

CollaborationGateway

September 4, 2026

CollaborationGateway

Kind: Class

Source: [atloria-monorepo/apps/api/src/collaboration/collaboration.gateway.ts](#)

Collaboration WebSocket Gateway for real-time document editing

Uses Yjs (CRDT) for automatic conflict resolution

Authentication: the handshake is verified with the same JWT the REST API uses (handshake.auth.token or an Authorization: Bearer header). Sockets that fail verification are disconnected, and the verified `sub` claim is the only source of user identity — event payload userIds are ignored.

Crash recovery: every applied Yjs update is appended to a Redis log and a snapshot is persisted on each save, so a process crash loses at most the updates that never reached Redis (instead of up to 30s of edits).

Implements: `OnGatewayInit`, `OnGatewayConnection`, `OnGatewayDisconnect`, `OnModuleDestroy`

Methods

Method	Signature	Returns
<code>afterInit</code>	<code>afterInit()</code>	<code>void</code>
<code>onModuleDestroy</code>	<code>onModuleDestroy()</code>	<code>Promise<void></code>
<code>handleConnection</code>	<code>handleConnection(client: Socket)</code>	<code>void</code>
<code>handleDisconnect</code>	<code>handleDisconnect(client: Socket)</code>	<code>void</code>
<code>handleJoinDocument</code>	<code>handleJoinDocument(data: JoinDocumentData, client: Socket)</code>	<code>Promise<DocumentStateResponse></code>

Method	Signature	Returns
handleUpdate	handleUpdate(data: DocumentUpdateData, client: Socket)	void
handleSyncStep1	handleSyncStep1(data: { documentId: string; stateVector: number[] }, _client: Socket)	{ update: number[] }
handleSyncStep2	handleSyncStep2(data: { documentId: string; update: number[] }, _client: Socket)	void
handleAwarenessUpdate	handleAwarenessUpdate(data: AwarenessUpdateData, client: Socket)	void
handleCursorUpdate	handleCursorUpdate(data: { documentId: string; cursor: { index: number; length: number } }, client: Socket)	void
handleLeaveDocument	handleLeaveDocument(data: LeaveDocumentData, client: Socket)	{ success: boolean }
getDocument	getDocument(documentId: string)	`Y.Doc
getActiveUsers	getActiveUsers(documentId: string)	CollaborationUser[]
getActiveDocumentCount	getActiveDocumentCount()	number
forceSave	forceSave(documentId: string)	Promise<boolean>

Properties

Property	Type
server	Server

Where it refuses work

- CollaborationGateway stops the work with WsException when !ydoc , in 3 places.

- `CollaborationGateway` stops the work with `WsException` when `!userId` — “Unauthenticated socket”.
- `CollaborationGateway` stops the work with `WsException` when `!documentId || !userName` — “Invalid join-document data: documentId and userName are required”.
- `CollaborationGateway` stops the work with `WsException` when `!hasAccess` — “Access denied to this document”.
- `CollaborationGateway` stops the work with `WsException` when `!ydoc || !awareness` — “Failed to initialize document”.
- `CollaborationGateway` stops the work with `WsException` when `!documentId || !update` — “Invalid update data: documentId and update are required”.

When something fails

- `CollaborationGateway` handles failure in 10 places: it lets it reach the caller in 5, logs it and continues in 3, and turns it into a return value in 2.

Referenced By

- `CollaborationModule` (MODULE_PROVIDES)