
OpenOversight Documentation

Lucy Parsons Labs

Nov 08, 2020

Contents:

1	Area Coordinator	1
2	Using the bulk upload feature	3
2.1	Preparation steps	3
2.2	Layout of the csv file	3
2.3	Explanation of the individual fields	4
2.4	Command-line options	5
2.5	The command to run on the server	5
3	Advanced csv import	7
3.1	Before you start	7
3.2	Explanation of the command	7
3.3	Examples	11
3.4	Local development flag <code>--force-create</code>	11
4	Indices and tables	13

CHAPTER 1

Area Coordinator

An area coordinator is a person with an account on OpenOversight that is charged with curating the collection for a particular department. There can be several area coordinators for a given department. The tasks they perform include:

- Add new officers
- Update officer information, such as new assignments or promotions
- Add private notes to particular officer profiles (visible to other area coordinators or administrators)
- Add a brief description to an officer's profile
- Add links and videos to officer's profiles
- Add incidents linked to one or more officers

Using the bulk upload feature

To facilitate adding rosters of new departments or updating existing departments there is a flask command that gives system administrators the ability to load the data from a csv file directly into the database.

Warning This process is not very robust at this point and there is risk of leaving the database in an inconsistent state. It is strongly recommended to back up the database before starting this operation and to run the command on a development server first and see if the results are as expected.

2.1 Preparation steps

- Create department if it does not exist
- Add ranks in hierarchical order, make sure all the ranks present in the csv are added

2.2 Layout of the csv file

The csv file can have the following fields:

```
department_id
unique_internal_identifier
first_name
last_name
middle_initial
suffix
gender
race
employment_date
birth_year
star_no
job_title
unit_id
```

(continues on next page)

(continued from previous page)

```
star_date
resign_date
salary
salary_year
salary_is_fiscal_year
overtime_pay
```

2.3 Explanation of the individual fields

2.3.1 General information:

- `department_id` id of department in the server database, for example 1 for Chicago Police Department can be found in url for that department: <https://openoversight.com/department/1>
- `unique_internal_identifier` a string or number that can be used to uniquely identify the officer, in departments in which the badge number stays with the officer using that number is fine, otherwise it is recommended to leave this blank and provide the `star_no` instead.
- `first_name` & `last_name` & `middle_initial`
- `suffix` one of Jr, Sr, II, III, IV, V
- `gender` one of M, F, Other or Not Sure
- `race` one of BLACK, WHITE, ASIAN, HISPANIC, NATIVE AMERICAN, PACIFIC ISLANDER, Other, Not Sure
- `employment_date` start of employment with this department
- `birth_year`

2.3.2 Current Employment information:

- `star_no` star or badge number, might be related to current rank
- `job_title` rank or title, needs to be added to this department verbatim or Not Sure
- `unit_id` id of unit within the department
- `star_date` (sic) start date of this assignment
- `resign_date` resignation date of this assignment

2.3.3 Salary information:

- `salary_year` year of which the salary information is provided
- `salary_is_fiscal_year` 'true' or 'false', salary information is on fiscal year basis vs. calendar year
- `salary` salary in given year
- `overtime_pay` overtime received in given year

2.3.4 Required fields

- `department_id`, `first_name`, `last_name`, `job_title` and either `star_no` or `unique_internal_identifier` are required.
- `employment_date`, `star_date` and `resign_date` can be either in `yyyy-mm-dd` or `mm/dd/yyyy` format - if the column is present the field cannot be left blank

2.4 Command-line options

- `--no-create` - For each line in the CSV, update an existing officer if one exists, but do not create any new officers. If an officer in the CSV is not already in OpenOversight, the line will be ignored.
- `--update-by-name` - Update officers by `first_name` and `last_name`. Useful when `unique_internal_identifier` and `star_no` are not available.
- `--update-static-fields` - Allow modifications to normally-static fields like `race`, `birth_year`, etc, which OpenOversight normally prevents from being modified. Values in the database will be overwritten with values in the CSV.

2.5 The command to run on the server

```
/usr/src/app/OpenOversight$ flask bulk-add-officers [/path/to/csv_file.csv]
```


CHAPTER 3

Advanced csv import

This is the documentation of the advanced csv import command `advanced_csv_import`. For the documentation of the less complicated bulk-upload command `bulk-add-officers` see [Using the bulk upload feature](#). `bulk-add-officers` accepts one csv containing information about the officer, including badge-number, jobs and salary and makes decisions on whether to update rows in the database or create new entries based on the existing data.

