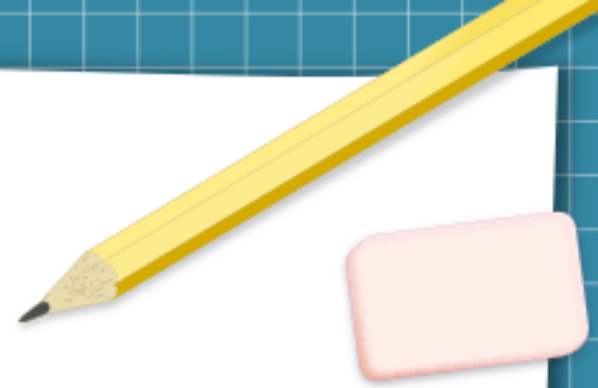




Content Providers

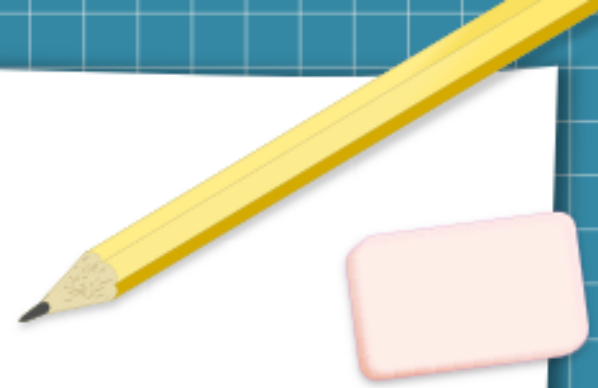
[https://developer.android.com/guide/topics/
providers/content-provider-basics](https://developer.android.com/guide/topics/providers/content-provider-basics)

Notes



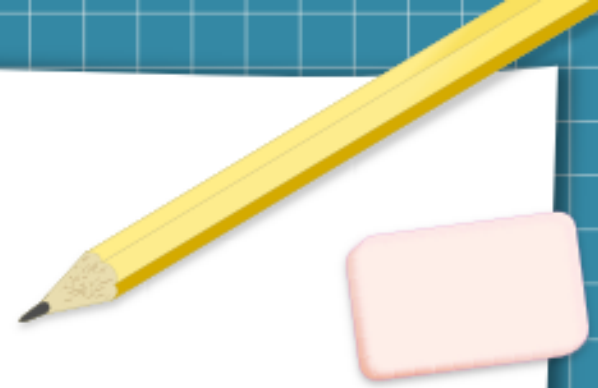
- A content provider manages access to a central repository of data.
- A provider is part of an Android application, which often provides its own UI for working with the data.
- However, content providers are primarily intended to be used by other applications, which access the provider using a provider client object.

Cont ...



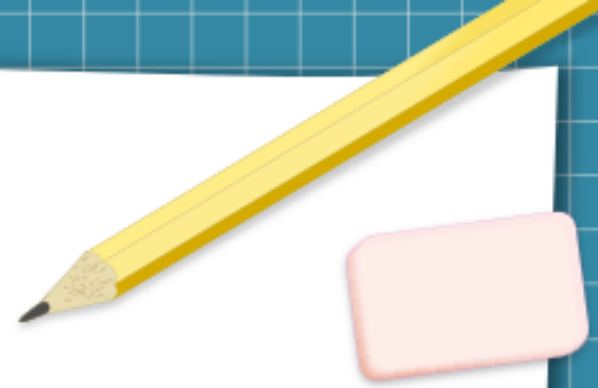
- Together, providers and provider clients offer a consistent, standard interface to data that also handles inter-process communication and secure data access.
- Typically you work with content providers in one of two scenarios; you may want to implement code to access an existing content provider in another application, or you may want to create a new content provider in your application to share data with other applications.

