Demo25-文件存储

1. 数据分为瞬时数据和持久数据，持久化技术可以让临时数据转变成持久数据
2. Android保存数据的持久化技术分3种： **文件存储，SharedPreferences存储，数据库存储**

3、最后一种就是网络服务器存储，但不属于Android持久化技术。

1. 文件存储适合存储简单的文本数据+二进制数据，使用的是BufferedWriter对象调用API .write(string)来实现存储: 使用的是BufferedReader调用.readLine()来读取文件存储的数据。

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

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

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<EditText

android:id="@+id/edit"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Type something here"

/>

</LinearLayout>

1. MainActivity.java

import android.content.Context;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;

import android.widget.EditText;

import android.widget.Toast;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

public class MainActivity extends AppCompatActivity {

private EditText edit;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

edit = (EditText) findViewById(R.id.edit);

String inputText = load();

if (!TextUtils.isEmpty(inputText)) {

edit.setText(inputText);

edit.setSelection(inputText.length());

Toast.makeText(this, "Restoring succeeded", Toast.LENGTH\_SHORT).show();

}

}

@Override

protected void onDestroy() {

super.onDestroy();

String inputText = edit.getText().toString();

save(inputText);

}

public void save(String inputText) {

FileOutputStream out = null;

BufferedWriter writer = null;

try {

out = openFileOutput("data", Context.MODE\_PRIVATE);

writer = new BufferedWriter(new OutputStreamWriter(out));

writer.write(inputText);

} catch (IOException e) {

e.printStackTrace();

} finally {

try {

if (writer != null) {

writer.close();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

public String load() {

FileInputStream in = null;

BufferedReader reader = null;

StringBuilder content = new StringBuilder();

try {

in = openFileInput("data");

reader = new BufferedReader(new InputStreamReader(in));

String line = "";

while ((line = reader.readLine()) != null) {

content.append(line);

}

} catch (IOException e) {

e.printStackTrace();

} finally {

if (reader != null) {

try {

reader.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

return content.toString();

}

}

save()

openFileOutput(“data”,Context.MODE\_PRIVATE)---》FileOutputStream---》OutputStreamWriter---》BufferedWriter---》.write(sth)

1. FileOutputStream 2、OutputStreamWriter 3、BufferedWriter.write(sth)

load()

openFileInput("data")---InputStreamReader---BufferedReader---readLine()