



Ejercicios: Express

Descripción

Modifique el ejercicio del chat <https://github.com/isis3710-uniandes/tutoriales/wiki/%5BNode.js%5D-WebSockets>:

1. Establezca un mecanismo de persistencia basado en archivos JSON almacenados en el servidor para guardar los mensajes enviados al chat.

2. Incluya los siguientes endpoints:

GET <http://localhost:3000/chat/api/messages>

Obtiene todos los mensajes que se han enviado al chat

GET <http://localhost:3000/chat/api/messages/1599868276352>

Obtiene el mensaje con el ts especificado

POST <http://localhost:3000/chat/api/messages>

Body

```
{
  "message": "New message",
  "author": "Juan José Rodríguez"
  "ts": "1599868276352"
}
```

Envía un nuevo mensaje al chat

PUT <http://localhost:3000/chat/api/messages>

Body

```
{
  "message": "Message updated",
  "author": "Juan José Rodríguez"
  "ts": "1599868276352"
}
```

Actualiza un mensaje

DELETE <http://localhost:3000/chat/api/messages/1599868276352>

Borra el mensaje con el ts especificado

3. Cuando se envíe un mensaje por el API, la aplicación desplegada en el browser debe actualizarse automáticamente.
4. Enviar los códigos de error al cliente cuando se ingresa, por ejemplo, un ts que no existe en la consulta, actualización o borrado de un mensaje.
5. Con la librería Joi, valide los campos en la creación y actualización así:

Todos son requeridos

El mensaje no puede tener menos de 5 caracteres

El autor debe tener un nombre y un apellido separados por un espacio