Overview



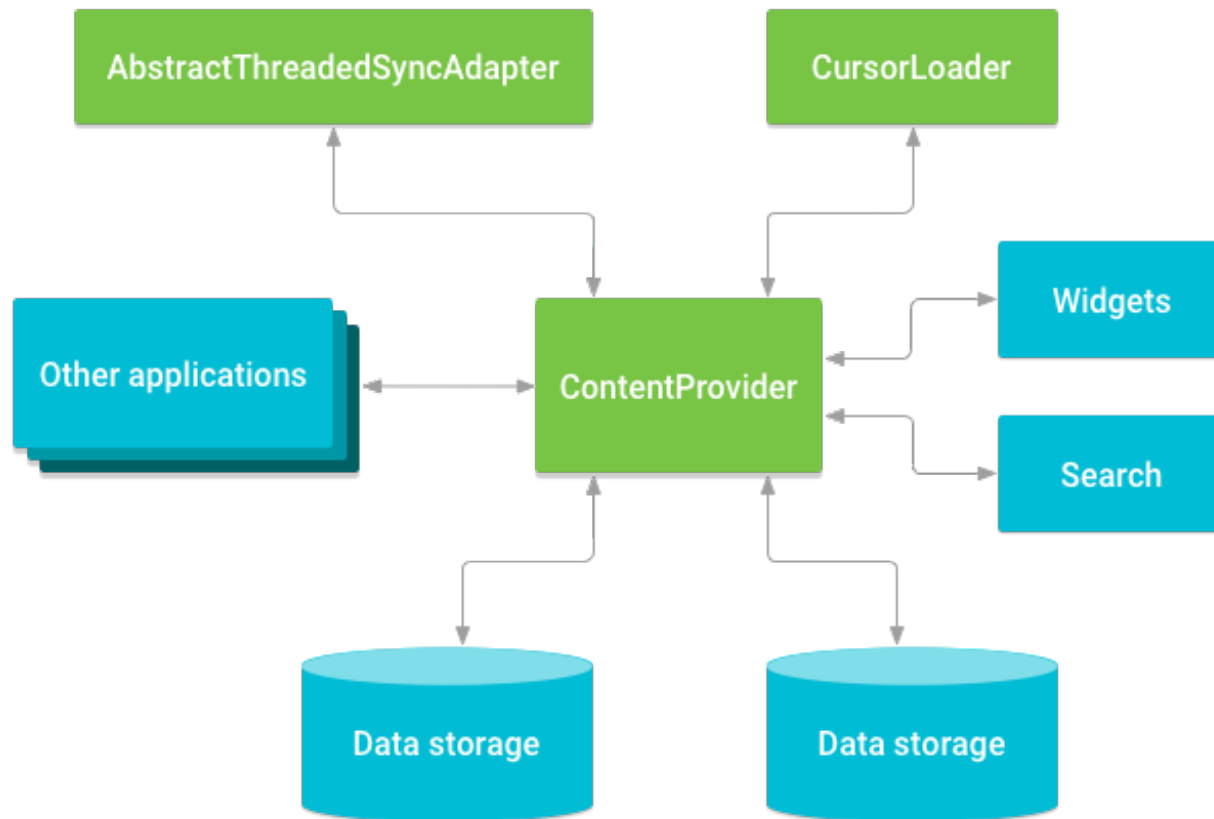
- A content provider presents data to external applications as one or more tables that are similar to the tables found in a relational database.
- A row represents an instance of some type of data the provider collects, and each column in the row represents an individual piece of data collected for an instance.

Cont ...



- A content provider coordinates access to the data storage layer in your application for a number of different APIs and components as illustrated in the figure below, these include:
 - Sharing access to your application data with other applications
 - Sending data to a widget
 - Returning custom search suggestions for your application through the search framework using `SearchRecentSuggestionsProvider`
 - Synchronizing application data with your server using an implementation of `AbstractThreadedSyncAdapter`
 - Loading data in your UI using a `CursorLoader`

Cont ...

