MYRA

Attendance System

We require to track the attendance of staff across our fulfilment centers, call center and the delivery fleet. We also require a report to be generated which gives the total minutes worked per day for each user.

A bit of background...

- Assume that each employee has access to a mobile-app which uses this set of APIs.
- Every day, each employee chooses to `start_day` as soon as they log in to the office premises.
- At the end of every day, the employee chooses to `stop_day` as they leave the office premises.
- In between, the employee can do `start_break` & `stop_break` to log their breaks (such as for lunch, coffee, etc.).

Requirements

- We need a system/API that, when given an `employee_id`, `start_date` and `end_date`, gives us the `total_minutes_worked` each day.
- The system should also perform appropriate data-validations. *E.g.*, it should handle the possibility of `stop_day` being done before `start_day`.
- Note that all the date times are in ISO format.

Included with the assignment is a Postman collection. Please make sure that your API conforms to the ones in that collection. You can download Postman here: https://www.getpostman.com and import the collection to see the API structure and the test the response.

Start Day

```
POST /attendance/
```

Request:

```
{
  "employee_id": 1,
  "date": "2017-05-1",
  "time": "10:05:35",
  "event": "start"
}
```

Response:

```
Code: 201

{
  "attendance": {
    "date":"2017-05-1",
    "start_time": "10:05:35",
    "end_time": null,
    "total_minutes_worked": 0
  }
}
```

End Day

```
POST /attendance/
```

Request:

```
{
  "employee_id": 1,
  "date": "2017-05-1",
  "time": "18:03:25",
  "event": "end"
}
```

Response:

```
Code: 200

{
  "attendance": {
    "date":"2017-05-1",
    "start_time": "10:05:35",
    "end_time": "18:04:12",
    "total_minutes_worked": 0
  }
}
```

Start Break

```
POST /break/
```

Request:

```
{
  "employee_id": 1,
  "date": "2017-05-1",
  "time": "12:10:35",
  "event": "start"
}
```

Response:

```
Code: 201

{
  "break": {
    "date":"2017-05-1",
    "start_time": "12:12:45",
    "end_time": null,
  }
}
```

End Break

`POST /break/`

Request:

```
{
  "employee_id": 1,
  "date": "2017-05-1",
  "time": "13:20:34",
  "event": "stop"
}
```

Response:

```
Code: 200

{
  "break": {
    "date":"2017-05-1",
    "start_time": "12:12:45",
    "end_time": "13:34:54"
  }
}
```

Listing attendance

`GET /attendance/?start_day=2017-05-1&end_day=2017-05-10&employee_id=1`

This API should return the attendance for all the days worked and the total minutes worked for each day. This will exclude all the break time taken during the work day.

Example: If start work is at 10:00 and stop is at 18:00, and one break is taken from 12:00 to 13:00, then the `total_minutes_worked` for that day = 420 minutes = 7 hours.

Response:

```
Code: 200

{
  "attendance": [
    {
      "date":"2017-05-1",
```

```
      "start_time": "10:05:35",
      "end_time": "18:04:12",
      "total_minutes_worked": 430
    },
    {
      "date":"2017-05-1",
      "start_time": "11:07:33",
      "end_time": "19:23:43",
      "total_minutes_worked": 442
    }
  ]
}
```

Expected load is 10 requests per second for each API. Response time should be within 100 ms.

Deliverable requirements

1. You can choose stack of your preference.
2. It's preferred to use macOS or Linux environment for coding if not then it must run on either of those.
3. It's mandatory to use Git for version control.
4. We expect you to send us a zip/tarball of your source code which should include Git metadata (the .git folder in the tarball so we can look at your commit logs and understand how your solution evolved).We expect you to do commits in regular interval.
5. It's mandatory to include a Readme file which must contain the following:
    - Overview of the tech stack, language etc.
    - Brief description of reason for using the specific tech stack
    - Infrastructure requirements for running your solution
    - Setup instructions, automated deployment of the program and dependencies to development and test environment is a plus
6. It's a huge plus if you have written unit test cases and automated them.
7. We are really interested in your application design/structure skills, so please solve the problem keeping this in mind.