



군산대학교
KUNSAN NATIONAL UNIVERSITY

자료 저장 및 관리 API

모바일 응용

남 광 우

안드로이드의 자료저장 관리방법

❖ 정보의 저장 방법

- ✓ Preference
- ✓ 파일시스템
- ✓ 내장 SQLite

Preference

❖ 선호설정(Preference) 지원 클래스

- ✓ android.content.SharedPreferences 인터페이스

❖ 기능

- ✓ 목적에 따라 전용 또는 공유 SharedPreferences 인스턴스를 얻는 방법
- ✓ SharedPreferences의 메서드들을 이용해서 원하는 항목의 값을 조회하는 방법
- ✓ SharedPreferences.Edit 객체를 이용하여 항목 변경 및 적용(commit)하는 방법


❖ SharedPreferences 인스턴스 얻기

- ✓ Activity의 전용 Preferences에 접근하는 방법
 - SharedPreferences ps = getPreferences(MODE_PRIVATE)
- ✓ 한 응용프로그램의 모든 Activity들이 공유하는 Prefereces에 접근
 - SharedPreferences ps = getSharedPreferences(“MySharedPref”,0)
 - 원하는 공유설정의 이름을 지정

Preference

❖ SharedPreferences 사용하기 – 생성

```
SharedPreferences pref = getSharedPreferences("Pref", Activity.MODE_PRIVATE);
```



MODE_PRIVATE
MODE_WORLD_READABLE
MODE_WORLD_WRITEABLE

Preference

❖ SharedPreferences 사용하기 – 조회

- ✓ key, value의 쌍으로 저장되어 있음
- ✓ key를 이용하여 조회

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
  
    SharedPreferences pref = getSharedPreferences("pref", Activity.MODE_PRIVATE);  
    EditText edit1 = (EditText)findViewById(R.id.EditText);  
    CheckBox check1 = (CheckBox)findViewById(R.id.CheckBox01);  
    CheckBox check2 = (CheckBox)findViewById(R.id.CheckBox02);  
  
    edit1.setText(pref.getString("editText", ""));  
    check1.setChecked(pref.getBoolean("check1", false));  
    check2.setChecked(pref.getBoolean("check2", false));  
}
```

Preference

❖ SharedPreferences 사용하기 – 조회 API

[contains\(String key\)](#) Checks whether the preferences contains a preference.

[edit\(\)](#) Create a new Editor for these preferences, through which you can make modifications to the data in the preferences and atomically commit those changes back to the SharedPreferences object.

[getAll\(\)](#) Retrieve all values from the preferences.

[getBoolean\(String key, boolean defValue\)](#) Retrieve a boolean value from the preferences.

[getFloat\(String key, float defValue\)](#) Retrieve a float value from the preferences.

[getInt\(String key, int defValue\)](#) Retrieve an int value from the preferences.

[getLong\(String key, long defValue\)](#) Retrieve a long value from the preferences.

[getString\(String key, String defValue\)](#) Retrieve a String value from the preferences.

[registerOnSharedPreferenceChangeListener\(SharedPreferences.OnSharedPreferenceChangeListener listener\)](#)
Registers a callback to be invoked when a change happens to a preference.

[unregisterOnSharedPreferenceChangeListener\(SharedPreferences.OnSharedPreferenceChangeListener listener\)](#)
Unregisters a previous callback.

Preference

❖ SharedPreferences 사용하기 – 추가 수정 삭제 PI

- ✓ Preferences의 edit() 를 통해 Editor를 받아와서 수정
- ✓ 수정후 commit() 실행

```
SharedPreferences.Editor editor = pref.edit(); // 에디터를 받아옵니다.
```

```
EditText edit1 = (EditText)findViewById(R.id.EditText);  
CheckBox check1 = (CheckBox)findViewById(R.id.CheckBox01);  
CheckBox check2 = (CheckBox)findViewById(R.id.CheckBox02);
```

```
editor.putString("editText", edit1.getText().toString()); // 데이터를 입력합니다.  
editor.putBoolean("check1", check1.isChecked());  
editor.putBoolean("check2", check2.isChecked());
```

```
editor.commit(); // 편집을 종료하고 저장합니다.
```


Preference

❖ SharedPreferences 사용하기 – 저장되는 파일의 위치

✓ 저장 위치

- data/data/패키지이름/shared_prefs 폴더에 위에서 지정한 Preference 이름으로 저장됨
 - Eclipse/DDMS로 보아함

com.androidhuman,Preferences	2009-05-24	01:12	drwxr-xr-x
lib	2009-05-24	01:12	drwxr-xr-x
shared_prefs	2009-05-24	01:13	drwxrwx--x
pref.xml	192 2009-05-26	00:41	-rw-rw----

```
<map>  
    <string name="String_Pref">고딕</String>  
    <int name="점수" value = "39"/>  
</map>
```

파일과 디렉토리 다루기

❖ File 사용하기 – 기본 디렉토리에 파일 만들고 쓰기

✓ 기본 디렉토리

– /data/data/<응용 프로그램 패키지 이름>/files/

