Developers

列表视图

本文内容

关键类

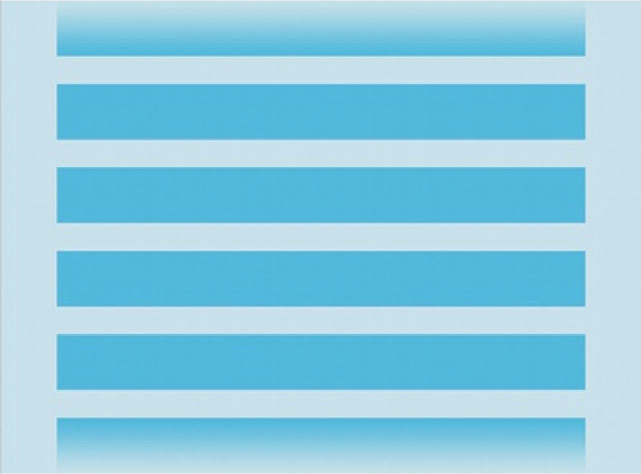
- ListView
- Adapter
- CursorLoader

另请参阅

- 加载器

ListView 是一个显示一列可滚动项目的视图组。 系统使用 Adapter 自动将列表项目插入列表，适配器从来源（例如数组或数据库查询）提取内容，并将每个项目结果转换为视图放置到列表中。

有关如何使用适配器动态插入视图的介绍，请阅读使用适配器构建布局。



使用加载器

使用 CursorLoader 是以异步任务形式查询 Cursor 的标准方式，可避免查询阻塞应用的主线程。 当 CursorLoader 接收到 Cursor 结果时，LoaderCallbacks 会收到对 onLoadFinished() 的回调，从中您可以使用新的 Cursor 更新 Adapter，然后列表视图会显示结果。

虽然 CursorLoader API 首先是在 Android 3.0（API 级别 11）中引入，但支持库中也有提供，因此您的应用可在使用这些 API 的同时，仍为 Android 1.6 或更高版本的设备提供支持。

如需了解有关使用 Loader 异步加载数据的详细信息，请参阅加载器指南。

示例

下例使用 ListActivity，该 Activity 包含一个 ListView 作为其仅有的默认布局元素。 它对联系人提供程序执行查询，以获取姓名和电话号码清单。

Activity 实现 LoaderCallbacks 接口，以使用 CursorLoader 为列表视图动态加载数据。

```
public class ListViewLoader extends ListActivity
        implements LoaderManager.LoaderCallbacks<Cursor> {

    // This is the Adapter being used to display the list's data
    SimpleCursorAdapter mAdapter;

    // These are the Contacts rows that we will retrieve
    static final String[] PROJECTION = new String[] {ContactsContract.Data._ID,
            ContactsContract.Data.DISPLAY_NAME};

    // This is the select criteria
    static final String SELECTION = "((" +
            ContactsContract.Data.DISPLAY_NAME + " NOTNULL) AND (" +
            ContactsContract.Data.DISPLAY_NAME + " != '' ))";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Create a progress bar to display while the list loads
        ProgressBar progressBar = new ProgressBar(this);
        progressBar.setLayoutParams(new LayoutParams(LayoutParams.WRAP_CONTENT,
                LayoutParams.WRAP_CONTENT, Gravity.CENTER));
        progressBar.setIndeterminate(true);
        getListView().setEmptyView(progressBar);

        // Must add the progress bar to the root of the layout
        ViewGroup root = (ViewGroup) findViewById(android.R.id.content);
        root.addView(progressBar);

        // For the cursor adapter, specify which columns go into which views
        String[] fromColumns = {ContactsContract.Data.DISPLAY_NAME};
        int[] toViews = {android.R.id.text1}; // The TextView in simple_list_item_1

        // Create an empty adapter we will use to display the loaded data.
        // We pass null for the cursor, then update it in onLoadFinished()
        mAdapter = new SimpleCursorAdapter(this,
                android.R.layout.simple_list_item_1, null,
                fromColumns, toViews, 0);
        setListAdapter(mAdapter);

        // Prepare the loader.  Either re-connect with an existing one,
        // or start a new one.
        getLoaderManager().initLoader(0, null, this);
    }

    // Called when a new Loader needs to be created
    public Loader<Cursor> onCreateLoader(int id, Bundle args) {
        // Now create and return a CursorLoader that will take care of
        // creating a Cursor for the data being displayed.
        return new CursorLoader(this, ContactsContract.Data.CONTENT_URI,
                PROJECTION, SELECTION, null, null);
    }

    // Called when a previously created loader has finished loading
    public void onLoadFinished(Loader<Cursor> loader, Cursor data) {
        // Swap the new cursor in.  (The framework will take care of closing the
        // old cursor once we return.)
        mAdapter.swapCursor(data);
    }

    // Called when a previously created loader is reset, making the data unavailable
    public void onLoaderReset(Loader<Cursor> loader) {
        // This is called when the last Cursor provided to onLoadFinished()
        // above is about to be closed.  We need to make sure we are no
        // longer using it.
        mAdapter.swapCursor(null);
    }

    @Override
    public void onListItemClick(ListView l, View v, int position, long id) {
        // Do something when a list item is clicked
    }
}
```

注：由于此示例在联系人提供程序中执行查询，因此，如果您想要试用此代码，您的应用必须在清单文件中请求 READ_CONTACTS 权限：

```
<uses-permission android:name="android.permission.READ_CONTACTS" />
```