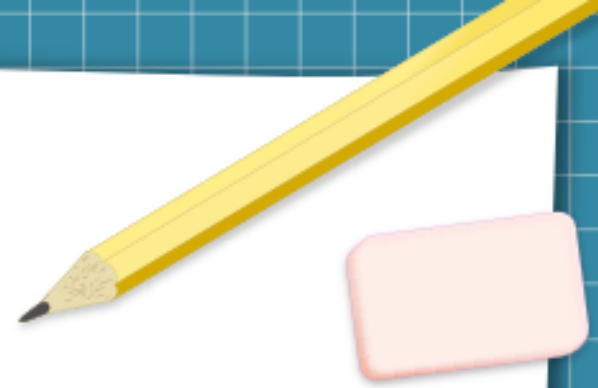


Accessing a provider



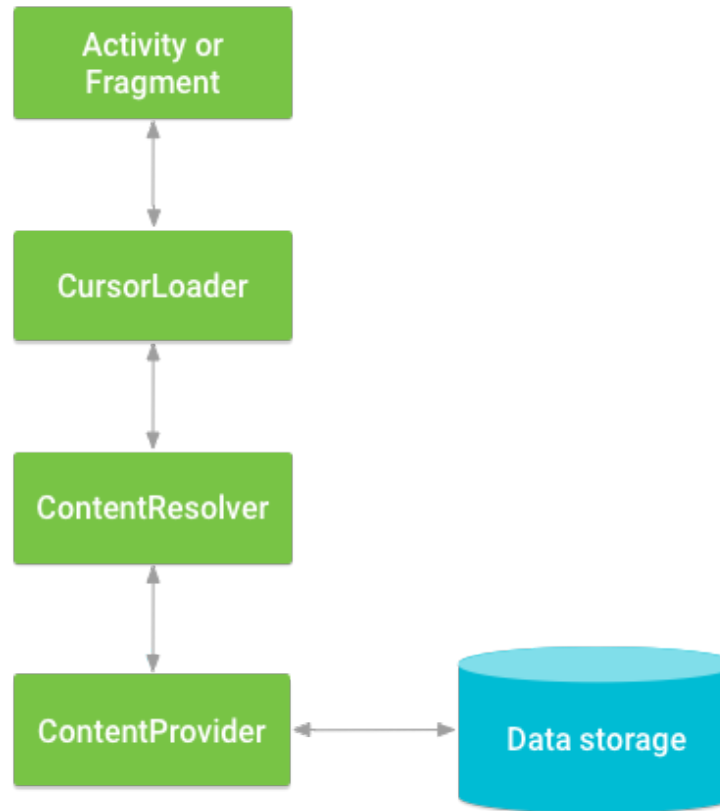
- When you want to access data in a content provider, you use the `ContentResolver` object in your application's `Context` to communicate with the provider as a client.
- The `ContentResolver` object communicates with the provider object, an instance of a class that implements `ContentProvider`.
- The provider object receives data requests from clients, performs the requested action, and returns the results.

Cont ...

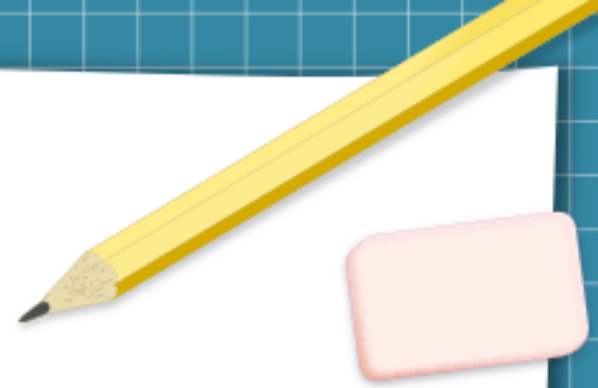


- A common pattern for accessing a `ContentProvider` from your UI uses a `CursorLoader` to run an asynchronous query in the background.
- The Activity or Fragment in your UI call a `CursorLoader` to the query, which in turn gets the `ContentProvider` using the `ContentResolver`.
- This allows the UI to continue to be available to the user while the query is running.

Cont ...

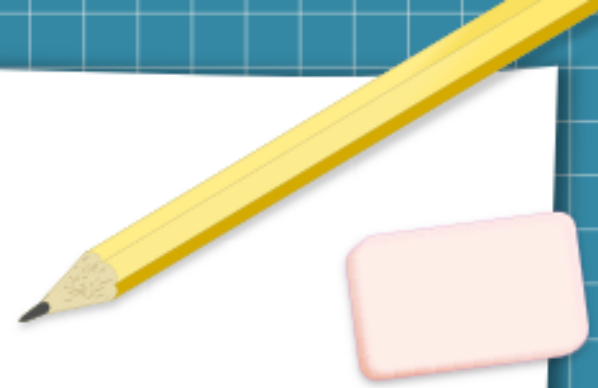


Cont ...



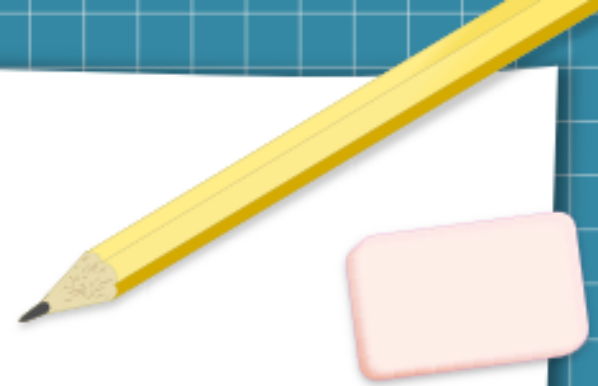
- One of the built-in providers in the Android platform is the user dictionary, which stores the spellings of non-standard words that the user wants to keep.
- See the table bellow for an example.

