

# Bravo Programming Challenge - Booking System

## Overview

This solution utilizes the following technologies in order to provide a booking system:

- Kotlin
- Spring Boot
- Spring Shell
- Maven

The system allows a motel admin to configure their inventory via command line interface. Rooms, reservation rules, and pricing rules can all be added on the fly in order to meet a customer's needs.

Once a motel is configured, customers can query and reserve inventory via a command line interface.

The entire system is built on RESTful principles including Uniform Interface, Layered System, Stateless, and Layered System. There is even HATEOAS present via tab completion and the `help` command.

## Setup

A command is available to set up an example customer, complete with a two-story motel, pricing rules, and reservation rules. See the steps below to get the system up and running.

*Note: Once the shell is running, the `help` command can be run to see all available APIs*

1. Run `./mvnw clean install && java -jar target/bravo-programming-challenge-1.0.jar` to build and run the shell application
2. Once you are presented with `shell:>` run `setup-customer-motel`

Now the system is ready to start taking reservations.

## Verifying System Behavior

Once the setup steps are complete, the following operations can be executed to verify the system is working as desired.

### Search With No Amenities

Let's start by searching for rooms without any pets or handicap accessibility.

```
> search-availability --check-in-date "10/22/2018" --check-out-date  
"10/23/2018" --number-of-beds 1
```

You should see the following message:

```
There are 2 rooms available for USD 50.00. See room details below.  
Room(roomNumber=B1, numberOfBeds=1, buildingLevel=2,  
handicapAccessible=false, petFriendly=false)  
Room(roomNumber=B2, numberOfBeds=1, buildingLevel=2,  
handicapAccessible=false, petFriendly=false)
```

Notice that only rooms without amenities are returned. This is because the system will prioritize non-amenity rooms when they are available and amenities are not included in the customer request.

The same query can be executed for 2 and 3 bed units and similar results should appear with updated pricing. Trying an invalid number of beds, like 5, should yield no results.

## Booking With No Amenities

Let's go ahead and make some reservations. The API is very similar to `search-availability`, the only difference being the initial command. Update your instruction to use `make-reservation` instead:

```
> make-reservation --check-in-date "10/22/2018" --check-out-date  
"10/23/2018" --number-of-beds 1
```

A success message should appear:

```
Room successfully reserved. USD 50.00 will be due at the time of arrival.
```

Our availability should be different now. Let's rerun our search query:

```
> search-availability --check-in-date "10/22/2018" --check-out-date  
"10/23/2018" --number-of-beds 1
```

Now only one room is available:

```
There are 1 rooms available for USD 50.00. See room details below.  
Room(roomNumber=B2, numberOfBeds=1, buildingLevel=2,  
handicapAccessible=false, petFriendly=false)
```

Let's go ahead and book that room too to see what happens. The same `make-reservation` command can be run (note: the up arrow will iterate through history inside of the shell).

```
> make-reservation --check-in-date "10/22/2018" --check-out-date  
"10/23/2018" --number-of-beds 1
```

Now, running the same `search-availability` command should yield new results:

```
There are 2 rooms available for USD 50.00. See room details below.  
Room(roomNumber=A1, numberOfBeds=1, buildingLevel=1,  
handicapAccessible=true, petFriendly=true)  
Room(roomNumber=A2, numberOfBeds=1, buildingLevel=1,  
handicapAccessible=true, petFriendly=true)
```

Since only amenity rooms are available, they are now presented to the customer and available to reserve. If you feel like it, go ahead and run the `make-reservation` command 2 more times to book up all of the 1 bed units for the night of the 22nd. When fully booked, both the `make-reservation` and `search-availability` command should return this message:

```
No rooms available that match that criteria.
```

## Booking A Handicap Accessible Room

Booking a handicap accessible room is as easy as adding the `--handicap-accessible` flag. Let's update our dates to get a reservationless time period and run a search:

```
> search-availability --check-in-date "10/23/2018" --check-out-date  
"10/24/2018" --number-of-beds 1 --handicap-accessible
```

As expected, handicap accessible rooms should be returned:

```
There are 2 rooms available for USD 50.00. See room details below.  
Room(roomNumber=A1, numberOfBeds=1, buildingLevel=1,  
handicapAccessible=true, petFriendly=true)  
Room(roomNumber=A2, numberOfBeds=1, buildingLevel=1,  
handicapAccessible=true, petFriendly=true)
```

To book a handicap accessible room:

```
> make-reservation --check-in-date "10/23/2018" --check-out-date  
"10/24/2018" --number-of-beds 2 --handicap-accessible
```

Running our availability search again reveals that there is one less room available:

```
There are 1 rooms available for USD 75.00. See room details below.  
Room(roomNumber=A4, numberOfBeds=2, buildingLevel=1,  
handicapAccessible=true, petFriendly=true)
```

## Booking A Room With Pets

To search or book with pets we will need to utilize the `--number-of-pets` flag. Let's start by updating our dates again and trying a search with 1 pet:

```
> search-availability --check-in-date "10/24/2018" --check-out-date  
"10/25/2018" --number-of-beds 1 --number-of-pets 1
```

The following results should be presented:

```
There are 2 rooms available for USD 70.00. See room details below.  
Room(roomNumber=A1, numberOfBeds=1, buildingLevel=1,  
handicapAccessible=true, petFriendly=true)  
Room(roomNumber=A2, numberOfBeds=1, buildingLevel=1,  
handicapAccessible=true, petFriendly=true)
```

Notice that the price is updated to reflect the \$20 charge for the pet. If the query is updated to 2 pets, that amount should jump to \$90. Increasing the length of stay does not apply additional pet fees.

A booking with pets can be completed by using the same flag:

```
> make-reservation --check-in-date "10/24/2018" --check-out-date  
"10/25/2018" --number-of-beds 1 --number-of-pets 1
```

Returning with the message:

```
Room successfully reserved. USD 70.00 will be due at the time of arrival.
```

## Max Pets Restriction

The maximum pet restriction can be tested by providing a number of pets greater than 2 to either the `make-reservation` or `search-availability` command:

```
> search-availability --check-in-date "10/24/2018" --check-out-date  
"10/26/2018" --number-of-beds 1 --number-of-pets 3
```

The following error should be presented:

```
3 is too many pets. Only 2 are allowed.
```

---

## Exiting Application Shell

To exit the application shell, simply run:

```
> exit
```

## Running Tests

The test suite can be run via the following command outside of the application shell:

```
./mvnw test
```

There are 40 tests in total and 100% line coverage on the service layer where all of the business logic resides.