Exercise 4: Documenting Query Parameters

For this exercise, you'll get a little practice documenting query parameters. Let's say that you are working for a company called BugSquisher that makes bug tracking software. They have an API that lets you create and manage bugs for quality assurance.

Information from developer team

In this exercise, you'll write up the query parameter table for the request that returns a list of bugs. You can choose which bugs will be returned by using query parameters. Here is the information on the query parameters from the developer team:

- startDate. The start date for when the bug was created. Format: YYYY-MM-DD.
- endDate. The end date for when the bug was created. Format: YYYY-MM-DD.
- priority. The priority value. (How important is it.) Integer of 1 to 4, where 1 is the highest priority.
- severity. The severity value. (How big the impact is.) Integer of 1 to 4, where 1 is the most severe.
- status. The status of the bug. Valid values: open, closed, duplicate, notabug
- start. For pagination, the starting index. Zero is the first bug on the list.
- total. For pagination, the total number of bugs to return

None of these query parameters are required. The default is all.

Create a query parameter table

Follow these steps:

1. Open up a word processor like Microsoft Word, Google Drive Docs, or Apple Pages.
2. Create a table with 5 columns and 8 rows.
3. On the header row, add these for the columns: Parameter, Description, Type, Required, and Notes.
4. Now fill in the rows with the information from above.

The challenge: the developer team gave you the very vague description of "The default is all". Developer teams are often not that skilled at communicating, and your job is to find a better way to say things. See if you can figure out an appropriate phrase for each of the default values. For example, for startDate, you could say "The default is the earliest recorded bug." For priority, you could say, "The default is all priorities." The total is especially tricky: think about what the default value should be that puts no limits on how many bugs you would get returned after the start index.

Create a sample request

Create a sample URL. It will start with:

GET https://api.bugsquisher.com/bug

Add query parameters for each of the parameters in the table and give them reasonable values.
Now, just above the sample request, add a one sentence description of what bugs it returns.
You might want to choose values that make this one sentence description as simple as possible.

You can see the documentation I created at: http://sdkbridge.com/ud/Exercise4Answers.pdf