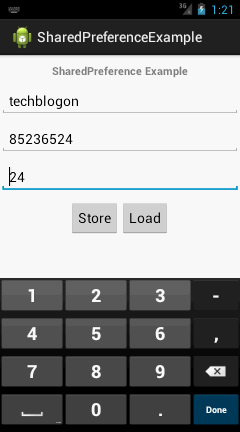
Store your application’s data on the web with your own network server. So that other users can access it when required.

#### ****Internal Storage****

Store your applications private data on the phone memory (device memory)

#### ****External Storage****

Store your application’s data on the shared external storage, so that other application can be access it if required.

[](http://techblogon.com/wp-content/uploads/2013/04/android-shared-preference-example.png)

Now it is enough for some basic idea about data storing in Android. Let’s come to our main focus e.g.  “Android SharedPreferences example with source code description”

To make this Android shared preference example very simple, we will use different Android primitive data types like: boolean, float, int, long, and string etc and store some user data into Android SharedPreference. Then we will retrieve (fetch) those stored data from Android SharedPreferences.

**Note:** Android SharedPreferences uses only **Key-Value pair** for storing data, that means, when we will store data to Android SharedPreferences, we will use a unique key and some corrosponding data with that key.

### Simple Steps for Android Shared Preferences Use

**Step 1:** Create a new Shared Preferences file in your Android application. We will open it if it is already exist. We will use the function getSharedPreferences() function from the inbuilt Android class.

**Step 2:**  For getting data from the Android SharedPreferences file, we will use functions like getString(),getInt() etc.

**Step 3:**  For storing the data, we need to use the Android Editor ClassSharedPreferences.Editor. Then we will store data into the Android SharedPreferences file using functions putString(),putInt() etc. then commit the whole data using commit() function.

### Android SharedPreferences Initialization

Android SharedPreferences can be fetch using the function getSharedPreferences(). This function need a context (your Activity context). so use your Activity context if you will call this from your activity class. If you call this from a non Activity class like Android services, then first store the context and use it for getSharedPreferences().Don’t worry we will discuss complete details in below example. this is only for the code snippet. Finally we need an Android editor class to save the changes in the Android SharedPreferences. Below is the code snippet.

First import the below package into your java file, where you want to use shared preference.

|  |  |
| --- | --- |
| 1  2  3  4  5 | SharedPreferences pref = getApplicationContext().getSharedPreferences("MyPref", 0); // 0 - for private mode  //here the first parameter is name of your pref file which will hold your data. you can give any name.  // Note here the 2nd parameter 0 is the default parameter for private access.    Editor editor = pref.edit(); // used for save data |

### Storing Data into Android SharedPreferences

After the above initialization code, we need to follow the below code snippet for storing different types of data in to Android SharedPreferences.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | editor.putBoolean("key\_name1", true); // Storing boolean value - true or false  editor.putString("key\_name2", "string-value"); // Storing string value  editor.putInt("key\_name3", "int value"); // Storing integer value  editor.putFloat("key\_name4", "float value"); // Storing float value  editor.putLong("key\_name5", "long value"); // Storing long value    editor.commit(); // commit changes into sharedpreferences file. |

**Note:** In the above code snippet, to make the code buildable please change “float value”, “long value” to the real float and long values.

### Fetching Data from Android SharedPreferences

Data can be fetch from saved preferences by calling [getString()](http://developer.android.com/reference/android/content/SharedPreferences.html" \l "getString(java.lang.String, java.lang.String)" \t "_blank) (For string), getInt() (for Int) method etc. Note that: these methods should be called using SharedPreferences not using Editor. Below is the code snippet.

|  |  |
| --- | --- |
| 1 | SharedPreferences pref = getApplicationContext().getSharedPreferences("MyPref", 0); // 0 - for private mode |
| 1  2  3  4  5  6  7 | // getting values from stored preferences  // If any value is not present in the preferences file, then the second parameter will be the default value - //In this case null  pref.getString("key\_name1", ""); // getting String  pref.getInt("key\_name2", ""); // getting Integer  pref.getFloat("key\_name3", null); // getting Float  pref.getLong("key\_name4", null); // getting Long  pref.getBoolean("key\_name5", true); // getting boolean |

### Deleting Data from Android SharedPreferences

If you want to delete some specific data from Android sharedpreferences, then we can call the function **remove(“key\_name”)** to delete that particular value. If you want to delete all the data, call ***clear()***function.

