This challenge utilizes some of the simple concepts that Globus uses on a regular basis. Four test files have been provided that can be loaded via Ajax to test your final code.  The JSON test files should not be changed.

The goal is to allow candidates to show us their skills.  We anticipate that some candidates may not be able to complete all the requirements, so we have prioritized them below.  Please proceed to implement the challenge in the order of these requirements.  Code clarity, organization and accuracy will be part of the evaluation.

We use Ember on our projects, but feel free to use any libraries or frameworks you like.  If building your project into ready-to-run JavaScript/CSS requires transpiling (e.g. Babel, Sass), please include a mechanism to do so (e.g. Grunt, Gulp, Broccoli, Make).

If there are any questions please send them to shifflett@globus.org and jaswilli@globus.org.

Deliverable: An e-mailed archive or a clonable git repository (e.g. on GitHub), with a README

Product Explanation:
This page is going to act as status tracker loading data.  The data is minimal, so some interpretation is going to be required.

Status can be translated as such:
INACTIVE: No start date
ERROR: Has a start date and an end date, but total != processed
SUCCESS: Has a start date and an end date, and total == processed
IN PROGRESS: Has a start date and no end date, and total != processed

Requirements:
A. The final rendering of the data should look like the design PNGs included in this challenge
Table Columns:
A1: Status: This is an interpreted column determined by the four statuses as follows:
- INACTIVE:Display the words "not started"
- SUCCESS: Display the words "Completed: <end_date>" (where <end_date> is replaced by the data for end date.)
- ERROR: Display the words "Halted: <end_date>" (where <end_date> is replaced by the data for end date.)
- IN PROGRESS: Display the words "Time Remaining: <remaining>" (where <remaining> is replaced by the data for remaining time.)
A2: Progress: This is a combination of two fields.  <processed>/<total>.

A3: User: This is a hyperlink that displays the user's full name where the link opens into a mail client.

A4: Request Date: This is just the request date.

A5: Nice Status: This appears below and across the rows above it. It is the status from the returned data object.

B. Successful tasks should have their "nice_status" highlight the word "success"

C. Failed tasks should have their "nice_status" highlight the word "fail" and "error"

D. Progress should be shown in byte notation rather than just raw bytes. Thus 1231245566 bytes translates to 1.15 GB.

E. Dates: All dates should be translated into the user's timezone. Start and End dates do not always match timezone, because start date refers to the origin server time and end date refers to the destination server time.

F. Remaining Time should show as hh:mm:ss for any remaining time less than 2 days. If remaining time is above 2 days, show it as "# days"

G. Some error handling should exist.

G1. test3.json cannot be parsed by JSON.parse(), this is an error.

G2. test4.json has a number of errors.

H. Sort Order: Inactive tasks, Tasks in progress. Secondary sort: Completion Date.

I. Keep the same case when highlighting words in Successful and Failed tasks.