

The simplest way to Connect CLAUDE to TradingView

Here is our one prompt that can connect claude to trading view and help you understand trading

Set up TradingView MCP for me from scratch. Follow these steps exactly:

****Step 1 – Check Prerequisites:****

- Verify Node.js 18+ is installed. Check `/usr/local/bin/node` and common paths. If not found, tell me to install it.
- Verify git is installed.
- Check if `/usr/local/bin` is in my PATH. If not, add `export PATH="/usr/local/bin:$PATH"` to my `~/.zshrc`.

****Step 2 – Clone & Install:****

- Clone `https://github.com/tradesdontlie/tradingview-mcp.git` to `~/tradingview-mcp`
- Run `npm install` in that directory

****Step 3 – Configure MCP:****

- Find my current working directory
- Create `.mcp.json` in my current working directory with this config:

```
```json
{
 "mcpServers": {
 "tradingview": {
 "command": "/usr/local/bin/node",
 "args": ["/Users/YOUR_USERNAME/tradingview-mcp/src/server.js"],
 "env": {
 "PATH": "/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin"
 }
 }
 }
}
```
```

Replace YOUR_USERNAME with my actual macOS username (get it from `whoami`).

****Step 4 – Create rules.json:****

Create `~/tradingview-mcp/rules.json` with:

```
```json
{
 "watchlist": ["BTCUSD", "ETHUSD", "SOLUSD", "AAPL", "TSLA", "NVDA"],
 "timeframes": ["1W", "1D", "4H"],
 "biasCriteria": {
 "bullish": [
 "RSI(14) above 50 and rising",
 "MACD histogram positive and increasing",

```

```

 "Price above 21 EMA and 50 EMA",
 "21 EMA above 50 EMA (golden alignment)"
],
 "bearish": [
 "RSI(14) below 50 and falling",
 "MACD histogram negative and decreasing",
 "Price below 21 EMA and 50 EMA",
 "21 EMA below 50 EMA (death alignment)"
],
 "neutral": [
 "Mixed signals across indicators",
 "Price choppy between 21 and 50 EMA",
 "RSI between 45-55 with no clear trend"
]
},
"riskRules": {
 "maxPositionSize": "2% of portfolio per trade",
 "stopLossRequired": true,
 "riskRewardMinimum": 2.0,
 "preSessionChecklist": [
 "Check for major news events",
 "Verify no earnings announcements today",
 "Check overall market sentiment (SPX/VIX)"
]
},
"twelveHrUpdate": {
 "enabled": true,
 "scheduleHours": [9, 21],
 "reportFormat": "structured",
 "includeKeyLevels": true,
 "includeVolumeAnalysis": true,
 "saveHistory": true,
 "historyDir": "~/tradingview-mcp/sessions"
}
}
```

```

****Step 5 – Add 12 Hr Update instructions to CLAUDE.md:****
Append the following to `~/tradingview-mcp/CLAUDE.md`:

```

```
12 Hr Update – Custom Watchlist Scanner

When the user asks for a "12 hr update" or "twelve hour update", follow this workflow:

Step-by-Step
1. Read `rules.json` in the project root to get the watchlist symbols,
timeframes, and bias criteria

```

2. For each symbol in `watchlist`:
  - a. `chart\_set\_symbol` to switch to that symbol
  - b. For each timeframe in `timeframes` (e.g., "W", "D", "240"):
    - `chart\_set\_timeframe` to switch
    - `quote\_get` for current price
    - `data\_get\_study\_values` for all indicator readings
    - `data\_get\_ohlcv` with `summary: true` for price stats
    - `data\_get\_pine\_lines` for support/resistance levels
  - c. Apply `biasCriteria` from rules.json to determine: Bullish / Bearish / Neutral
3. Compile all results into a structured report
4. Save to `~/tradingview-mcp/sessions/` as `YYYY-MM-DD\_HH-mm.json`
5. Display the formatted report to the user

### ### Report Format

Each asset should show: Symbol | Bias | Price | Key Level | Per-Timeframe Breakdown | Indicator Values | Watch Points

### ### Summary Section

End with a summary grouping assets by Bullish/Bearish/Neutral, plus any risk checklist items from `rules.json.riskRules`

```

Step 6 – Create sessions directory:

```
```bash
mkdir -p ~/tradingview-mcp/sessions
```
```

Step 7 – Install TradingView Desktop (if not already installed):

- Check if TradingView.app exists in /Applications
- If not, tell me to download it from <https://www.tradingview.com/desktop/> and install it

Step 8 – Launch TradingView with CDP:

- Kill any existing TradingView process
- Launch with: `open -a "TradingView" --args --remote-debugging-port=9222`
- Wait for CDP to be ready by polling `http://localhost:9222/json/version`
- Confirm CDP is live

Step 9 – Set up 12 Hr Update cron:

Create a recurring cron job at 9:03 AM and 9:03 PM that runs the 12 hr update – scanning all watchlist assets and delivering a structured report.

Step 10 – Verify:

- Run `tv\_health\_check` via CLI to confirm connection
- Tell me to restart Claude Code so the MCP tools load natively
- After restart, I should be able to say "Run tv_health_check" and it works

When done, show me a summary of everything that was set up and how to use it.

