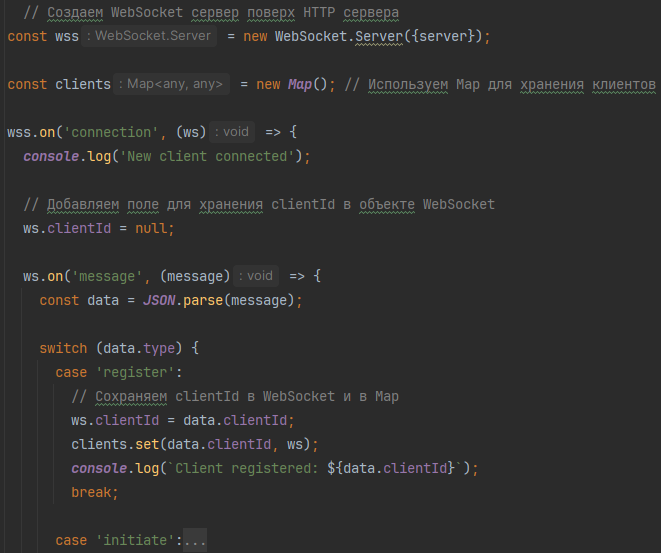
Не понятно это: wss.on('connection', (ws) => { console.log('New client connected'); ws.on('message', (message) => { try { Тут, у тебя wss.on('message', находится внутри wss.on('connection', Странная вложенность события в событие.

**ChatGPT**

Да, вложенность событий в данном случае абсолютно логична. Давайте разберёмся:

1. **wss.on('connection', ...)** — это обработчик события, которое возникает, когда новый клиент подключается к WebSocket-серверу. Каждый клиент имеет свой собственный объект WebSocket (в данном случае это ws), который предоставляет методы для взаимодействия с этим конкретным клиентом.
2. **ws.on('message', ...)** — это обработчик события для конкретного подключенного клиента. Он реагирует на сообщения, которые отправляет клиент.

Именно поэтому **ws.on('message', ...)** должен находиться внутри **`wss.on('connection', ...)**, поскольку он относится к индивидуальному клиенту, который только что подключился.



**Почему так сделано?**

* **wss.on('connection', ...)** срабатывает при каждом новом подключении клиента и создаёт индивидуальное пространство обработки (**ws**).
* Вложенное **ws.on('message', ...)** гарантирует, что каждое сообщение обрабатывается только в контексте конкретного клиента.

Если бы **ws.on('message', ...)** находилось вне **`wss.on('connection', ...)**, то вы бы пытались обрабатывать события **всех клиентов** сразу, что не соответствует архитектуре WebSocket-соединений.