Cont ...



- `mCursor = getContentResolver().query(`
- `UserDictionary.Words.CONTENT_URI,` // The content URI of the words table
- `mProjection,` // The columns to return for each row
- `mSelectionClause,` // Selection criteria
- `mSelectionArgs,` // Selection criteria
- `mSortOrder);` // The sort order for the returned rows

Content URIs



- A content URI is a URI that identifies data in a provider.
- Content URIs include the symbolic name of the entire provider (its authority) and a name that points to a table (a path).
- When you call a client method to access a table in a provider, the content URI for the table is one of the arguments.

Retrieving data from the provider

A yellow pencil with a pink eraser is positioned diagonally in the top right corner of the slide, pointing towards the title.

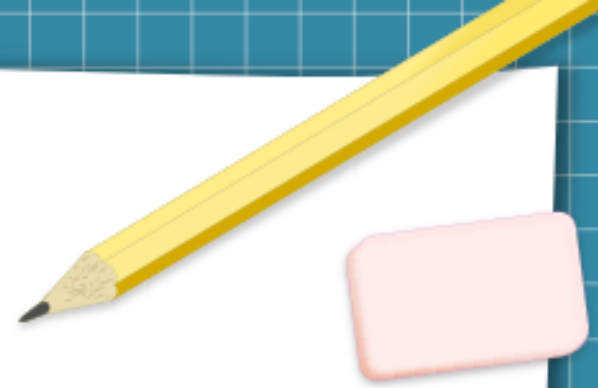
- To retrieve data from a provider, follow these basic steps:
 - Request the read access permission for the provider.
 - Define the code that sends a query to the provider.

Requesting read access permission



- To retrieve data from a provider, your application needs "read access permission" for the provider.
- You can't request this permission at run-time; instead, you have to specify that you need this permission in your manifest, using the `<uses-permission>` element and the exact permission name defined by the provider.
- When you specify this element in your manifest, you are in effect "requesting" this permission for your application. When users install your application, they implicitly grant this request.

Cont ...



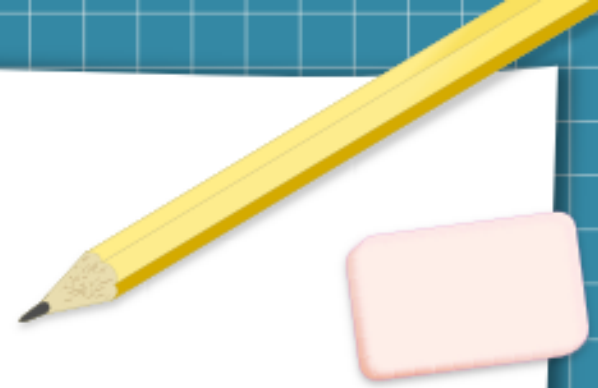
- To find the exact name of the read access permission for the provider you're using, as well as the names for other access permissions used by the provider, look in the provider's documentation.

Constructing the query



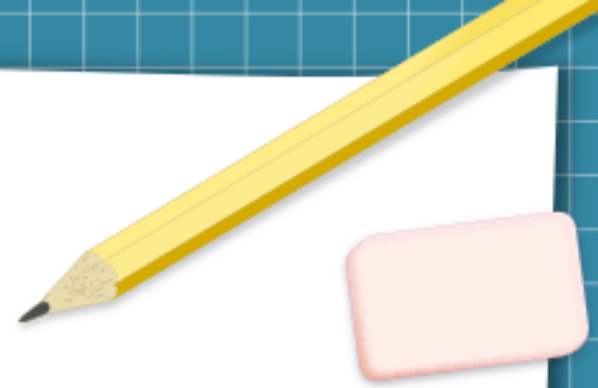
- // A "projection" defines the columns that will be returned for each row
- String[] mProjection =
- {
- UserDictionary.Words._ID, // Contract class constant for the _ID column name
- UserDictionary.Words.WORD, // Contract class constant for the word column name
- UserDictionary.Words.LOCALE // Contract class constant for the locale column name
- };
- // Defines a string to contain the selection clause
- String mSelectionClause = null;
- // Initializes an array to contain selection arguments
- String[] mSelectionArgs = {""};

Cont ...