|  |  |
| --- | --- |
| 1  2  3 | editor.remove("key\_name1"); // will delete the key named key\_name1  editor.remove("key\_name2"); // will delete key named key\_name2  editor.commit(); // commit above changes |

If you want to clear (delete) all data at a time then you can use the below code snippet.

|  |  |
| --- | --- |
| 1  2 | editor.clear(); // clear all data.  editor.commit(); // commit changes |

### Create Project: Android SharedPreferences Example

Now coming to the Android SharedPreferences Example. You can use the below example to learn about Android Shared Preferences. We will get some input data (we have used 3 fields : Name, Employee ID and Age) from user. Then we will save it into the shared preferences file, After that we will retrieve those values from the android SharedPreferences file. To make the example very simple, we will use 2 buttons (**Store**and **Load**) for this purpose. Also i have implemented a Simple ***SharedPrefManager*** Class. which will perform all Android SharedPreferences operations for you. You can use and edit this java file (SharedPrefManager Class) as per your requirement. Only what you need to do is: replace the package name ‘package com.techblogon.sharedpreferenceexample;‘ to your application’s package name.

1. Create a project with project name: **SharedPreferenceExample**

2. Fill Application Name: **SharedPreferenceExample**

3. Fill Package Name as: **package com.techblogon.sharedpreferenceexample**;

4. I have used SDK version ***Android 4.0.3*** and Eclipse Version***Indigo***. But you can use any version.

5. Add below xml file (activity\_main.xml) into your project’s ***res/layout***folder. or you can copy the xml file contents.

This the default xml file in the project. here we used 3 EditText field to get user data to store into the sharedpreferences file. Also added 2 buttons (**Store** and **Load**) to make the operation simple.

|  |  |
| --- | --- |
| 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 | <LinearLayout 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:orientation="vertical"      tools:context=".MainActivity">       <TextView        android:id="@+id/textView"        android:layout\_width="match\_parent"        android:layout\_height="wrap\_content"        android:gravity="center\_horizontal"        android:textStyle="bold"        android:layout\_marginTop="10dp"        android:hint="SharedPreference Example"        android:ems="10" >      </TextView>        <EditText          android:id="@+id/editTextEnterName"  android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content"          android:layout\_marginTop="10dp"          android:maxLength="30"          android:hint="Enter Your Name">          <requestFocus />      </EditText>    <EditText          android:id="@+id/editTextEnterEid"  android:layout\_width="fill\_parent"          android:layout\_height="wrap\_content"          android:layout\_marginTop="10dp"          android:maxLength="8"          android:inputType ="number"          android:hint="Enter Your Employee ID">     </EditText>          <EditText          android:id="@+id/editTextEnterAge"  android:layout\_width="fill\_parent"  android:layout\_marginTop="10dp"  android:inputType ="number"  android:maxLength="2"          android:layout\_height="wrap\_content"          android:hint="Enter Your Age">     </EditText>      <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:orientation="horizontal"  android:layout\_marginTop="10dp"      android:gravity="center\_horizontal">        <Button          android:id="@+id/buttonStore"  android:layout\_height="wrap\_content"  android:layout\_width="wrap\_content"          android:onClick="onClickStore"          android:text="Store" />       <Button          android:id="@+id/buttonLoad"  android:layout\_height="wrap\_content"  android:layout\_width="wrap\_content"          android:layout\_toRightOf="@+id/buttonStore"          android:onClick="onClickLoad"          android:text="Load" />       </RelativeLayout>>    </LinearLayout> |

6. Add below Java file (SharedPrefManager.java)  in you project’s /***src*** folder.

This file contains a class  ‘SharedPrefManager’,which will handle all Android SharedPreferences operations. You just want to call those functions form where you want to store/load data in to the shared preference.  All functions are static in nature, so no need to create any instance of the class, just call these functions using the class name with .dot operator. For example if you want to save data then just call : SharedPrefManager.StoreToPref();

