CoinDesk Legacy News API (v1) — Full Reference & Usage Guide

1. Overview

The CoinDesk Legacy News API provides access to live and historical news articles metadata, sources, and categories. It is structured into **four key endpoints**, each serving a different purpose for news aggregation and filtering.

---

2. Endpoints & Details

(A) Latest News Articles (`news_v1_article_list`)

Purpose: Fetches a list of articles with metadata.

Key Parameters: - `limit` (int): Number of results to return (default usually 10–20). - `offset` (int): For pagination. - `from` (timestamp): Fetch articles from this datetime onward. - `to` (timestamp): Fetch articles up to this datetime. - `category` (string): Filter by category (see category endpoint). - `source` (string): Filter by specific source(s).

Response Schema:

```
{
  "articles": [
    {
      "id": "12345",
      "title": "Sample Headline",
      "summary": "Short snippet",
      "url": "https://www.coindesk.com/article-link",
      "image_url": "https://cdn.coindesk.com/image.jpg",
      "published_at": "2025-09-29T16:00:00Z",
      "source": "CoinDesk",
      "category": "Markets"
    }
  ],
  "count": 20,
  "offset": 0,
  "total": 1000
}
```

Configuration / Use Case: - Build infinite scroll news feeds (using `offset`). - Incrementally fetch new news (using `from`). - Filter dashboards by category or source.

(B) List News Feeds (`news_v1_source_list`)

Purpose: Returns all available news sources.

Response Schema:

```
{
  "sources": [
    {"id": "coindesk", "name": "CoinDesk"},
    {"id": "reuters", "name": "Reuters"}
  ]
}
```

Configuration / Use Case: - Populate source filter dropdowns in UI. - Attribute news articles to their publishers. - Filter articles by trusted sources only.

(C) News Article Categories (`news_v1_category_list`)

Purpose: Returns all available article categories.

Response Schema:

```
{
  "categories": [
    {"id": "markets", "name": "Markets"},
    {"id": "policy", "name": "Policy"},
    {"id": "defi", "name": "DeFi"}
  ]
}
```

Configuration / Use Case: - Populate filters or tabs in dashboards (Markets, Regulation, DeFi). - Enable user personalization (choose preferred categories).

(D) List News Feeds and Categories (`news_v1_feed_category_list`)

Purpose: Returns combined mapping of sources and their categories.

Response Schema:

```
{
  "feeds": [
    {
      "source": "coindesk",
      "categories": ["markets", "policy"]
    },
    {
      "source": "reuters",
      "categories": ["global", "crypto"]
```

```
    }
  ]
}
```

Configuration / Use Case: - Useful when building structured filters (e.g., show only Reuters + Markets). - Helps design hierarchical dropdowns (Source → Categories). - Configure alert systems based on specific source-category pairs.

---

3. Unified Data Model (Recommended Schema)

Article Object: - `id`: Unique identifier - `title`: Headline - `summary`: Snippet / excerpt - `url`: Article link - `image_url`: Article image link - `published_at`: Timestamp - `source`: Source ID - `source_name`: Source name (via source list) - `category`: Category ID - `category_name`: Category name (via category list)

---

4. Configuration & Best Practices

- **Incremental Fetching:** Always use `from` to minimize redundancy.
- **Pagination:** Use `offset` to implement infinite scroll.
- **Filters:** Integrate both sources and categories for personalization.
- **Caching:** Cache source and category lists (rarely change).
- **Error Handling:** Implement retries and backoff for rate limits.
- **Deduplication:** Use `id` or `url` to prevent duplicates.

---

5. Practical Examples

Example 1: Fetch Latest 20 Articles

```
GET /data/news/v1/article/list?limit=20
```

Example 2: Fetch Articles Since Last Fetch Time

```
GET /data/news/v1/article/list?from=2025-09-29T00:00:00Z
```

Example 3: Get Sources for Dropdown

```
GET /data/news/v1/source/list
```

Example 4: Get Categories for UI Filters

```
GET /data/news/v1/category/list
```

Example 5: Get Feeds & Categories Together

```
GET /data/news/v1/feed_category/list
```

---

6. Integration Opportunities

- **Dashboards**: Build category-filtered news widgets.
- **Notifications**: Push alerts for breaking news by category.
- **Analytics**: Track volume of news by category over time.
- **Social Automation**: Auto-post trending articles from selected sources.

This full reference covers all 4 endpoints, their parameters, response schemas, and recommended usage patterns.