Lesson-7

Shared Preferences & Fragments

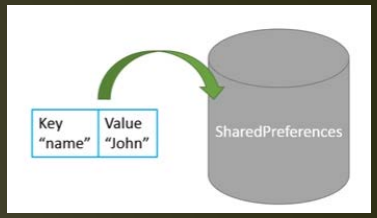
Shared Preferences

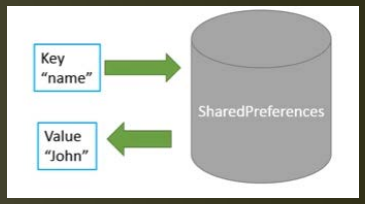
Introduction

- There are several options in android to save persistent application data. The solution you choose depend upon the requirement of the project such as, whether data should be private to your application or accessible to other application also. Following are the options which can be used for storing the data.
  - **Shared Preferences:** Store data as key-value pairs.
  - **Internal Storage:** Store data on the device memory. Data will not be accessed by other application.
  - **External Storage:** Store public data on the shared external storage. Data will be accessed by other application.
  - **SQLite Databases:** Store structured data in a private database.
  - **Network Connection:** Store data on the Web with your own network server.

SharedPreferences

- Android provides SharedPreferences object to help the developer to save simple application data.
- Using the SharedPreferences object, the developer can save and retrieve the desired data through the use of key/value pairs.
- The developer needs to specify a key for the data that is required to be saved, and then both the key and its value will be saved automatically to an XML file.
- It enables the developer to save primitive data types such as boolean, float, long, string, and so on.
- The SharedPreferences class is present in the android.content package and the developer needs to import the class to work with an object of the class.





Applications of SharedPreferences

- Last user entered in your application
- Store the last updates of date and time
- Credentials – Remember user details like user name and password
- Location catching – Identify the last location
- Store the login pattern

SharedPreferences

- Thus, before using the SharedPreferences object, use the getSharedPreferences() method to obtain an instance of the SharedPreferences object, passing it the name of the preference file as String, as well as the operating file creation mode as int that is MODE_PRIVATE(takes o - Zero) as shown below.

```
SharedPreferences prefs = getSharedPreferences(prefName, MODE_PRIVATE);
```

- By using the above object(pref) only can read the data from the pref. To write the data into the pref, you have to create an object for SheredPreferences.Editior.
- This is done by invoking the edit() method on the object of the SharedPreferences class as shown

```
SharedPreferences.Editor editor = prefs.edit();
```

- Next, the developer can add key/value pairs to the Editor object by invoking its various methods such as putBoolean(), putString(), putLong(), putInt(), or putFloat()(which method to use depends on the type of data that is being saved) as shown below

```
editor.putFloat("temperature", 85);
editor.putBoolean("authenticated", true);
editor.putString("username", "Wei-Meng Lee");
```

- Finally, the commit() method of the Editor class is invoked to commit the changes back to the SharedPreferences object so that the values can be saved to persistent storage. Code snippet 5 displays the use of commit()method.

```
editor.commit();
```

- Retrieving data from the SharedPreferences object is similar. First, one needs to obtain an instance of the SharedPreferences object and then use the various methods (depending on the type of data you are retrieving) to retrieve the values based on the keys as shown

prefs = getSharedPreferences(prefName, MODE_PRIVATE);

To retrieve the value use this syntax : **pref.getXXX("key",default_value)**

float temperature = prefs.getFloat("temperature", 50);

