

Dobavljači sadržaja

Mobilne aplikacije

Dobavljači sadržaja

- Dobavljači sadržaja (ContentProvider) enkapsuliraju podatke omogućavajući da im se pristupi na standardizovan način
- Omogućava razmenu podataka između komponenti koje se nalaze u različitim procesima i obezbeđuju bezbedan (a obično i perzistentan) pristup podacima.

Dobavljači sadržaja

- Pružaju podatke drugim komponentama u formi jedne ili više tabela (slično relacionalnom modelu podataka)
- Vrsta u tabeli predstavlja instancu entiteta, a kolona njegovo svojstvo

Dobavljači sadržaja

Table 1: Primer dobavljenih podataka.

word	app id	frequency	locale	_ID
mapreduce	user1	100	en_US	1
precompiler	user14	200	fr_FR	2
applet	user2	255	fr_CA	3
const	user1	255	pt_BR	4
int	user5	100	en_UK	5

Dobavljači sadržaja

- Podatke opisuje URI i MIME tip
- URI (`content://user_dictionary/words`) se sastoji iz šeme (`content`), imena dobavljača sadržaja (`user_dictionary`) i imena tabele (`words`)
- Pojedinačnoj vrsti se može pristupiti dodavanjem njenog identifikatora na ovaj URI (`content://user_dictionary/words/1`)
- MIME tip specificira tip sadržaja (`text/plain`, `text/pdf`, `image/jpeg`, itd.)

Dobavljači sadržaja

- Za pristup podacima koje pruža dobavljač sadržaja koristi se ContentResolver klasa
- Ova klasa omogućava izvršavanje CRUD operacija nad (perzistentnim) skladištem podataka
- Pri tome se dobavljač sadržaja ne mora nalaziti u istom procesu, a automatski je obezbeđen bezbedan pristup podacima

Dobavljači sadržaja

Podacima se pristupa u dva koraka:

- Zatraže se odgovarajuća prava pristupa
- Izvrši se upit nad dobavljačem sadržaja

Dobavljači sadržaja

- Da bi mogla da pristupi podacima, komponenta mora da ima "read access permission" nad odgovarajućim provajderom
- Ovo pravo pristupa se mora specificirati u `AndroidManifest.xml` (nije ga moguće specificirati u toku izvršavanja aplikacije)

Prava pristupa

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3     <uses-permission android:name="android.permission.READ_USER_DICTIONARY">
4     <uses-permission android:name="android.permission.WRITE_USER_DICTIONARY">
5     </manifest>
6
```

Dobavljači sadržaja

- Upit se postavlja na sličan način na koji se postavlja SQL upit
- Sadrži URI, spisak kolona koje treba vratiti (projekciju), uslov koji vraćene vrste treba da zadovolje (selekciju) i način sortiranja rezultata

Upit

```
String[] projection = {  
2    UserDictionary.Words._ID,  
    UserDictionary.Words.WORD,  
4    UserDictionary.Words.LOCALE  
};  
6  
String selectionClause = null;  
8  
String[] selectionArgs = {" "};  
10  
String sortOrder = null;  
12  
cursor = getContentResolver().query(  
14    UserDictionary.Words.CONTENT_URI,  
    projection ,  
16    selectionClause ,  
    selectionArgs ,  
18    sortOrder);
```

Dobavljači sadržaja

- Na sličan način na koji je moguće pristupiti podacima, moguće ih je i promeniti.

Dobavljači sadržaja

```
// Demonstrates the usage of insert method
2 Uri newUri;

4 ContentValues newValues = new ContentValues();
  newValues.put(UserDictionary.Words.APP_ID, "example.user");
6 newValues.put(UserDictionary.Words.LOCALE, "en_US");
  newValues.put(UserDictionary.Words.WORD, "insert");
8 newValues.put(UserDictionary.Words.FREQUENCY, "100");

10 newUri = getContentResolver().insert(
    UserDictionary.Word.CONTENT_URI,
12   newValues);
```

Dobavljači sadržaja

```
// Demonstrates the usage of update method
2 ContentValues updateValues = new ContentValues();
  updateValues.putNull(UserDictionary.Words.LOCALE);
4
  String selectionClause = UserDictionary.Words.LOCALE + " LIKE ?";
6 String[] selectionArgs = {"en_%"};

8 int rowsUpdated = getContentResolver().update(
  UserDictionary.Words.CONTENT_URI,
10   updateValues,
  selectionClause,
12   selectionArgs);
```

Dobavljači sadržaja

