



آموزش برنامه نویسی اندروید در محیط اندروید استودیو

ذخیره اطلاعات با استفاده از SharedPreferences

مدرس : سید مهدی مطهری

www.android-studio.ir



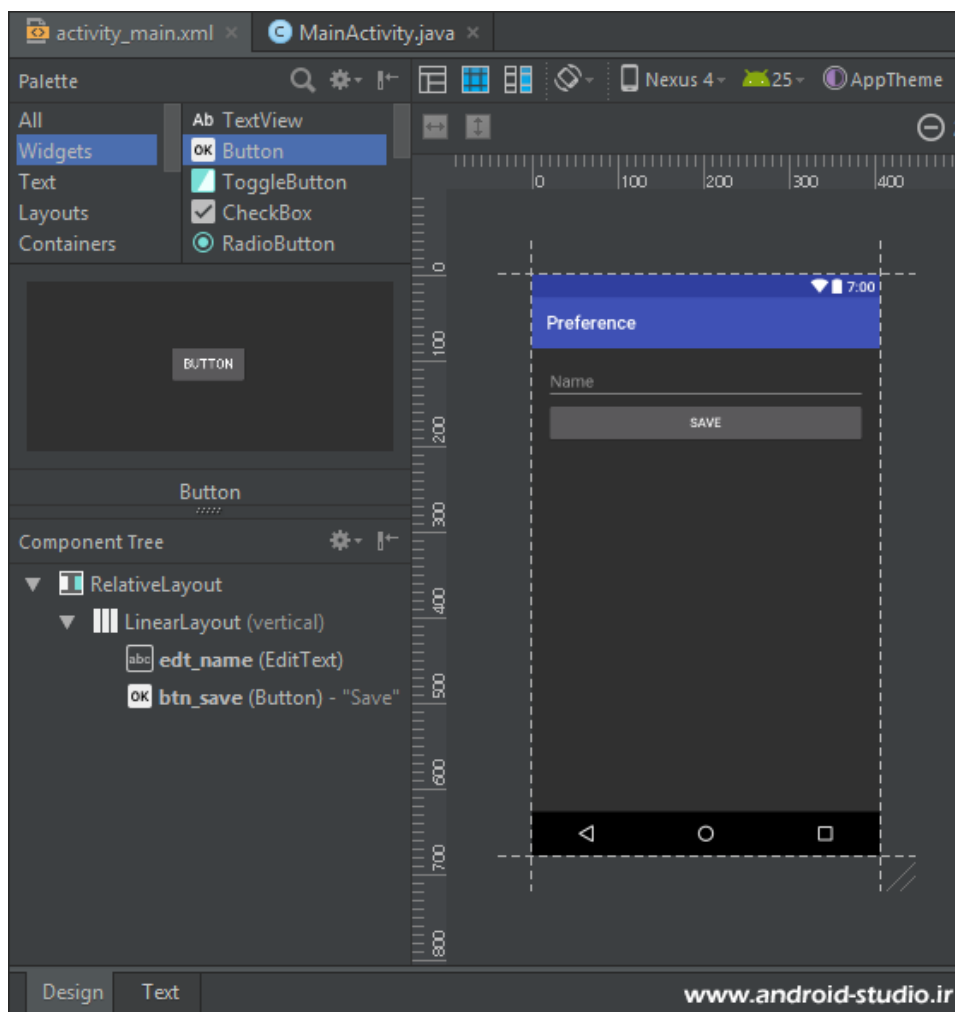
به نام خدا

در جلسه گذشته به بررسی دیتابیس در اندروید پرداختیم و با نحوه ذخیره اطلاعات در دیتابیس آشنا شدیم. اما در اندروید روش های دیگری هم برای ذخیره اطلاعات در نظر گرفته شده که در این جلسه به معرفی SharedPreferences می پردازیم. اینکه هرکدام از روش های ذخیره سازی در چه مواردی کاربرد دارد عمدتاً به سلیقه توسعه دهنده بستگی دارد اما عموماً در کاربردهای متفاوت، با توجه به ویژگی آن، یک گزینه نسبت به دیگری رایج تر بوده و ارجح است. به عنوان مثال برای ذخیره تنظیمات اپلیکیشن (فونت و سبک برنامه، حالت شب/روز، رنگ پس زمینه و...)، ثبت اطلاعات کاربر، نحوه نمایش و چیدمان لیست ها، فعال یا غیرفعال بودن دریافت نوتیفیکیشن ها، و به طور کلی داده های با حجم کم، معمولاً از SharedPreferences استفاده می شود. SharedPreferences به ما اجازه میدهد تا اطلاعات را با فرمت Key/Value (کلید/مقدار) ذخیره و نگهداری کنیم. این اطلاعات با فرمت xml ذخیره شده و تا زمانی که اپلیکیشن از روی سیستم عامل حذف نشده و یا کاربر به صورت دستی در قسمت تنظیمات برنامه ها، Data ی اپلیکیشن را حذف نکند، باقی می ماند.

برای دستیابی به Preference ها به سه متد دسترسی داریم:

- getPreferences()
- getSharedPreferences()
- getDefaultSharedPreferences()

در این مبحث من از متد getSharedPreferences() استفاده می کنم. قبل از ادامه توضیحات، یک پروژه جدید می سازم که نام آن را Preference گذاشته ام. در ابتدا ساده ترین حالت را در نظر گرفته و تنها یک فیلد ورودی متن (EditText) و یک Button به اکتیویته activity_main.xml اضافه می کنم تا کاربر نام خود را وارد کرده و ذخیره نماید:



در قدم بعد، این دو Widget را در MainActivity.java تعریف می کنم.



:activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context="ir.android_studio.preference.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <EditText
            android:id="@+id/edt_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Name"
            android:inputType="textPersonName" />

        <Button
            android:id="@+id/btn_save"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Save" />

    </LinearLayout>
</RelativeLayout>
```

:MainActivity.java

```
package ir.android_studio.preference;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    EditText edtName;
    Button btnSave;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        edtName = (EditText) findViewById(R.id.edt_name);
        btnSave = (Button) findViewById(R.id.btn_save);
    }
}
```



حالا از اینترفیس SharedPreferences یک نمونه می سازم. من نام shPref را برای این نمونه در نظر گرفتم:

```
SharedPreferences shPref;
```

در ادامه shPref را مساوی متد getSharedPreferences() قرار می دهم:

```

activity_main.xml x MainActivity.java x
MainActivity onCreate ()
1 package ir.android_studio.preference;
2
3 import android.content.SharedPreferences;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.widget.Button;
7 import android.widget.EditText;
8
9 public class MainActivity extends AppCompatActivity {
10
11     EditText edtName;
12     Button btnSave;
13     SharedPreferences shPref;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19
20         edtName = (EditText) findViewById(R.id.edt_name);
21         btnSave = (Button) findViewById(R.id.btn_save);
22         shPref = getSharedPreferences(...
23
24     }
25 }
26
String name, int mode
www.android-studio.ir

```

ملاحظه می کنید این متد دو پارامتر ورودی می گیرد. اولی name و دومی mode. پارامتر نخست مربوط به نام فایل است که برای ذخیره سازی داده ها استفاده می شود و اختیاری است. من MyPref نام می کنم (یک متغیر از نوع String با نام MyPref و مقدار MyPrefers ساخته ام. برای موارد بعدی نیز متغیر تعریف می کنم تا از وارد کردن دستی نامها جلوگیری شود که در نهایت منجر به افزایش سرعت و کاهش خطا می شود). پارامتر دوم مربوط به mode (روش) عملیاتی است.



انواع مد ها را بررسی می کنیم:

- MODE_PRIVATE: مد پیش فرض که فایل محتوی داده ها تنها با فراخوانی اپلیکیشن در دسترس خواهد بود.
- MODE_WORLD_READABLE: با این مد سایر اپلیکیشن ها نیز می توانند داده ها را بخوانند که از حیث امنیت ممکن است مشکل ساز شود. (این مد از API 17 به بعد deprecated (منقضی) شده است)
- MODE_WORLD_WRITEABLE: این مد به سایر اپلیکیشن ها اجازه می دهد تا اطلاعات Preference برنامه ما را ویرایش کنند که مانند مورد قبل به لحاظ امنیتی مشکل ایجاد نموده و از API 17 به بعد deprecated شده است.
- MODE_MULTI_PROCESS: این مد تعدیل preferences را چک خواهد کرد حتی اگر Shared Preference قبلا اجرا شده باشد. (این مد از API 23 به بعد deprecated شده است)
- MODE_APPEND: به واسطه این مد، preferences های جدید با preferences های قبلی ترکیب خواهد شد.

ما از MODE_PRIVATE استفاده می کنیم:

```
shPref = getSharedPreferences("MyPref", Context.MODE_PRIVATE);
```

خواسته ما از برنامه این است که پس از وارد کردن نام و لمس کلید Save توسط کاربر، نام وارده ذخیره شده و با بستن و اجرای مجدد اپلیکیشن، اطلاعات وارد شده نمایش داده شود. ابتدا متد مربوط به دکمه btnSave را کامل کرده و در ادامه توضیح می دهیم.



```

package ir.android_studio.preference;

import android.content.Context;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    EditText edtName;
    Button btnSave;
    SharedPreferences shPref;
    public static final String MyPref = "MyPrefers";
    public static final String Name = "nameKey";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        edtName = (EditText) findViewById(R.id.edt_name);
        btnSave = (Button) findViewById(R.id.btn_save);
        shPref = getSharedPreferences(MyPref, Context.MODE_PRIVATE);

        btnSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String n = edtName.getText().toString();
                SharedPreferences.Editor sEdit = shPref.edit();
                sEdit.putString(Name, n);
                sEdit.apply();

                Toast.makeText(MainActivity.this, "Saved", Toast.LENGTH_LONG).show();

            }
        });
    }
}

```

در خط اول متغیری از جنس String با نام n تعریف کردم که با دستور

```
edtName.getText().toString()
```

Text ورودی کاربر را از edtName گرفته و در خود ذخیره می کند (توسط `getText` رشته ورودی دریافت و به وسیله `toString` به رشته از جنس String تبدیل می شود).



حالا به یک ادیتور نیاز داریم تا به واسطه آن اطلاعات را ذخیره و یا ویرایش کنیم. یک نمونه از SharedPreferences.Editor با نام sEdit ساخته که با shPref.edit() آماده ذخیره و یا ویرایش اطلاعات است.

در مرحله بعد باید محتویات درون n را به Name منتقل کنیم. این کار توسط متد putString انجام می شود:

```
sEdit.putString(Name, n);
```

همانطور که از نام متد پیداست، برای ذخیره و یا ویرایش داده های از نوع String بکار می رود. به همین ترتیب برای مقادیر صحیح عددی از متد putInt، مقادیر boolean متد putBoolean، مقادیر شناور متد putFloat و مقادیر long متد putLong استفاده می شود.

در نهایت توسط متد commit() تغییرات اعمال شده ذخیره می شود:

```
sEdit.commit();
```

اندروید استودیو به من توصیه می کند به جای commit از متد apply استفاده کنم، بنابراین sEdit.apply() را جایگزین کردم (تفاوتهایی بین این دو وجود دارد که در صورت تمایل با جستجوی عبارت "commit() vs apply()" به جوابهای خوبی خواهید رسید. برای زیبایی کار یک پیغام از نوع Toast به دکمه اضافه کردم تا پس از لمس دکمه عبارت Saved به کاربر نمایش داده شود (در مبحث [دیتابیس](#) با Toast آشنا شدیم). تا اینجای کار اگر کد ما مشکلی نداشته باشد اطلاعات وارد شده باید به درستی ذخیره شود. اما هنوز نیاز به یک دستور شرطی داریم تا بعد از خروج کاربر از برنامه و ورود مجدد به آن، چک کند اگر مقادیری در Preference ذخیره شده در محل موردنظر نمایش داده شود. به کد زیر دقت کنید:

```
if (shPref.contains(Name)) {
    edtName.setText(shPref.getString(Name, null));
}
```

این کد می گوید اگر shPref حاوی Name است، توسط متد getString مقدار آنرا گرفته و توسط متد setText روی ویجت edtName نمایش بدهد.



مشابه متدهای set، متدهای get شامل getString، getInt، getBoolean، getFloat و getLong برای بازبازی داده های ذخیره شده در اختیار ما قرار گرفته است. این متدها نیز دو پارامتر ورودی دارند که اولی کلید مربوط به داده ذخیره شده و پارامتر دوم مقدار پیش فرض است. پیش فرض به این معنی که اگر در کلید مربوطه (به عنوان مثال Name) داده ای ذخیره نشده بود، مقداری که اینجا تعیین کرده ایم به کاربر نمایش داده شود. یک مثال میزنم:

```
edtName.setText(shPref.getString(Name, "Name not saved!"));
```

در حالت بالا اگر قبلا در Name چیزی ذخیره نشده باشد، پس از اجرای اپلیکیشن در ویجت edtName عبارت Name not saved! قرار می گیرد. اما چون من قبلا در لایه رابط کاربری مقدار Name را به عنوان hint برای edtName در نظر گرفته ام، از این پارامتر صرف نظر کرده و مقدار null قرار داده ام. البته به جای null رشته خالی یعنی "" هم می توان استفاده کرد:

```
edtName.setText(shPref.getString(Name, ""));
```



کد کامل MainActivity:

```
package ir.android_studio.preference;

import android.content.Context;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    EditText edtName;
    Button btnSave;
    SharedPreferences shPref;
    public static final String MyPref = "MyPrefers";
    public static final String Name = "nameKey";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        edtName = (EditText) findViewById(R.id.edt_name);
        btnSave = (Button) findViewById(R.id.btn_save);
        shPref = getSharedPreferences(MyPref, Context.MODE_PRIVATE);

        btnSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String n = edtName.getText().toString();
                SharedPreferences.Editor sEdit = shPref.edit();
                sEdit.putString(Name, n);
                sEdit.apply();

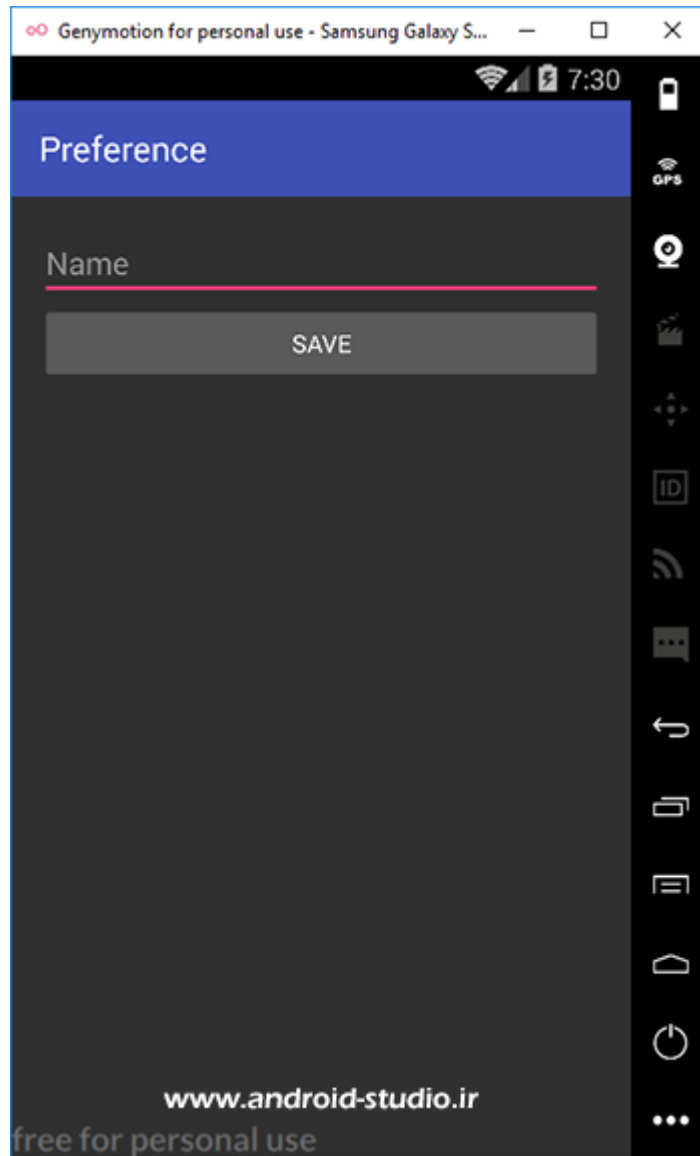
                Toast.makeText(MainActivity.this, "Saved", Toast.LENGTH_LONG).show();

            }
        });

