Lab 03-04 Grade 98/100

horizontal line

# Issue Tracker - Phase 3

This lab is a continuation of the Issue Tracker project that we started previously. In this assignment we will implement search endpoints for **users** and **bugs.**

# Use your existing GitHub Repository

Use your **awd1111-issue-tracker** repository that you created earlier.

**Do not create a new repository!**

# 

***See exercises on the following pages…***

# 

# Exercise 1. User List Endpoint (50 pts) –2 pts

**GET /api/user/list**

* Upgrade this route to provide additional search functionality.
* The following data **may** be provided via **query parameters,** and read from **req.query:**
  + **keywords**
    1. This parameter is optional, do nothing if it is falsy.
    2. If it is not falsy, then use the [**$text**](https://docs.mongodb.com/manual/text-search/) operator to search for users containing the provided string.
    3. Add a [**wildcard text index**](https://docs.mongodb.com/manual/core/index-text/) using the MongoDB shell to support these queries.
  + **role**
    1. This parameter is optional, do nothing if it is falsy.
    2. If it is not falsy, then only show users with the provided role.
  + **maxAge** 
    1. Measured in days since the user was created.
    2. This parameter is optional, do nothing if it is falsy.
    3. If it is not falsy, then use the **$gte** operator to show only users that were created after a calculated date.
  + **minAge**
    1. Measured in days since the user was created.
    2. This parameter is optional, do nothing if it is falsy.
    3. If it is not falsy, then use the **$lt** operator to show only users that were created before a calculated date.
  + **sortBy**
    1. Sort the results based on this selection.
    2. Allowed Options: **"givenName", "familyName", "role", "newest", "oldest"**
    3. Default to sorting by the **"givenName"** mode as described below.
    4. These options must sort the results as described in the list below:
       - **givenName:** given name ascending, family name ascending, created date ascending
       - **familyName:** family name ascending, given name ascending, created date ascending
       - **role:** role ascending, given name ascending, family name ascending, created date ascending
       - **newest:** created date descending
       - **oldest:** created date ascending -2pts First record should have the oldest creation date. This does not seem to be the case.
    5. Implement this logic with either a **switch** statement or **if** statements.
    6. *The extra fields listed above are to make the sort order stable.*
  + **pageSize**
    1. Specifies the number of items per page to be displayed.
    2. Default to **5,** if not provided.
    3. Use **.limit()** or **$limit** to limit the number of results returned to the page size.
  + **pageNumber** (default: 1, which page of results, first page is page 1)
    1. Specifies which page of results that the client wants.
    2. The first page is page **1.**
    3. Default to **1,** if not provided.
    4. Use **.skip()** or **$skip** to skip over the required number of items. *(Remember to consider the pageSize.)*
* **All of the above query parameters are optional. All combinations of these query parameters must be supported!**
* **If you haven't yet, you will need to add the following indexes to support these queries:**
  + [**wildcard text index**](https://docs.mongodb.com/manual/core/index-text/)
  + created date: 1
  + given name: 1, family name: 1, created date: 1
  + family name: 1, given name: 1, created date: 1
  + role: 1, given name: 1, family name: 1, created date: 1

***More exercises on the following pages…***

# 

# Exercise 2. Bug List Endpoint (50 pts)

**GET /api/bug/list**

* Upgrade this route to provide additional search functionality.
* The following data **may** be provided via **query parameters,** and read from **req.query:**
  + **keywords**
    1. This parameter is optional, do nothing if it is falsy.
    2. If it is not falsy, then use the [**$text**](https://docs.mongodb.com/manual/text-search/) operator to search for users containing the provided string.
    3. Add a [**wildcard text index**](https://docs.mongodb.com/manual/core/index-text/) using the MongoDB shell to support these queries.
  + **classification**
    1. This parameter is optional, do nothing if it is falsy.
    2. If it is not falsy, then only show bugs with the provided classification.
  + **maxAge**
    1. Measured in days since the bug was created.
    2. This parameter is optional, do nothing if it is falsy.
    3. If it is not falsy, then use the **$gte** operator to show only bugs that were created after a calculated date.
  + **minAge**
    1. Measured in days since the bug was created.
    2. This parameter is optional, do nothing if it is falsy.
    3. If it is not falsy, then use the **$lt** operator to show only bugs that were created before a calculated date.
  + **closed**
    1. Default to show all bugs
    2. If closed is **true,** then **show** all closed bugs.
    3. If closed is **false,** then **show** all open bugs.
  + **sortBy**
    1. Sort the results based on this selection.
    2. Allowed Options: **"newest", "oldest", "title", "classification", "assignedTo", "createdBy"**
    3. Default to sorting by the **"newest"** mode as described below.
    4. These options must sort the results as described in the list below:
       - **newest:** created date descending
       - **oldest:** created date ascending
       - **title:** title ascending, created date descending
       - **classification:** classification ascending, created date descending
       - **assignedTo:** assigned to name ascending, created date descending
       - **createdBy:** created by name, created date descending
    5. Implement this logic with either a **switch** statement or **if** statements.
    6. *The extra fields listed above are to make the sort order stable.*
  + **pageSize**
    1. Specifies the number of items per page to be displayed.
    2. Default to **5,** if not provided.
    3. Use **.limit()** or **$limit** to limit the number of results returned to the page size.
  + **pageNumber** (default: 1, which page of results, first page is page 1)
    1. Specifies which page of results that the client wants.
    2. The first page is page **1.**
    3. Default to **1,** if not provided.
    4. Use **.skip()** or **$skip** to skip over the required number of items. *(Remember to consider the pageSize.)*
* **All of the above query parameters are optional. All combinations of these query parameters must be supported!**
* **If you haven't yet, you will need to add the following indexes to support these queries:**
  + [**wildcard text index**](https://docs.mongodb.com/manual/core/index-text/)
  + created date: -1
  + title: 1, created date: -1
  + classification: 1, created date: -1
  + assigned to name: 1, created date: -1
  + created by name: 1, created date: -1

***Fin.***