



Deals/API Intern 2022

Here are the two tasks we want you to try. Your solution should be:

- working, meeting given requirements,
- tested,
- documented, so that we know how to use it and how to run your tests,
- clean and readable

Task 1

OpenX joins two type of clients - Buyers and Sellers. Buyers buy ads, Sellers (aka Publisher) sell space to present those ads. OpenX obeys the standard of telling other ad-tech parties about the Sellers we work with at <https://openx.com/sellers.json>, under .sellers key. Example entry in that list:

```
{
  "seller_id": "540154109",
  "name": "Ascendium Co.",
  "domain": "ascendium.com",
  "seller_type": "INTERMEDIARY",
  "is_passthrough": 0
}
```

Seller in that list has its own Sellers when its type ("seller_type") is of "INTERMEDIARY" or "BOTH". For example, at <https://ascendium.com/sellers.json> you will find Ascendium's list of sellers and the same story repeats. This whole tree of sellers under sellers, under sellers, etc, is called *Supply Chain*.

Prepare a solution that will present whole Supply Chain tree for OpenX in an easy to read way:

- user should be able to see if a domain is direct or indirect Seller,
- user should be able to tell what's the maximum depth of our Supply Chain.

Task 2

Assume you are given set of files listing events in people's calendars.

File with such a content:

```
2022-06-01 12:00:00 - 2022-06-01 12:59:59
2022-06-01 15:20:00 - 2022-06-01 15:49:59
```

means that a person on 2022-06-01 is busy for one hour at 12:00 and busy for half an hour at 15:20.

Additionally, if a line in that file consists only of a date (like "2022-07-01") then it means a person is busy whole day. Separate file for each person.

Prepare a script that will print out the soonest date in the future when at least desired amount of people are available for given amount of time. Script should accept a named parameter `--duration-in-minutes` which defines for how many minutes people should be available. Minimum number of people that must be available should be defined by `--minimum-people` argument. Script should read people's calendars from `*.txt` files in the directory provided as an `--calendars` string.

Example

Assuming it's 2022-07-01 09:00:00 now and:

- file `/in/alex.txt` consists of:

```
2022-07-02 13:15:00 - 2022-06-01 13:59:59
```
- file `/in/brian.txt` consists of:

```
2022-07-01
2022-07-02 00:00:00 - 2022-07-02 12:59:59
```

Calling the script:

```
python3 find-available-slot.py --calendars /in --duration-in-minutes 30 --minimum-people 2
```

should print out:

```
2022-07-02 14:00:00
```