

Exercise

The code you submit for this exercise is intended as a starting point to help us better understand how you work through engineering problems. There is no preferred solution, but you'll want to be able to walk us through your implementation and explain the technical choices you made.

You can solve this problem using any combination of open-source technologies you'd like. There is no specific requirement for state persistence as part of this exercise, i.e you are not required to use MySQL or a NoSQL database. Demonstrating a solid understanding of object-oriented software development and architecture is more important than the specific languages or technologies used.

When we evaluate your submission we'll be looking at general code quality and awareness of development patterns. Because we're building out a lot of new infrastructure, it's very important to us that candidates write code that is well modularized so it can easily be extended and built upon in the future -- even for an exercise.

This exercise is not intended to take more than a few hours, if you find that you have questions please ask for clarification sooner rather than later. In the ReadMe, you can address what additional steps you would take to improve your code if given unlimited time.

Please use a GitHub repo to share your response with us.

Instructions

- Setup a GitHub repo for this sample application.
- Develop code that meets the application requirements.
- Include a ReadMe.md file that addresses the questions at the bottom of the requirements and includes any thoughts on additional work you would like to do to this, if you had unlimited time.
- Read all of the directions in this requirements carefully and please make sure you make commits at regular logical intervals.
- When you're done the exercise please email us a link to the repo.
- Have fun.

Scenario

Hi and welcome to team Gilded Rose! As you know, we are a small, magical inn with a prime location in a prominent city run by a friendly innkeeper named Allison. She insists on running a top notch inn, and keeping 5 star reviews is crucial to her business. As such, she doesn't want to overbook her rooms, or have rooms booked too soon after they have been vacated. The cleaning gnomes won't have time to keep everything tidy if the turnover rate is too quick. Allison requests that you build a version one of a room booking system for her inn. Since the system is a v1, you don't have to implement everything that would be needed for an actual system in this exercise -- you only have to do what we ask for in the specification.

Some background on the business....

The inn has **FOUR** rooms

- **ONE** room sleeps **TWO** people and has **ONE** storage space
- **ONE** room sleeps **TWO** people and has **ZERO** storage space
- **ONE** room has **TWO** storage spaces and sleeps **ONE** person
- **ONE** room sleeps **ONE** person and has **ZERO** storage space

Cleaning

- All rooms must be cleaned after being occupied and prior to being rented again.
- The gnomes cleaning squad needs **ONE** hour per room plus **THIRTY** minutes per person in the room to clean it.
- There is only **ONE** gnome cleaning squad, but we might hire more.
- Legally the gnomes are only allowed to work **EIGHT** hours per day, but we're always lobbying the local lord to let the gnomes work more. (We're really sure they want to)
- The gnomes start time is flexible but must be contiguous.

Guests can not store their luggage in another guest's room.

Being that this is a shared space inn, guests might be in shared rooms if that's the most profitable solution.

The cost is calculated per person according to this formula:

- $(\text{Base room cost} / \text{number of people in the room}) + (\text{base storage costs} * \text{number of items stored})$.
- Base room cost is 10 Gold, storage cost is 2 Gold.

Specification:

Your application needs to:

- Expose a RESTful API which
 - Has an endpoint for listing room availability
 - Based on guests bookings, luggage storage requirements and cleaning availability
 - Has an endpoint for booking rooms
 - Based on guests bookings, luggage storage requirements and cleaning availability this should return a valid room number, to confirm the booking or an error if the room is not bookable for some reason.
 - Has an endpoint for management to obtain the required gnome schedule for each day.
- Maximize the number of guests able to stay in the inn for each night
 - The application can not reassign guest rooms after a guest has been confirmed.

Your application does not need to:

- The application does not need a user interface, it only needs to accept input over the API.
- Does not need to be highly performant, i.e don't be concerned about fast storage.

ReadMe.md Requirements

The ReadMe should be a Markdown file. Your ReadMe should include:

- Information on how to setup our environments to run your application.
- Information on how to run your application.

Additionally, your ReadMe should answer the following questions:

- At a high level, how does your system work?
- How would we extend your system if we had to add more rooms, more business logic constraints, more gnomes?
- What documentation, websites, papers, etc. did you consult for this assignment?

- What third-party libraries or other tools does your application use? How did you choose each library or framework you used?
- How long did you spend on this exercise? If you had unlimited time to spend on this, how would you spend it and how would you prioritize each item?
- If you were going to implement a level of automated testing to prepare this for a production environment, how would you go about doing so?