

Zerodha Error:

```
ZERODHA_SOCKET_ERROR {
  userId: 'f295974c-14b6-4fe0-8799-56f769110c6d',
  message: ErrorEvent {
    target: WebSocket {
      _events: [Object: null prototype],
      _eventsCount: 4,
      _maxListeners: undefined,
      readyState: 2,
      protocol: "",
      _binaryType: 'arraybuffer',
      _closeFrameReceived: false,
      _closeFrameSent: false,
      _closeMessage: "",
      _closeTimer: null,
      _closeCode: 1006,
      _extensions: {},
      _receiver: null,
      _sender: null,
      _socket: null,
      _isServer: false,
      _redirects: 0,
      url: 'wss://ws.kite.trade/?api_key=...&access_token=...&uid=1767337640990',
      _req: [ClientRequest],
      [Symbol(shapeMode)]: false,
      [Symbol(kCapture)]: false
    },
    type: 'error',
    message: 'Unexpected server response: 403',
    error: Error: Unexpected server response: 403
      at ClientRequest.<anonymous>
(C:\Users\ADMIN\OneDrive\Documents\Files\VisionX\BullTrek\node_modules\kiteconnect\node_modules\ws\lib\websocket.js:595:7)
      at ClientRequest.emit (node:events:519:28)
      at ClientRequest.emit (node:domain:489:12)
      at HTTPParser.parserOnIncomingClient [as onIncoming] (node:_http_client:716:27)
      at HTTPParser.parserOnHeadersComplete (node:_http_common:117:17)
      at TLSSocket.socketOnData (node:_http_client:558:22)
      at TLSSocket.emit (node:events:519:28)
      at TLSSocket.emit (node:domain:489:12)
      at addChunk (node:internal/streams/readable:561:12)
      at readableAddChunkPushByteMode (node:internal/streams/readable:512:3)
    }
  }
}
```

Code:

```
import { KiteTicker } from "kiteconnect";
import { StocksExchange } from "@prisma/client";
import {
  handleZerodhaOrderUpdate,
  handleZerodhaTradeUpdate,
} from "../../services/stocks/exchangeSocketServices/zerodhaSocketService";
import { logEvent } from "../../utils";
import { SocketManager } from "../../socketManagement";

type ZerodhaCredentials = {
  apiKey: string;
  accessToken: string;
};

export const ZerodhaOrderHandler = {
  connect(userId: string, credentials: ZerodhaCredentials): any {
    const { apiKey, accessToken } = credentials;

    const ticker = new KiteTicker({
      api_key: apiKey,
      access_token: accessToken,
    });

    ticker.on("connect", () => {
      logEvent("CONNECTED", {
        userId,
        exchange: StocksExchange.ZERODHA,
        market: "ORDERS",
      });
    });

    ticker.on("order_update", (order) => {
      console.log("ZERODHA_ORDER_UPDATE", { userId, order });

      handleZerodhaOrderUpdate(order, userId);
    });

    // ticker.on("trade", (trade) => {
    //   console.log("ZERODHA_TRADE_UPDATE", { userId, trade });

    //   handleZerodhaTradeUpdate(trade, userId);
    // });

    ticker.on("reconnect", (attempt, delay) => {
      console.log("ZERODHA_RECONNECTING", { userId, attempt, delay });
    });
  }
};
```

```

    ticker.on("noreconnect", () => {
        console.log("ZERODHA_NO_RECONNECT", { userId });
        SocketManager.removeSocket(userId, StocksExchange.ZERODHA, "ORDERS");
    });

    ticker.on("disconnect", () => {
        console.log("ZERODHA_DISCONNECTED", { userId });
    });

    ticker.on("error", (err) => {
        console.log("ZERODHA_SOCKET_ERROR", {
            userId,
            message: err?.message || err,
        });
    });

    ticker.on("close", () => {
        logEvent("CLOSED", {
            userId,
            exchange: StocksExchange.ZERODHA,
            market: "ORDERS",
        });
    });

    ticker.connect();

    SocketManager.registerSocket(
        userId,
        StocksExchange.ZERODHA,
        "ORDERS",
        ticker,
        "stock"
    );

    return ticker;
},
};

```