Rice Stock Data MCP Server

Setup Guide for Claude Desktop Integration

Kerry Back

2025-09-14

This is a Model Context Protocol (MCP) server for accessing the Rice Business Stock Market Data Portal through Claude Desktop. Its features include:

- Natural language queries for stock market data
- Access to comprehensive financial metrics and indicators
- Rice University email verification for access control
- Seamless integration with Claude Desktop

Step 1: Install Node.js

- Visit nodejs.org and install the LTS version

- Optional: Verify installation by opening Terminal/Command Prompt and running:

```
node --version
npm --version
```

Both commands should return version numbers.

Step 2: Get Your Access Token

1. Visit the [Rice Business Stock Market Data Portal](#) to request an access token (if you do not already have one)
2. Verify your Rice University email address
3. Copy your personal access token

Step 3: Configure Claude Desktop

1. Open Claude Desktop
2. Go to **Settings → Developer → Edit Config** This should open File Explorer (on Windows) at the `claude_desktop_config.json` file. Right-click on it and select **Edit with Notepad**.
3. Copy and past the following into the file (if you already have a `mcpServers` section, add the `rice-stock-data` section to it):

```
{
  "mcpServers": {
    "rice-stock-data": {
      "command": "npx",
      "args": ["@kerryback/rice-stock-data@1.0.7"],
      "env": {
        "USER_ACCESS_TOKEN": "your_actual_access_token_here",
        "APP_URL": "https://data-portal.rice-business.org"
      }
    }
  }
}
```

4. Replace `"your_actual_access_token_here"` with your actual access token from Step 1 (keep the quotation marks)
5. Save the configuration file
6. Restart Claude Desktop completely. **Closing Claude Desktop does not restart it.** Instead, find the Claude Desktop icon in the system tray. Right-click and choose Quit.

Step 4: Using the MCP Server

Once configured, you can ask Claude questions about stock market data. For example, "Using the Rice stock data portal, …"

- "Get the history of Apple's adjusted closing price beginning in 2020."
- "Get all insider trades for Tesla for the past two years." "
- "Find the current median PE ratio of mid-cap oil and gas stocks." "

- "Get the history of DuPont ratios for all biotech stocks for the past five years."
- "Get Tesla's ROE for each of the past 12 quarters on a trailing 4-quarters basis."

Table Descriptions

Detailed descriptions of the database tables and the variables in each table can be found here. Here are some general issues to be aware of:

- To filter on a categorical variable, the exact values of the categorical variable must be determined. It can help to ask Claude: "list the possible values of variable x in table y".
- The SEP table contains three types of prices. The `open`, `high`, `low`, `close` and `volume` variables are split adjusted. The `closeunadj` variable is not split adjusted. The `closeadj` variable is both split and dividend adjusted (like yfinance), so total returns including dividends can be computed (to a sufficient degree of precision) as percent changes in `closeadj`.
- Financial statement data is available in the SF1 table either as reported or including restatements. Data from both 10Ks and 10Qs is available. As reported 10K data is selected by filtering on `dimension='ARY'`. As reported 10Q data is selected by filtering on `dimension='ARQ'`. The most recent data including restatements is selected by filtering on `dimension='MRY'` or `dimension='MRQ'`. Data is also available quarter by quarter on a trailing four quarters basis as `dimension='ART'` or `dimension='MRT'`.
- The `reportperiod` variable in the SF1 table is the end of the fiscal period. The `datekey` variable is the date at which the statement was filed (for `ARx` data). There are also `fiscalperiod` and `calendardate` variables.