Grow with Google Challenge Scholarship
Lesson 8 Notes

Setup QuizExample

Before you start

Before you move on to building your QuizExample app, please do the following:

1. **Download the DroidTermsExample App -** You **need** to have this app on your phone and you need to run it once. By running this app, your phone will store word and definition data locally on your phone. To get the app, you can download the app from **here**.
2. **Review the Following Concepts -** For this lesson you will need to understand the following concepts from the SQLiteDatabase lesson:

- Contract Class
- Projection, selection, selection args, sort order
- Cursor Class

QuizExample Code

The code for this app can be found in the **Lesson08-Quiz-Example** folder of the **Toy App Repository**.
If you need to a refresher on how the code is organized, please refer to the **concept where we introduced the code flow**.

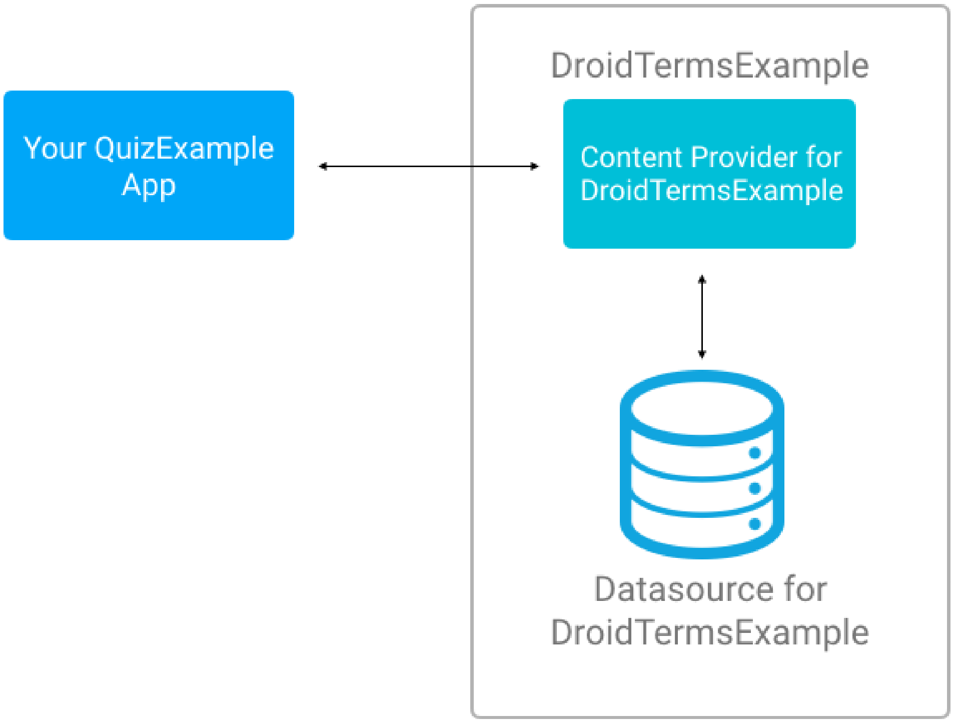
Explanation of QuizExample

Right now, QuizExample is a simple app that switches between two states. The changes made were:

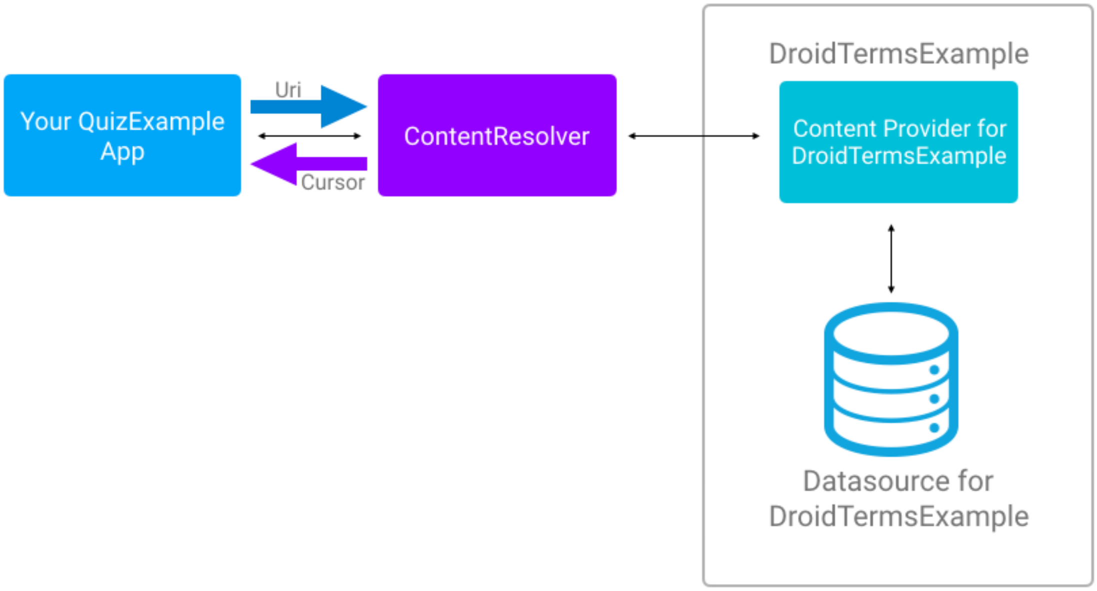
- Include a library named `droidtermsprovider` that has a class in it `DroidTermsExampleContract`
- Includes a layout for displaying the word and definition data with a button
- Includes strings for the buttons
- Includes two states: `STATE_HIDDEN` and `STATE_SHOWN` and a variable called `mCurrentState` which keeps track of the current state
- Includes a method called onButtonClick which is triggered when the button is clicked and toggles between the two states
- Includes two mostly empty helper methods `nextWord` and `showDefinition` that switch button text between two states.
  You'll use the code that has to do with state to keep track of whether the definition is showing or not.

