

# TÀI LIỆU API: ITAPIA API GATEWAY v1.0

URL Cơ sở (Base URL): /api/v1

Đây là tài liệu dành cho các API công khai do API Gateway của hệ thống ITAPIA cung cấp. API này là điểm vào duy nhất để tương tác với các dịch vụ phân tích và dữ liệu. Tất cả các response đều trả về dưới định dạng JSON.

## 1. Đối tượng Dữ liệu (Data Objects / Schemas)

### 1.1. TickerMetadata

Đối tượng này chứa thông tin tĩnh, mô tả về một mã chứng khoán. Nó được đính kèm trong mọi response để cung cấp ngữ cảnh.

- Schema:

```
{
  "ticker": "string",
  "company_name": "string | null",
  "exchange_code": "string",
  "currency": "string",
  "timezone": "string",
  "sector_name": "string",
  "data_type": "'daily' | 'intraday' | 'news'"
}
```

### 1.2. PriceDataPoint

Đối tượng này đại diện cho một điểm dữ liệu giá (một cây nến).

- Schema:

```
{
  "open": "number | null",
  "high": "number | null",
  "low": "number | null",
  "close": "number | null",
  "volume": "integer | null",
  "timestamp": "integer"
}
```

- Chi tiết: timestamp là Unix timestamp tính bằng giây (UTC).

### 1.3. PriceFullPayload

Đây là cấu trúc response hoàn chỉnh cho các yêu cầu về giá.

- Schema:

```
{
  "metadata": "TickerMetadata",
  "datas": [
    "PriceDataPoint"
  ]
}
```

### 1.4. NewsPoint

Đối tượng này đại diện cho một bản tin.

- Schema:

```
{
  "news_uuid": "string",
  "title": "string",
  "summary": "string | null",
  "provider": "string | null",
  "link": "string | null",
  "publish_ts": "integer | null",
  "collect_ts": "integer"
}
```

- **Chi tiết:** `publish_ts` và `collect_ts` là Unix timestamp tính bằng giây (UTC).

### 1.5. NewsFullPayload

Cấu trúc response hoàn chỉnh cho các yêu cầu về tin tức.

- **Schema:**

```
{
  "metadata": "TickerMetadata",
  "datas": [
    "NewsPoint"
  ]
}
```

### 1.6. SectorPayload

Đối tượng này đại diện cho thông tin của một nhóm ngành.

- **Schema:**

```
{
  "sector_code": "string",
  "sector_name": "string"
}
```

## 2. Các Endpoints

### 2.1. Nhóm Metadata

GET `/metadata/sectors`

Lấy danh sách đầy đủ tất cả các nhóm ngành được hỗ trợ trong hệ thống. Hữu ích cho việc xây dựng các bộ lọc hoặc dropdown trên giao diện người dùng.

- **Tham số:** Không có.
- **Response Thành công (200 OK):**
  - **Body:** `Array<SectorPayload>`
  - **Ví dụ:**

```
[
  {
    "sector_code": "TECH",
    "sector_name": "Technology"
  },
  {
    "sector_code": "FIN",
    "sector_name": "Financials"
  }
]
```

### 2.2. Nhóm Dữ liệu Giá (Prices)

GET `/prices/daily/{ticker}`

Lấy dữ liệu giá lịch sử hàng ngày cho một mã cổ phiếu, được sắp xếp từ mới nhất đến cũ nhất.

- **Tham số Đường dẫn (Path Parameters):**
  - `ticker` (string, bắt buộc): Mã cổ phiếu (ví dụ: `AAPL`).
- **Tham số Truy vấn (Query Parameters):**
  - `skip` (integer, tùy chọn, mặc định: 0): Số lượng bản ghi cần bỏ qua.
  - `limit` (integer, tùy chọn, mặc định: 500): Số lượng bản ghi tối đa cần lấy.
- **Response Thành công (200 OK):**
  - **Body:** `PriceFullPayload`
- **Response Lỗi:**
  - `404 Not Found`: Nếu `ticker` không tồn tại.

GET `/prices/daily/sector/{sector_code}`

Lấy dữ liệu giá lịch sử hàng ngày cho tất cả các cổ phiếu thuộc một nhóm ngành cụ thể.

- **Tham số Đường dẫn (Path Parameters):**
  - `sector_code` (string, bắt buộc): Mã nhóm ngành (ví dụ: `TECH`).

- **Tham số Truy vấn (Query Parameters):**
  - `skip` (integer, tùy chọn, mặc định: 0): Số lượng bản ghi cần bỏ qua cho **mỗi ticker**.
  - `limit` (integer, tùy chọn, mặc định: 100): Số lượng bản ghi tối đa cần lấy cho **mỗi ticker**.
- **Response Thành công (200 OK):**
  - **Body:** `Array<PriceFullPayload>` (Một danh sách, mỗi phần tử là dữ liệu của một cổ phiếu).
- **Response Lỗi:** Trả về một danh sách rỗng `[]` nếu `sector_code` không hợp lệ hoặc không có ticker nào thuộc ngành đó.

GET `/prices/intraday/last/{ticker}`

Lấy điểm dữ liệu giá trong ngày **gần nhất** của một cổ phiếu.

- **Tham số Đường dẫn (Path Parameters):**
  - `ticker` (string, bắt buộc): Mã cổ phiếu.
- **Response Thành công (200 OK):**
  - **Body:** `PriceFullPayload` (Mảng `datas` sẽ chỉ chứa một phần tử).
- **Response Lỗi:**
  - `404 Not Found`: Nếu `ticker` không tồn tại hoặc không có dữ liệu intraday cho ticker đó.

GET `/prices/intraday/history/{ticker}`

Lấy toàn bộ lịch sử giá trong ngày (thường là dữ liệu của ngày giao dịch hiện tại, được giới hạn bởi kích thước stream trong Redis).

- **Tham số Đường dẫn (Path Parameters):**
  - `ticker` (string, bắt buộc): Mã cổ phiếu.
- **Response Thành công (200 OK):**
  - **Body:** `PriceFullPayload`
- **Response Lỗi:**
  - `404 Not Found`: Nếu `ticker` không tồn tại hoặc không có dữ liệu intraday cho ticker đó.

### 2.3. Nhóm Tin tức (News)

GET `/news/{ticker}`

Lấy danh sách các tin tức gần đây cho một mã cổ phiếu, được sắp xếp từ mới nhất đến cũ nhất.

- **Tham số Đường dẫn (Path Parameters):**
    - `ticker` (string, bắt buộc): Mã cổ phiếu.
  - **Tham số Truy vấn (Query Parameters):**
    - `skip` (integer, tùy chọn, mặc định: 0): Số lượng tin tức cần bỏ qua.
    - `limit` (integer, tùy chọn, mặc định: 10): Số lượng tin tức tối đa cần lấy.
  - **Response Thành công (200 OK):**
    - **Body:** `NewsFullPayload`
  - **Response Lỗi:**
    - `404 Not Found`: Nếu `ticker` không tồn tại.
-