

OctaPy's Back-End Technical Requirements Document

Author/s:	Sameh M. A. AbuRadi, Assistance: Celine Crisp
Document Version:	Version 1.27
Last Date Modified:	25 February '21
Last Reviewed Date:	

General Information

Purpose - *Purpose of the document*

This is a technical requirement document to outline the technical requirements of OctaPy's implementation. This is a list of product goals including their timeframes and feature categories.

Scope - *Scope of the document and project*

This is to scope the MVP version of OctaPy, its features, their implementation requirement and dependencies.

Acronyms and/or Definitions:

- MVP - Minimum Viable Product
- BETA - Second Version, Following The MVP Lunch
- AuX - Authentication Features/Functions
- OvX - Overview Features/Functions
- AnX - Analytics Features/Functions
- AcX - Activities Features/Functions
- SeX - Settings Features/Functions
- UMX - User Management Features/Functions
- IOX - API Features/Functions
- DBX - Database Tasks
- *Ricket* - *recordTicket*

Project Overview

Project Document Dependencies:

1. The Scope of All Scopes
 - a. Web UI Features
 - i. Overview
 - ii. Analytics
 - iii. Activities (including Notifications)
 - iv. Settings
 - v. Users Management
 - b. System Features
 - c. API Features
2. OctaPy WireFrames (Version 3.0)

Roles and Responsibilities:

- Marcel: General Manager & Product Owner
- Sameh: Software Development & Team Lead

Features and User Impact

Feature/Function List

Au1, POST /signUp
Au2, GET /verify/{verificationKey}
Au3, POST /signIn
Au4, POST /resetPwd
Au5, POST /apiReq
Ov1, GET /overview/views/analytics
Ov2, GET /overview/views/activities
Ov3, GET /overview/views/statistics
An1, GET /analytics/views/callTimeGraph/{filtersOpts}
An2A, GET /analytics/views/widgets/{filtersOpts}
An2B, GET /analytics/views/widget/{Id}/{filtersOpts}
An4, GET /analytics/views/actionsListByTicketNo/{filtersOpts}/{search}
An5, GET /analytics/views/actionsListByCleintId/{filtersOpts}/{search}
An6, GET /analytics/views/actionByTicketNo/{TicketNo}/{Ricket}
An7, GET /analytics/views/actionByCleintId/{CleintId}/{Ricket}
Ac1, GET /activities/views/logs
Ac2, GET /activities/views/multiBarGraphs
Ac3, GET /activities/views/pieCharts
Se1, GET /settings/profile
Se2, GET /settings/standards
Se3, GET /settings/constraints
Se4, GET /settings/getDevKeyList
Se5, POST /settings/modify
Se6, POST /settings/addDevKey
Se7, POST /settings/modifyDevKey
Se8, POST /settings/delDevKey
UM1, GET /userMgmt/getUsersList
UM2, GET /userMgmt/searchUsersList/{...}
UM3, GET /userMgmt/getUserProfile/{userId}
UM4, POST /userMgmt/modifyUserProfile
I01, POST /record/call
I02, POST /record/text
I03, GET /report/{Ricket}

The highlighted in green are the end points that fall within the API scope,

The expected timeframe for them is 31 working days as explained below:

7 days: Developers, Interactions, ActivitiesLog & WordSequence Documents
Design & Implementation

1 day : Se4, GET /settings/getDevKeyList
1 day : Se6, POST /settings/addDevKey & Se8, POST /settings/delDevKey
1 day : Se7, POST /settings/modifyDevKey
1 day : Au5, POST /apiReq
12 days: I01, POST /record/call
4 days: I02, POST /record/text
4 days: I03, GET /report/{Ricket}

Feature/Function & Requirement

POST /signUp

<- (Au1)

Expected Parameters:

- *companyName, (String)*
- *email, (String, to-be-verified)*
- *phone, (String, to-be-verified)*
- *password, (String, to-be-verified)*

Expected Response(s):

