
CS 1998 Project 4

Due Sunday 4/19, 11:59 PM

Let's recreate one of AppDev's apps!

The first Android app that AppDev created and shipped is Eatery! After watching the video lecture for this week, you'll realize that, at its core, Eatery is a simple app to build. Even the JSON file that you'll be parsing is the same file that powers Eatery's backend. Some of the details in the file can leave you scratching your head (like how menus are represented), but we tried our best to guide you through the data. Best of luck!

Quest Objectives

You are tasked with building a simple version of Eatery. Basic requirements are as follows, but you are free (and encouraged) to experiment with additional features!

1. Parse the attached JSON file using the method shown in the demo or any third-party libraries. To meet the basic requirement, you must retrieve information about at least 3 dining halls (Becker, North Star, and one of your choice) and 5 cafes (non-dining halls). For each eatery, you must retrieve the fields:
 - *"name"*
 - *"about"* or *"aboutShort"*
 - *"campusArea"* > *"descrshort"*
 - *"location"*
 - *"diningItems"* for cafes and *"operatingHours"* > *"event"* > *"menu"* for dining halls
 - Note: you'll notice that the menu for dining halls is highly nested. You only need to parse the *"items"* field of the **dinner** menu for a single day.
2. Create models for the parsed information. The only requirement for this is that your RecyclerView must take in an ArrayList of the model that you created, not a primitive Java type. We recommend that you create a parent class for all eateries, and two child classes, one for dining halls and one for cafes.
3. Add a RecyclerView to the main activity with your list of eateries. Each cell in the list must contain an image, eatery's name, and campus area. Feel free to add any other details.

4. Add two filter buttons to the top of the activity for dining halls and cafes. Clicking on the filter buttons should filter the list accordingly.
5. Add functionality to the RecyclerView so that clicking on an eatery opens a new activity (we'll call this "menu activity"). This activity needs to contain an image, eatery's name, location, and menu:
 - For cafes, display the list of "*diningItems*" in a ListView.
 - For dining halls, display the list of "*item*" in a ListView. If the menu is missing, display the text "No menu information available"

Once you've finished, **please read the [website](#) for how to submit your project**. We've written up a set of important guidelines on the format in which to submit your project.

Good luck, and have fun!

Side Quests

New optional side quests unlocked! Once again, these are entirely **optional**.

- Payment Methods (limited time event)
 - Parse the "payMethods" field in the JSON file and display icons for each payment method on both activities.
- Search Bar (limited time event)
 - Add a search bar to the main activity. The list of eateries should change on all inputs of the search bar. For example, if you search "house", the search result should show "Becker House Dining" and "Cook House Dining" (assuming that your list contains these two eateries).
 - The search results should be displayed in a RecyclerView (can reuse same one from #3) and clicking on a cell should open up the menu activity with all the requirements from #5.
- A Balanced Meal Plan (limited time event)
 - Parse the "category" and "item" fields for every meal represented on the JSON (for Becker and North Star) for **one day** of your choice.
 - Reformat the menu to display both the categories and items.
 - Add filter buttons on the menu activity to represent all the meal types. The menu should change accordingly. You'll notice that Cornell sorts meals into five categories: breakfast, brunch, lunch, lite lunch, dinner.
 - It's up to you to decide what to do with the "brunch" and "lite lunch" information (if it shows up)

- A Younger, More Attractive Language (permanently unlocked)
 - Complete this assignment in Kotlin: the future of Android Development, a more flexible and elegant language, and something we simply didn't have enough time to cover!

References

For this assignment, you may need to figure out a few things on your own. Some classes that may be helpful for this assignment include:

For basic functionality:

- ListView
- RecyclerView
- JsonObject/JsonArray
- Intent
- Bundle

For challenge problems:

- Kotlin
- SearchView

Feedback Form

After you finish this assignment, don't forget to fill out the weekly feedback form at <https://forms.gle/JEmkSzEAKL3wBREj8>.

Academic Integrity

All University-standard Academic Integrity guidelines should be followed. This includes proper attribution of any resources found online, including anything that may be open-sourced by AppDev. The University guidelines for Academic Integrity can be found [here](#).