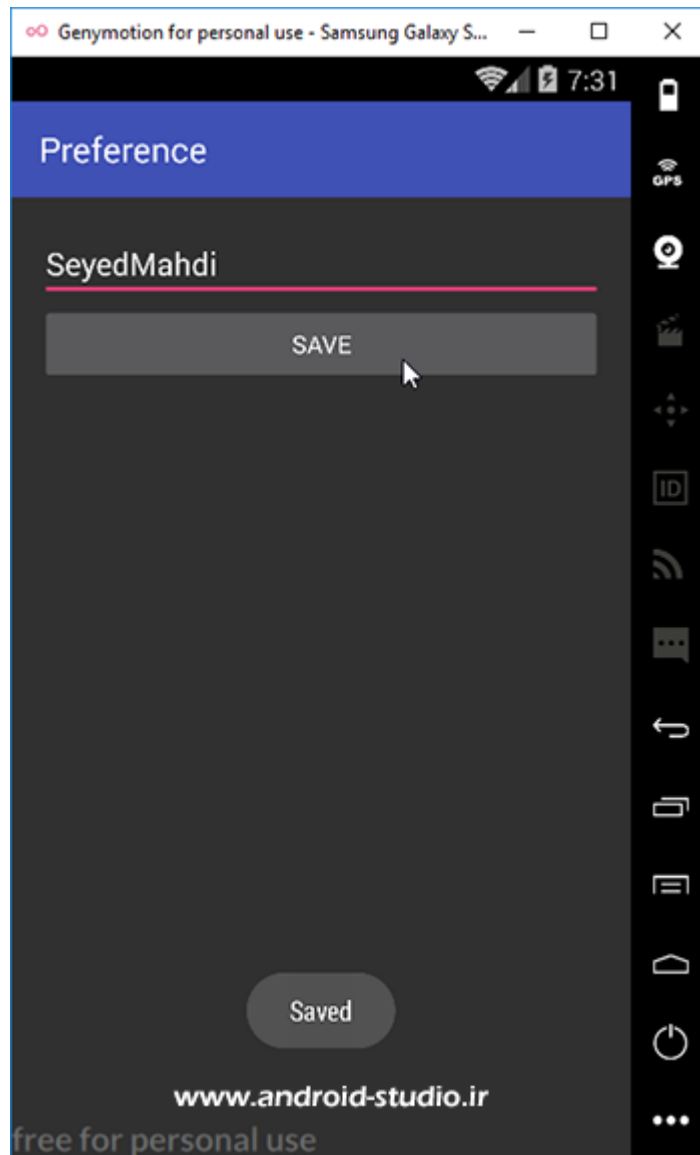
        if (shPref.contains(Name)) {
            edtName.setText(shPref.getString(Name, null));
        }
    }
}
```



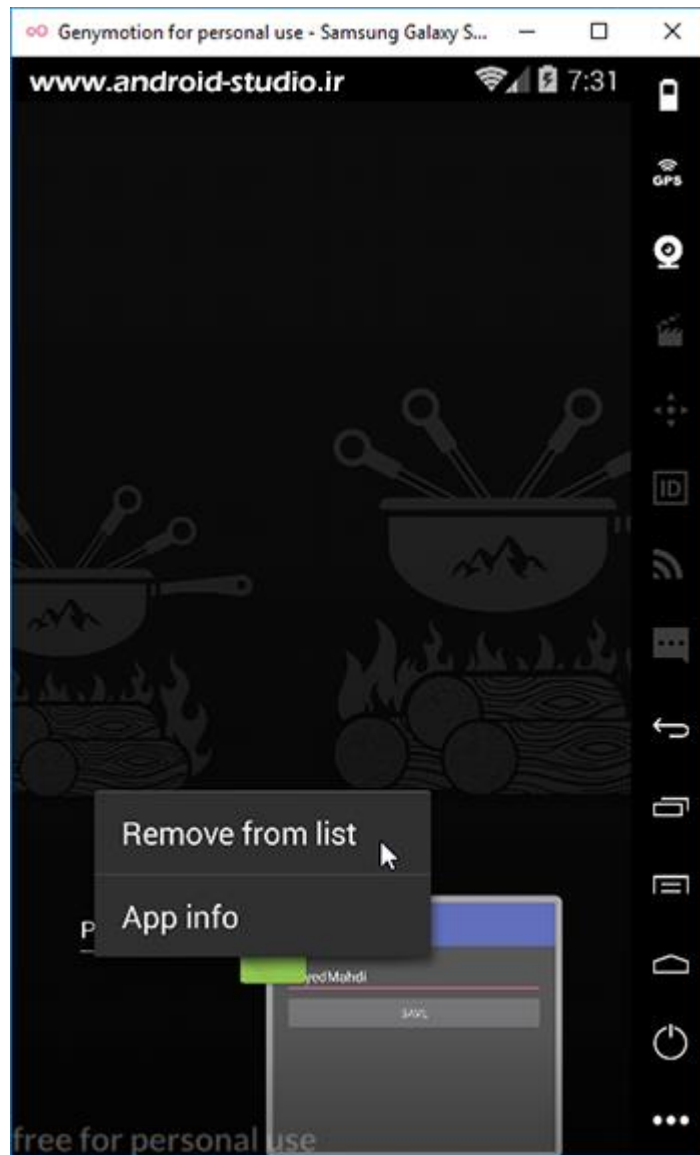
پروژه را اجرا می کنم:



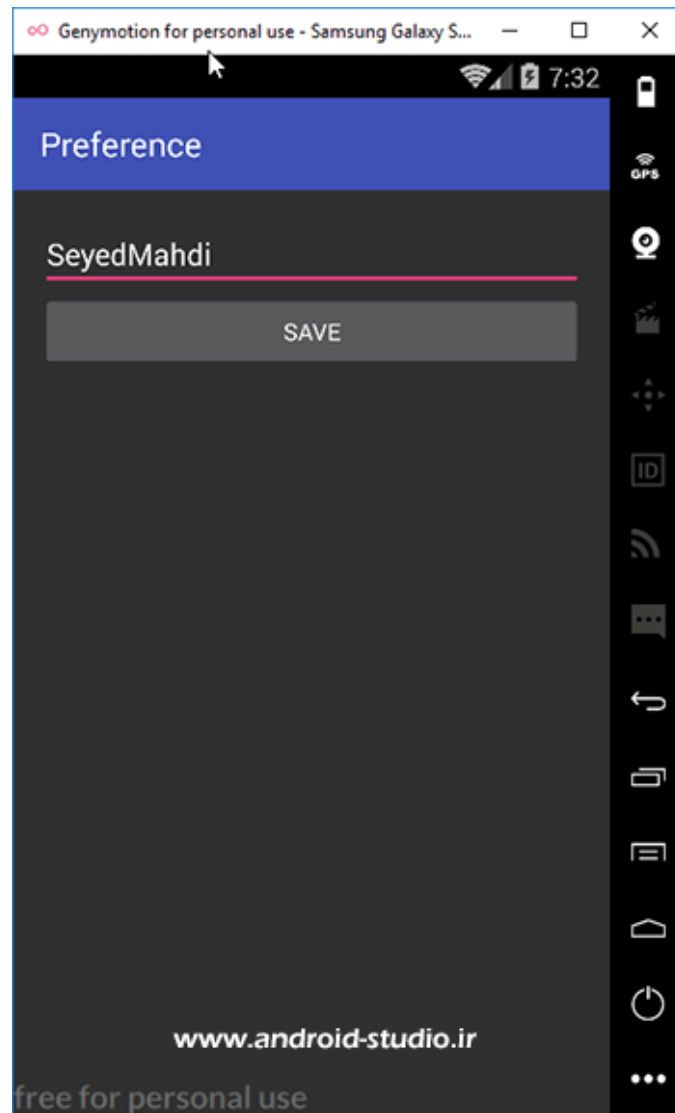
یک نام وارد کرده، ذخیره می کنم:



از اپلیکیشن خارج می شوم:



مجدد اجرا می کنم:



برنامه به درستی عمل کرد و نام ذخیره شده در EditText نمایش داده شد. اگر نام فعلی را تغییر داده و مجدد ذخیره کنیم، جایگزین نام قبلی خواهد شد.

دو فیلد ورودی دیگر با عنوان نام خانوادگی و سن نیز به برنامه اضافه می کنیم:



```
package ir.android_studio.preference;

import android.content.Context;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    EditText edtName;
    EditText edtFamily;
    EditText edtAge;
    Button btnSave;
    SharedPreferences shPref;
    public static final String MyPref = "MyPrefers";
    public static final String Name = "nameKey";
    public static final String Family = "familyKey";
    public static final String Age = "ageKey";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        edtName = (EditText) findViewById(R.id.edt_name);
        edtFamily = (EditText) findViewById(R.id.edt_family);
        edtAge = (EditText) findViewById(R.id.edt_age);
        btnSave = (Button) findViewById(R.id.btn_save);
        shPref = getSharedPreferences(MyPref, Context.MODE_PRIVATE);

        btnSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String n = edtName.getText().toString();
                String f = edtFamily.getText().toString();
                int a = Integer.parseInt(edtAge.getText().toString());

                SharedPreferences.Editor sEdit = shPref.edit();
                sEdit.putString(Name, n);
                sEdit.putString(Family, f);
                sEdit.putInt(Age, a);
                sEdit.apply();

                Toast.makeText(MainActivity.this, "Saved", Toast.LENGTH_LONG).show();

            }
        });

        if (shPref.contains(Name)) {
            edtName.setText(shPref.getString(Name, null));
        }

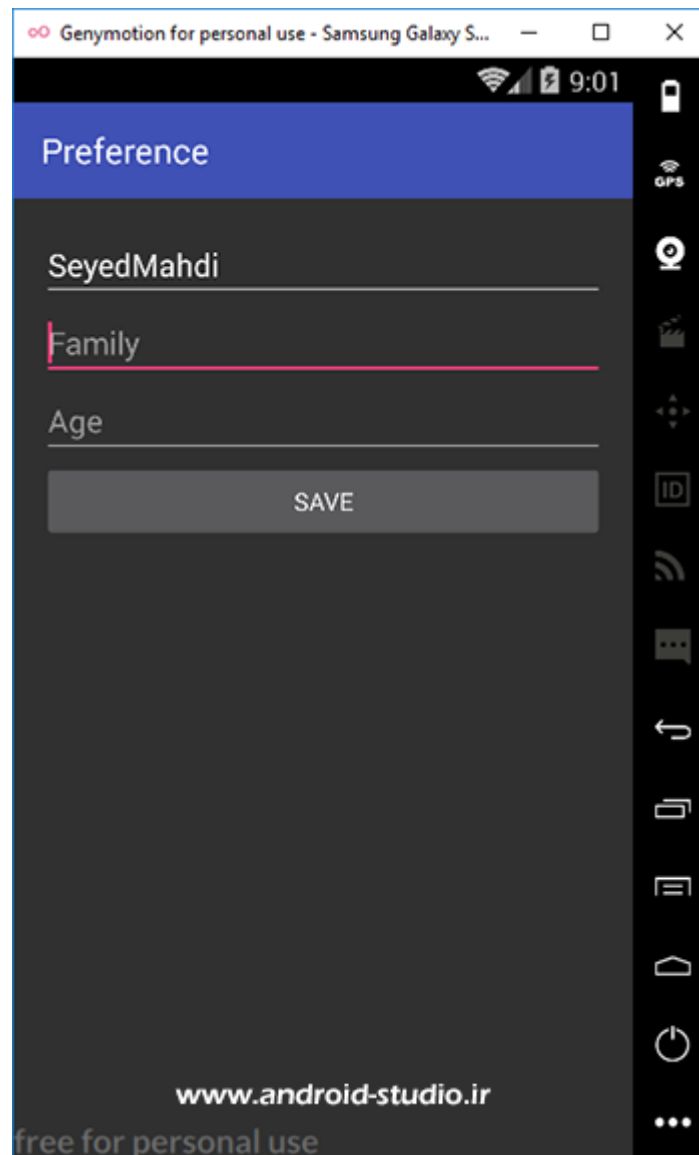
        if (shPref.contains(Family)) {
            edtFamily.setText(shPref.getString(Family, null));
        }

        if (shPref.contains(Age)) {
            edtAge.setText(String.valueOf(shPref.getInt(Age, 0)));
        }

    }
}
```



پروژه را مجدد اجرا می کنم:



اطلاعات را وارد کرده و روی دکمه save کلیک می کنم که ذخیره اطلاعات با موفقیت انجام می شود. مقدار int به تنهایی نمی تواند توسط setText به ویجت ارسال شود. راه حل ساده این است که یک رشته خالی به آن اضافه کنیم:

```
edtAge.setText(shPref.getInt(Age, 0) + "");
```

اما من روشی را انتخاب می کنم که اصولی باشد. به اینصورت که توسط String.valueOf() عدد صحیح به String تبدیل شده و سپس به edtAge فرستاده می شود. عدد صفر هم به عنوان مقدار پیش فرض تعیین شده.



برای سادگی و مختصر شدن کد بهتر بود سن را هم از نوع String تعریف کرده و فقط در رابط کاربری، فیلد موردنظر محدود به کاراکترهای عددی می شد. با این حال بهانه ای شد تا از متد putInt و موارد مربوط به تبدیل String به Integer نیز استفاده شود.

اما در SharedPreferences دو متد دیگر نیز در اختیار داریم:

- remove("KeyName"): توسط این متد می توان دیتا یا دیتاهای مدنظر را حذف کرد.
- clear(): با اجرا شدن این متد کلیه اطلاعات ذخیره شده حذف می گردد.

دو دکمه جدید با نام های "Remove Name and Family" و "Reset" به برنامه اضافه می کنم. تصمیم دارم با کلیک دکمه Remove، نام و نام خانوادگی و با کلیک Reset تمامی اطلاعات حذف شود. متد مربوط به دکمه ها را به صورت زیر تکمیل کردم:

```
btnRemove.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        SharedPreferences.Editor rEdit = shPref.edit();
        rEdit.remove(Name);
        rEdit.remove(Family);
        rEdit.apply();

        Toast.makeText(MainActivity.this, "Removed!", Toast.LENGTH_LONG).show();
    }
});

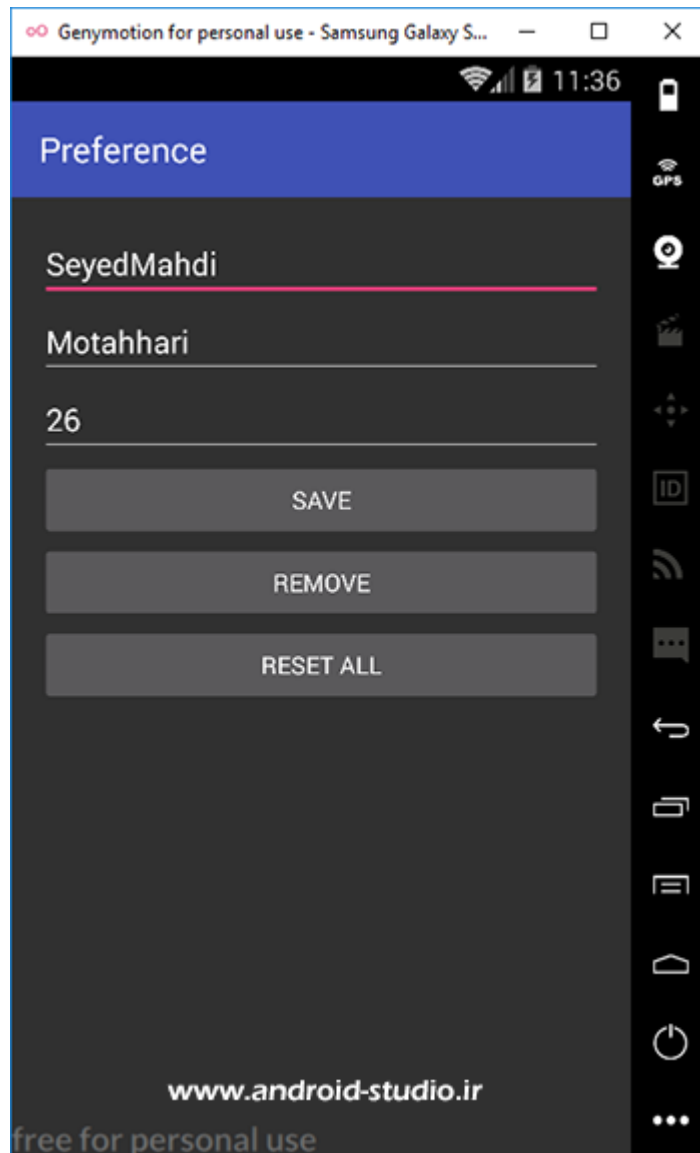
btnClear.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        SharedPreferences.Editor cEdit = shPref.edit();
        cEdit.clear();
        cEdit.apply();