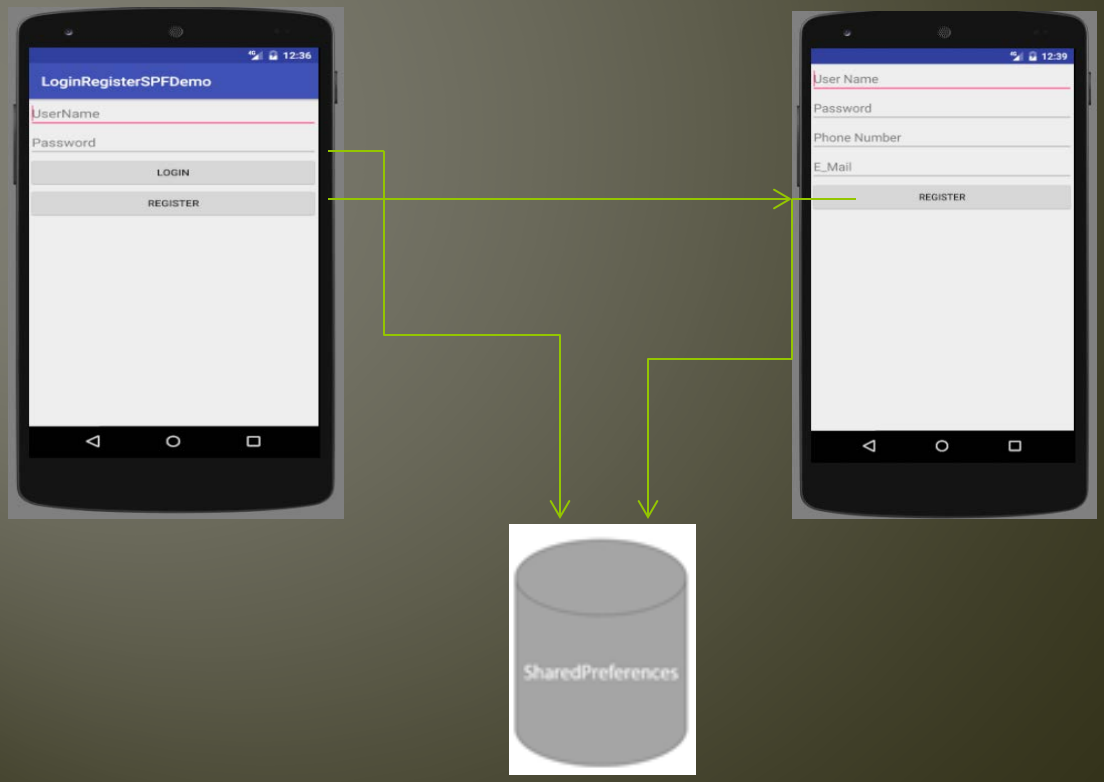
boolean authenticated = prefs.getBoolean("authenticated", false);

String username = prefs.getString("username", "");

- Internally SPF will maintain the data in a XML file, we can explore the SharedPreferences xml file using ADM [Android Device Monitor]
- The file will be saved under the directory data/data/<application_packagename>/shared preference folder. This can be viewed using DDMS perspective.
- You can view it by clicking Tools → Android→Android Device Monitor → file_explorer → data → data → pkg_name → spf_name.xml
- By using SharedPreferences, we can maintain huge amount of data , but for every value we have to give a unique key its difficult to remember & assign new keys for every value , that's why SharedPreferences is preferred to maintain the limited amount of data, if you want to maintain huge amount of data Android is preferred to use SQLite DB.

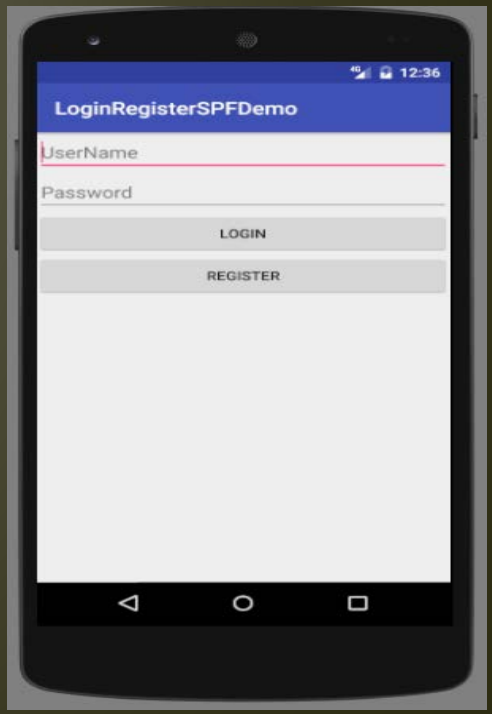
Hands-on-Example

Problem Requirement : Design an application which preform, if the user already exists they can directly click the Login button else by clicking Register button to register the information. Here we are using SharedPreferences to store the information of the user.



