

Android-OrmLiteContentProvider potsetnik

Instalacija biblioteke se izvodi dodavanjem naziva u build.gradle file (takodje mora biti instalirana ormlight android biblioteka):

```
dependencies {  
    compile "com.tojc.ormlite.android:ormlite-content-provider-library:1.0.4@aar"  
}
```

Da bi ispravno koristili biblioteku moramo definisati par novih klasa i neke postojeće proširiti. Klasu koja opisuje bazu podataka ne moramo menjati. Medjutim klase koje opisuju tabele moramo proširiti sa par elemenata.

```
@DatabaseTable(tableName = AccountContract.TABLE_NAME)  
@DefaultContentUri(authority = AccountContract.AUTHORITY, path =  
AccountContract.CONTENT_URI_PATH)  
@DefaultContentMimeTypeVnd(name = AccountContract.MIMETYPE_NAME, type =  
AccountContract.MIMETYPE_TYPE)  
public class Account {  
    ...  
}
```

ne obavezno ali možemo dodati klasu Contract koja nasluedjuje BaseContract i koja u sebi sadži opis svih potrebnih elemenata da bi ContentProvider radio.

```
public final class AccountContract implements BaseColumns {  
  
    private AccountContract() {  
        // Utility constructor  
    }  
  
    public static final String TABLE_NAME = "Account";  
  
    public static final String AUTHORITY =  
"com.tojc.ormlite.android.ormlitecontentprovider.sample";  
  
    public static final String CONTENT_URI_PATH = "accounts";  
  
    public static final String MIMETYPE_TYPE = "accounts";  
    public static final String MIMETYPE_NAME =  
"com.tojc.ormlite.android.ormlitecontentprovider.sample.provider";  
  
    public static final int CONTENT_URI_PATTERN_MANY = 1;  
    public static final int CONTENT_URI_PATTERN_ONE = 2;  
  
    public static final Uri CONTENT_URI = new  
Uri.Builder().scheme(ContentResolver.SCHEME_CONTENT).authority(AUTHORITY).appendPa  
th(CONTENT_URI_PATH).build();
```

```

    public static final String NAME = "name";
}

```

Takođe moramo napraviti klasu koja opisuje naš ContentProvider:

```

public class SampleProvider extends OrmLiteSimpleContentProvider<SampleHelper> {
    @Override
    protected Class<SampleHelper> getHelperClass() {
        return SampleHelper.class;
    }

    @Override
    public boolean onCreate() {
        setMatcherController(new MatcherController())//
            .add(Account.class, SubType.DIRECTORY, "",
AccountContract.CONTENT_URI_PATTERN_MANY)//
            .add(Account.class, SubType.ITEM, "#",
AccountContract.CONTENT_URI_PATTERN_ONE));
        return true;
    }
}

```

takođe AndroidManifest.xml fajl morat proširiti za provider element koji opisuje naš novokreirani ContentProvider

insert u tabelu koristeći ContentProvider

```

ContentValues values = new ContentValues();
values.clear();
values.put(AccountContract.NAME, "Yamada Tarou");
getContentResolver().insert(AccountContract.CONTENT_URI, values);

```

insert u tabelu koristeći ContentProvider

```

Cursor c = getContentResolver().query(AccountContract.CONTENT_URI, null, null, null,
null);
c.moveToFirst();
do {
    for (int i = 0; i < c.getColumnCount(); i++) {
        Log.d(getClass().getSimpleName(), c洗getColumn洗(i) + " : " + c.getString(i));
    }
} while (c.moveToNext());
c.close();

```

Za selekciju možemo koristiti i dodatna svojstva Cursor-a ako žimo da filtriramo sadržaj ili sortiramo.

Content Provider:

[http://androidexample.com/Content_Provider_Basic/index.php?
view=article_discription&aid=120](http://androidexample.com/Content_Provider_Basic/index.php?view=article_discription&aid=120)

Android-OrmLiteContentProvider:

<https://github.com/jakenjarvis/Android-OrmLiteContentProvider/blob/master/README.md>