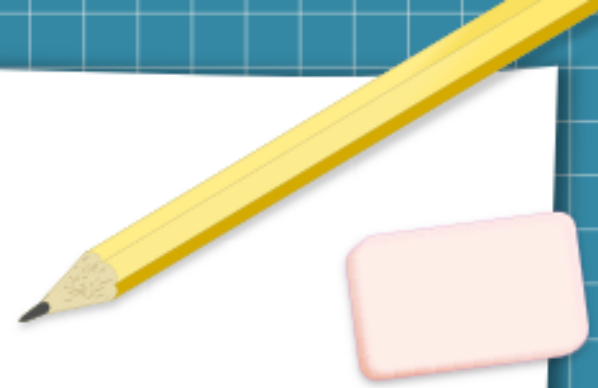
- // Gets a word from the UI
- `mSearchString = mSearchWord.getText().toString();`
- // Remember to insert code here to check for invalid or malicious input.
- // If the word is the empty string, gets everything
- `if (TextUtils.isEmpty(mSearchString)) {`
 - // Setting the selection clause to null will return all words
 - `mSelectionClause = null;`
 - `mSelectionArgs[0] = "";`
- `} else {`
 - // Constructs a selection clause that matches the word that the user entered.
 - `mSelectionClause = UserDictionary.Words.WORD + " = ?";`
 - // Moves the user's input string to the selection arguments.
 - `mSelectionArgs[0] = mSearchString;`
- `}`

Cont ...



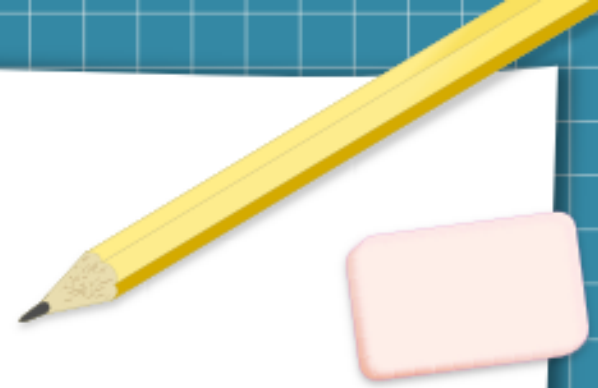
- // Does a query against the table and returns a Cursor object
- mCursor = getContentResolver().query(
• UserDictionary.Words.CONTENT_URI, // The content URI of the words
table
- mProjection, // The columns to return for each row
- mSelectionClause, // Either null, or the word the user entered
- mSelectionArgs, // Either empty, or the string the user entered
- mSortOrder); // The sort order for the returned rows

Cont ...



- `// Some providers return null if an error occurs, others throw an exception`
- `if (null == mCursor) {`
- `/*`
- `* Insert code here to handle the error. Be sure not to use the cursor! You may want to`
- `* call android.util.Log.e() to log this error.`
- `*`
- `*/`
- `// If the Cursor is empty, the provider found no matches`
- `} else if (mCursor.getCount() < 1) {`
- `/*`
- `* Insert code here to notify the user that the search was unsuccessful. This isn't necessarily`
- `* an error. You may want to offer the user the option to insert a new row, or re-type the`
- `* search term.`
- `*/`
- `} else {`
- `// Insert code here to do something with the results`
- `}`

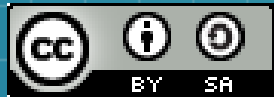
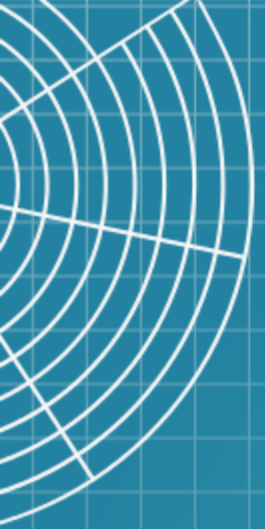
Cont ...