|  |  |
| --- | --- |
| 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  80  81  82  83  84  85  86  87  88  89  90  91  92 | package com.techblogon.sharedpreferenceexample;  import android.content.Context;  import android.content.SharedPreferences;    // all methods are static , so we can call from any where in the code  //all member variables are private, so that we can save load with our own fun only  public class SharedPrefManager {  //this is your shared preference file name, in which we will save data  public static final String MY\_EMP\_PREFS = "MySharedPref";    //saving the context, so that we can call all  //shared pref methods from non activity classes.  //because getSharedPreferences required the context.  //but in activity class we can call without this context  private static Context mContext;    // will get user input in below variables, then will store in to shared pref  private static String mName = "";  private static String mEid = "";  private static int mAge = 0;    public static void Init(Context context)  {  mContext = context;  }  public static void LoadFromPref()  {  SharedPreferences settings = mContext.getSharedPreferences(MY\_EMP\_PREFS, 0);  // Note here the 2nd parameter 0 is the default parameter for private access,  //Operating mode. Use 0 or MODE\_PRIVATE for the default operation,  mName = settings.getString("Name",""); // 1st parameter Name is the key and 2nd parameter is the default if data not found  mEid = settings.getString("EmpID","");  mAge = settings.getInt("Age",0);  }  public static void StoreToPref()  {  // get the existing preference file  SharedPreferences settings = mContext.getSharedPreferences(MY\_EMP\_PREFS, 0);  //need an editor to edit and save values  SharedPreferences.Editor editor = settings.edit();  editor.putString("Name",mName); // Name is the key and mName is holding the value  editor.putString("EmpID",mEid);// EmpID is the key and mEid is holding the value  editor.putInt("Age", mAge); // Age is the key and mAge is holding the value    //final step to commit (save)the changes in to the shared pref  editor.commit();  }    public static void DeleteSingleEntryFromPref(String keyName)  {  SharedPreferences settings = mContext.getSharedPreferences(MY\_EMP\_PREFS, 0);  //need an editor to edit and save values  SharedPreferences.Editor editor = settings.edit();  editor.remove(keyName);  editor.commit();  }    public static void DeleteAllEntriesFromPref()  {  SharedPreferences settings = mContext.getSharedPreferences(MY\_EMP\_PREFS, 0);  //need an editor to edit and save values  SharedPreferences.Editor editor = settings.edit();  editor.clear();  editor.commit();  }    public static void SetName(String name)  {  mName =name;  }  public static void SetEmployeeID(String empID)  {  mEid = empID ;  }  public static void SetAge(int age)  {  mAge = age;  }    public static String GetName()  {  return mName ;  }  public static String GetEmployeeID()  {  return mEid ;  }  public static int GetAge()  {  return mAge ;  }  } |

7. Add the below default Activity Java file (MainActivity.Java) , which contains all GUI part and will use all SharedPreferences functions from this class.

In this class we called SharedPrefManager.Init(this)function from the OnCreate() of the activity. Then we will Store and Load user data when user click on the buttons called **Store** and **Load**. Here you can find functions like onClickLoad() and onClickStore(), where we will store and load user data. So first we will get user data and pass all user data to the SharedPrefManagerclass using it’s own functions, then we willstore all data at a time using SharedPrefManager class’s function SharedPrefManager.StoreToPref(). Similarly we will use SharedPrefManager.LoadFromPref().