Activity_main.xml

```
<LinearLayout xmlns:android=
"http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    >
     <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="UserName"
        android:id="@+id/et1" />
     <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:id="@+id/et2" />
<Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Login"
        android:onClick="login"
         />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Register"
        android:onClick="register"
        />
</LinearLayout>
```



Register.xml

```
<LinearLayout xmlns:android=
"http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical">
   <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="User Name"
      android:id="@+id/et1" />
   <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Password"
      android:inputType="textPassword"
      android:id="@+id/et2" />
   <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Phone Number"
      android:id="@+id/et3" />
<EditText

android:layout_width="match_par
ent"
android:layout_height="wrap_cont
ent"
      android:hint="E_Mail"
      android:id="@+id/et4" />
<Button
android:layout_width="match_par
ent"
android:layout_height="wrap_cont
ent"
   android:text="Register"
   android:onClick="register" />
</LinearLayout>
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
  EditText et1,et2;
    @Override
    protected void onCreate(Bundle savedInstanceState)
         {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        et1 = (EditText)findViewById(R.id.et1);
        et2 = (EditText)findViewById(R.id.et2);
 }
```

MainActivity.java

```
// Check whether the user exist already in the SharedPreferences.
    public void login(View view){
        // To get the SharedPreferences using its name
        SharedPreferences spf = getSharedPreferences("login", Context.MODE_PRIVATE);
        String name = spf.getString("name","no value"); // key, value pair. Here by default name is not found assign "
no value"
        String pwd = spf.getString("pass","no value");
        if(et1.getText().toString().equalsIgnoreCase(name) &&
                et2.getText().toString().equalsIgnoreCase(pwd)){
            Toast.makeText(getApplicationContext(),"Success",Toast.LENGTH_LONG).show();
        }
        else{
            Toast.makeText(getApplicationContext(),"Fail to Login",Toast.LENGTH_LONG).show();
        }
    }
    public void register(View view){
        // We are navigating through to the RegisterActivty from ManinActivity using Intent
        Intent i = new Intent();
        i.setComponent(new ComponentName(this,RegisterActivity.class));
        startActivity(i);
    }
}
```

RegisterActivity.java

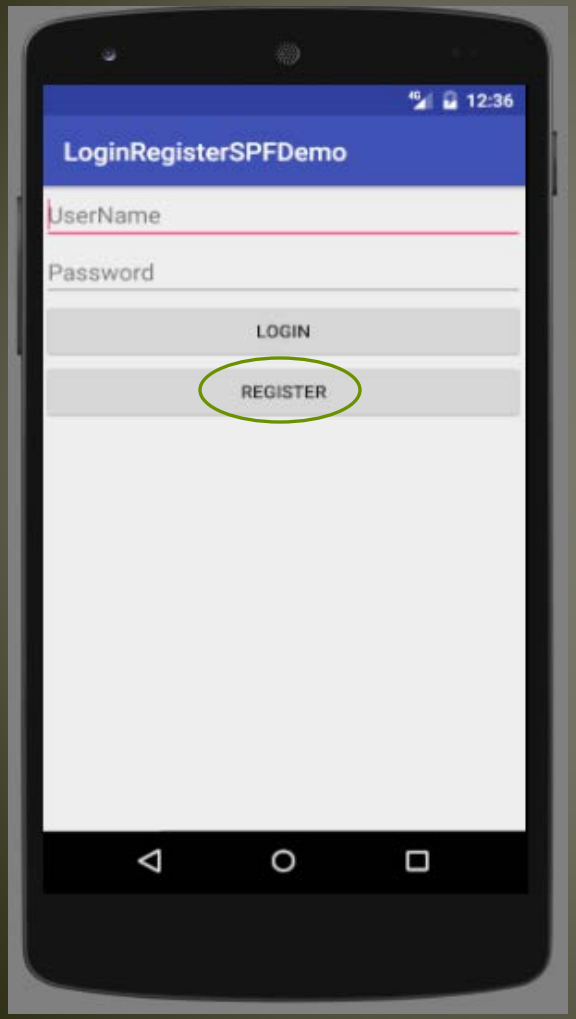
```
public class RegisterActivity extends Activity {
    EditText et1,et2,et3,et4;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.register);
        et1 = (EditText)findViewById(R.id.et1);
        et2 = (EditText)findViewById(R.id.et2);
        et3 = (EditText)findViewById(R.id.et3);
        et4 = (EditText)findViewById(R.id.et4);
    }
```

