

Smart Host Coding Challenge

Build a room occupancy optimization tool for one of our hotel clients! Our customer has a certain number of free rooms each night, as well as potential guests that would like to book a room for that night.

Our hotel clients have two different categories of rooms: Premium and Economy. Our hotels want their customers to be satisfied: they will not book a customer willing to pay over EUR 100 for the night into an Economy room. But they will book lower paying customers into Premium rooms if these rooms would be empty and all Economy rooms will be filled by low paying customers. Highest paying customers below EUR 100 will get preference for the “upgrade”. Customers always only have one specific price they are willing to pay for the night.

Please build a small API that provides an interface for hotels to enter the numbers of Premium and Economy rooms that are available for the night and then tells them immediately how many rooms of each category will be occupied and how much money they will make in total. Potential guests are represented by an array of numbers that is their willingness to pay for the night.

Use the following raw json file as mock data for potential guests in your tests (include the downloaded file in the project or get and parse it from the github directly): [gist](#)

Test results you should get:

Test 1

Free Premium rooms: 3

Free Economy rooms: 3

Usage Premium: 3 (EUR 738)

Usage Economy: 3 (EUR 167)

Test 2

Free Premium rooms: 7

Free Economy rooms: 5

Usage Premium: 6 (EUR 1054)

Usage Economy: 4 (EUR 189)

Test 3

Free Premium rooms: 2

Free Economy rooms: 7

Usage Premium: 2 (EUR 583)

Usage Economy: 4 (EUR 189)

Test 4

Free Premium rooms: 10

Free Economy rooms: 1

Usage Premium: 7 (EUR 1153)

Usage Economy: 1 (EUR 45)

Please share the project with us through Github or any other Git based code sharing platform (i.e. Gitlab, Bitbucket etc.).

Requirements for a valid solution:

- Tracking progress through Git commits
- Minimal readme explaining how to run the project and tests
- Tests/TDD (at least use the ones specified above)
- A working algorithm implemented in Java 11 or higher
- Clean code structure and formatting
- Thoughtful naming of variables and functions
- Go for high code quality rather than completeness if you feel time pressure

Nice to have:

- A restful JSON API implemented with Spring Boot or a comparable framework
- Anything else you would like to add

You don't need to do every part of the challenge perfectly. Most of all try to show your strengths. And if you feel unable to finish the project in time or can not fulfill any of the requirements for whatever reason, please let us know before you start working so that we can find another solution.