```
// Demonstrates the usage of delete method
2 String selectionClause = UserDictionary.Words.APP_ID + " LIKE ?";
  String[] selectionArgs = {"user"};
4
  int rowsDeleted = getContentResolver().delete(
6    UserDictionary.Words.CONTENT_URI,
    selectionClause,
8    selectionArgs);
```

Dobavljači sadržaja

- Razlikuju se sistemski i aplikacioni dobavljači sadržaja
- Sistemski dobavljači sadržaja su uključeni u Android (Browser, Calendar, CallLog, Contacts, MediaStore, Settings, UserDictionary, itd.)
- Aplikacione dobavljače sadržaja pišu programeri koji pišu i ostale komponente aplikacije

Dobavljači sadržaja

Pravljenje aplikacionih dobavljača sadržaja sastoji se iz nekoliko koraka:

- Prvo treba odrediti na koji način će se skladištiti podaci
- Zatim treba naslediti `ContentProvider` klasu i deklarirati provajder u `AndroidManifest.xml`
- Na kraju treba definisati odvojenu `Contract` klasu u kojoj se nalazi ime provajdera, tabela i kolona, kao i prava pristupa

Dobavljači sadržaja

Podaci se mogu skladištiti u bilo kojoj vrsti skladišta. Međutim, trebalo bi skladištiti:

- Strukturirane podatke u SQLite
- Slike (i ostale podatke u binarnom obliku koji zauzimaju dosta prostora) u sistemu datoteka
- Takođe je moguće koristiti klase iz java.net paketa za skladištenje podataka na mreži

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3     <application ... >
4         <provider
5             android:name=". ExampleContentProvider "
6             android:authorities="ftn.mobilneaplikacije.ExampleContentProvider" />
7         </application>
8 </manifest>
```

Dobavljači sadržaja

- Prilikom nasleđivanja `ContentProvider` klase obavezno treba implementirati `query()`, `insert()`, `update()`, `delete()`, `getType()` i `onCreate()` metode
- Ove metode imaju iste potpise kao i odgovarajuće metode `ContentResolver` klase
- Sve metode osim metode `onCreate()` moraju biti thread-safe!
- Izbegavati izvršavanje dugačkih operacija u `onCreate()` metodi!
- Ne postoji `onDestroy` metoda (dobavljači sadržaja postoje od početka do kraja procesa)

ExampleContentProvider.java

```
1 public class ExampleContentProvider extends ContentProvider {
2     @Override
3     public boolean onCreate() {
4         // ...
5     }
6
7     @Override
8     public String getType(Uri uri) {
9         // ...
10    }
11
12    @Override
13    public Uri insert(Uri uri, ContentValues values) {
14        // ...
15    }
16
17    @Override
18    public Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder) {
19        // ...
20    }
21
22    @Override
23    public int update(Uri uri, ContentValues values, String selection, String[] selectionArgs) {
24        // ...
25    }
26
27    @Override
28    public int delete(Uri uri, String selection, String[] selectionArgs) {
29        // ...
30    }
31 }
32
```

Dobavljači sadržaja

- Contract klasa predstavlja "ugovor" između dobavljača sadržaja i aplikacija koje ga koriste
- Omogućava pristup dobavljaču sadržaja, čak i ako se njegova implementacija promeni (URI, imena tabela, imena kolona, itd.)
- To je javna finalna klasa koja sadrži konstante koje odgovaraju URI-u koji identifikuje podatke, njihovom MIME tipu, imenima tabela i imenima kolona
- Contract klasa takođe olakšava posao programerima zato što obično sadrži konstante koja se lako pamte

ExampleContract.java

```

1 public final class ExampleContract {
2
3     // The authority of the content provider.
4     public static final String AUTHORITY = "ftn.mobilneaplikacije.ExampleContentProvider";
5
6     // The content URI for the top-level authority.
7     public static final Uri CONTENT_URI = Uri.parse("content://" + AUTHORITY);
8
9     // Constants for an example table.
10    public final class ExampleTable {
11
12        // The content URI for the top-level authority.
13        public static final Uri TABLE_NAME = "TABLE_NAME";
14
15        // The content URI for the top-level authority.
16        public static final Uri TABLE_URI = Uri.parse("content://" + AUTHORITY + "/" + TABLE_NAME);
17
18        // The mime type of the content provider.
19        public static final Uri MIME_TYPE = "MIME_TYPE";
20
21        // The mime type of the content provider.
22        public static final Uri COLUMN_NAME = "COLUMN_NAME";
23
24        // The content URI for the top-level authority.
25        public static final Uri COLUMN_URI = Uri.parse("content://" + AUTHORITY + "/" + TABLE_NAME + "/" + COLUMN_NAME);
26    }
27 }
28

```