Loaders

Why use Loaders?

- Loaders provide an interface for loading data asynchronously
- Loaders can monitor data source and deliver new results when content changes

The Cast

LoaderManager

One loader manager per Activity / Fragment for management of multiple loaders

LoaderManager.LoaderCallbacks

Callback interface for interacting with the LoaderManager

CursorLoader

Performs actual loading of data from Cursor

AsyncTaskLoader

Provides an AsyncTask to do work in background

Implementation

Subclass AsyncTaskLoader instead of CursorLoader

When implementing without a Content Provider

Create abstract method loadCursor()

Our AsyncTaskLoader Subclass will be an abstract class designed to be subclassed. In our subclass we'll override loadCursor for specific implementations

Subclass AsyncTaskLoaderSubclass and implement loadCursor()

In Activity or Fragment, implement LoaderManager.LoaderCallbacks<Cursor>

- `onCreateLoader(int id, Bundle args)` - Return a new CursorLoader
- `onLoadFinished(Loader<D> loader, D data)` - Set adapter with cursor data
- `onLoaderReset(Loader<D> loader)` - Set adapter to null