- *if successful, STATUS: OK*
- *otherwise:*
 - *if email isn't valid, STATUS: KO, errMsg: "email isn't valid"*
 - *if phone isn't valid, STATUS: KO, errMsg: "phone isn't valid"*
 - *if password isn't valid, STATUS: KO,*
errMsg: "password is too short, minimum length is 12 characters" OR
errMsg: "password is too long, maximum length is 128 characters" OR
errMsg: "pwd must contain at least one: 1-9, A-Z, a-z character" OR
errMsg: "system failure, please try again"

Estimated development time: a day of business

Dependencies: DB1, DB2

Requirements:

- *A password encryption function,*
- *an insert function,*
- *an account verification emailing function*

GET /verify/{verificationKey}

<- (Au2)

Expected Response(s):

- *redirecting to the signIn form whether successful or not*

Estimated development time: half a day of business

Dependencies: DB1, DB2

Requirements:

- *A modify function*

POST /signIn

<- (Au3)

Expected Parameters:

- *email, (String, to-be-verified)*
- *password, (String, to-be-verified)*

Expected Response(s):

- *if successful, STATUS: OK, accessToken: (a 32 Chars Long String)*
- *otherwise:*
 - *STATUS: KO,*
errMsg: "credentials didn't match" OR
errMsg: "system failure, please try again"

Estimated development time: a day of business

Dependencies: DB1, DB2, Au1

Requirements:

- A password decryption function,
- a select function

POST /resetPwd

<- (Au4)

Expected Parameters:

- email, (String, to-be-verified)

Expected Response(s):

- if successful, STATUS: OK
- otherwise:
 - STATUS: KO,
 - errMsg: "system failure, please try again"

Estimated development time: a day of business

Dependencies: DB1, DB2, Au1, Au3

Requirements:

- A password decryption function,
- a password encryption function,
- a password modification function,
- a password modification emailing function

POST /apiReq

<- (Au5)

Expected Parameters:

- email, (String, to-be-verified)
- userName, (String, to-be-verified)
- devKey, (String, to-be-verified)

Expected Response(s):

- if successful, STATUS: OK, devToken: (a 32 Chars Long String)
- otherwise:
 - STATUS: KO,
 - errMsg: "credentials didn't match" OR
 - errMsg: "system failure, please try again"

Estimated development time: a day of business

Dependencies: DB1, DB2, Au1, Au3

Requirements:

- A select function,
- a modification function

GET /overview/views/analytics

<- (Ov1)

Expected Parameters:

- accessToken: (a 32 Chars Long String)*
- filtersOpts: {} // to be discovered*

Expected Response(s):

- if successful;*
 - STATUS: OK,*
 - DATA: {}, AVG & SUM Widgets*
 - AVG number of calls to resolve an issue*
 - SUM of time "on a call" <- format: HH:mm*
 - AVG number of insignificant words per minute*
 - SUM of time of insignificant words/phrases*
 - AVG number of seconds of silence per minute*
 - SUM of time of silent*
 - AVG number of interruptions per minute*
 - SUM of interruption times*
- otherwise:*
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"*
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: two days of business

Dependencies: Auth, DB1, DB2, DB4, An2, Fltr

Requirements:

- A select function*

GET /overview/views/activities

<- (Ov2)

Expected Parameters:

- accessToken: (a 32 Chars Long String)*
- filtersOpts: {} // to be discovered*

Expected Response(s):

- if successful;*
 - STATUS: OK,*
 - DATA: {}, Logs List of Widgets, each contains:*
 - Endpoint, format: VERB /endpoint*
 - Timestamp, format: DD/MM/YYYY, HH:mm:ss.MS*
 - Status, OK / KO*
- otherwise:*
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"*
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: a day of business

Dependencies: Auth, DB1, DB2, DB5, Ac1, Fltr

Requirements:

- A select function*

GET /overview/views/statistics

<- (Ov3)

Expected Parameters:

- accessToken: (a 32 Chars Long String)
- filtersOpts: {} // to be discovered

Expected Response(s):

- if successful;
 - STATUS: OK,
 - DATA: {}, List of Chart Data
 - Resolved Tickets of ALL Tickets (percentage)
 - On-Hold Time of ALL Call Time in Mins (percentage)
 - Audio Calls of ALL Actions (percentage)
 - Successful Requests to Unsuccessful Requests Ratio
 - Incoming Requests to Outgoing Ones (Cron Jobs) Ratio
- otherwise:
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: two days of business

