**Run Book for Data Engineer Exam**

**Version Control and Approval**

**Version History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Reviewer** | **Version Summary** |
| 0 | 02-Mar-2019 | Niranjan Reddy Rachamala |  | Initial version |

**Contents**

[1. Introduction 2](#_Toc2449963)

[2. Execution steps 2](#_Toc2449964)

[2.1 Generate JSON for each record of the given input csv file 2](#_Toc2449965)

[2.2 Generate metrics for given input csv file 2](#_Toc2449966)

[2.3 Generate Summary report 2](#_Toc2449967)

# Introduction

This document gives the execution steps to provide insight into the Enron event history data provided for data engineer test.

The Enron event history data in included (.csv, adapted from the widely-used publicly available data set).

Two CSV files will be provided in the source directory, the smaller of the two files will be used for testing purposes. Assume a clean dataset, but feel free to write code that would handle bad data.

The columns contain:

\* \*\*time\*\* - time is Unix time (in milliseconds)

\* \*\*message identifier\*\*

\* \*\*sender\*\*

\* \*\*recipients\*\* - pipe-separated list of email recipients

\* \*\*topic\*\* - always empty

\* \*\*mode\*\* - always "email"

# Execution steps

## 2.1 Generate JSON for each record of the given input csv file

Execute the python program to generate the json files for each record of the given csv file.

python parse\_daily\_enron\_file.py source/<data files in csv format>

**example**:

python parse\_daily\_enron\_file.py source/enron-event-history-20180202.csv

## 2.2 Generate metrics for given input csv file

Execute the below script to generate the output metrics for sender and receiver.

sh get\_metrics.sh source/<data files in csv format>

**example**:

sh get\_metrics.sh source/enron-event-history-20180202.csv

## 2.3 Generate Summary report

Execute the below script to generate the summary report having the number of messages sent and received by each person.

python Summary\_Report.py