



创建自己的内容提供者

1. 实训目的

- (1) 掌握 ContentProvider 类的 6 个方法;
- (2) 掌握内容 URI 的格式;
- (3) 掌握 UriMatcher 类的使用。

2. 实训要求

- (1) 在 DatabaseTest 项目中, 新建一内容提供者 DatabaseProvider;
- (2) 新建项目 ProviderTest, 访问并操作 DatabaseTest 中的数据;
- (4) 完成实验报告。

3. 实训指导

- (1) 在 DatabaseTest 项目中, 新建一内容提供者 DatabaseProvider

```
public class DatabaseProvider extends ContentProvider {  
    public static final int BOOK_DIR=0;  
    public static final int BOOK_ITEM=1;  
    public static final int CATEGORY_DIR=2;  
    public static final int CATEGORY_ITEM=3;  
    public static final String  
AUTHORITY="com.example.cyy.databasetest.provider";  
    private static UriMatcher uriMatcher;  
    private MyDatabaseHelper dbHelper;  
    //添加希望匹配的几种 URI 格式  
    static {  
        uriMatcher=new UriMatcher(UriMatcher.NO_MATCH);  
        uriMatcher.addURI(AUTHORITY, "book", BOOK_DIR);  
        uriMatcher.addURI(AUTHORITY, "book/#", BOOK_ITEM);  
        uriMatcher.addURI(AUTHORITY, "category", CATEGORY_DIR);  
        uriMatcher.addURI(AUTHORITY, "category/#", CATEGORY_ITEM);  
    }  
    public DatabaseProvider() {
```



```
}

@Override
public int delete(Uri uri, String selection, String[]
selectionArgs) {
    // Implement this to handle requests to delete one or more
rows.
    SQLiteDatabase db=dbHelper.getWritableDatabase();
    int deleteRows=0;
    switch (uriMatcher.match(uri)) {
        case BOOK_DIR:
            deleteRows=db.delete("Book", selection, selectionArgs);
            break;
        case BOOK_ITEM:
            String bookId=uri.getPathSegments().get(1);
            deleteRows=db.delete("Book", "id=?", new
String[] {bookId});
            break;
        case CATEGORY_DIR:
            deleteRows=db.delete("Category", selection, selectionArgs);
            break;
        case CATEGORY_ITEM:
            String categoryId=uri.getPathSegments().get(1);
            deleteRows=db.delete("Category", "id=?", new
String[] {categoryId});
            break;
        default:
            break;
    }
}
```



```
    }  
    return deleteRows;  
}  
  
@Override  
public String getType(Uri uri) {  
    // TODO: Implement this to handle requests for the MIME type  
of the data  
    // at the given URI.  
    switch (uriMatcher.match(uri)) {  
        case BOOK_DIR:  
            return  
"vnd.android.cursor.dir/vnd.com.example.cyy.databasetest.provider.boo  
k";  
        case BOOK_ITEM:  
            return  
"vnd.android.cursor.item/vnd.com.example.cyy.databasetest.provider.bo  
ok";  
        case CATEGORY_DIR:  
            return  
"vnd.android.cursor.dir/vnd.com.example.cyy.databasetest.provider.cat  
egory";  
        case CATEGORY_ITEM:  
            return  
"vnd.android.cursor.item/vnd.com.example.cyy.databasetest.provider.ca  
tegory";  
    }  
    return null;  
}
```



```
@Override
public Uri insert(Uri uri, ContentValues values) {
    // TODO: Implement this to handle requests to insert a new
row.

    SQLiteDatabase db=dbHelper.getWritableDatabase();
    Uri uriReturn=null;
    switch (uriMatcher.match(uri)) {
        case BOOK_DIR:
        case BOOK_ITEM:
            long newBookId=db.insert("Book", null, values);
            //将内容 URI 解析为 Uri 对象

            uriReturn=Uri.parse("content://" +AUTHORITY+"/book/"+newBookId);
                break;
        case CATEGORY_DIR:
        case CATEGORY_ITEM:
            long newCategoryId=db.insert("Category", null, values);

            uriReturn=Uri.parse("content://" +AUTHORITY+"/category/"+newCategoryId
);
                break;
        default:
            break;
    }
    return uriReturn;
}

@Override
public boolean onCreate() {
```



```
// TODO: Implement this to initialize your content provider
on startup.

//创建 MyDatabaseHelper 类的实例，完成数据库的创建或升级操作
dbHelper=new
MyDatabaseHelper (getContext (), "BookStore.db", null, 2);
return true;
}

@Override
public Cursor query(Uri uri, String[] projection, String
selection,
                    String[] selectionArgs, String sortOrder) {
// TODO: Implement this to handle query requests from
clients.
    SQLiteDatabase db=dbHelper.getReadableDatabase ();
    Cursor cursor=null;
    //uriMatcher.match(uri)，将一个 Uri 对象传入，返回值是某个能够
    匹配这个 Uri 对象所对应的自定义代码，利用这个代码可以判断出调用方期望
    访问的哪张表的数据。
    switch (uriMatcher.match(uri)) {
        case BOOK_DIR:
            cursor=db.query ("Book", projection, selection, selectionArgs, null, null, s
            ortOrder);
            break;
        case BOOK_ITEM:
            //uri.getPathSegments () 将内容 URI 权限之后的部分以
            “/” 符号分割，分割后的结果放入字符串列表中，第 0 个位置为路径，第 1 个
            位置为 id
```



```
        String bookId=uri.getPathSegments().get(1);
        cursor=db.query("Book", projection, "id=?", new
String[] {bookId}, null, null, sortOrder);
        break;
        case CATEGORY_DIR:

cursor=db.query("Category", projection, selection, selectionArgs, null, nu
ll, sortOrder);

        break;
        case CATEGORY_ITEM:

        String categoryId=uri.getPathSegments().get(1);
        cursor=db.query("Category", projection, "id=?", new
String[] {categoryId}, null, null, sortOrder);

        break;
        default:
        break;
    }
    return cursor;
}

@Override
public int update(Uri uri, ContentValues values, String
selection,

        String[] selectionArgs) {

    // TODO: Implement this to handle requests to update one or
more rows.

    SQLiteDatabase db=dbHelper.getWritableDatabase();

    int updateRows=0;
    switch (uriMatcher.match(uri)) {
```



```
        case BOOK_DIR:

updateRows=db.update("Book", values, selection, selectionArgs);
            break;
        case BOOK_ITEM:
            String bookId=uri.getPathSegments().get(1);
            updateRows=db.update("Book", values, "id=?", new
String[] {bookId});
            break;
        case CATEGORY_DIR:

updateRows=db.update("Category", values, selection, selectionArgs);
            break;
        case CATEGORY_ITEM:
            String categoryId=uri.getPathSegments().get(1);
            updateRows=db.update("Category", values, "id=?", new
String[] {categoryId});
            break;
        default:
            break;
    }
    return updateRows;
}
}
```

