

Problem-2

Task description

Your task is to implement a service with the ability to communicate through Socket.IO.

Requirements

Using Socket.IO, implement the following behaviour:

- On a connection event, the newly connected client should receive an event message with the payload:

```
{
  "message": "Hello guest!"
}
```
- On a set-name event with the following payload:

```
{
  "name": "John"
}
```

 - the name is bound to the socket for later use.
- On a connection event, every other connected socket should receive an event message with the payload:

```
{
  "message": "Say hello to the new guest!"
}
```
- On a disconnection of socket event, every other connected socket should receive an event message with the payload:

```
{
  "message": "John has left the place!"
}
```

 - or, if the socket has not set its name with a set-name event:

```
{
  "message": "Guest has left the place!"
}
```
 - where "John" is equal to the name property assigned to the socket.

- When one of the connected clients sends a message, an event of type message should be emitted to every connected client except for the sender, with the message content prefixed by the name and a colon.
 - When user "John" emits an event with the following payload:


```
{
  "message": "What time is it?"
}
```
 - the emitted message to everyone except for the sender will be:


```
{
  "message": "John: What time is it?"
}
```
 - or (in the case of no name being set):


```
{
  "message": "Guest: What time is it?"
}
```
- When emitting a direct-message event with the following data passed in the payload:


```
{
  "to": "Jason",
  "message": "How are you?"
}
```

 - where to contains the name of the receiver of the message and message contains any text that is to be sent to the receiver, the message from the message field should be sent to the user whose name is equal to the value of the to property. The message text should be sent in the message field. It should be emitted as a message event with payload:


```
{
  "message": "How are you?"
}
```
 - If there is no user with the name from the to field then the following message event should be sent to the socket that emitted the event:


```
{
  "message": "No such user!"
}
```
- The case where there are multiple users with the same name does not have to be handled.

Hints

- The dependencies used in this task are in the following versions:

```
{  
  "dependencies": {  
    "express": "4.17.1",  
    "socket.io": "2.3.0"  
  }  
}
```