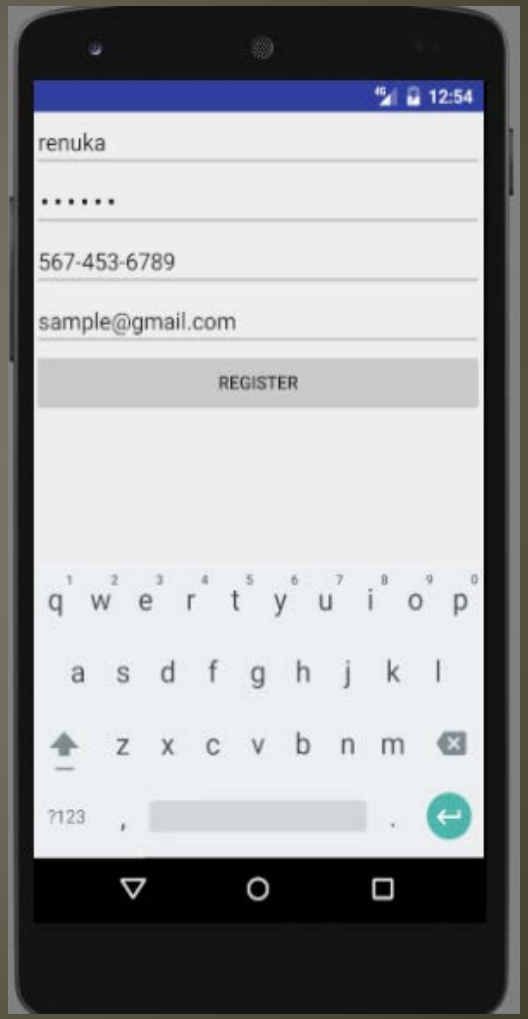
```
Register this activity in AndriodManifest.xml
<activity android:name=".RegisterActivity"></activity>
```

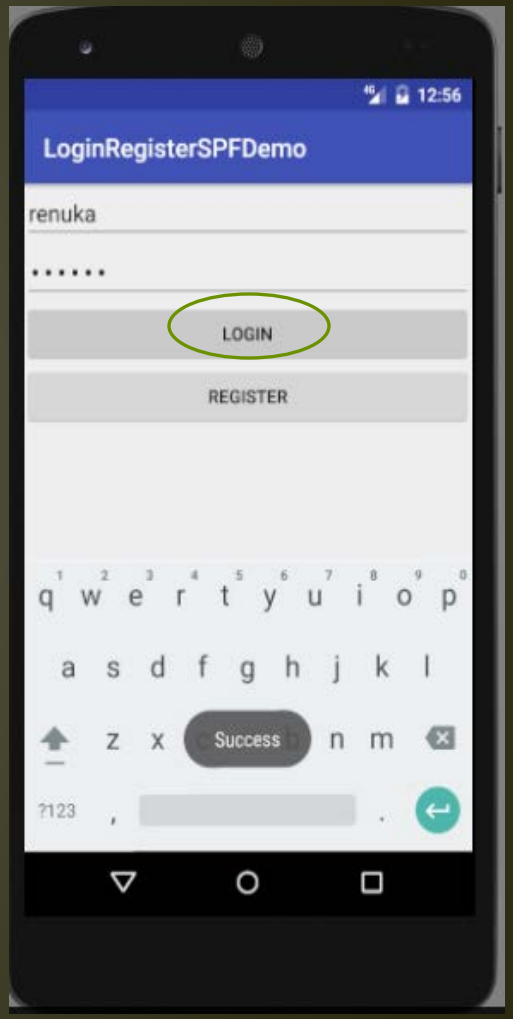
RegisterActivity.java

```
// Write the data into SharedPreferences when the user clicks Register Button
    public void register(View view){
        // Create and Read the SharedPreferences
        SharedPreferences spf = getSharedPreferences("login",
                                                    Context.MODE_PRIVATE);
        // To write a data using SharedPreferences Object
        SharedPreferences.Editor spe = spf.edit();
        // Using put method to write the data in SharedPreferences
        spe.putString("name",et1.getText().toString());
        spe.putString("pass",et2.getText().toString());
        spe.putString("phone",et3.getText().toString());
        spe.putString("email",et4.getText().toString());
        spe.commit();
        Toast.makeText(this,"Registered",Toast.LENGTH_LONG).show();
        // Once finished writing we need to go back to the main activity to show the Login
        finish(); // automatically destroy the activity and give the visibility of Main Activity
    }
}
```

