

Description

Your task is to design an API for an automated charging station management system.

Our station management system consists of 3 entities: Station, StationType, and Company.

- Company(id, name). A company can have multiple child companies
- A Station(id, name) belongs to a Company(id, name)
- A Station has one StationType(id, name, maxPower)

This database schema you can use as the starting point. Please feel free to add or modify more fields/columns if needed.

- Feel free to use any Javascript framework.
- Feel free to choose any kind of SQL database that fits you the best.

Task 1

- Design CRUD APIs for managing stations, station types, companies.
- We also want to have an endpoint that takes a company id and responds with data (stationId, stationName, maxPower) about all stations that belong to the given company and its child companies

Task 2

We want to design a script parser that translates user inputs into controlling commands that are sent to stations, and responses with charging power in time-series data format

Your task is to implement an API for that parser.

The script starts with **Begin** and ends with **End**. Valid commands are:

- Start station **<stationId>|all**: start charging one station or all stations. When **all** is given, the command applies to all stations in the system
- Stop station **<station-id>|all**: stop charging one station or all stations. When **all** is given, the command applies to all stations in the system
- Wait **<time-in-second>**: do nothing and wait for the given period in seconds

The response of this API should report the current state of the system step by step:

- Charging station ids, and the total charging power of the charging stations grouped by company
- All charging station ids
- Total charging power of all charging stations

Please refer to the example for more details

Notes:

- A station can only be in 2 states: charging or not charging, and when it is charging it consumes maxPower from its station type

- If a charger belongs to a child company, it also reports stationId and charging power to the parent companies
- No need to report **Wait** steps.

Example

Given:

- 3 company: company 1, company 2, company 3
- Company 2, 3 are child companies of company 1
- Company 1 owns stations 5
- Company 2 owns stations 2,3
- Company 3 owns stations 1,4
- All stations have 1 stationType with **maxPower** = 10

Request:

```
Begin
Start station 1
Wait 5
Start station 2
Wait 10
Start station all
Wait 10
Stop station 2
Wait 10
Stop station 3
Wait 5
Stop station all
End
```

Response:

```
{
  data: [
    {
      step: "Begin"
      timestamp: <unix-timestamp-of-step-1>,
      companies: [],
      totalChargingStations: [],
      totalChargingPower: 0
    },
    {
      step: "Start station 1"
      timestamp: <unix-timestamp-of-step-2>,
      companies: [
        {
          id: 1,
          chargingStations: [1],
```

```

        chargingPower: 10
      },
      {
        id: 3,
        chargingStations: [1],
        chargingPower: 10
      },
    ],
    totalChargingStations: [1],
    totalChargingPower: 10
  },
  {
    step: "Start station 2"
    timestamp: <unix-timestamp-of-step-3 = timestamp-of-step-2 +
5seconds>,
    companies: [
      {
        id: 1,
        chargingStations: [1,2],
        chargingPower: 20
      },
      {
        id: 2,
        chargingStations: [2],
        chargingPower: 10
      },
      {
        id: 3,
        chargingStations: [1],
        chargingPower: 10
      },
    ],
    totalChargingStations: [1,2],
    totalChargingPower: 20
  },
  ...and so on
]
}

```

Notes: feedbacks and communication are very welcome, if you think the specifications are not clear or if you have any questions, please feel free to contact us