

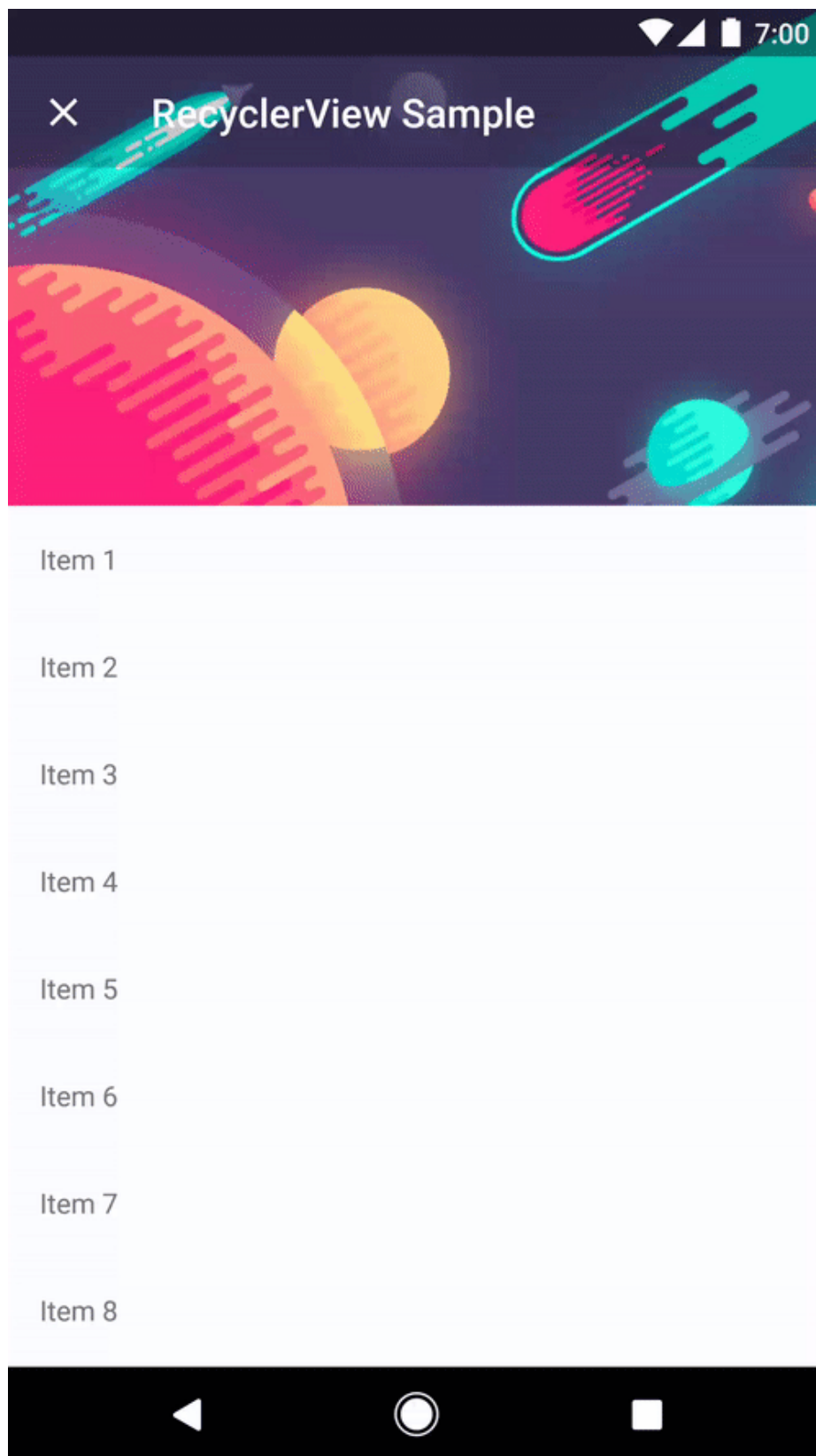
Android Drag Dismiss Activity

Another implementation of the drag-to-dismiss [Activity](#) pattern. This one is inspired by Nick Butcher's [Plaid](#) implementation.

The project has a simple API and is a pretty powerful and beautiful implementation of the pattern. This library is used in some of my apps (Talun for Twitter and Pulse SMS). It has been abstracted from Jacob Klinker and I's [article-android](#) library, which is an awesome readability style in-app web browser.

This library provides an elastic layout that can dismiss an [Activity](#) with a [RecyclerView](#) or a regular [Activity](#). Whenever the user swipes up or down, from the bottom or the top of the content, the [DragDismissActivity](#) will finish.

Including It In Your Project



This project was designed to be very easy to implement on top of your existing [Activities](#).

To include it in your project, add this to your module's `build.gradle` file:

```
1 dependencies {
2     ...
3     compile 'com.klinkerapps:drag-dismiss-activity:1.7.0'
4 }
```

Note: The normal way to implement the drag-dismiss functionality is by extending the two provided [Activities](#): [DragDismissActivity](#) and [DragDismissRecyclerViewActivity](#). If you would rather not do that, I have provided a delegate for each of these use-cases, that you can use: [DragDismissDelegate](#) and [DragDismissRecyclerViewDelegate](#). To see an example of the delegate's usage, check out the [AbstractDragDismissActivity](#).