Running the Code





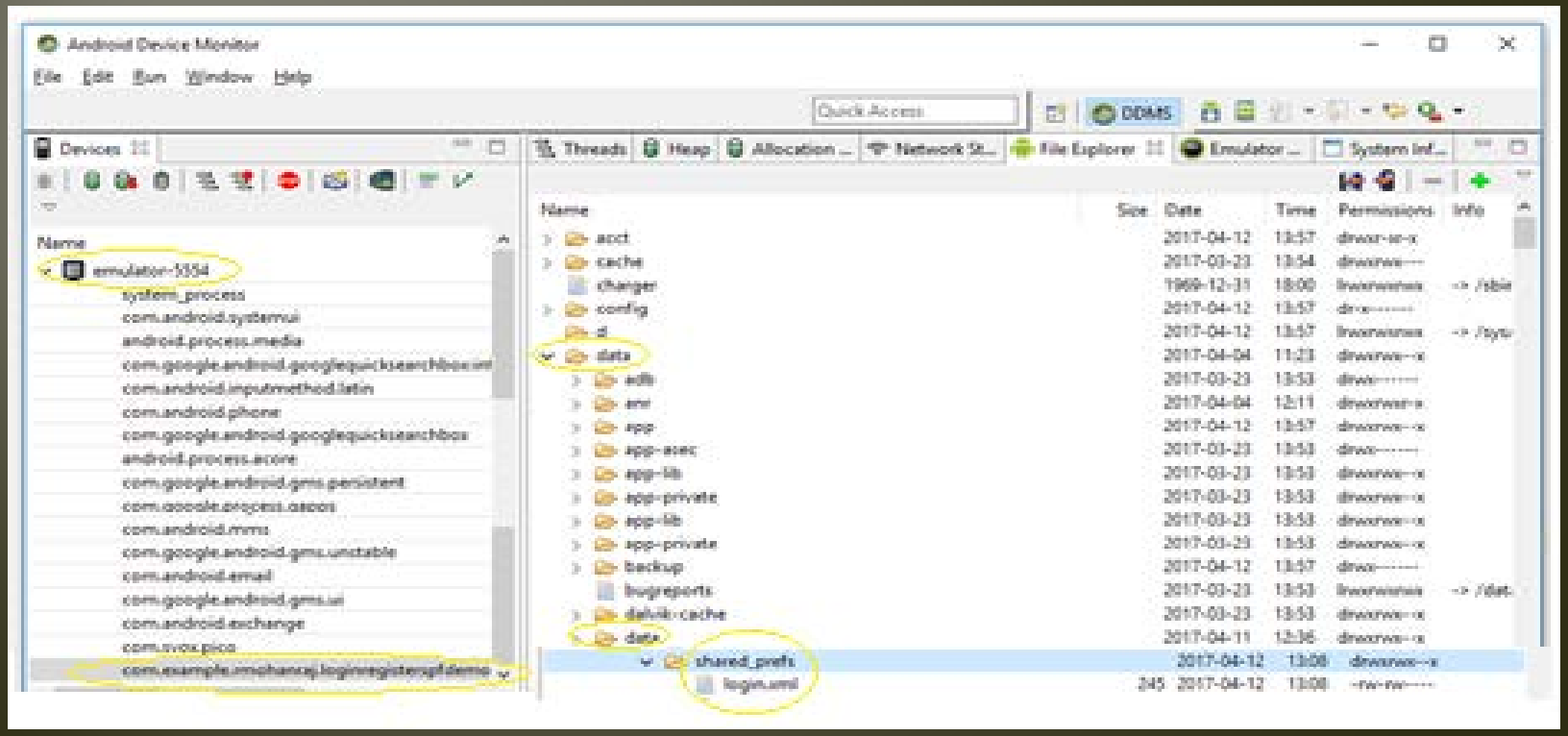


Demo Folder from Lesson6

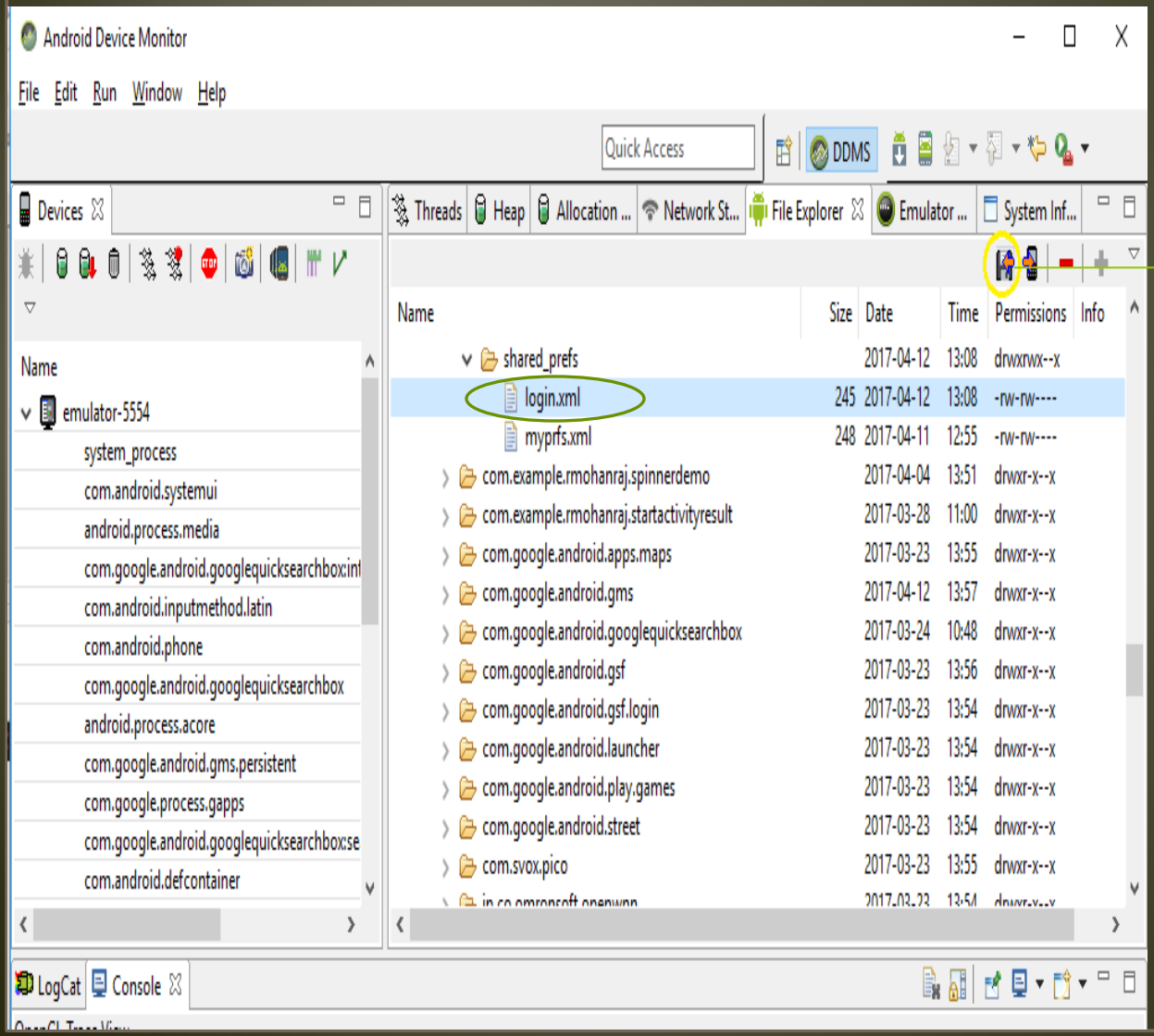
- LoginRegisterSPF
- SharedPreferencesDemo

View the SharedPreferences XML File

- Open the File Explorer
  - Tools → Android→Android Device Monitor → file_explorer → data → data → pkg_name → spf_name.xml
- Run your code through Emulator



View the SharedPreferences XML File



Click to pull the file in your local drive

login.xml

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="name">renuka</string>
    <string name="phone">45674342424</string>
    <string name="email">test@gmail.com</string>
    <string name="pass">renuka</string>
</map>
```