|  |  |
| --- | --- |
| 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  80  81  82  83  84 | package com.techblogon.sharedpreferenceexample;    import android.os.Bundle;  import android.app.Activity;  import android.view.Menu;  import android.view.View;  import android.widget.EditText;  import android.widget.Toast;    public class MainActivity extends Activity {    @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);    SharedPrefManager.Init(this);    }  public void onClickStore(View v) {    //get user input first, then store. we will use our SharedPrefManager Class functions  EditText editTextName,editTextEmpID,editTextAge;  editTextName=(EditText)findViewById(R.id.editTextEnterName);  editTextEmpID=(EditText)findViewById(R.id.editTextEnterEid);  editTextAge=(EditText)findViewById(R.id.editTextEnterAge);    //convert EditText to string  String srtTextName = editTextName.getText().toString();  String srtTextEmpID = editTextEmpID.getText().toString();  String strTextAge = editTextAge.getText().toString();    if(0!= srtTextName.length())  SharedPrefManager.SetName(srtTextName); // need string value so convert it  if(0 !=srtTextEmpID.length())  SharedPrefManager.SetEmployeeID(srtTextEmpID); // need string value so convert it  if(0 != strTextAge.length())  SharedPrefManager.SetAge(Integer.parseInt(strTextAge)); // need integer value so convert it    //now save all to shared pref, all updated values are now available in SharedPrefManager class, as we set above  SharedPrefManager.StoreToPref();    //reset all fields to blank before load and update from sharedpref  EditText tv = null;  tv = (EditText)findViewById(R.id.editTextEnterName);          tv.setText("");          tv = (EditText)findViewById(R.id.editTextEnterEid);          tv.setText("");          tv = (EditText)findViewById(R.id.editTextEnterAge);          tv.setText("");    Toast.makeText(this, "Data Successfully Stored to SharedPreference", Toast.LENGTH\_LONG).show();    }  public void onClickLoad(View v) {    //Get all values from SharedPrefference file  SharedPrefManager.LoadFromPref(); // all values are loaded into corresponding variables of SharedPrefManager class    //Now get the values form SharedPrefManager class using it's static functions.  String strTextName,strTextEmpID;  int iTextAge;  strTextName = SharedPrefManager.GetName();  strTextEmpID = SharedPrefManager.GetEmployeeID();  iTextAge =SharedPrefManager.GetAge();    //Now we can show these persistent values on our activity (GUI)  EditText tv = null;  tv = (EditText)findViewById(R.id.editTextEnterName);          tv.setText(strTextName);          tv = (EditText)findViewById(R.id.editTextEnterEid);          tv.setText(strTextEmpID);          tv = (EditText)findViewById(R.id.editTextEnterAge);          tv.setText(String.valueOf(iTextAge));    Toast.makeText(this, "Data Successfully Loaded from SharedPreference", Toast.LENGTH\_LONG).show();  }  @Override  public boolean onCreateOptionsMenu(Menu menu) {  // Inflate the menu; this adds items to the action bar if it is present.  getMenuInflater().inflate(R.menu.activity\_main, menu);  return true;  }  } |

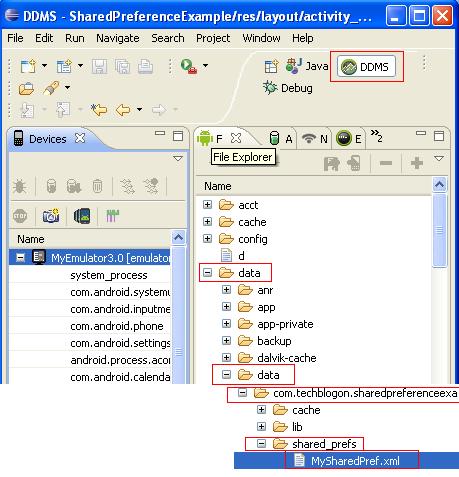
8. Finally the Manifest File.

|  |  |
| --- | --- |
| 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 | <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android"      package="com.techblogon.sharedpreferenceexample"      android:versionCode="1"      android:versionName="1.0" >        <uses-sdk          android:minSdkVersion="8"          android:targetSdkVersion="16" />        <application          android:allowBackup="true"          android:icon="@drawable/ic\_launcher"          android:label="@string/app\_name"          android:theme="@style/AppTheme" >          <activity              android:name="com.techblogon.sharedpreferenceexample.MainActivity"              android:label="@string/app\_name" >              <intent-filter>                  <action android:name="android.intent.action.MAIN" />                    <category android:name="android.intent.category.LAUNCHER" />              </intent-filter>          </activity>      </application>    </manifest> |

9. That’s it. Build and run the application on Android device/emulator.

### Tips on Android Shared Preference Uses

Sometimes you might want to delete the old SharedPreferences file. If you want to delete the old one and create a new one, then you need to know the file location where this file is created for your application. **Goto DDMS->File Explorer**. Please refer to the below image.

[](http://techblogon.com/wp-content/uploads/2013/04/android-sharedpreference-file-location-in-device.jpg)

### Some Android SharedPreferences Issues and Solutions

**Error:** The new data overwrites the old data in Android Shared Preference

**Solution:** Android calls ***clear()***function, when you save (commit) data. so all your existing data will be erased. To prevent that, first you have to get all old data and save those in some variables, then add you new data and save all at a time. You can use Android Vector and Android Array for this purpose.

**Error:**Data not saved in Android Shared Preference.

**Solution:**Sometimes the above issue happens with me also, there are several reasons. However delete the preference file and create a new one might solve your problem. Make sure to save your date before delete.