```
import java.io.FileOutputStream;
```

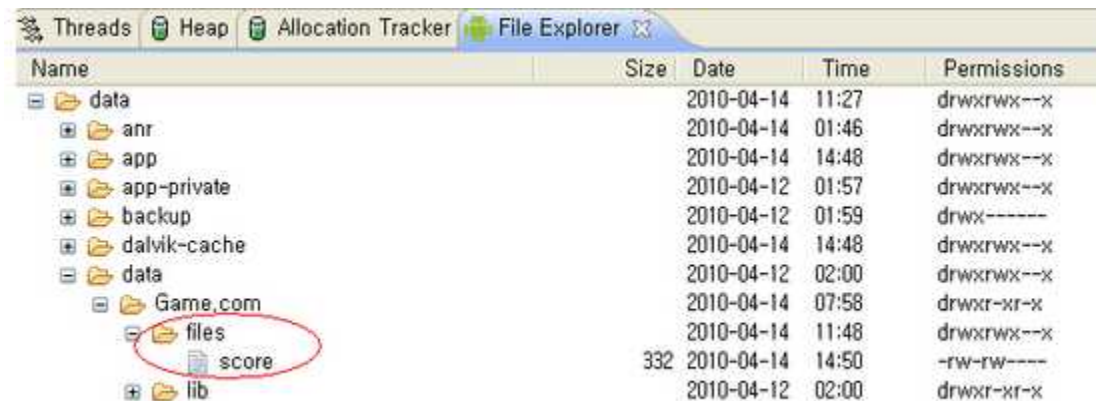
```
FileOutputStream fos;
```

```
String strFileContents = “파일 기록 테스트”;
```

```
fos = openFileOutput(“testfile.txt”, MODE_PRIVATE );
```

```
fos.write(strFileContents.getBytes());
```

```
fos.close();
```



Name	Size	Date	Time	Permissions
data		2010-04-14	11:27	drwxrwx--x
anr		2010-04-14	01:46	drwxrwx--x
app		2010-04-14	14:48	drwxrwx--x
app-private		2010-04-12	01:57	drwxrwx--x
backup		2010-04-12	01:59	drwx-----
dalvik-cache		2010-04-14	14:48	drwxrwx--x
data		2010-04-12	02:00	drwxrwx--x
Game.com		2010-04-14	07:58	drwxr-xr-x
files		2010-04-14	11:48	drwxrwx--x
score	332	2010-04-14	14:50	-rw-rw----
lib		2010-04-12	02:00	drwxr-xr-x

파일과 디렉토리 다루기

❖ File 사용하기 – 기본 디렉토리에 파일 추가 모드로 쓰기

```
fos = openFileOutput("testfile.txt", MODE_APPEND);
```

❖ File 사용하기 – 기본 디렉토리에서 파일 읽기

```
FileInputStream fis = openFileInput(filename);  
StringBuffer sBuffer = new StringBuffer();  
DataInputStream dataIO = new DataInputStream( fis);  
String strLine = null;  
  
while( (strLine = dataIO.readLine()) != null ) {  
    sBuffer.append(strLine + "\n");  
}  
  
dataIO.close();  
fis.close();
```

파일과 디렉토리 다루기

❖ File 사용하기 – Java Object Stream 으로쓰기

```
public Score(Context ct)
{
    context = ct; // 생성자에서 콘텍스트를 받는다
}
// 파일에서 score 얻기
private void ReadScore() throws Exception
{
    FileInputStream fis = null;
    fis = context.openFileInput(filePath);
    ObjectInputStream ois = new ObjectInputStream(fis);
    vScoreInfo = (Vector)ois.readObject();
    ois.close();
    fis.close();
}
// 파일에 score 쓰기
private void WriteScore() throws Exception
{
    FileOutputStream fos = null;
    fos = context.openFileOutput(filePath, context.MODE_PRIVATE); // mode : 파일이 없으면 생성됨
    ObjectOutputStream oos = new ObjectOutputStream(fos);
    oos.writeObject(vScoreInfo);
    oos.close();
    fos.close();
}
```

파일과 디렉토리 다루기

❖ File 사용하기 – SDCARD에 저장하기

- ✓ android.os.Environment 클래스를 이용

```
try {  
    File root = Environment.getExternalStorageDirectory();  
    if (root.canWrite()){  
        File gpxfile = new File(root, "gpxfile.gpx");  
        FileWriter gpxwriter = new FileWriter(gpxfile);  
        BufferedWriter out = new BufferedWriter(gpxwriter);  
        out.write("Hello world");  
        out.close();  
    }  
} catch (IOException e) {  
    Log.e(TAG, "Could not write file " + e.getMessage());  
}
```

파일과 디렉토리 다루기

❖ File 사용하기 –XML 리소스 사용하기

```
InputStream in=getResources().openRawResource(R.raw.words);
DocumentBuilder builder=DocumentBuilderFactory
    .newInstance()
    .newDocumentBuilder();
Document doc=builder.parse(in, null);
NodeList words=doc.getElementsByTagName("word");
for (int i=0;i<words.getLength();i++) {
    items.add(((Element)words.item(i)).getAttribute("value"));
}
in.close();
}
```

파일과 디렉토리 다루기

❖ File 사용하기 –XML 리소스 사용하기 – 샘플 파일

```
<words>
  <word value="lorem" />
  <word value="ipsum" />
  <word value="dolor" />
  <word value="sit" />
  <word value="amet" />
  <word value="consectetuer" />
  <word value="adipiscing" />
  <word value="elit" />
  <word value="morbi" />
  <word value="ante" />
  <word value="porttitor" />
  <word value="sodales" />
  <word value="pellentesque" />
  <word value="augue" />
  <word value="purus" />
</words>
```

word.xml