The advanced csv upload is for the most part a way to copy data for one department into the database with as little as possible logic added on. So the tables provided in csv form represent the data that will be inside the sql tables after running the command. (With a few exceptions for many-to-many relationships and auxiliary models like location and license plates)

3.1 Before you start

CSV uploads should always be tested locally or in other non-production environments and it is strongly recommended to have the database backed up before running the command. The command is designed to fail early and will only commit the changes if it didn't encounter any problems. The command however is pretty powerful and can therefore lead to data loss and inconsistencies if the provided csv files are not prepared correctly.

3.2 Explanation of the command

```
/usr/src/app/OpenOversight$ flask advanced-csv-import --help
Usage: flask advanced-csv-import [OPTIONS] DEPARTMENT_NAME
```

```
Add or update officers, assignments, salaries, links and incidents from
csv files in the department DEPARTMENT_NAME.
```

```
The csv files are treated as the source of truth. Existing entries might
be overwritten as a result, backing up the database and running the
command locally first is highly recommended.
```

(continues on next page)

(continued from previous page)

```
See the documentation before running the command.
```

```
Options:
```

```
--officers-csv PATH
--assignments-csv PATH
--salaries-csv PATH
--links-csv PATH
--incidents-csv PATH
--force-create      Only for development/testing!
--help             Show this message and exit.
```

The command expects one mandatory argument the department name. This is to reduce the chance of making changes to the wrong department by mixing up files. Then there are 5 options to include paths to officers, assignments, salaries, incidents and links csv files. Finally there is a `--force-create` flag that allows to delete and overwrite existing entries in the database. This is only supposed to be used in non-production environments and to allow replication of the data of another (in most cases production) instance to local environments to test the import command locally first. More details on that flag at the end of the document: [Local development flag `--force-create`](#).

3.2.1 General overview of the csv import:

The following lists the header fields that each csv can contain. If the csv includes any other fields, the command will fail. However the fields are not case-sensitive and spaces are treated as `_`. So `Officer ID` can be used instead of `officer_id`.

All optional fields can be left blank and will be inserted as NULL or empty string as appropriate. **Warning:** When updating a record a field that is left blank might overwrite an existing record. This can only be prevented by not including the column in the csv at all.

Formats

- **date:** The date should be provided in `YYYY-MM-DD` format.
- **time:** Time should be provided in `HH:MM:SS` 24h-format in the respective timezone.

The `id` field

Each csv corresponds to a table in the OpenOversight database. And each csv file has to include `id` as a field in the table. That field has one primary purpose: If the field is blank, it is assumed that that row is a new entry. If the field contains a number however, it is assumed that a record with that particular `id` already exists in the database and the record will be updated according to the provided fields. Finally in the case of officers and incidents there is a third option where the `id` field can contain a string that starts with `#`. This also indicates a new record, but that new record can be referenced in other provided tables. (for example as the `officer_id` in the salaries csv)

3.2.2 Officers csv

- **Required:** `id`, `department_name`
- **Optional:** `last_name`, `first_name`, `middle_initial`, `suffix`, `race`, `gender`, `employment_date`, `birth_year`, `unique_internal_identifier`
- **Ignored:** `badge_number`, `job_title`, `most_recent_salary` (Unused but command will not fail when field is present)

Details:

- `department_name` Name of department exactly as in the server database. This needs to match the department name provided with the command.
- `unique_internal_identifier` A string or number that can be used to uniquely identify the officer, in departments in which the badge number stays with the officer using that number is fine. Can and should be left blank if no such number is available.
- `first_name` & `last_name` Will be inserted into the database as is.
- `middle_initial` Usually up to one character, but can be more.
- `suffix` Choice of Jr, Sr, II, III, IV, V.
- `gender` Choice of M, F, Other.
- `race` One of BLACK, WHITE, ASIAN, HISPANIC, NATIVE AMERICAN, PACIFIC ISLANDER, Other.
- `employment_date` *Date* representing the start of employment with this department.
- `birth_year` Integer representing the birth year of the officer.

