

直接回答

以下是如何使用 Node.js 連接到 OpenRouter 進行虛擬貨幣技術分析的程式範例與教學指南。OpenRouter 提供 AI 模型的統一 API，因此需要從其他來源（如 CoinGecko）獲取虛擬貨幣數據，然後使用 OpenRouter 的 AI 模型進行分析。以下是簡化的步驟與說明，適合初學者理解。

關鍵要點

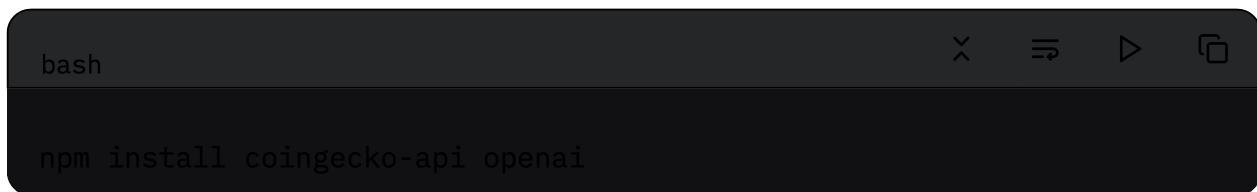
- **解決方案**：使用 CoinGecko API 獲取虛擬貨幣歷史價格數據，並通過 OpenRouter 的 AI 模型進行技術分析。
- **不確定性**：OpenRouter 的 AI 模型（如大型語言模型）並非專為技術分析設計，結果可能不如傳統技術指標精確，但可提供自然語言的市場洞察。
- **爭議**：AI 模型在金融分析中的可靠性存在討論，建議將其用於輔助決策，而非唯一依據。
- **簡單易用**：Node.js 程式碼使用現成的庫，易於設置和運行。

設置環境

¹ **註冊 OpenRouter**：在 OpenRouter 網站 創建帳戶並獲取 API 金鑰。

² **安裝 Node.js 套件**：

- 安裝 `coingecko-api` 用於獲取虛擬貨幣數據。
- 安裝 `openai` 用於與 OpenRouter 的 API 交互。
- 運行以下命令：



```
bash
npm install coingecko-api openai
```

獲取虛擬貨幣數據

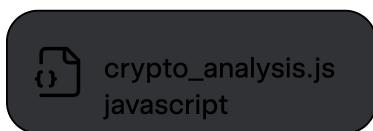
使用 CoinGecko API 獲取比特幣過去 30 天的每日價格數據。程式碼會返回開盤價、最高價、最低價和收盤價（OHLC）等數據。

使用 AI 進行技術分析

將獲取的數據格式化為提示（prompt），並通過 OpenRouter 的 API 發送到 AI 模型，請求進行技術分析。例如，提示可以是： 這是比特幣過去 30 天的每日價格數據，請分析當前趨勢並提供買

注意事項

- **模型選擇**：OpenRouter 提供多種模型（如 `text-davinci-003`），但需試驗以找到最適合金融分析的模型。
- **結果限制**：AI 模型可能無法精確計算技術指標（如 RSI、MACD），建議結合傳統分析工具。
- **錯誤處理**：確保程式碼包含錯誤處理，以應對 API 請求失敗的情況。



行內顯示

詳細報告

以下是使用 Node.js 連接到 OpenRouter 進行虛擬貨幣技術分析的全面指南，涵蓋設置、程式碼實現、限制與建議。此報告旨在為您提供完整的背景資訊、技術細節以及實用建議，確保您能夠成功實現需求。

背景與需求分析

您希望使用 Node.js 開發一個程式，通過 OpenRouter 進行虛擬貨幣（如比特幣）的技術分析，並特別提到「AI Coding 用」。根據研究，OpenRouter 是一個統一 API 平台，提供對多種大型語言模型（LLMs）的訪問，但不直接提供虛擬貨幣數據。因此，解決方案需要結合外部數據源（如 CoinGecko）來獲取價格數據，並利用 OpenRouter 的 AI 模型進行分析。「AI Coding」可能指使用 AI 輔助編碼或分析，假設您希望 AI 模型直接分析數據並提供技術洞察。

技術分析通常涉及計算指標（如移動平均線、相對強弱指數 RSI、MACD）或識別圖表形態（如 K 線圖）。然而，LLMs 更擅長自然語言處理，可能無法精確計算數學指標，但可以根據提供的數據生成趨勢描述或交易信號建議。以下是實現的詳細步驟。

