

Sharing Data Between Application With Content Providers

* Explain Content Provider in Android.

⇒ Content Providers in Android are components that manages access to a structured set of data.

Content Provider allowing to share securely data between application and manage access permissions.

They serve as an abstraction layer between application and the underlying data source.

It allows to perform CRUD operations on a data source using ContentResolver.

Content Provider enable data sharing between different application in secure manner.

=> Steps to create a Content Provider.

1 Define Data Contract:

Define the structure of your data using a Data Contract classes.

2 Implement Content Provider:

Create a class that extends 'ContentProvider' and implement CRUD operation.

3 Declare Provider in Manifest:

Declare your Content Provider in the AndroidManifest.xml File using the '<provider>' element.

4 Use Content Resolver:

Use 'ContentResolver' in your application to interact with the Content Provider and perform data operations.

=> Example:

-> Data Contract Class:

```
public class MyDataContract
{
    public static final String
        AUTHORITY = "com.example.myapplication.provider";

    public static final Uri content-url =
        Uri.parse("content://"+AUTHORITY+
            "/mydata");
}
```

-> Manifest Declaration:

```
<provider
    android:name=".MyContentProvider"
    android:authorities="com.example.myapplication.provider"
    android:exported="true" />
```

-> Content Provider:

```
public class MyContentProvider extends
    ContentProvider
```

```
{
```

```
    private MyDatabaseHelper dbHelper;
```

```
    public boolean onCreate()
```

```
{
```

```
        base
```

```
        dbHelper = new MyDatabaseHelper
            (getContext());
```

```
        return true;
```

```
}
```

```
    public Uri insert(Uri uri, ContentValues values)
```

```
{
```

```
        SQLiteDatabase db = dbHelper.
            getWritableDatabase();
```

```
        long id = db.insert("my table", null,
            values);
```

```
        return ContentUris.withAppended
            Id(MyDataContract.CONTENT_URI,
            id);
```

```
}
```

```
}
```