Glofox is a saas platform for boutiques, studios, and gyms which allows business owners to manage their courses, classes, members, memberships etc,. There are a couple of stories outlined below which you have to implement.

Story - Create Classes

As a studio owner i want to create classes for my studio so that my members can attend classes

Acceptance Criteria

- Implement an API to create classes(`/classes`). Assume this api doesn't need to have any authentication to start with.
- Few bare minimum details we need to create classes are - class name, start_date, end_date, capacity. For now, assume that there will be only one class per given day. Ex: If a class by name pilates starts on 1st Dec and ends on 20th Dec, with capacity 10, that means Pilates has 20 classes and for each class the maximum capacity of attendance is 10.
- No need to save the details in any database. Maintain an in memory array or a file to save the info. (If you want to use the database, that's fine as well).
- Use Restful standards and create the api endpoint with proper success and error responses.

Story - Book for a class

As a member of a studio, I can book for a class, so that I can attend a class.

Acceptance Criteria

- Implement an API endpoint (`/bookings`). Assume this api doesn't need to have any authentication to start with.
- Few bare minimum details we need for reserving a class are - name(name of the member who is booking the class), date(date for which the member want to book a class)
- No need to save the details in DB. If you can maintain the state in an in memory array or a file is good to start with. But no constraints if you want to use a database to save the state.
- Use REST standards and think through basic api responses for success and failure.
- No need to consider the scenario of overbooking for a given date. Ex: 14th Dec having a capacity of 20 , but the number of bookings can be greater than 20.

What we look for

Just because you take an easy approach won't set you back in our eyes. Easy might just mean you're busy - we understand. Either way, we look at the same areas:

- Approach taken
- Architecture
- Code structure and clarity
- Performance
- Testing
- Include a readme file if there is setup required for your code to run, or if there is something you would like to comment on.

You can pass on the finished code either as a zip file or link us to any of hosted VCS like github/bitbucket/gitlab etc,.