**Robust server structure: Major error types & handling assignment**

**Instructions**

Your task is to extend the given API server, which stores "Notes", to enhance the error handling functionality. It has the following existing route handlers:

* GET /notes returns an array of notes
* POST /notes creates a new notes
* GET /notes/:noteId returns a single note by id

**Existing Files**

You should only need to edit code inside src/app.js.

Use the existing data files located in src/data for the responses. Feel free to add or remove data from the files as necessary, but keep the same shape of the data.

**Tasks**

1. Modify the existing error handler so that it:
   * returns a 500 status code by default, or the status code stored in the status property of the error object
   * returns a JSON response formatted like { error: "<error-message-here>" }
2. Move the validation logic in the GET /notes/:noteId route handler into its own middleware function. In the event of an error, your validation middleware *should* call next() with an error object (i.e., { status: ..., message: ...}) as the argument so that it can be passed along to the error handler.

**Hint**: if the text property is missing from the body of the POST, return the following error message: "A 'text' property is required."

1. Move the validation logic in the POST /notes route handler into its own middleware function. In the event of an error, your validation middleware *should* call next() with an error object (i.e., { status: ..., message: ...}) as the argument so that it can be passed along to the error handler.