3.2.3 Assignments csv

- Required: `id`, `officer_id`, `job_title`
- Optional: `badge_number`, `unit_id`, `start_date`, `resign_date`

Details:

- `officer_id` Number referring to `id` of existing officer or string starting with # referring to a newly created officer in the provided officers csv.
- `badge_number` Any string that represents the star or badge number of the officer. In some departments this number changes with the assignment.
- `job_title` The job title, needs to be created for that department.
- `unit_id` Id of existing unit within the department.
- `start_date` Start *date* of this assignment.
- `resign_date` End *date* of this assignment.

3.2.4 Salaries csv

- Required: `id`, `officer_id`, `salary`, `year`
- Optional: `overtime_pay`, `is_fiscal_year`

Details:

- `officer_id` Integer referring to `id` of existing officer or string starting with # referring to a newly created officer in the provided officers csv
- `salary` Number representing the officer's salary in the given year.
- `year` Integer, the year this salary information refers to.

- `overtime_pay` Number representing the amount of overtime payment for offer in given year.
- `is_fiscal_year` Boolean value, indicating whether the provided year refers to calendar year or fiscal year. The values `true`, `t`, `yes` and `y` are treated as “yes, the salary is for the fiscal year”, all others (including blank) as “no”

3.2.5 Incidents csv

- Required: `id`, `department_name`
- Optional: `date`, `time`, `report_number`, `description`, `street_name`, `cross_street1`, `cross_street2`, `city`, `state`, `zip_code`, `creator_id`, `last_updated_id`, `officer_ids`, `license_plates`

Details:

- `department_name` Name of department exactly as in the server database. This needs to match the department name provided with the command.
- `date` *Date* of the incident
- `time` *Time* of the incident
- `report_number` String representing any kind of number assigned to complaints or incidents by the police department.
- `description` Text description of the incident.
- `street_name` Name of the street the incident occurred, but should not include the street number.
- `cross_street1`, `cross_street2` The two closest intersecting streets.
- `city`, `state`, `zip_code` State needs to be in 2 letter abbreviated notation.
- `creator_id`, `last_updated_id` Id of existing user shown as responsible for adding this entry.
- `officer_ids` Ids of officers involved in the incident, separated by `|`.
 - Each individual id can either be an integer referring to an existing officer or a string starting with `#` referring to a newly created officer.
 - Example: `123|#C1|1627` for three officers, one with id 123, one with 1627 and one whose record was created in the officers csv and whose id-field was the string `#C1`.
- `license_plates` all license plates involved in the incident. If there is more than one, they can be separated with a `|`.
 - Each license plate consists of the license plate number and optionally a state in abbreviated form separated by an underscore `_`.
 - Example: `ABC123_IL|B991` for one license plate with number ABC123 from Illinois and one with number B991 and no associated state.

3.2.6 Links csv

- Required: `id`, `url`
- Optional: `title`, `link_type`, `description`, `author`, `creator_id`, `officer_ids`, `incident_ids`

Details:

- `url` Full url of the link starting with `http://` or `https://`.
- `title` Text that will be displayed as the link.
- `description` A short description of the link.
- `link_type` Choice of `Link`, `YouTube Video` and `Other Video`.
- `author` The source or author of the linked article, report, video.
- `creator_id` Id of existing user shown as responsible for adding this entry.
- `officer_ids` Ids of officer profiles this link should be visible on, separated by `|`. See same field in incidents above for more details.
- `incidents_ids` Ids of incidents this link should be associated with, separated by `|`. Just like `officer_ids` this can contain strings starting with `#` to refer to an incident created in the incident csv.

3.3 Examples

Example csvs can be found in the repository under `OpenOversight/tests/test_csvs`.

3.4 Local development flag `--force-create`

This flag changes the behavior when an integer is provided as `id`. Instead of updating an existing record, a new record will be created and assigned the given `id`. If a record with that `id` already exists in the database, it will be deleted before the new record is created.

This functionality is intended to be used to import csv files downloaded from [OpenOversight download page](#) to get a local copy of the production data for one department in the local development database.

OpenOversight is a Lucy Parsons Labs project to improve law enforcement accountability through public and crowd-sourced data. We maintain a database of officer demographic information and provide digital galleries of photographs. This is done to help people identify law enforcement officers for filing complaints and in order for the public to see work-related information about law enforcement officers that interact with the public.

CHAPTER 4

Indices and tables

- `genindex`
- `search`