

## Backend - Technical Challenge

### Overview

A company has a business process that requires processing of text files from external customers.

Some information about the text file:

- Each line of the text file is an individual record and can be processed independently of the other records in the text file.
- The size of the file can range from a few KB to a few GB.
- Here is the format of each line:

```
[Date in YYYY-MM-DDThh:mm:ssZ Format][space][Email Address][space][Session Id in GUID format]
```

For example:

```
2020-12-04T11:14:23Z jane.doe@email.com 2f31eb2c-a735-4c91-a122-b3851bc87355
```

- The lines are ordered by the Date field in ascending order.
- Sample files can be found here: <https://bitbucket.org/ROKT/backend-technical-test>

### How to submit your solution

In an effort to prevent rejected or bounced emails due to zip files, we have a Google form for you to upload your solution.

Google form URL: <https://forms.gle/L3w5czKvgvn7M7ow9>

Complete the form with your email address and a single zip file containing your project. Please **do not** include any build artifacts or sensitive information, like your API key, in the zip file. **Treat the zip file like you would a public GitHub repo.**

**Please make this a single zip file named: `\${yourName}-\${roleAppliedFor}.zip`**

### Requirements

Please write an application to parse the text files above.

- Inputs:
  - Path to the file
  - From date time
  - To date time
- Output: Parsed entries within the date time range inclusively, in JSON format. Example:

```
[  
  {  
    "eventTime": "2000-01-01T03:05:58Z",  
    "email": "test123@test.com",  
    "sessionId": "97994694-ea5c-4da7-a4bb-d6423321ccd0"  
  },  
  {  
    "eventTime": "2000-01-01T04:05:58Z",  
    "email": "test456@test.com",  
    "sessionId": "97994694-ea5c-4da7-a4bb-d6423321ccd1"  
  }  
]
```

## Notes and Hints

Some notes about the challenge:

- Please write the application in a language that you are most comfortable with. It can be either a console app or a Web API.
- Please feel free to make any assumptions that are not specified above, just be sure to tell us the assumptions you made when submitting your application.
- The submitted code should be production-grade code. Your submission will be judged on code quality as well as correctness.
  - Should include testing code.
  - Should be robust enough to handle failure cases.
- Package the final code in a zip file. Please do not include any binaries files in the zip, e.g. libraries, compiled code...
- Please include a file named *“readme.md”* with instructions on how to compile and run the application as well as any assumptions you’ve made.
- Making the application portable would be a plus, i.e. dockerizing the solution.

A couple of hints on how to proceed (those are just hints, please go as far as you like):

- Make it work first!
- Think about edge cases.
- Think about how to write the application with constant memory consumption regardless of file size.
- Think about how to utilise multiple CPU cores.
- Run through your application with the provided test files.