

Sauce Ruby Assessment

This test is designed to allow for the creative demonstration of skills for a software engineer in the context of Sauce - a company with data at its core and the need to frequently retrieve, process and transport data. This test is designed to be a simplified, but representative example of one of many challenges you can expect to face on a daily basis when joining the fine ensemble here at Sauce.

The challenge

You have been provided with a JSON file of data of 1,000 users.
It is an aggregated view of publicly available data from Instagram.

The goal is to read in the data, massage it according to the requirements, and save the massaged data to JSON files of no more than 100 entries each.

You are welcome to use any libraries in your implementation.

The JSON file

The provided 1000-users.json is in the JSON Array format and consists of 1,000 entries.
Each entry follows the schema as demonstrated below:

```
{
  "_id": { "$oid": "5aa104e0f20e84e6104cecd6" },
  "id": "868192463",
  "type": "add",
  "username": "misterfox_cafe",
  "bio": "Cafe 🦊 Bar 🍺 Events Ringwood Golf Course 🌿",
  "followed_by": 1213,
  "mentions": [
    "huntgram",
    "misterfox_cafe",
    "natalitza_k",
    "ellerycohen",
    "bmtone",
    "littlebertha",
    "kisasun",
    "balancebymonica",
    "melbournebreakfastdiary",
    "ringwoodgolf"
  ],
  "hashtags": [
```

```
        "events",
        "notyourtypicalgolfcafe",
        "ringwood",
        "golfcourse",
        "golf",
        "brioche",
        "freerange",
        "takeaway",
        "bar",
        "organic"
    ]
}
```

Requirements

- The output must be in multiple files of max 100 entries
- Data must be massaged to the following rules:
 - Empty arrays must be removed
 - The `_id` field (note the underscore) must be removed, leaving just the id
 - Anything but alpha-numeric characters and hashtags must be removed from the bio field
 - Provide clear usage instructions to allow us to easily verify the solution
- Optional Bonus Points:
 - write tests to demonstrate the workings of the data massaging
 - Be able to support very large JSON files by streaming the file in/out
 - generate a report upon completion of the process, which can include data such as: process duration, average followers/followers, most followed users, average number of mentions etc