

# Parking Lot

**Vague, high-level problem statement, as delivered in an interview:**

Design/Model a parking lot

**What that typically means:** Model a parking lot and design a REST API for a company that operates a parking lot. Client of that software is "UI" that other colleagues of yours will build on top of your API.

As usual, key is to start simple. Assume a single-story parking lot for cars only. Once you get the basic structure going, then add more vehicle types (e.g. Trucks, RVs (takes up multiple spots), bikes) differentiation (e.g. SUV, Compact) and floors. Then add queues, time limits and payments. And then think about multiple parking lots (will need you to add an admin layer).

You can also harness the power of tech and make it easier for people to park and exit. E.g. assign a parking spot and put it back into the pool once a car exits.

## **Deliverables:**

1. A class diagram, showing relationships with each other where appropriate. Classes should show state and methods. Use any convenient notation. UML is more widely known.

2. Main() method, showing how you'll initialize your system and start using it.

3. Identify the design pattern.

4. A REST API

Extension-1: Can your model be used to implement software for a 3-car garage?

Extension-2: What are the right data-structures to use, if you want to find an empty spot in constant time?