Dependencies: Auth, DB1, DB2, DB4, DB5, An3, Ac3, Fltr

Requirements:

- Multiple select functions

GET /analytics/views/callTimeGraph/{filtersOpts}

<- (An1)

Expected Parameters:

- accessToken: (a 32 Chars Long String)

Expected Response(s):

- if successful;
 - STATUS: OK,
 - DATA: {}, Lists for Diagram Data
 - AVG time of calls along timeframe <- Array of Floats
 - MAX time of calls along timeframe <- Array of Floats
 - MIN time of calls along timeframe <- Array of Floats
 - MAX time of calls' threshold <- Float
 - MIN time of calls' threshold <- Float
- otherwise:
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: two days of business

Dependencies: Auth, DB1, DB2, DB4, Fltr

Requirements:

- A select function,
- data functions (w/ Pandas)

GET /analytics/views/widgets/{filtersOpts} <- (An2A)

Expected Parameters:

- accessToken: (a 32 Chars Long String)

Expected Response(s):

- if successful;
 - STATUS: OK,
 - DATA: {}, an array of widget content... each Widget contains:
 - WidgetID
 - The indicator label / name <- String
 - The indicator clarification (Optional) <- String
 - The indicator reading <- String (could be a number / percentage)
 - Time related difference ratio <- Integer
 - The diagram category <- String (Options: graph, barChart ...)
 - The diagram feed <- Array of Floats
 - The diagram clarification <- String (example: "In Last 24 Hours")
- otherwise:
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: two days of business

Dependencies: Auth, DB1, DB2, DB4, Fltr

Requirements:

- A select function,
- data functions (w/ Pandas)

GET GET /analytics/views/widget/{Id}/{filtersOpts} <- (An2B)

Expected Parameters:

- accessToken: (a 32 Chars Long String)

Expected Response(s):

- if successful;
 - STATUS: OK,
 - DATA: {},
 - The indicator label / name <- String
 - The indicator clarification (Optional) <- String
 - The diagram category <- String (Options: graph, barChart ...)
 - The diagram feed <- Array of Floats
 - The diagram clarification <- String (example: "In Last 24 Hours")
- otherwise:
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: two days of business

Dependencies: Auth, DB1, DB2, DB4, Fltr

Requirements:

- A select function,
- data functions (w/ Pandas)

GET /analytics/views/actionsListByTicketNo/{filtersOpts}/{search} <- (An4)

Expected Parameters:

- *accessToken: (a 32 Chars Long String)*

Expected Response(s):

- *if successful;*
- *STATUS: OK,*
- *DATA: {}, Tickets List of Widgets, each contains:*
 - *A Ticket-Number*
 - *The number of interactions between brackets*
 - *“Resolved” / “Unresolved” status*
 - *The most mentioned keywords*
in the form of Chips in a Scrollable-Invisible-Box
 - *List of Rickets included in the ticket*
- *otherwise:*
 - *if accessToken isn’t valid, STATUS: KO, errMsg: “unauthorized”*
 - *otherwise, STATUS: KO, errMsg: “system failure, please try again”*

Estimated development time: two days of business

Dependencies: Auth, DB1, DB2, DB4, Fltr, IO1, IO2, IO3

Requirements:

- *A select function*

GET /analytics/views/actionsListByCleintId/{filtersOpts}/{search} <- (An5)

Expected Parameters:

- *accessToken: (a 32 Chars Long String)*

Expected Response(s):

- *if successful;*
- *STATUS: OK,*
- *DATA: {}, Tickets List of Widgets, each contains:*
 - *A Client-Id*
 - *The number of interactions between brackets*
 - *“Resolved” / “Unresolved” status*
 - *The most mentioned keywords*
in the form of Chips in a Scrollable-Invisible-Box
 - *List of Rickets included in the ticket*
- *otherwise:*
 - *if accessToken isn’t valid, STATUS: KO, errMsg: “unauthorized”*
 - *otherwise, STATUS: KO, errMsg: “system failure, please try again”*

Estimated development time: two days of business

Dependencies: Auth, DB1, DB2, DB4, Fltr, IO1, IO2, IO3

Requirements:

- *A select function*

GET /analytics/views/actionByTicketNo/{TicketNo}/{Ricket} <- (An6)
GET /analytics/views/actionByCleintId/{CleintId}/{Ricket} <- (An7)

Expected Parameters:

- accessToken: (a 32 Chars Long String)

Expected Response(s):

- if successful;
 - STATUS: OK,
 - DATA: {}, Interactions List of Widgets, each contains:
 - The category (Call OR Email)
 - Timestamp, format: DD/MM/YYYY, HH:mm:ss.MS
 - if the interaction is a call:
 - An audio player with a timestamped log of detected events,
possible detected events:
 - Silence
 - Noise
 - Escalation
 - Interruption (multi-channelled calls only)
 - Promise
 - Critical Info (in compliance with PCI)
 - Competitor Mention
 - Legal Action Mention
 - Call for Action
 - Lack of Understanding
 - Insignificant Talk
 - Profanity
 - Client Hold Request (multi-channelled calls only)
 - if the interaction is a text:
 - A text scrollable-invisible-box with highlighted events,
possible detected events:
 - Promise
 - Critical Info (in compliance with PCI)
 - Competitor Mention
 - Legal Action Mention
 - Call for Action
 - Profanity
 - otherwise:
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: five days of business

Dependencies: Auth, DB1, DB2, DB4, DB5, Fltr, IO1, IO2, IO3

Requirements:

- A select function

GET /activities/views/logs

<- (Ac1)

Expected Parameters:

- accessToken: (a 32 Chars Long String)
- filtersOpts: {} // to be discovered

Expected Response(s):

- if successful;
 - STATUS: OK,
 - DATA: {}, Logs List of Widgets, each contains:
 - Endpoint, format: VERB /endpoint
 - Timestamp, format: DD/MM/YYYY, HH:mm
 - Status, OK / KO
 - Content
 - expands into a floating view when clicked
- otherwise:
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: a day of business

Dependencies: Auth, DB1, DB2, DB5, Fltr, IO1, IO2, IO3

Requirements:

- A select function

GET /activities/views/multiBarGraphs

<- (Ac2)

Expected Parameters:

- accessToken: (a 32 Chars Long String)
- filtersOpts: {} // to be discovered

Expected Response(s):

- if successful;
 - STATUS: OK,
 - DATA: {}, Lists for Diagram Data
 - Successful & Unsuccessful Requests along timeframe
 - Incoming Requests & Outgoing Ones (Cron Jobs) along timeframe
- otherwise:
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: two days of business

Dependencies: Auth, DB1, DB2, DB5, Fltr, IO1, IO2, IO3

Requirements:

- A select function

GET /activities/views/pieCharts <- (Ac3)

Expected Parameters:

- *accessToken: (a 32 Chars Long String)*
- *filtersOpts: {} // to be discovered*

Expected Response(s):

- *if successful;*
- *STATUS: OK,*
- *DATA: {}, List of Chart Data*
 - *Successful Requests to Unsuccessful Requests Ratio*
 - *Incoming Requests to Outgoing Ones (Cron Jobs) Ratio*
- *otherwise:*
 - *if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"*
 - *otherwise, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: two days of business

Dependencies: Auth, DB1, DB2, DB5, Fltr, IO1, IO2, IO3

Requirements:

- *A select function*

GET /settings/profile <- (Se1)

Expected Parameters:

- *accessToken: (a 32 Chars Long String)*

Expected Response(s):

- *if successful;*
- *STATUS: OK,*
- *DATA: {}, Profile Data*
 - *Company's Name*
 - *Email Address*
 - *Contact Number*
- *otherwise:*
 - *if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"*
 - *otherwise, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: a day of business

Dependencies: Auth, DB1, DB2

Requirements:

- *A select function*

GET /settings/standards <- (Se2)

Expected Parameters:

- *accessToken: (a 32 Chars Long String)*

Expected Response(s):

- *if successful;*
- *STATUS: OK,*
- *DATA: {}, Standards Settings Data*
 - *Silence defined Length (ms)*
 - *Interruption defined Length (ms)*
 - *Whether to list actions by TicketNo OR by CleintId*

- otherwise:
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: a day of business

