**Steps:**

* For end users a zip of format KoreChatAnalyzer\_<timestamp>.zip will be shared.
* copy the zip file to the server into a location (For eg: /data directory)
* unzip the zip file using the following command:  
   command: unzip KoreChatAnalyzer\_2021-10-12T17\:31\:24.zip -d KoreChatAnalyzer
* it will create a directory KoreChatAnalyzer in /data folder
* go into the folder  
   command: cd KoreChatAnalyzer
* Prerequisite:
  + in createindex.js two mongo commands are listed, please run them on mongo following the below steps.
  + log into mongo terminal
  + select koredbt001 database  
     command: use koredbt001
  + now run the commands individually  
     command: db.transitionrecords.createIndexes([{"timestamp":1, "metaInfo.channel": 1}],{"name":"timestamp\_1\_metaInfo.channel\_1","background":true})  
    command: db.transitionrecords.createIndexes([{"metaInfo.identity":1,"timestamp":1}],{"name":"metaInfo.identity\_1\_timestamp\_1","background":true})
  + exit the mongo
* in current directory (/data/KoreChatAnalyzer) create a python 3.6 virtual environment  
   command: virtualenv venv --python=python3.6
* activate the virtual environment  
   command: source venv/bin/activate
* modify config.json with necessary constant parameters like MONGO\_URL which does not need to be passed as command line param every time. (config parameters are explained later in below section)
* modify rules.json based on your requirements.
* now run the main.py file to generate report  
   command: python main.py config=config.json start\_time=2021-10-12T00:00:00 end\_time=2021-10-12T12:00:00
* the report will be generated in data folder inside current directory by default with name start\_time-end\_time.xlsx

**Config:**

* config parameters can be either set in config.json file or can also be passed as command line arguments while running the program.
* the command line argument are passed as below:  
   Eg - python key1=value1 key2=value2 ...
* The config file name is to be passed as a command line argument.  
   Eg - python main.py config=/tmp/config.json
* Any other command line arguments passed along with 'config=' will override the config set in the config file.  
   Eg - if START\_TIME and END\_TIME are defined in config by default the program will consider that period to analyze the calls. However if needed to be tested for another time range we need not modify the config file each time and those parameters can be passed while running the main script.  
   command: python main.py config=config.json start\_time=2021-10-12T00:00:00 end\_time=2021-10-12T12:00:00
* it is better to keep constant parameters like mongo url in config.json file and other dynamic parameters like start\_time and end\_time as command line parameters.
* different configs supported:

| **config.json key** | **cmd line param key** | **Description** |
| --- | --- | --- |
| MONGO\_URL | mongo | Mongo url to connect with mongo db. |
| START\_TIME | start\_time | The datetime from which the chat must be analyzed.  Formats - ‘YYYY-mm-ddTHH:MM:SS’(2021-10-12T00:00:00), ‘YYYY-mm-ddTHH:MM:SSTZ(2021-10-12T00:00:00+0530)  (default - past hour starting -> can be set via duration config) |
| END\_TIME | end\_time | The datetime to which the chat must be analyzed.  Formats - ‘YYYY-mm-ddTHH:MM:SS’(2021-10-12T00:00:00), ‘YYYY-mm-ddTHH:MM:SSTZ(2021-10-12T00:00:00+0530)  (default - past hour ending) |
| MONGO\_TIME\_OUT | mongo\_time\_out | Time in ms for which the program must wait for mongo connection, |
| LOG\_FOLDER | log\_folder | Path to which logs to be written (by default log folder will be created in the current directory) |
| DATA\_FOLDER | data\_folder | Path to which the reports to be generated (by default data folder will be created in the directory) |
| RULES\_FILE\_PATH | rules\_file\_path | Path for rules.json file |
|  | config | Path for config.json file |
| CHANNEL | channel | To analyze chat for specific channels (by default all channels are considered. For IVR calls only analysis set the value to **ivrVoice**) |
| DURATION | duration | Time in minutes to be considered as upper boundary before start time (default 60 min) |
| END\_TIME\_CUTOFF | end\_time\_cutoff | Time in minutes to be considered as lower boundary before end time (default 10 min) |
| TRANSITION\_RECORDS\_DB | transition\_records\_db | DB name where transitionrecords collection is present |
| TRANSITION\_RECORDS\_COLLECTION | transition\_records\_collection | Collection name for transitionrecords |