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. This is send to the device at the time of reservation. There is a background task that pushes a master schedule down to every machine every 30 minutes to ensure data integrity.

When a patron makes a reservation they enter the reservation date, time cell phone number and email address. An available machine is selected and the reservation is made; a*ssume 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 code. 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
    - Locks machine via **Machine API**
- Machine API
    - The API should be designed to service a collection of devices using the machine number to locate the correct device
    - Assume this logic wraps a Device SDK given to you with the following interface

```
public interface IMachineDevice{
        bool Lock(DateTime reservationDateTime);
        void Unlock();
}
```

-
    - Lock Machine
        - Takes in machine number and DateTime of reservation
        - Returns true if the machine was unlocked and could be locked at the specified DateTime, else false
            - Do not attempt to lock the machine via the IoT device when reserving. This should be handled by the background task
    - Unlock Machine
        - Takes in machine number
- Claim a Reservation
    - Takes in
        - PIN

        - Machine ID
    - If the PIN matches the active reservation for the machine
        - Updates the reservation in the DB as used
        - Unlocks machine via the **Machine API**
    - 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
- 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**
- Background Task
    - Every 30 minutes push the next 6 hours of reservations to each IoT device
        - This is done via the `IMachineDevice` interface
    - 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 mocking 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.

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?