Dependencies: Auth, DB1, DB2

Requirements:

- A select function

GET /settings/constraints <- (Se3)

Expected Parameters:

- accessToken: (a 32 Chars Long String)

Expected Response(s):

- if successful;
 - STATUS: OK,
 - DATA: {}, Constraints Settings Data
 - Expected Number of Tickets; Daily, Weekly & Monthly
 - Thresholds
 - Minimum Call Length (sec)
 - Maximum Call Length (sec)
- otherwise:
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: a day of business

Dependencies: Auth, DB1, DB2

Requirements:

- A select function

GET /settings/getDevKeyList <- (Se4)

Expected Parameters:

- accessToken: (a 32 Chars Long String)

Expected Response(s):

- if successful;
 - STATUS: OK,
 - DATA: {}, Developers List of Widgets, each contains:
 - Developer's username
 - passKey
 - Status (Activated OR Deactivated)
- otherwise:
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: a day of business

Dependencies: Auth, DB1, DB2, DB3

Requirements:

- A select function

POST /settings/modify

<- (Se5)

Expected Parameters:

- *accessToken: (a 32 Chars Long String)*
- *profile/companyName: (String, Optional)*
- *profile/email: (String, Optional)*
- *profile/contactNo: (String, Optional)*
- *profile/password: (String, Optional) <- if included, requires:*
- *profile/currentPwd: (String, Dependent)*
- *standards/silenceLen: (Integer, Optional)*
- *standards/interruptionLen: (Integer, Optional)*
- *standards/listBy: (String, Optional)*
- *constrains/expTicketNo/daily: (Integer, Optional)*
- *constrains/expTicketNo/weekly: (Integer, Optional)*
- *constrains/expTicketNo/monthly: (Integer, Optional)*
- *constrains/thresholds/minCALLLen: (Integer, Optional)*
- *constrains/thresholds/maxCALLLen: (Integer, Optional)*

Expected Response(s):

- *if successful, STATUS: OK*
- *otherwise:*
 - *if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"*
 - *if password is included and the current password didn't match
STATUS: KO, errMsg: "password didn't match, please try again"*
 - *if password is included and the new password didn't qualify
STATUS: KO, errMsg: "password didn't qualify, please try again"*
 - *if the datatype of a parameter didn't match
STATUS: KO, errMsg: "unexpected datatype, please try again"*
 - *else,
STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: six days of business

Dependencies: Auth, DB1, DB2,

Requirements:

- *Multiple modification functions*

POST /settings/addDevKey

<- (Se6)

Expected Parameters:

- *accessToken: (a 32 Chars Long String)*
- *username, String, to-be-verified*

Expected Response(s):

- *if successful, STATUS: OK*
- *otherwise:*
 - *if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"*
 - *if username isn't valid, STATUS: KO, errMsg: "username isn't valid"*
 - *otherwise, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: half a day of business

Dependencies: Auth, DB1, DB2, DB3

Requirements:

- *An insert function*

POST /settings/modifyDevKey

<- (Se7)

Expected Parameters:

- accessToken: (a 32 Chars Long String)*
- username, String, to-be-verified*

Expected Response(s):

- if successful, STATUS: OK*
- otherwise:*
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"*
 - if username isn't valid, STATUS: KO, errMsg: "username isn't valid"*
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: two days of business

Dependencies: Auth, DB1, DB2, DB3

Requirements:

- A modification function*

POST /settings/delDevKey

<- (Se8)

Expected Parameters:

- accessToken: (a 32 Chars Long String)*
- username, String, to-be-verified*

Expected Response(s):

- if successful, STATUS: OK*
- otherwise:*
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"*
 - if username isn't found, STATUS: KO, errMsg: "username isn't found"*
 - otherwise, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: half a day of business

Dependencies: Auth, DB1, DB2, DB3

Requirements:

- A deletion function*

GET /userMgmt/getUsersList

<- (UM1)

Expected Parameters:

- accessToken: (a 32 Chars Long String)*

Expected Response(s):

- if successful,*
 - STATUS: OK*
 - DATA: {}, Users List of Widgets, each contains:*
 - Company Name*
 - Email*
 - userId*
 - Additional Info*
- otherwise:*
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"*
 - else, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: a day of business

