websockets

WebSockets Study Guide

What is a WebSocket?

A WebSocket is:

a persistent real-time communication connection between two systems.

Unlike normal REST APIs:

- the connection stays open
- both systems can send data anytime

Simple Explanation

REST API

Client asks for information repeatedly.

Example:

```
"Any new messages?"
"Any new messages?"
"Any new messages?"
```

WebSocket

Connection stays open continuously.

Server pushes updates instantly:

```
"New message received"
```

without client repeatedly asking.

Why WebSockets Exist

WebSockets enable:

real-time communication

Examples:

- live chat
- agent status updates
- notifications
- voice/video signaling
- stock market feeds
- multiplayer gaming

Common WebSocket Use Cases

Use Case	Example
Live Chat	Customer-agent messaging
Agent Presence	Agent online/offline updates
Notifications	Real-time alerts
Voice/Video Apps	Session signaling
Monitoring Dashboards	Live metrics
Collaboration Tools	Real-time updates

How WebSockets Work

Step 1 — Client Connects

Example:

```
Browser → WebSocket Server
```

Step 2 — Connection Upgraded

Initial connection starts using HTTP/HTTPS.

Then upgraded to:

WebSocket protocol

Step 3 — Persistent Connection Established

Connection remains:

continuously open

Both sides can send data anytime.

WebSocket Communication Flow

```
Client ↔ WebSocket Server
```

Bidirectional communication.

REST API vs WebSocket

REST API	WebSocket
Request/response	Persistent connection
Stateless	Stateful

REST API	WebSocket
Client initiates requests	Both sides send data
Polling often required	Real-time updates
Short-lived connection	Long-lived connection

Simple Analogy

REST API

Like sending emails:

- request
- response
- connection closes

WebSocket

Like phone call:

- connection stays active
- both sides talk anytime

Example WebSocket Flow

```
Customer sends message
        ↓
Agent instantly receives message
        ↓
Agent replies immediately
        ↓
```

```
Customer sees reply in real time
```

No repeated API polling needed.

WebSocket URLs

Instead of:

```
https://
```

WebSockets use:

```
ws://
```

Secure WebSockets:

```
wss://
```

Secure WebSockets

WSS = WebSocket Secure

Equivalent of:

HTTPS for WebSockets

Uses:

- TLS encryption
- secure communication

VERY important in:

- banking
- enterprise platforms
- customer interactions

Why WebSockets Matter In Glia

Glia heavily relies on:

- real-time chat
- live agent updates
- voice interactions
- customer engagement

WebSockets are commonly used for:

- live interaction synchronization
- agent presence
- messaging
- session updates

WebSockets vs Webhooks

WebSocket	Webhook
Persistent connection	One-time event notification
Real-time bidirectional	Event-based push
Both sides communicate	Source system pushes
Continuous session	Stateless notification

WebSockets vs REST APIs

REST API	WebSocket
Request/response	Continuous communication
Polling required for updates	Instant updates
Better for CRUD operations	Better for real-time apps

CRUD:

- Create
- Read
- Update
- Delete

Common WebSocket Troubleshooting

Problem 1 — Connection Fails

Possible causes:

- firewall blocking
- proxy issues
- incorrect endpoint
- TLS/certificate issue

Problem 2 — Connection Drops

Possible causes:

- timeout
- network instability
- load balancer session timeout

Problem 3 — TLS/WSS Failure

Symptoms:

- secure connection rejected
- certificate errors

Check:

- TLS certificates
- HTTPS/WSS configuration

Problem 4 — Authentication Failure

Some WebSockets require:

- bearer token
- session authentication
- OAuth validation

Result: connection rejected.

Problem 5 — High Latency

Symptoms:

- delayed chat messages
- slow updates

Possible causes:

- network congestion
- overloaded server
- scaling issues

WebSocket Troubleshooting Flow

Step 1 — Validate Endpoint

Check:

```
ws://
```

or:

```
wss://
```

Step 2 — Validate Connectivity

Check:

- firewall
- proxy

- load balancer
- port access

Step 3 — Validate Authentication

Check:

- bearer token
- OAuth session
- API credentials

Step 4 — Validate TLS/WSS

Check:

- certificate validity
- TLS compatibility
- HTTPS/WSS support

Step 5 — Review Logs

Check:

- WebSocket handshake logs
- disconnect reasons
- timeout events
- session logs

Common Interview Questions

“What is a WebSocket?”

Good Answer:

> “A WebSocket is a persistent bidirectional communication protocol that enables real-time data exchange between systems over a continuously open connection.”

“Difference between REST API and WebSocket?”

Good Answer:

> “REST APIs use a request/response model with short-lived connections, while WebSockets maintain a persistent connection that supports real-time bidirectional communication.”

“Why use WebSockets?”

Good Answer:

>

> “WebSockets are ideal for real-time applications such as live chat, notifications, voice signaling, and agent presence updates because they eliminate the need for continuous polling.”

“Difference between WebSocket and webhook?”

Good Answer:

> “WebSockets maintain a persistent real-time communication session, while webhooks are one-time event notifications triggered when specific events occur.”

“How would you troubleshoot WebSocket issues?”

Good Answer:

> “I would validate endpoint connectivity, verify firewall and proxy access, confirm authentication and TLS configuration, review handshake and disconnect logs, and isolate any network or session timeout issues.”

Important Security Concepts

Secure WebSockets should use:

WSS (WebSocket Secure)

This provides:

- TLS encryption
- secure communication
- token protection

Easy Memory Trick

REST API = Request/Response

Webhook = Event Notification

WebSocket = Live Continuous Conversation

Important Terms To Know

Term	Meaning
WebSocket	Persistent real-time connection
WSS	Secure WebSocket
Bidirectional	Both sides communicate
Persistent Connection	Connection remains open
Polling	Repeated API requests
Real-Time	Instant communication
Handshake	Initial connection setup
Stateful	Maintains session state

Revision #1
Created 21 May 2026 00:48:53 by Cesar Gzz
Updated 21 May 2026 00:48:58 by Cesar Gzz