解決方案概述

解決方案分為三個主要部分：

¹ **數據獲取**：使用 CoinGecko API 獲取虛擬貨幣的歷史價格數據。

² **AI 分析**：通過 OpenRouter 的 API，將數據發送到 AI 模型進行技術分析。

3 Node.js 整合：使用 Node.js 編寫程式，整合上述功能。

步驟 1：設置環境

1 註冊 OpenRouter 帳戶：

- 訪問 OpenRouter 網站，創建帳戶。
- 在帳戶設置中生成 API 金鑰，用於程式中的身份驗證。

2 安裝 Node.js 與套件：

- 確保已安裝 Node.js（建議版本 16 或更高）。
- 在專案目錄中初始化 Node.js 專案：

```
bash
npm init -y
```

- 安裝所需套件：

```
bash
npm install coingecko-api openai
```

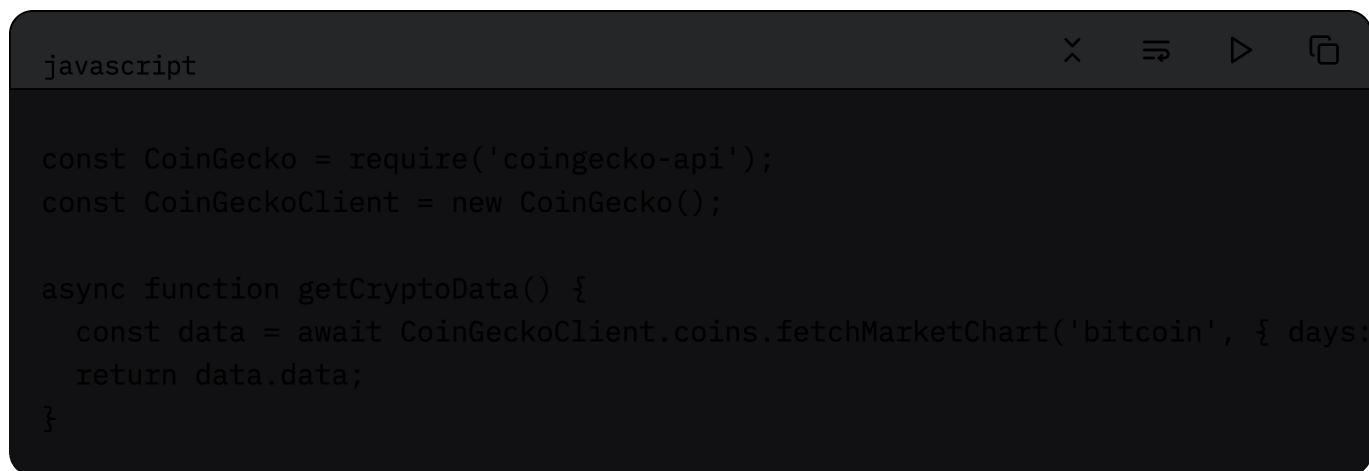
- coingecko-api：Node.js 的 CoinGecko API 封裝，無需額外依賴。

- `openai`：OpenAI 的 Node.js SDK，與 OpenRouter 的 API 相容。

步驟 2：獲取虛擬貨幣數據

CoinGecko 提供免費的 API，無需 API 金鑰即可使用（但有速率限制，免費計劃為每分鐘 30 次呼叫）。您可以使用 `fetchMarketChart` 方法獲取比特幣過去 30 天的每日價格數據，包含時間戳與價格。

以下是範例程式碼片段（已包含在完整程式中）：



```
javascript
const CoinGecko = require('coingecko-api');
const CoinGeckoClient = new CoinGecko();

async function getCryptoData() {
  const data = await CoinGeckoClient.coins.fetchMarketChart('bitcoin', { days: 30 });
  return data.data;
}
```

- **輸出格式**：返回的數據包含 `prices`（價格）、`market_caps`（市值）和 `total_volumes`（交易量）。`prices` 是一個陣列，每個元素為 [時間戳, 價格]。
- **可選參數**：您可以修改 `days` 參數（例如 7 表示 7 天）或指定其他貨幣（如 `ethereum`）。

步驟 3：配置 OpenRouter 的 AI API

OpenRouter 的 API 與 OpenAI 的 API 相容，因此可以使用 OpenAI 的 Node.js SDK，只需將 `basePath` 設置為 OpenRouter 的端點 <https://openrouter.ai/api/v1>。您需要提供 OpenRouter 的 API 金鑰。

範例配置：

javascript

```
const { Configuration, OpenAIApi } = require("openai");

const configuration = new Configuration({
  apiKey: "your_openrouter_api_key",
  basePath: "https://openrouter.ai/api/v1"
});
const openai = new OpenAIApi(configuration);
```

步驟 4：使用 AI 進行技術分析

將 CoinGecko 獲取的數據格式化為文字提示，發送到 AI 模型。提示應清晰描述任務，例如要求模型分析價格趨勢或提供買賣信號。以下是範例提示：

text

以下是比特幣過去 30 天的每日價格數據：[[時間戳1, 價格1], [時間戳2, 價格2], ...]。請進行技術分析。

範例程式碼片段（已包含在完整程式中）：

javascript

```
async function analyzeData(data) {
  const prompt = `以下是比特幣過去 30 天的每日價格數據：${JSON.stringify(data.price_history)}`;

  const completion = await openai.createCompletion({
    model: "text-davinci-003",
    prompt: prompt,
    max_tokens: 1000,
  });
  console.log(completion.data.choices[0].text);
}
```

