

Intro, Sharing, Search

Joe Rogers
Android 310

Welcome

Welcome to Android 310

Instructor: Joe Rogers

TA: Jerry Frost

Site: <https://catalyst.uw.edu/workspace/joeroger/49369/>

Assignments

There will be 2-3 assignments worth 60pts each. With possible bonus opportunities. The assignments will be multi-week with steps or phases each week.

No Late Work, so don't wait until last week to start to catch up...

Grades

This is a satisfactory/unsatisfactory course.

Roughly 60% (36pts) per assignment needed to pass.

May have a curve depending on how things go, but don't count on it.

Attendance

- Expect you to attend each week. Gives opportunity for questions.
- However, if work, job, illness, etc takes you out of town, or makes you unavailable, recordings will be available.

Questions

- Please ask questions. If the question is off topic or too involved I may defer to the end of class. If I don't get to it, ask in the discussion forum.
- Jerry will be monitoring for online questions and will get my attention and relay the question.

Goals

- Build a “weather” app using an existing api
 - Allows for learning to read/process network data
 - Look up user’s current location to find weather near them.
 - Look up other “cities” to allow user to monitor other locations.
 - Utilize a map (possibly unrelated to the weather app)
 - Techniques for loading data efficiently.

Additional Goals

- Utilizing sensors
- Administration, Accounts
- Webviews if have time
- The developer console and publishing

Sharing

Sharing between apps

- Something Android has always supported.
- Allows an app to hand off to an app that does it better vs doing everything itself.
 - Examples: Sending email, navigation, taking picture, showing a web page
- Relatively easy to setup and do.

Sending share via standard intent

- Essentially form an intent using ACTION_SEND.
- Set the type to be “text/plain” if sending text
- Use the EXTRA_TEXT to provide the text to share.
- Can alternately provide EXTRA_TEXT_HTML for an html formatted version of the text. This will be ignored if the receiver doesn't understand HTML.

Sharing files

- Must use a content provider and Uri to provide the file to another app.
- Use FileProvider class in the support library that does most of the work, if you don't want to roll your own.
- The provider should provide the proper mime type via getType().

Invoking the intent

- If you just want to share, possibly using the user's favorite handler for `ACTION_SEND`, then just call `startActivity()`.
- If you want to allow user to “always” choose which app, wrap intent with `Intent.createChooser()` and the OS will present matching apps to the user.

Using an ShareActionProvider

- Configure the provider in the “menu” of the activity or fragment managing the provider.
 - If using support lib, likely need to use the compat version of the action provider.
- Retrieve the provider, and update the intent.
- When user sees provider in action bar, the provider will show last picked, plus give user option to choose other apps.

Receiving share intents

- Allow your app to be the recipient of shared data if it makes sense. I.e if building an email client, makes sense to handle “email” requests.
- In the manifest, register an intent filter for the type of action to handle on your activity. If ACTION_SEND also register the mime types you can handle.
- In the activity, you will now need to handle the registered “action” and do something appropriate with the data.

Demo

- Demo
 - ShareActionProvider via Content Detail Fragment
 - Receiving a share via Top Level Activity and the Receive Share Fragment

Search

Search

- Allow the user to search your app for “content”. Can be done with both a database, or a network call.
- Done via the SearchView (primarily) in the action bar or toolbar.
- Easy, but can be tricky due to number of steps involved.

Search Activity

- Either create or designate an activity to be the “target” of the search.
 - Not necessarily to handle the search in 3.0+, but still needed to just make work.
 - Essentially needs to handle ACTION_SEARCH.
- If using default suggestions, this activity will receive the search intent.

Searchable XML

- Create a searchable.xml file
 - Indicates basic data/configuration for your search.
 - Add voice search, register with OS, etc
 - Also used to indicate content provider for suggestion drop down.

Configure Search View

- Add SearchView as an ActionView to the menu of fragment or activity visualize the search.
 - Again if using support library, you should use the support version of the SearchView.
 - Doesn't have to be the “target” of the search, but if a different activity, you will need to create a ComponentName to reference the search activity.

Configure SearchView cont.

- In onCreateOptionsMenu, get reference to SearchView and provide reference to the searchable info via the search activity.
- You will need the SearchManager to retrieve this data.
- Optionally, set up a listener for query requests if handling some or all manually.

ContentProvider (optional)

- If using suggestions managed by the SearchView, add additional URI match patterns to the content provider.
- Update the query() method to handle these new search URIs.
 - This may involve aliasing columns to match the expected search columns to fill adapter and intent.
 - Use LIKE (standard) or MATCH (FTS) to find data.

Database (optional)

- If have several columns of text to search or a large dataset, it is more efficient to use an FTS table
- Update the table creation statements to use FTS. Only able to be done at creation time.
- Again use the MATCH keyword to search a FTS table.

Flavors of search

- If use the built in suggestion capability, as you type letters you will see a drop down indicating the match values.
- The query is done automatically in a separate thread, so if a content provider reading the database/web will work without any additional threads.

Manually handling query

- If already showing a “list” maybe better to just “filter” the list based on search vs using a drop down.
- Implement the `OnQueryTextListener()` and override `onQueryTextChange()` to restart your loader with an updated selection.

Demo

Demo

- Show steps to configure
- Show “dropdown” version and handling
- Show updating list version.

Resources

Resources Sharing

- Common Intents
- Map/Navigation Intents
- Sharing Simple Data
- Sharing Files

Search

- Search (focus on SearchView instructions)
- SearchView (includes customization options)
- SuggestionTable mapping
- FTS tables