

Senior Back-end Engineer Test Project

Create a simple Blog Post application that allows commenting. Comments can also be commented but up to the 3rd layer only.

Requirements:

- Create the comment system REST API with Laravel 5.5+ with MySQL as the database.
- Do NOT use Laravel's Eloquent ORM, you are only allowed to use the query builder or raw SQL.
- A comment has only two required fields: 1) commenter's name and 2) the comment message.
- Sub-comments are up to the 3rd layer only (3 layers from the root comment).
- The API must be able to create, update, delete comments.
- You can assume that the application has only one Blog Post and there is no need for user authentication.
- An API end-point to list a Blog Post's comments and its sub-comments (comment tree).

Comments example output:

```
[
  {
    "id": 654,
    "name": "John Smith",
    "message": "Hi Jane, please read this post.",
    "comments": [
      {
        "id": 655,
        "name": "Jane Johnson",
        "message": "John, thanks for mentioning me here.",
        "comments": [
          {
            "id": 657,
            "name": "Jon Snow",
            "message": "I know nothing about this.",
            "comments": []
          },
          {
            "id": 658,
            "name": "Oberyn Martel",
```

```

        "message": "There is no justice here in the capital.",
        "comments": []
    }
]
},
{
    "id": 656,
    "name": "Arya Stark",
    "message": "I like this post too.",
    "comments": []
}
]
},
{
    "id": 790,
    "name": "Sandor Clegane",
    "message": "I understand that if any more words come pouring out your
mouth, I'm gonna have to eat every chicken in this room.",
    "comments": []
}
]

```

What we are looking for:

- Clean Code following PSR Standards
- Organized business logic
- Optimized SQL
- Test-driven development and Unit Tests
- Object-oriented design
- Software design patterns

Important notes:

- Start an empty repository in GitHub, commit the base Laravel to the master branch and then start work. (so when we read your code, we can separate it from Laravel base code).
- Don't make the project complicated; there is no need for user login, live broadcasting, and notifications.
- You don't need to create a User and Post model.
- You only have 24 hours to finish this project, and time begins upon receiving the email (but if you need more time, please don't hesitate to negotiate).
- If you have questions, please feel free to ask.