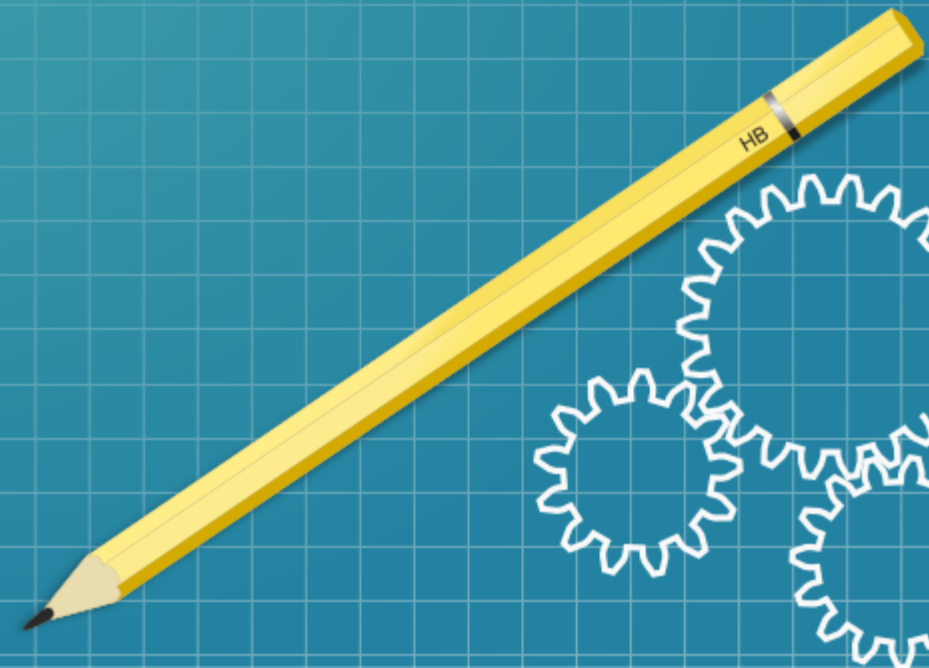
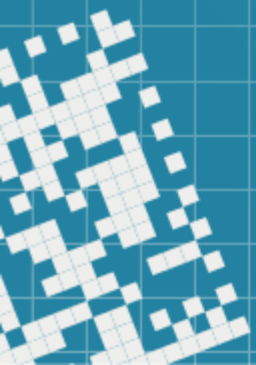
- This query is analogous to the SQL statement:
 - `SELECT _ID, word, locale FROM words WHERE word = <userinput> ORDER BY word ASC;`

Contacts Provider

- <https://developer.android.com/training/contacts-provider/retrieve-names>



This work is licensed under a Creative Commons
Attribution-ShareAlike 3.0 Unported License.
It makes use of the works of Mateus Machado Luna.

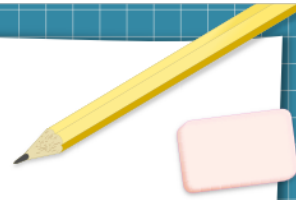




Content Providers

[https://developer.android.com/guide/topics/
providers/content-provider-basics](https://developer.android.com/guide/topics/providers/content-provider-basics)

Notes



- A content provider manages access to a central repository of data.
- A provider is part of an Android application, which often provides its own UI for working with the data.
- However, content providers are primarily intended to be used by other applications, which access the provider using a provider client object.

Cont ...

- Together, providers and provider clients offer a consistent, standard interface to data that also handles inter-process communication and secure data access.
- Typically you work with content providers in one of two scenarios; you may want to implement code to access an existing content provider in another application, or you may want to create a new content provider in your application to share data with other applications.

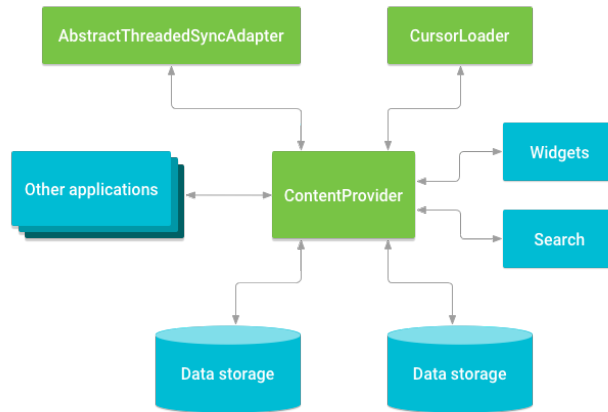
Overview

- A content provider presents data to external applications as one or more tables that are similar to the tables found in a relational database.
- A row represents an instance of some type of data the provider collects, and each column in the row represents an individual piece of data collected for an instance.

Cont ...

- A content provider coordinates access to the data storage layer in your application for a number of different APIs and components as illustrated in the figure below, these include:
 - Sharing access to your application data with other applications
 - Sending data to a widget
 - Returning custom search suggestions for your application through the search framework using `SearchRecentSuggestionsProvider`
 - Synchronizing application data with your server using an implementation of `AbstractThreadedSyncAdapter`
 - Loading data in your UI using a `CursorLoader`

Cont ...



