API Documentation for Kikkoman

Arthur

2025-09-18

Table of Contents

1. Introduction

This document describes the REST API for the Kikkoman project. All endpoints return JSON responses and accept JSON payloads where applicable.

2. Authentication

Not required since the API is only exposed internally as a AI back-end service without any external access.

3. Error Handling

All error responses follow this pattern:

```
{
 "error": "Description of error",
 "code": 123
}
```

4. Endpoints

4.1. /metadata/documents

- **Method:** GET
- **Description:** Returns an aggregated summary of source statistics.

***** Request

```
{
 "output_language": "english"
}
```

- `output_language`: (string) Either "en" or "jp."
  - The choice determines the language used for the source, brand and language name strings.

***** Response

```
{
 "document_count": 293595,
 "term_count": 1429925,
 "data": {
   "brands": [
      {"name": "unknown",
       "count": 285916
      },
      {"name": "kikkoman",
       "count": 7636
      }
   ],
   "languages": [
      {"name": "english",
       "count": 248811
      },
      {"name": "spanish",
       "count": 21972
      }
   ],
   "platforms": [
      {"name": "x",
       "count": 273130
      },
      {"name": "amazon",
       "count": 14989
      }
   ]
 }
}
```

- `document_count`: (integer) Total number of ingested documents (posts).
- `term_count`: (integer) Total number of extracted keywords. .
- `data`: (list) Total counts of various languages, platforms (sources) and brands.
  - `brands`: (list) A list of brands and their counts.
  - `languages`: (list) A list of languages and their counts. Currently: English, German, Spanish, Italian, Polish and French.

- `platforms`: (list) A list of sources/platforms and their counts.

4.2. /metadata/terms

- **Method:** POST
- **Description:** Global counts of items.

***** Request

```
{
 "output_language": "en"
}
```

- `output_language`: (string) Either "en" or "jp."

***** Response

```
{"data": [
  {"class": "sauces",
   "topic": "sauces"
   "count": 254432
  },
  {"class": "vegetables",
   "topic:" "ingredients",
   "count": 72391
  },
  {"class": "herbs",
   "topic": "ingredients",
   "count": 41025
  }
]
}
```

- For each item in the list ("data"), we give:
  - `class`: (string) Class name (see the section on taxonomy below).
  - `topic`: (string) The topic name (see the section on the taxonomy below).
  - `count`: (integer) Lifetime counts of this class.

4.3. /terms/filter

- **Method:** POST
- **Description:** Returns items ("keywords," "terms") filtered by various criteria and bucketed by intervals.

***** Request

```
{
 "language": "any",
 "brand": "any",
 "platform": "x",
 "interval": "yearly",
 "iso_date_from": "2023-01-01",
 "iso_date_to": "2025-12-31",
 "output_language": "en",
 "entity_filter": ["chicken"],
}
```

- `language`: (string) Source language or "any" (= all).
- `brand`: (string) Brand or "any" (= all).
- `platform`: (string) Source platform or "any" (= all).
- `interval`: (string) Size of the data buckets. "none" means all data is in one bucket. Legal values: "none," "daily", "monthly," "quarterly" and "yearly."
- `iso_date_from`: (string) Start of time window (ISO 8601, YYYY-MM-DD).
- `iso_date_to`: (string) End of time window (ISO 8601, YYYY-MM-DD).
- `output_language`: (string) Either "en" or "jp."
- `entity_filter`: (string) List of terms to filter results by, or empty list.

***** Response

- With an interval specified ("interval" != "none"), "data" is a list, containing each interval as an object.

```
{
"data": [
   {"iso_date_from": "2023-01-01",
    "iso_date_to": "2023-12-31",
    "results": [
       {"rank": 1,
        "class": "sauces",
        "topic":, "sauces",
        "count": 7160,
        "skew": 0.02814,
        "document_frequency": 1.04404
       },
       {"rank": 2,
        "class": "poultry",
        "topic": "ingredients",
        "count": 6903,
        "skew": 0.35209,
        "document_frequency": 1.00656
       }
    ]
  }
]
}
```

- With no interval, "data" is a JSON object representing the whole period.

```
{"data":
  {
    "iso_date_from": "2023-01-01",
    "iso_date_to": "2025-12-31",
    "results": [
       {"rank": 1,
        "class": "sauces",
        "topic": "sauces",
        "count": 17801,
        "skew": 0.06996,
        "document_frequency": 1.02564
       }
    ]
  }
}
```

- `results`: (list) A list of keywords ordered by rank (1 being the highest).
  - `rank`: (integer) Rank, calculated based on raw counts.
  - `counts`: (integer) Occurrence of this item in the set of documents specified by the filter/periodization.
  - `skew`: (float) Measures the degree to which the item is overrepresented in the filter/periodization set.
  - `document_frequency`: (float) Intended as proportion of documents in the filtered set which contain the item.
  - `class`: (string) See below for taxonomy.
  - `topic`: (string) See below for taxonomy.
- Document frequency can behave oddly due to aggregation. It is a work in progress, as is skew.

4.4. /terms/deltas

- **Method:** POST
- **Description:** Change in item frequency between this year (2025) and the last (2024).

***** Request

```
{
 "output_language": "en"
}
```

***** Response

```
{
"data": [
  {
     "class": "sous vide",
     "topic": "preparation",
     "delta": 0
  },
  {
     "class": "oats",
     "topic": "ingredients",
     "delta": -12
  }
]
}
```

- `delta`: Change in counts in 2025 as compared to 2024 ("oats" has fallen by 12, "sous vide" hasn't changed).

4.5. /brands/signatures

- **Method:** POST
- **Description:** Show the top 6 items that are associated with each brand.

***** Request

```
{
 "output_language": "en"
```

```
}
```

***** Response

```
{
 "data": [
   {
      "brand": "kikkoman",
      "signatures": [
         {
           "class": "sauces",
           "topic": "sauces",
           "weight": 0.01067
         },
         {
           "class": "vegetables",
           "topic": "ingredients",
           "weight": 0.00417
         }
      ]
   }
}
```

- `data`: (list) A list of JSON objects, one for each brand that the application handles.
  - `signatures`: (list) A list of items with their class, topic and weight.
  - `weight`: (float) The weight of this item for the Brand in question (association strength).

5. Common JSON Keys

Key	Description
iso_date_from	Start of time window (ISO 8601, YYYY-MM-DD)
iso_date_to	End of time window (ISO 8601, YYYY-MM-DD)
count	Integer, total count (summary/statistics)
class	Class name of item
topic	Topic name of item
brand	One of the tracked brands
output_language	Either "en" or "jp" (language of output)
language	Source language

6. Reference tables

Source languages
English
German
Spanish
Italian
Polish
French

- Extraction is less good for the non-English languages,

Currently accepted brands
kikkoman
bachans
sweetbabyrays
kinders
unknown_brand

- "unknown_brand" is by far the largest currently.

Platform
x
youtube
amazon
reddit
shorts

- Note that no Reddit posts occur in the system.

7. Taxonomy

- The system extracts mentions of cooking terminology from texts in 6 languages.

- Cooking terminology is aggregated on three levels:
  - Topic: "preparation," "ingredients," and "sauces."
  - Class: +25 different, broad classes of preparation techniques and ingredients (reference TBA).
  - Term: +340 distinct terms, which are grouped into classes.

8. Changelog

- 2025-09-29: Working.
- 2025-09-18: Initial version

9. Contact

For further questions, email: [arthur.hard@elife.co.jp]