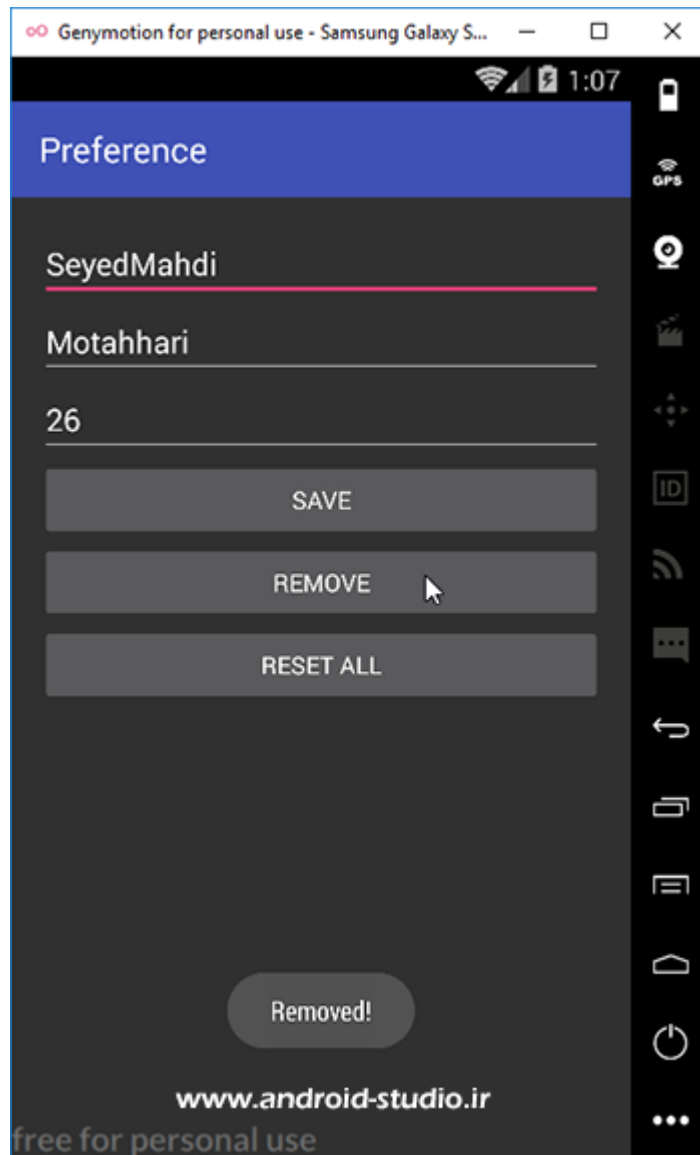
        Toast.makeText(MainActivity.this, "All data cleared!", Toast.LENGTH_LONG).show();
    }
});
```

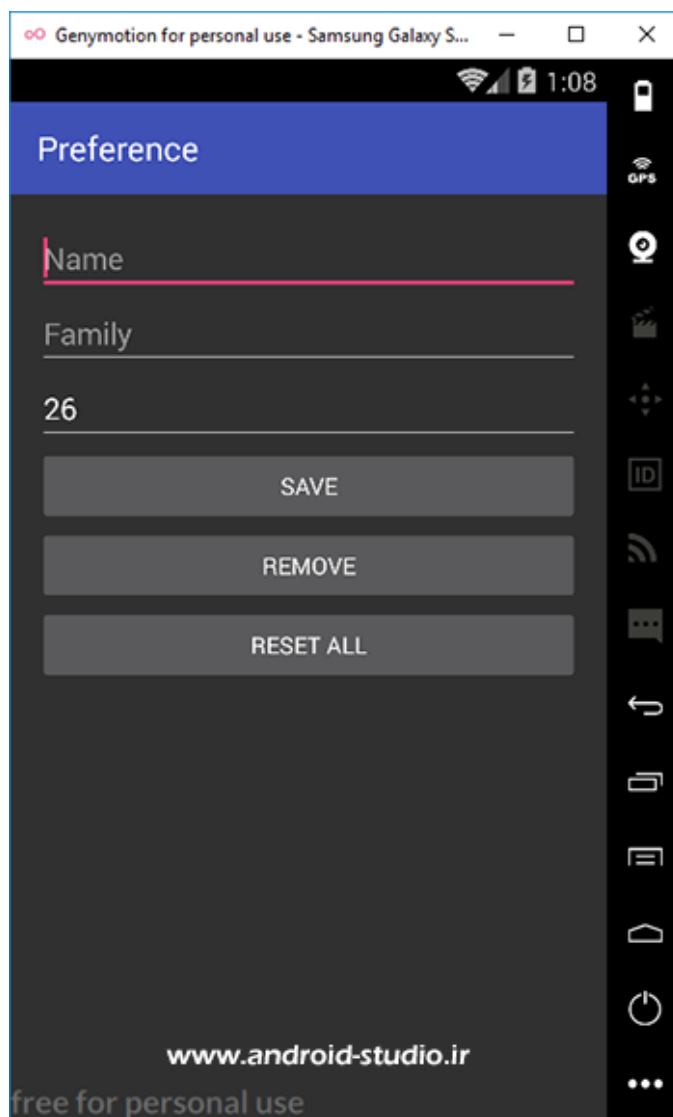
مانند دکمه save ابتدا یک نمونه ساخته و سپس متدهای موردنیاز فراخوانی شده است. توجه داشته باشید در صورتی که متد apply() یا commit() اضافه نشود، تغییرات اعمال نخواهد شد.

پروژه را اجرا می کنم:

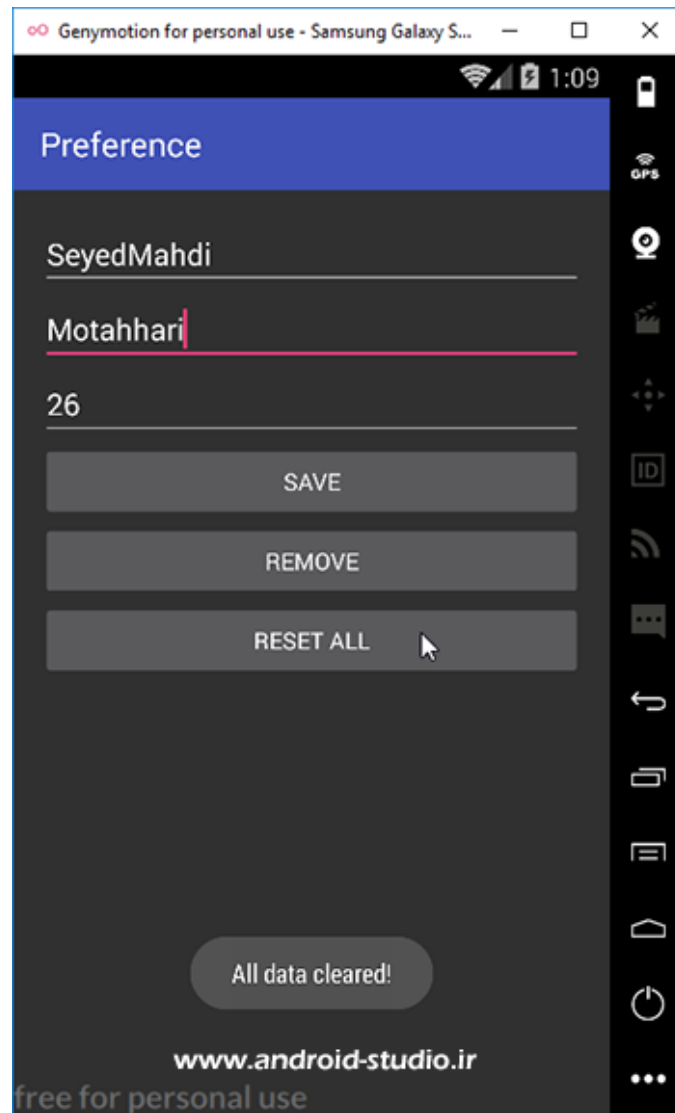


اطلاعاتی که قبلاً ذخیره شده بود نمایش داده می شود. انتظار دارم با زدن گزینه Remove و سپس بستن و اجرای مجدد برنامه، نام و نام خانوادگی حذف شده باشد:

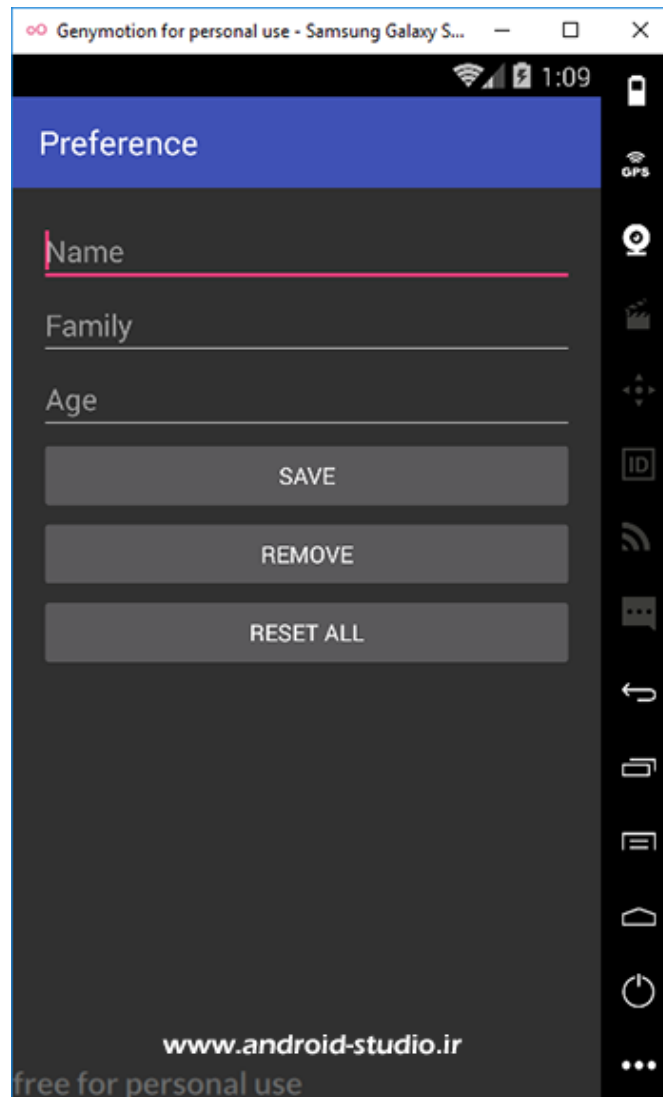




اطلاعات فیلدهای نام و نام خانوادگی با موفقیت حذف شد. در مرحله بعد برای اطمینان از عملکرد صحیح برنامه، ابتدا نام و نام خانوادگی را مجدد ذخیره کرده و در نهایت روی Reset all کلیک می‌کنم:



متد `clear()` نیز به درستی عمل کرده و پس از خروج از برنامه و اجرای مجدد، اطلاعاتی که وارد کرده بودم نمایش داده نمی شود.



مانند دیتابیس SQLite امکان دسترسی به فایل Preference نیز وجود دارد.

از منوی **Tools > Android Device Monitor** پنجره **Android Device Monitor** را باز می‌کنم. در سمت چپ (Device) نام اپلیکیشن را انتخاب کرده سپس در سمت راست در تب **File Manager** مسیر زیر را دنبال می‌کنم:

Data > data > [Application name] > shared_prefs



The screenshot displays the Android Studio interface. The top toolbar includes icons for File Explorer, Emulator, and System. The File Explorer view shows a tree structure of files and folders. The 'MyPrefs.xml' file is highlighted. The LogCat window at the bottom shows the URL 'www.android-studio.ir'.

فایل با همان نامی که تعیین کرده بودم ایجاد شده است. توسط گزینه Pull a file from a device فایل MyPrefs.xml را روی رایانه ذخیره و باز می کنم:

```

1 <?xml version='1.0' encoding='utf-8' standalone='yes' ?>
2 <map>
3   <string name="nameKey">SeyedMahdi</string>
4   <string name="familyKey">Motahhari</string>
5   <int name="ageKey" value="26" />
6 </map>
7

```

length: 202 lines: 7 Ln: 7 Col: 1 Sel: 0|0 Unix (LF) UTF-8 INS

مشاهده می کنید تمامی اطلاعات درون این فایل ذخیره شده است.



:activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context="ir.android_studio.preference.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <EditText
            android:id="@+id/edt_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Name"
            android:inputType="textPersonName" />

        <EditText
            android:id="@+id/edt_family"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Family"
            android:inputType="textPersonName" />

        <EditText
            android:id="@+id/edt_age"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Age"
            android:inputType="number" />

        <Button
            android:id="@+id/btn_save"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Save" />

        <Button
            android:id="@+id/btn_remove"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Remove" />

        <Button
            android:id="@+id/btn_clear"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Reset All" />
    </LinearLayout>
</RelativeLayout>
```




MainActivity.java

```

package ir.android_studio.preference;

import android.content.Context;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    EditText edtName;
    EditText edtFamily;
    EditText edtAge;
    Button btnSave;
    Button btnRemove;
    Button btnClear;
    SharedPreferences shPref;
    public static final String MyPref = "MyPrefers";
    public static final String Name = "nameKey";
    public static final String Family = "familyKey";
    public static final String Age = "ageKey";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        edtName = (EditText) findViewById(R.id.edt_name);
        edtFamily = (EditText) findViewById(R.id.edt_family);
        edtAge = (EditText) findViewById(R.id.edt_age);
        btnSave = (Button) findViewById(R.id.btn_save);
        btnRemove = (Button) findViewById(R.id.btn_remove);
        btnClear = (Button) findViewById(R.id.btn_clear);
        shPref = getSharedPreferences(MyPref, Context.MODE_PRIVATE);

        btnSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String n = edtName.getText().toString();
                String f = edtFamily.getText().toString();
                int a = Integer.parseInt(edtAge.getText().toString());

                SharedPreferences.Editor sEdit = shPref.edit();
                sEdit.putString(Name, n);
                sEdit.putString(Family, f);
                sEdit.putInt(Age, a);
                sEdit.apply();

                Toast.makeText(MainActivity.this, "Saved", Toast.LENGTH_LONG).show();

            }
        });

        btnRemove.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                SharedPreferences.Editor rEdit = shPref.edit();
                rEdit.remove(Name);
                rEdit.remove(Family);
            }
        });
    }
}

```



```
        rEdit.apply();

        Toast.makeText(MainActivity.this, "Removed!", Toast.LENGTH_LONG).show();
    }
});

btnClear.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        SharedPreferences.Editor cEdit = shPref.edit();
        cEdit.clear();
        cEdit.apply();

        Toast.makeText(MainActivity.this, "All data cleared!",
Toast.LENGTH_LONG).show();
    }
});

if (shPref.contains(Name)) {
    edtName.setText(shPref.getString(Name, null));
}

if (shPref.contains(Family)) {
    edtFamily.setText(shPref.getString(Family, null));
}

if (shPref.contains(Age)) {
    edtAge.setText(String.valueOf(shPref.getInt(Age, 0)));
}
}
}
```

توجه : سورس پروژه درون پوشه Exercises قرار داده شده است

با ارائه انتقادات و پیشنهادات خود، ما را در ارائه آموزش های بهتر یاری فرمائید.

www.android-studio.ir