The Big Picture

Right now your understanding of a ContentProvider is this:


Your QuizExample App
DroidTermsExample
Content Provider for DroidTermsExample
Datasource for DroidTermsExample

The fuller diagram includes the concepts shown here:


Your QuizExample App
Uri
Cursor
ContentResolver
DroidTermsExample
Content Provider for DroidTermsExample
Datasource for DroidTermsExample

General Steps for Using a ContentProvider

You will take the following steps:

1. Get permission to use the ContentProvider.
2. Get the ContentResolver
3. Pick one of four basic actions on the data: query, insert, update, delete
4. Identify the data you are reading or manipulating to create a URI
5. In the case of reading from the ContentProvider, display the information in the UI

Quiz: TVTime

For this quiz and the next, we're going to talk about a fictional app called TvTime that has a content provider. We'll apply what you've learned about communicating with providers to this app. The content provider for TVTime is a little more complicated than the DroidTermsExample app. TVTime stores two different types of data, a list of the user's favorite actors and a list of their favorite tv shows. It stores both of these using a content provider.

QUIZ QUESTION

You have a content provider which provides access to actor and TV show information stored on the phone. You have code to create and view data about TV shows and the actors in them.

Now you want the code to remove a TV show. What must you be sure to do?

- Change the content authority to allow for deleting
- Change the contract class to create a URI for deleting
- Add the READ permission for the Content Provider
- Call the delete method on the ContentResolver

Solution Explanation

Correct Answer: Call the delete method on the ContentResolver

If you are trying to specify the "action" you're performing on the data, you must pick the correct method you call on the ContentResolver. For this removing TV shows example, the code might look like this:

```
ContentResolver resolver = getContentResolver();
resolver.delete(TVTimeContract.CONTENT_URI, null, null);
```

Why the Other Answers Were Incorrect

Incorrect Answer: Change the content authority to allow for deleting and *Change the contract class to create a URI for deleting*

There are a few reasons why the first two answers are incorrect. If you are using a content provider, the contract class and the authority will be provided for you so that you can use them. You cannot change what the content authority or the contract class is. You're essentially using someone else's API to grab data, and you don't have control over the format the API expects.

Also, you don't encode the "action" you're trying to take in the URI; again, that is reserved for the method you call on the ContentResolver

Incorrect Answer: Add the READ permission for the Content Provider

It’s possible you’ll need to add permissions, but in this case, since you’re doing a delete action (which is writing to the content provider) the permission needed would be a WRITE permission.

QUIZ QUESTION

You currently have the code to display a list of TV shows (seen below), and now you want to display a list of actors. What, at the very least, would you change in the following code?

```
//Current code
ContentResolver resolver = getContentResolver();
mData = resolver.query(TVContract.Shows.CONTENT_URI, null, null, null, null);
```

○ Look at the TVContract and pick a different URI

○ Change app permissions

○ Change the method called on the ContentResolver

○ Change the ContentResolver class to the correct ContentProvider class to access the data directly

Solution Explanation

Correct Answer: Look at the TVContract and pick a different URI

If you are changing the content you are trying to access in the ContentProvider, you need to change the URI, so the first answer is correct. To query for actors, your code might look like this:

```
ContentResolver resolver = getContentResolver();
mData = resolver.query(TVContract.Actors.CONTENT_URI, null, null, null, null);
```

Why the Other Answers Were Incorrect

Incorrect Answer: Change app permissions

You are not accessing a different content provider and because you already are displaying TV shows, you must have read access, so you do not need to change permissions to query for a list of actors.

Incorrect Answer: Change the method called on the ContentResolver

The action you are taking in both cases is reading some data, which is the query method. You would change the method called on the ContentResolver if you wanted to change the type of action you were doing, for example, adding data instead of reading data. But you don't want to do that, so no need to change this.

Incorrect Answer: Change the ContentResolver to the correct ContentProvider to access the data directly

You always need to access a content provider using the content resolver, so you should never attempt to do this.

Calendar App

For this final quiz, let's assume you need to interact with one of the Google maintained Content Providers, the Calendar Provider. Below are 6 statements. Which would you use to change the color of the user's calendars to be red? Assume that this "values" variable has the information that describes new or changed data.

Hint: if you're wondering what the difference is in URIs, check out these links to the documentation for **DroidTermsExampleContract** and **CalendarContract**.

QUIZ QUESTION

Which statement would you use to turn a user's calendars colors red?

○ resolver.query(CalendarContract.CONTENT_URI, null, null, null, null);
○ resolver.update(CalendarContract.Calendars.CONTENT_URI, values, null, null);
○ resolver.insert(CalendarContract.CONTENT_URI, values);
○ resolver.update(CalendarContract.Events.CONTENT_URI, values, null, null);
○ resolver.query(CalendarContract.Calendars.CONTENT_URI, null, null, null, null);
○ resolver.delete(CalendarContract.Events.CONTENT_URI,null,null);

Solution Explanation

Correct Answer: resolver.update(CalendarContract.Calendars.CONTENT_URI, values, null, null);

```
resolver.update(CalendarContract.Calendars.CONTENT_URI, values, null, null);
```

Is the statement you would use. Since you want to change pre-existing information, you use **update**. Then you specify what you want to update by using **CalendarContract.Calendars.CONTENT_URI** as the first argument. Next you pass in the variable values, which defines what content to change. Note that **CalendarContract** is a Contract class that stores data associated with the CalendarProvider.