

Vincere API Integration – Usage

We have developed a package named vincere-api which will have vincere API wrapper to make it convenient for end user to use the script without any modification in the code file.

The script accepts required details through command line arguments.

System Requirements:

The python3.x should be installed. After installation of python run following command to install pre-requisite.

```
$pip install -r requirements.txt
```

The requirements.txt file is present at the root level in code repository.

```
$python main.py -h
```

```
usage: main.py [-h] [-m] [-c CANDIDATE] [-d DELETE [DELETE ...]] [-r REASON]
               [-i INDUSTRIES [INDUSTRIES ...]] [-e EXPERTISE [EXPERTISE ...]]
               [-s SUB [SUB ...]] [-f FUNCTIONAL_ID] [-l COMPANY_COUNT] [-v]
```

optional arguments:

-h, --help show this help message and exit
-v, --verbose Print verbose logging on screen

Requests:

-m, --master Generate Master Records
-c CANDIDATE, --candidate CANDIDATE
 Candidate ID
-d DELETE [DELETE ...], --delete DELETE [DELETE ...]
 Space separated candidate list.
-r REASON, --reason REASON
 Reason to delete candidates.
-i INDUSTRIES [INDUSTRIES ...], --industries INDUSTRIES [INDUSTRIES ...]
 Space separated industries list.
-e EXPERTISE [EXPERTISE ...], --expertise EXPERTISE [EXPERTISE ...]
 Space separated function expertise list.
-s SUB [SUB ...], --sub SUB [SUB ...]
 Space separated Sub function expertise list.
-f FUNCTIONAL_ID, --functional FUNCTIONAL_ID
 Functional Expertise ID.
-l COMPANY_COUNT, --company_count COMPANY_COUNT
 Company count.

Operation wise command

Operation	Command
Get note_on This command will get note_on data for the given candidate.	\$ python main.py -c <<candidate_id>> i.e. \$ python main.py -c 70468
Set Company count This command will set company count for the given candidate	\$ python main.py -c <<candidate_id>> -l <<company_count>> i.e. \$ python main.py -c 70468 -l 5
Set Industries This command will set list of industries for a given candidate.	\$ python main.py -c <<candidate_id>> -i <<Space separated industry ids>> i.e. \$ python main.py -c 70468 -i 4 5 8 40
Set Functional Expertise This command will set list of functional expertise for a given candidate.	\$ python main.py -c <<candidate_id>> -i <<Space separated functional_expertise ids>> i.e. \$ python main.py -c 70468 -e 4 5 8 40
Set sub Functional Expertise This command will set list of sub functional expertise for a given candidate.	\$ python main.py -c <<candidate_id>> -f <<functionan_expertise_id> -s <<Space separated sub functional_expertise ids>> i.e. \$ python main.py -c 70468 -f 40 -s 14 15 140
Delete Candidates The function will delete multiple candidates in a single run.	\$ python main.py -d <<Space separate candidate ids>> -r <<reason>> i.e. \$ python main.py -d 70465 3456 78908 -r "No more valid candidates."
Generate Master The above command requires IDs to be passed for many of the pre-defined parameters. Currently we need List of Industries, Functional Expertise and Sub Functional Expertise .	\$python main.py -m