# Parking lot management API

Design a parking lot API using object-oriented principles.

Design and code a Parking log management API following the rules described. It is important that you apply your knowledge on coding best practices, design patterns, clean code, etc, so try to polish it as much as you can.

## Parking rules

1. The parking lot has multiple levels. Each level has multiple rows of spots.
2. The parking lot can park motorcycles, cars, and buses.
3. The parking lot has motorcycle spots, compact spots, and large spots.
4. A motorcycle can park in any spot.
5. A car can park in either a single compact spot or a single large spot.
6. A bus can park in five large spots that are consecutive and within the same row. It cannot park in small spots.

## API requirements

We need this rules to be included in an API, in the following way:

- When a new vehicle arrives to the parking, it has to be assigned a free spot or set of spots where to park.
- Once the vehicle is parked in the notified spot or set of spots, those have to be marked as occupied.
- One vehicle can be changed to a different location within the parking.
- The vehicle can leave the parking at any time, and the assigned spot or set of spots have to be released.
- The parking has to provide, by request, its complete status, giving the list of spots, the status and location of every spot, and in case of being occupied, which vehicle type is parked.
- The parking has to provide, by request, the amount of vehicles along with their type, per level and totalized.

## Assumptions

You can consider the following assumptions:

- A standard parking has 3 levels
    - Level 1: 100 spots distributed in 10 rows of 10 consecutive spots each
    - Level 2: 80 spots distributed in 7 rows of 10 spots, and 1 row of 8 spots
    - Level 3: 75 spots distributed in 6 rows of 10 spots, 1 row of 8 spots, and 1 row of 7 spots
- API has to be web oriented, that is, HTTP, REST, or whatever other standard you deem appropriate.
- API has to be coded in Java 8+.
- Consider to add a software project manager like Maven, Gradle or any other of your knowledge. You can add any needed dependency to the project file.
- It is **not** mandatory to deliver a complete implementation, but try to go as further as you can in the agreed time.

- Code structure and organization is of great importance, so please, do your best in this sense.
- This is an offline technical test which means we cannot give you direct support, so if you have any doubt, take a decision and let us know in a comment block, or in the email with your solution attached. We will take your assumptions and decision into consideration.

Many thanks for your time and good luck!! We are looking forward to having a look to your solution ☺