(2) 新建项目 **ProviderTest**，访问并操作 **DatabaseTest** 中的数据。

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```



```
android:orientation="vertical">
```

```
<Button
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="add to book"  
    android:id="@+id/add_data"/>
```

```
<Button
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="query from book"  
    android:id="@+id/query_data"/>
```

```
<Button
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="update book"  
    android:id="@+id/update_data"/>
```

```
<Button
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="delete data"  
    android:id="@+id/delete_data"/>
```

```
</LinearLayout>
```

```
MainActivity
```

```
    public class MainActivity extends AppCompatActivity {  
        private String newId;  
        private MyDatabaseHelper dbHelper;
```




```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button addData=(Button) findViewById(R.id.add_data);
    //添加数据
    addData.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Uri
uri=Uri.parse("content://com.example.cyy.databasetest.provider/book");
            ContentValues values=new ContentValues();
            values.put("name", "A Clash of Kings");
            values.put("author", "George Martin");
            values.put("pages", 1040);
            values.put("price", 22.87);
            Uri newUri=getContentResolver().insert(uri, values);
            newId=newUri.getPathSegments().get(1);
        }
    });

    Button queryData=(Button) findViewById(R.id.query_data);
    queryData.setOnClickListener(new View.OnClickListener() {
        @Override
        //查询数据
        public void onClick(View view) {
            Uri
uri=Uri.parse("content://com.example.cyy.databasetest.provider/book");
```



Cursor

```
cursor=getContentResolver().query(uri,null,null,null,null);
    if(cursor!=null){
        while(cursor.moveToNext()){
            String
name=cursor.getString(cursor.getColumnIndex("name"));
            String
author=cursor.getString(cursor.getColumnIndex("author"));
            int
pages=cursor.getInt(cursor.getColumnIndex("pages"));
            double
price=cursor.getDouble(cursor.getColumnIndex("price"));
            Log.d("MainActivity","book name is "+name);
            Log.d("MainActivity","book author is
"+author);
            Log.d("MainActivity","book pages is "+pages);
            Log.d("MainActivity","book price is "+price);
        }
        cursor.close();
    }
});
Button updateData=(Button)findViewById(R.id.update_data);
updateData.setOnClickListener(new View.OnClickListener() {
    @Override
    //更新数据
    public void onClick(View view) {
        Uri
uri=Uri.parse("content://com.example.cyy.databasetest.provider/book/");
```



```
+newId);  
  
        ContentValues values=new ContentValues();  
        values.put("name", "A Storm of Swords");  
        values.put("pages", 1216);  
        values.put("price", 24.9);  
        getContentResolver().update(uri, values, null, null);  
    }  
});  
  
Button deleteData=(Button) findViewById(R.id.delete_data);  
deleteData.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Uri  
uri=Uri.parse("content://com.example.cyy.databasetest.provider/book/"  
+newId);  
        getContentResolver().delete(uri, null, null);  
    }  
});  
}  
}
```

```
01-25 11:14:04.527 3174-3174/com.example.cyy.provider D/MainActivity: book name is A Clash of Kings  
01-25 11:14:04.527 3174-3174/com.example.cyy.provider D/MainActivity: book author is George Martin  
01-25 11:14:04.527 3174-3174/com.example.cyy.provider D/MainActivity: book pages is 1040  
01-25 11:14:04.527 3174-3174/com.example.cyy.provider D/MainActivity: book price is 22.87
```

图 2 点击添加数据、查询数据后的运行结果

```
01-25 11:17:17.906 3174-3174/com.example.cyy.provider D/MainActivity: book name is A Storm of Swords  
01-25 11:17:17.906 3174-3174/com.example.cyy.provider D/MainActivity: book author is George Martin  
01-25 11:17:17.907 3174-3174/com.example.cyy.provider D/MainActivity: book pages is 1216  
01-25 11:17:17.907 3174-3174/com.example.cyy.provider D/MainActivity: book price is 24.9
```

图 3 点击更新数据后、查询数据后的运行结果