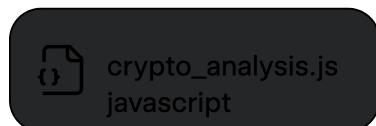
• **模型選擇**：範例使用 `text-davinci-003`，但 OpenRouter 提供多種模型（如 GPT-4、

Claude)。您可以在 OpenRouter 模型頁面 查看可用模型並選擇適合的模型。

- **提示設計**：提示應具體且結構化。例如，您可以要求模型分析特定指標（如移動平均線交叉）或圖表形態（如頭肩頂）。
- **輸出**：模型將返回自然語言的分析結果，例如「價格呈現上升趨勢，建議在回調至 50 日均線時買入」。

完整程式碼

以下是整合所有功能的完整程式碼，包含錯誤處理：



行內顯示

運行程式

1 將程式碼保存為 `crypto_analysis.js`。

2 將 `your_openrouter_api_key` 替換為您的 OpenRouter API 金鑰。

3 在終端運行：

A screenshot of a terminal window titled 'bash'. The window has a dark theme with light-colored text. In the main pane, the command `node crypto_analysis.js` is visible. The terminal interface includes standard window controls at the top right.

4 程式將輸出 AI 模型的分析結果，例如趨勢描述或交易建議。

限制與建議

- **LLM 的局限性**：

- 大型語言模型（如 `text-davinci-003`）擅長自然語言處理，但未專為技術分析訓練，可能無法精確計算指標或識別複雜圖表形態。
- 研究表明，LLMs 在金融分析中的應用（如 CryptoTrade 論文）主要集中在情緒分析或結合鏈上/鏈下數據，而非純技術分析。
- 建議將 AI 分析作為輔助工具，結合傳統技術分析方法。

- **數據格式：**

- CoinGecko 的免費 API 提供每日數據，若需更高頻率（如每小時），可能需要付費計劃或使用其他 API（如 Binance）。
- 確保提示中的數據格式清晰，過長的數據可能超出模型的 token 限制。

- **模型選擇：**

- OpenRouter 提供多種模型，部分模型（如 GPT-4 或 Claude）可能更適合金融分析。請參考 [OpenRouter 模型頁面](#)。
- 若需圖像分析（如K線圖），選擇支持圖像處理的模型（如 Google Gemini），但這需要額外處理圖表數據。

- **增強分析：**

- 使用 Node.js 庫（如 `technicalindicators`）計算技術指標，然後將結果發送給 AI 模型進行解釋。例如：

```
bash
npm install technicalindicators
```

```
javascript
const TechnicalIndicators = require('technicalindicators');
const prices = data.prices.map(p => p[1]);
const sma = TechnicalIndicators.SMA.calculate({ period: 20, values: pri
```

- 將指標結果加入提示，例如：「這是比特幣的 20 日簡單移動平均線數據：\${sma}，請解釋其意義。」

- **錯誤處理：**

- 程式碼已包含基本錯誤處理，但建議添加速率限制檢查（CoinGecko 免費計劃為每分鐘

30 次)。

- 確保 API 金鑰安全，避免公開在程式碼中（可使用環境變數）。

進階應用

- **情緒分析**：結合新聞或 X 平台的數據（如 LLM 情緒分析研究），讓 AI 模型分析市場情緒。
- **自動交易**：將 AI 分析結果與交易 API（如 Binance）整合，實現自動化交易策略（如 LLM 自動交易博客）。
- **圖表分析**：若需分析K線圖，可將價格數據轉為圖表（使用 `chart.js`），並使用支持圖像的模型進行分析。

相關資源

- **CoinGecko API**：詳細文檔請見 CoinGecko API。
- **OpenRouter 文檔**：快速入門請見 OpenRouter Quickstart。
- **n8n 工作流**：參考 n8n AI 虛擬貨幣分析，了解如何結合數據與 AI 分析。
- **技術指標庫**：若需計算指標，可使用 `technicalindicators`。

結論

本指南提供了一個使用 Node.js、CoinGecko 和 OpenRouter 進行虛擬貨幣技術分析的實用解決方案。雖然 LLMs 在技術分析中的應用仍處於探索階段，但其自然語言處理能力可為交易者提供有價值的洞察。建議您試驗不同的模型與提示，並結合傳統分析方法以提高準確性。

關鍵引用

- CoinGecko API Documentation
- OpenRouter Quickstart Guide
- OpenAI Node.js SDK
- n8n AI-Powered Crypto Analysis Workflow
- CryptoTrade: LLM-based Cryptocurrency Trading

- LLMs in Cryptocurrency Sentiment Analysis
- LLM for Automated Trading Blog