Replacing an Activity

This library is meant to replace your [AppCompatActivity](#). I will set up all the drag-dismiss features for you, and wrap your content in a boilerplate UI that contains a [Toolbar](#) and a [ScrollView](#) for your content.

[DragDismissActivity](#) is easy to implement:

1. Within your [AndroidManifest.xml](#), on your [activity](#) elements replace whatever theme you are currently using with `android:theme="@style/DragDismissTheme`.
2. Instead of extending [AppCompatActivity](#), extend the [DragDismissActivity](#).
3. You won't want to override [Activity#onCreate](#). Instead, override [DragDismissActivity#onCreateContent](#). This method acts a bit like [Fragment#onCreateView](#):

```
1 @Override
2 protected View onCreateContent(LayoutInflater inflater, ViewGroup
3     parent, Bundle savedInstanceState) {
4     View v = inflater.inflate(R.layout.activity_scrollable, parent,
5         false);
6     // do your normal view setup and Activity#onCreate lifecycle
7     // actions here,
8     // instead of within the Activity#onCreate.
9     return v;
10 }
```

For a very simple example, please see [DismissableActivityNormalContent](#) Sample.

Replacing an Activity with a RecyclerView

If you are using an `Activity` that just has a `RecyclerView` in it, then this library can do all the heavy-lifting for you, all you will need to do is set it up with a `RecyclerView.Adapter` and a `LayoutManager`.

`DragDismissRecyclerViewActivity` is easy to implement as well:

1. Within your `AndroidManifest.xml`, on your `activity` elements replace whatever theme you are currently using with `android:theme="@style/DragDismissTheme`.
2. Instead of extending `AppCompatActivity`, extend the `DragDismissRecyclerViewActivity`.
3. You won't want to override `Activity#onCreate`. Instead, to set up your `RecyclerView`, override `DragDismissRecyclerViewActivity#setupRecyclerView`:

```
1 @Override
2 protected void setupRecyclerView(RecyclerView recyclerView) {
3     recyclerView.setLayoutManager(new LinearLayoutManager(this));
4     recyclerView.setAdapter(new SampleAdapter());
5
6     // do any other RecyclerView or Activity setup that needs to be
7     done
8 }
```

For a very simple example, please see `DismissableActivityRecyclerView` Sample.

Customizing the Boilerplate UI

After implementing the steps above, you will get your old `Activity`, with drag-to-dismiss functionality at the top and the bottom of the layout, great! You probably want to customize it further though.

I have provided a `DragDismissBundleBuilder` that allows you to add customization to the `DragDismissActivity` through extras on an `Intent`:

```
1 Intent dragDismissActivity = new Intent(this, MyDragDismissActivity.
2     class);
3 new DragDismissBundleBuilder(context)
4     .setTheme(DragDismissBundleBuilder.Theme.LIGHT) // LIGHT (default),
5     DARK, BLACK, DAY_NIGHT, SYSTEM_DEFAULT
6     .setPrimaryColorResource(R.color.colorPrimary) // defaults to a
7     semi-transparent black
8     .setSupportActionBar(true) // defaults to true
9     .setShouldScrollToolbar(true) // defaults to true
```

```
9      .setFullscreenOnTablets(false)           // defaults to false, tablets
      will have padding on each side
10     .setDragElasticity(DragDismissIntentBuilder.DragElasticity.NORMAL)
      // Larger elasticities will make it easier to dismiss.
11     .setDrawUnderStatusBar(false)           // defaults to false. Change to
      true if you don't want me to handle the content margin for the
      Activity. Does not apply to the RecyclerView Activities
12     .build(dragDismissActivity);
13
14 // do anything else that you want to set up the Intent
15 // dragDismissActivity.putBoolean("test_bool", true);
16
17 startActivity(dragDismissActivity);
```

Adding a ProgressBar

Have content that won't be loaded immediately? No problem.

While content is loading, you can call `DragDismissActivity#showProgressbar` to automatically show the progress indicator. When the content is done loading, just call `DragDismissActivity#hideProgressbar` to remove it.

For a simple example usage, check out the `DismissableActivityNormalContent` sample.

Contributing

Originally based off of [klinker41's](#) work for `article-android`.

Please fork this repository and contribute back using pull requests. Features can be requested using issues. All code, comments, and critiques are greatly appreciated.

Changelog

The full changelog for the library can be found [here](#).

License

```
1 Copyright 2019 Luke Klinker
2
3 Licensed under the Apache License, Version 2.0 (the "License");
4 you may not use this file except in compliance with the License.
5 You may obtain a copy of the License at
6
```

```
7   http://www.apache.org/licenses/LICENSE-2.0
8
9   Unless required by applicable law or agreed to in writing, software
10  distributed under the License is distributed on an "AS IS" BASIS,
11  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied
12  .
13  See the License for the specific language governing permissions and
14  limitations under the License.
```