

# DayTwo Backend Home Assignment

Thank you for your interest in joining DayTwo!

Shield's Superhero Database:

## Background Story:

Congratulations! You have been recruited by S.H.I.E.L.D. (Strategic Homeland Intervention, Enforcement, and Logistics Division) to assist in the development of a cutting-edge microservice that will serve as the backbone of our superhero database. As you may know, our world is filled with extraordinary individuals with incredible abilities, and it's our duty to keep track of them all.

Your mission, should you choose to accept it, is to create a microservice with crucial endpoints: including adding new superheroes to our database, retrieving information about them, setting a communication channel between them and receiving information about that communication.

## Part 1: Creating the Microservice

Your microservice should be developed using either JavaScript (Node.js) or TypeScript, whichever you're most comfortable with. Here are the details of the endpoints:

1. **POST /superheroes** This endpoint should allow users to add new superheroes to our database. The request body should contain JSON data representing the superhero's information, including their name, alias, powers, and any additional details you deem necessary.
2. **GET /superheroes/:id** This endpoint should retrieve information about a specific superhero based on their unique identifier (ID). Users should be able to provide the ID of the superhero they're interested in, and the microservice should respond with the corresponding superhero's details.

## Part 2: Set Timer Endpoint

Now, let's create a timer that'll trigger the communication between superheroes and the base. Here are the details of the endpoint:

1. **POST /timers** This endpoint should allow the superhero to send a message to a url after X amount of time.
  - a. The request body should contain JSON data including hours, minutes, seconds, message, url and the superhero UUID.
  - b. The response should include the message id and time left until it will be triggered.
  - c. If the superhero UUID isn't valid, return an error message.

- d. After the timer ends, trigger a POST request to the url with the timer id, the message and the superhero name in the body.
2. **GET /timers/:id** This endpoint should retrieve information about a specific timer based on their id. The response should return the id and time left (amount of seconds until the timer triggers).
  - a. If the timer already expired, return 0 in the time left.
  - b. The trigger timer should not be canceled by process restarts. Any timer that should be triggered in the time the server was down should be triggered once it back up.
  - c. In case of horizontal scalability the timer must be triggered only once.

**Additional Guidelines:**

- Ensure that your code is well-structured, easy to read, and follows best practices for JavaScript/TypeScript development.
- Implement error handling to gracefully manage potential issues such as invalid requests or database connectivity problems.
- Feel free to add any additional features or improvements to enhance the functionality of the microservice.
- Document your code thoroughly, including instructions on how to run the microservice locally and interact with its endpoints.

**Submission:**

Once you've completed the assignment, submit your code along with any relevant documentation or instructions. We're excited to see your superhero database in action! Good luck!