Accessing a provider

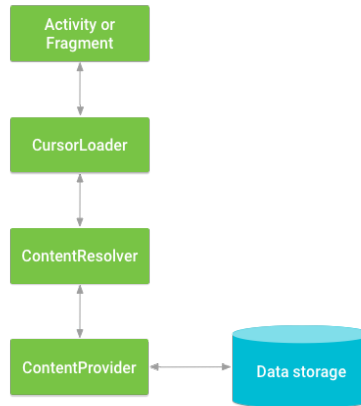


- When you want to access data in a content provider, you use the ContentResolver object in your application's Context to communicate with the provider as a client.
- The ContentResolver object communicates with the provider object, an instance of a class that implements ContentProvider.
- The provider object receives data requests from clients, performs the requested action, and returns the results.

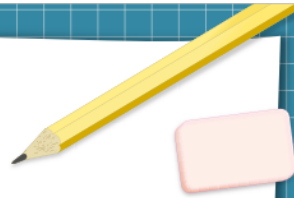
Cont ...

- A common pattern for accessing a `ContentProvider` from your UI uses a `CursorLoader` to run an asynchronous query in the background.
- The Activity or Fragment in your UI call a `CursorLoader` to the query, which in turn gets the `ContentProvider` using the `ContentResolver`.
- This allows the UI to continue to be available to the user while the query is running.

Cont ...

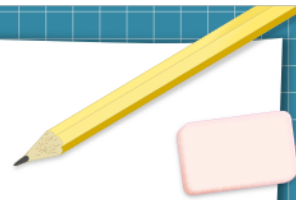


Cont ...



- One of the built-in providers in the Android platform is the user dictionary, which stores the spellings of non-standard words that the user wants to keep.
- See the table bellow for an example.

Cont ...



- `mCursor = getContentResolver().query(`
- `UserDictionary.Words.CONTENT_URI,` // The content URI of the words table
- `mProjection,` // The columns to return for each row
- `mSelectionClause,` // Selection criteria
- `mSelectionArgs,` // Selection criteria
- `mSortOrder);` // The sort order for the returned rows

Content URIs



- A content URI is a URI that identifies data in a provider.
- Content URIs include the symbolic name of the entire provider (its authority) and a name that points to a table (a path).
- When you call a client method to access a table in a provider, the content URI for the table is one of the arguments.

Retrieving data from the provider



- To retrieve data from a provider, follow these basic steps:
 - Request the read access permission for the provider.
 - Define the code that sends a query to the provider.

Requesting read access permission



- To retrieve data from a provider, your application needs "read access permission" for the provider.
- You can't request this permission at run-time; instead, you have to specify that you need this permission in your manifest, using the <uses-permission> element and the exact permission name defined by the provider.
- When you specify this element in your manifest, you are in effect "requesting" this permission for your application. When users install your application, they implicitly grant this request.

Cont ...

- To find the exact name of the read access permission for the provider you're using, as well as the names for other access permissions used by the provider, look in the provider's documentation.

Constructing the query

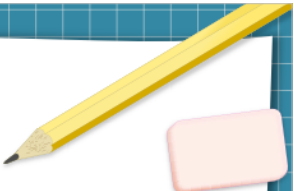


- // A "projection" defines the columns that will be returned for each row
- String[] mProjection =
- {
- UserDictionary.Words._ID, // Contract class constant for the _ID column name
- UserDictionary.Words.WORD, // Contract class constant for the word column name
- UserDictionary.Words.LOCALE // Contract class constant for the locale column name
- };

- // Defines a string to contain the selection clause
- String mSelectionClause = null;

- // Initializes an array to contain selection arguments
- String[] mSelectionArgs = {""};

Cont ...

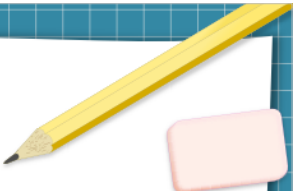


- // Gets a word from the UI
- `mSearchString = mSearchWord.getText().toString();`
- // Remember to insert code here to check for invalid or malicious input.
- // If the word is the empty string, gets everything
- `if (TextUtils.isEmpty(mSearchString)) {`
- // Setting the selection clause to null will return all words
- `mSelectionClause = null;`
- `mSelectionArgs[0] = "";`
- `} else {`
- // Constructs a selection clause that matches the word that the user entered.
- `mSelectionClause = UserDictionary.Words.WORD + " = ?";`
- // Moves the user's input string to the selection arguments.
- `mSelectionArgs[0] = mSearchString;`
- `}`

Cont ...

