



TDDBUDDY.COM

# Laundry Reservation

---

## *The Kata*

Your local Laundromat, *Wunda Wash*, is revamping their facilities and have asked you to help them implement an IoT solution to modernize the Laundry industry. The IoT device has one function; allow patrons to access a reserved washing machine when arriving at the facility.

The IoT device uses an electronic lock to only allow the patron who has reserved the machine access to it. The lock is unlocked via a PIN. Reservation details are sent to the device at the time of reservation. And re-synced every 30 minutes via a scheduled task on the server. The scheduled task pushes the next 6 hours of data each execution.

When a patron makes a reservation, they enter the reservation date, time, their cell phone number and email address. An available machine is selected and the reservation is made; *assume a machine is always available to be reserved and that a user may only have a single active reservation at a time.*

Once the reservation is saved to the DB an email is sent with a machine number, reservation ID and 5-digit PIN. The PIN is entered into the IoT device to access the reservation and unlock the machine.

Write the following functionality:

- Create Reservation
  - Takes in
    - Reservation date and time
    - Cell phone number
    - Email address
  - Sends confirmation email with a machine number, reservation ID and a 5 digit PIN
    - Assume there are 25 machines and one will always be available, randomly pick one when making the reservation
  - Saves reservation to the DB
  - Send lock instruction to selected machine via **Machine API**
    - Reservation Id
    - Machine number
    - Reservation date and time
    - PIN
- Machine API
  - The API should be designed to service a collection of devices using the machine number to locate the correct device
  - Assume API logic wraps a Device SDK given to you with the following interface

```
public interface IMachineDevice{
    bool Lock(string reservationId, DateTime reservationDateTime, int pin);
    void Unlock(string reservationId);
}
```
  - Lock Machine
    - Takes in machine number, DateTime of reservation and reservationId
      - *Lock(reservationId, machineNumber, reservationDateTime)*
    - Returns true if the machine was unlocked and could be locked at the specified DateTime via the SDK
    - If the reservationId already exist, then update the pin and DateTime returning true.
    - In all other cases return false.



TDDBUDDY.COM

- Unlock Machine
  - Takes in machine number and reservationId
    - *Unlock(machineNumber, reservationId)*
- Claim a Reservation
  - Takes in
    - Machine ID
    - PIN
  - If the PIN matches the active reservation for the machine
    - Updates the reservation in the DB as used
    - IoT device unlocks machine
  - After 5 failed attempts to enter the PIN send an SMS to the parton's cell phone with a new pin and update the reservation to reflect this new PIN so they may try again
    - Ensure the Lock Machine method is called so the IoT device receives the updated pin.
      - ReseavationId
      - DateTime
      - New Pin
- Cancel a Reservation
  - Takes in
    - Reservation ID
  - Sends email notifying user of cancellation
    - Be sure to include the reservation ID in the email
  - Updates the reservation in the DB as canceled
  - Unlocks machine via the **Machine API**
    - reservationId
- Background Task
  - Every 30 minutes push the next 6 hours of reservations to each IoT device
    - This is done via the **Machine API's Lock Machine method**
  - If the machine cannot be Locked for the given DateTime
    - Pick a new machine
    - Send an email with the updated machine number and new PIN

### Hints

This is a test double kata - focus on testing one action per test when building out the business logic. If you want to ensure all the methods were called in the correct order do so in a single test per business requirement. Considering using a Guid for the reservation ID.

You may also choose to write real implementations for each test double. This will take much longer but will provide a rich real-world experience to the exercise.

### Bonus

Add the ability to reserve both a washing machine and dryer when making a reservation.

How much code can you re-use?

Did you pick generic domain driven names for your classes?