BE - Recruitment Task (II)

Below you will find a description and a download for the development task to be provided to the candidate.

The candidate is expected to complete the task on his public personal repository such as Gitlab, and provide DemandScience, read access to it so we may review her/his work.

Backend Developer task

Design and implement an application (in Java/GoLang), that exposes the following endpoints. The uploaded files can be stored in the file system and in-memory datatype can be used to persist the historical data.

1. `/API/upload` - will be used to upload a `text/CSV` file and return a unique ID in the following format.
   On upload, the application will parse through the file, and add a new flag(`true/false`) column at the very end of each non-empty row (ignoring the header).

   `true` - indicates the row has at least one field that is a valid email address (using a simple regex or a library)
   `false` - indicates the row doesn't have any email address.

   The updated CSV file is to be stored and can be retrieved later via the second endpoint.

   Expected responses:
   1. `OK`, response:
   2. `{`
   3. `  "id" : "a225eb00-0907-4273-92ca-5faadeefae5f"`
      `}`

   4. `BAD` response
   5. `{`
   6. `  "error": "{failure-reason}"`
      `}`

2. `/API/download/{id}`

   If the request is made with a valid and completed job `{id}`, a `200` response is expected with data as a blob. In the case of jobs in progress, a `423` response is expected.

   In the case of an invalid ID, a `400` response is expected.