- // Does a query against the table and returns a Cursor object
- mCursor = getContentResolver().query(
 - UserDictionary.Words.CONTENT_URI, // The content URI of the words
 - table
 - mProjection, // The columns to return for each row
 - mSelectionClause, // Either null, or the word the user entered
 - mSelectionArgs, // Either empty, or the string the user entered
 - mSortOrder); // The sort order for the returned rows

Cont ...



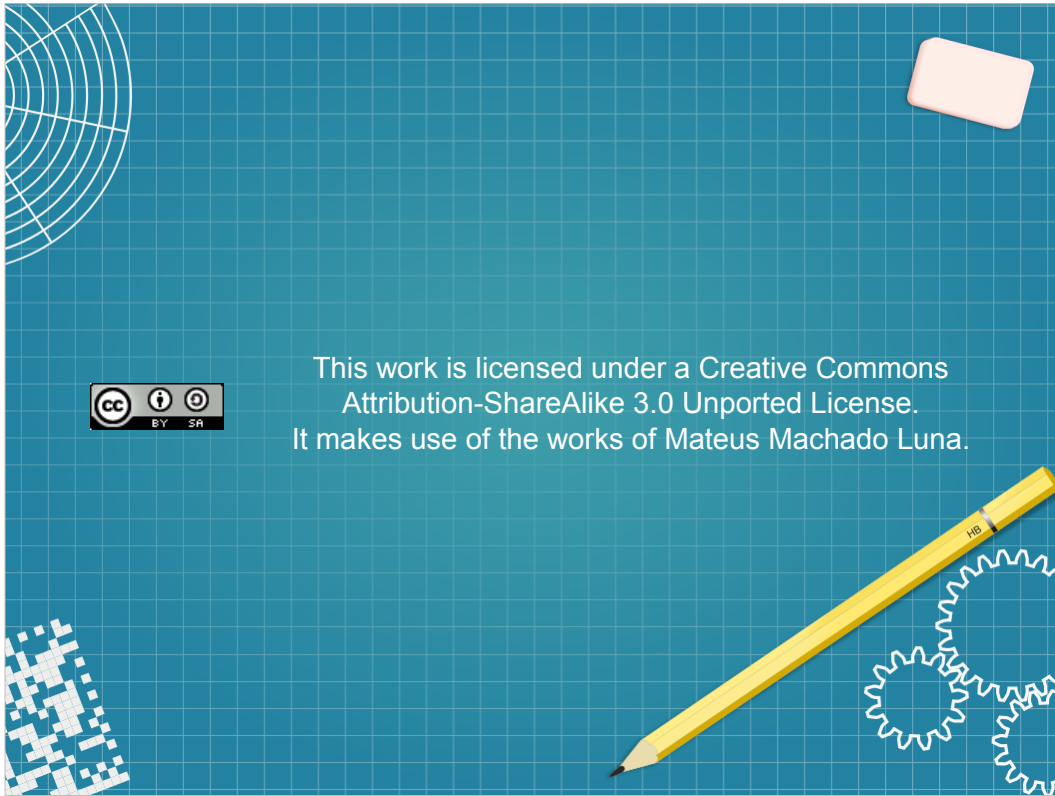
```
• // Some providers return null if an error occurs, others throw an exception
• if (null == mCursor) {
•     /*
•     * Insert code here to handle the error. Be sure not to use the cursor! You may want to
•     * call android.util.Log.e() to log this error.
•     *
•     */
• // If the Cursor is empty, the provider found no matches
• } else if (mCursor.getCount() < 1) {
•     /*
•     * Insert code here to notify the user that the search was unsuccessful. This isn't necessarily
•     * an error. You may want to offer the user the option to insert a new row, or re-type the
•     * search term.
•     */
• } else {
•     // Insert code here to do something with the results
• }
```

Cont ...

- This query is analogous to the SQL statement:
 - `SELECT _ID, word, locale FROM words WHERE word = <userinput> ORDER BY word ASC;`

Contacts Provider

- <https://developer.android.com/training/contacts-provider/retrieve-names>



This work is licensed under a Creative Commons
Attribution-ShareAlike 3.0 Unported License.
It makes use of the works of Mateus Machado Luna.