1 Assignment

Summary: Write code that parses the tvshow JSON data and then write a collection of utility functions that allow you to filter and summarize the data in various ways.

Fork the starting git repository using the URL below. You'll note that the provided repository doesn't include files for the code that you are supposed to write. Instead, you should create those files and name them appropriately. We do provide a 'data' directory and a Data class that provides some utilities for loading those JSON files into your program.

`https://classroom.github.com/a/7tDKR9se`

Step 0: Getting the Data

The JSON data for this assignment can be found at the following website.

`https://github.com/jdorfman/awesome-json-datasets#tv-shows`

This git repo has some links to JSON files for some shows. It also shows how the links are constructed and points options to others. The data all comes from the following website.

`http://www.tvmaze.com/api`

You are welcome to work with any shows that interest you. You should experiment with more than just one while doing this assignment.

Step 1: Parsing the JSON: Use the Gson library, as demonstrated in lecture on Thursday, to parse the provided JSON from tvmaze. We recommend that you first focus on parsing individual tv show episode JSON objects, by writing tests where you parse them from Strings that are hard-coded in your test class (i.e., cut-and-paste one episode from one of the JSON files). Once you can successfully parse a single episode's worth of data, try parsing a small JSON array of episodes (e.g. again cut-and-paste a small collection of episodes' JSON - this should be wrapped in square brackets) and then finally try parsing a whole show object

Step 2: Loading JSON from Files:

Note: Java's libraries for reading and writing files are not so straight-forward, so we've provided you some utilities in a Data class. Please read through this code so that you understand what it is doing.

Use the provided Data utilities to create your own utility function(s) that load the tvshow JSON files and parse the JSON they contain to produce Java objects.

Step 3: Filtering methods Write a collection of utility methods that each take a collection of parsed objects and return a subset of the original collection filtered by a given criterion. Your methods should be designed in such a way that you can take the output of one method and pass it in as the input of another method, in any order. You should write at least four methods. Some examples that you may use are:

1. returns all of the episodes from a given season
2. returns all of the episodes with an episode name contains a given String
3. returns all of the episodes with an airdate in a given year

The other method can be of your choice. Have fun with them. Do something interesting!

Step 4: Aggregation methods Write a collection of methods for aggregating data across a collection of episodes (that are potentially a filtered subset of all of the episodes in the tv show, using the above methods). You should write at least four methods examples could include the following:

1. returns the total number of episodes in a given collection of episodes
2. returns the average length of an episode in a given collection of episodes
3. return the maximum length of any episode in a given collection of episodes

Process: In the above, we've provided you 4 progressive steps for breaking down this assignment. Each of these is a small piece of functionality that you can (1) write tests for, (2) implement and get working, and (3) commit and push to github. We expect to see these commits.

Design and Style: As always, use the best design and coding style that you are familiar with. In particular, for this assignment, continue to focus on using good names (you have a lot of freedom in naming in this assignment) and code layout (as directed by Chapter 4 of the text book and section 4 of the Google Java style guide).