Dependencies: Auth, DB1, DB2, DB3

Requirements:

- A select function

GET /userMgmt/searchUsersList/{...}

<- (UM2)

Expected Parameters:

- *accessToken: (a 32 Chars Long String)*

Expected Response(s):

- *if successful,*
 - *STATUS: OK*
 - *DATA: {}, Users List of Widgets, each contains:*
 - *Company Name*
 - *Email*
 - *userId*
 - *Additional Info*
- *otherwise:*
 - *if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"*
 - *else, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: two days of business

Dependencies: Auth, DB1, DB2, DB3, Fltr

Requirements:

- A select function

GET /userMgmt/getUserProfile/{userId}

<- (UM3)

Expected Parameters:

- *accessToken: (a 32 Chars Long String)*

Expected Response(s):

- *if successful,*
 - *STATUS: OK*
 - *DATA: {}, Profile Data*
 - *profile/companyName*
 - *profile/email*
 - *profile/contactNo*
 - *standards/silenceLen*
 - *standards/interruptionLen*
 - *standards/listBy*
 - *constrains/expTicketNo/daily*
 - *constrains/expTicketNo/weekly*
 - *constrains/expTicketNo/monthly*
 - *constrains/thresholds/minCALLLen*
 - *constrains/thresholds/maxCALLLen*
- *otherwise:*
 - *if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"*
 - *else, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: a day of business

Dependencies: Auth, DB1, DB2, DB3?

Requirements:

- A select function

POST /userMgmt/modifyUserProfile

<- (UM4)

Expected Parameters:

- accessToken: (a 32 Chars Long String)
- profile/companyName: (String, Optional)
- profile/email: (String, Optional)
- profile/contactNo: (String, Optional)
- profile/password: (String, Optional)
- standards/silenceLen: (Integer, Optional)
- standards/interruptionLen: (Integer, Optional)
- standards/listBy: (String, Optional)
- constrains/expTicketNo/daily: (Integer, Optional)
- constrains/expTicketNo/weekly: (Integer, Optional)
- constrains/expTicketNo/monthly: (Integer, Optional)
- constrains/thresholds/minCALLLen: (Integer, Optional)
- constrains/thresholds/maxCALLLen: (Integer, Optional)

Expected Response(s):

- if successful, STATUS: OK
- otherwise:
 - if accessToken isn't valid, STATUS: KO, errMsg: "unauthorized"
 - if password is included and the new password didn't qualify
STATUS: KO, errMsg: "password didn't qualify, please try again"
 - if the datatype of a parameter didn't match
STATUS: KO, errMsg: "unexpected datatype, please try again"
 - else,
STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: six days of business

Dependencies: Auth, DB1, DB2, DB3?

Requirements:

- Multiple modification functions

POST /record/call

<- (IO1)

Expected Parameters:

- devToken: (a 32 Chars Long String)
- 1stChannelFile: (Binary)
- 2ndChannelFile: (Binary, Optional)
- departmentLabel: (String, Optional)
- AgentId: (String, Optional)
- ClientId: (String, Optional)
- TicketNo: (String, Optional)

Expected Response(s):

- if successful, STATUS: OK, Ricket: e.g. 2020120211461900045C
- otherwise:
 - if devToken isn't valid
STATUS: KO, errMsg: "unauthorized"
 - if 1stChannelFile didn't match .WAV format
STATUS: KO, errMsg: "binary needs to be in .WAV format"
 - else, STATUS: KO, errMsg: "system failure, please try again"

Estimated development time: twelve days of business

Dependencies: Auth, DB1, DB2, DB3, DB4, DB5, word-sequences pools

Requirements:

- Pydub's function; split_on_silence, Silence:
absence of volume above X dBFS (X: audio.dBFS - 15)
for longer than Y seconds (default Y: .5 / 500ms)
(Who-Said-What?; for both Agent & Client)
- a Speech-to-Text 3rd Party API function, (to discover)
- _thread function to fork the process and run it parallelly
- a noise detection function, Noise:
when an audio chunk isn't transcribed by the Speech-to-Text API
(Who-Said-What?; for both Agent & Client)
- an escalation detection function, Escalation:
when the pitch is detected to be higher than the median* over a time
period longer than Y, where the difference in audio is Z dBFS.
X, Y, Z are to be specified after discovery.
* median pitch is calculated along the call excluding silence & noise
(Who-Said-What?; for both Agent & Client)
- an interruption detection function, Interruption:
when speech-identified-audio is detected on clashing timestamps
of both channels (for more than Y seconds, default Y: .75 / 750ms)
(Who-Interrupted-Who?; for both Agent & Client)
- a promises detection function, Promises:
when the future tense is used (to be tracked by time statement/s)
(Who-Said-What?; for both Agent & Client)
- a word-sequences detection function, word-sequences pools:
 - Critical Info (in compliance with PCI)
 - Competitor Mention
 - Legal Action Mention
 - Call for Action
 - Lack of Understanding
 - Profanity
 - Insignificant Talk
 - Client Hold Request(Who-Said-What?; for both Agent & Client)

POST /record/text

<- (I02)

Expected Parameters:

- *devToken: (a 32 Chars Long String)*
- *TextContent: (String)*
- *departmentLabel: (String, Optional)*
- *AgentId: (String, Optional)*
- *ClientId: (String, Optional)*
- *TicketNo: (String, Optional)*

Expected Response(s):

- *if successful, STATUS: OK, Ricket: e.g. 2020120211197000019T*
- *otherwise:*
 - *if devToken isn't valid*
STATUS: KO, errMsg: "unauthorized"
 - *else, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: four days of business

Dependencies: Auth, DB1, DB2, DB3, DB4, DB5, word-sequences pools

Requirements:

- *_thread function to fork the process and run it parallelly*
- *a promises detection function, Promises:*
when the future tense is used (to be tracked by time statement/s)
- *a word-sequences detection function, word-sequences pools:*
 - *Critical Info (in compliance with PCI)*
 - *Competitor Mention*
 - *Legal Action Mention*
 - *Call for Action*
 - *Profanity*
 - *Insignificant talk*

GET /report/{Ricket}

<- (I03)

Expected Parameters:

- *devToken: (a 32 Chars Long String) // OR ?*
- *Ricket: (String)*

Expected Response(s):

- *if successful*
 - *STATUS: OK,*
 - *DATA: {}, Profile Data <- to be defined*
- *otherwise:*
 - *if devToken isn't valid*
STATUS: KO, errMsg: "unauthorized"
 - *else, STATUS: KO, errMsg: "system failure, please try again"*

Estimated development time: four days of business

Dependencies: Auth, DB1, DB2, DB3, DB4, DB5, Fltr

Requirements:

- *A select function*

Performance Requirements

A section to define what is expected from the feature - failure contingencies, redundancy requirements, etc

- The loading of a dashboard page shouldn't exceed 500ms time*
- The browser caching memory shouldn't exceed x MB ...*
- Passing all unit tests before pushing updates to production is a must*

Additional Requirements

DB1 Task: Database Integration , time estimated: two business days
DB2 Task: User Profile Document Design , time estimated: a day of business
DB3 Task: Developers Document Design , time estimated: a day of business
DB4 Task: Interactions Document Design , time estimated: a day of business
DB5 Task: ActivitiesLog Document Design , time estimated: a day of business
DB6 Task: Word Sequence Document Design , time estimated: a day of business

+

Auth: An authentication function for Ov, An, Ac, Se, UM, IO features
Fltr: A search & filtration mechanism to be defined and implemented

Testing Procedure

Unit testing using plain Python & its web library "request" is the method of testing, each one of the expected responses (whether it being a success or an error response) is to be tested at least once within the unit testing script, these tests are to be immigrated into the CI/CD environment at a later stage of MVP delivery / deployment.

Hardware and Software Requirements

Infrastructure:

- AWS Servers & Lambda Functions, AtlasDB & Company's Assets

Software:

- Python (v3.9.0+), Flask, Pandas, Pydub, LibROSA...
- An Emailing API (SendGrid), a speech-to-text API (AWS Transcribe)

Notes:

- The user authentication could be modularly taken from YOOTOK
- Remaining to be defined:
 - search & filtration mechanism
 - notification mechanism
 - ...

Summary

...