The Open Retailing API Data Dictionary

May 30, 2024

Version 1.0

Document Summary

This paper describes the benefits of creating and maintaining the Joint API Data Dictionary, the system of core definitions provided by the Open Retailing project that will ensure consistent data terminology used in applicable information technology systems. Consistency allows developers to create interoperable APIs (Application Programming Interfaces) quickly and effectively. Open Retailing serves the Transportation, Energy, and Convenience retail industry, and the API Data Dictionary streamlines the construction of interfaces and data structures for that industry. Though the dictionary is commonly used in Open Retailing Standards, it is proposed for use in other standards, as well as in proprietary industry interfaces within organizations, resulting in user benefits throughout all of these cases.

What follows is a brief history of data dictionaries, with a summary of the most noticeable benefits of using them, along with plans for making the dictionary more visible and useful to developers, as well as a description of the Open Retailing processes used to maintain the dictionary and keep it aligned with industry needs. Please see the appendix for the current status of the project.

“Open Retailing” is a joint project of the International Forecourt Standards Forum (IFSF) and Conexxus.

Contributors

David Ezell, Conexxus

Revision History

Revision Date	Revision Number	Revision Editor(s)	Revision Changes
May 30, 2024	V 1.0	Linda Toth, Conexxus	Release version
February 19, 2024	Draft 0.13	David Ezell, Conexxus Linda Toth, Conexxus	Resolve legal review comments
February 12, 2024	Draft 0.12	Linda Toth, Conexxus	General clean up, formatting and added Table of Contents.
January 24, 2024	DRAFT 0.11	David Ezell, Conexxus	Updated with changes as discussed with Carl Jones, Kees Mouws, Linda Toth.
November 30, 2023	DRAFT 0.10	David Ezell, Conexxus	Reformatted to use the joint Open Retailing template.

Copyright Statement

Disclaimers

IF YOU ACQUIRE THIS DOCUMENT FROM CONEXXUS, THE FOLLOWING DISCALIMER STATEMENT APPLIES:

Table of Contents

1 Introduction

Consistent data terminology makes information exchange reliable and effective. Today's computer and information technology (IT) systems exchange enormous quantities of data, and terminological consistency in the names for data, in the meanings of these names, and expected formats for the data, make the content much more useful to both machines and to humans. A data dictionary ensures consistency in naming and content of data.

1.1 History of terminology, structures, and controls

Taxonomies[i] have long existed for naming "things," associating a detailed description of a thing to its name, and defining relationships between the things named, i.e., how things are alike or different. Taxonomies have been used in biology for several centuries. Further, the name, description, and class associated with a "thing" is actually a separate thing in itself. The distinctions and commonalities between ideas about things versus physical things is as old as Plato[ii].

More recently, comparable naming systems have evolved in information technology. Data dictionaries[iii] apply taxonomic principles to data as a "thing." In a data dictionary, names are given to specific kinds of data, definitions that apply to the data, and attributes of the data. As a very simple example, a "postal-address-street" would be a name, "locates a house or business on a street" would be a description, and "alphanumeric data up to 40 characters" would be attributes. Because the name, description, and attributes are "data about data," these items are often referred to as "metadata." Together, the name, definition, and attributes comprise a data definition.

While there are many applications for data dictionaries in information technology, building a dictionary for data that flows around and through standardized interfaces between systems is especially useful. These standardized interfaces are Application Programming Interfaces (APIs[iv]), and the API Data Dictionary provides a common data language for use in creating those APIs. When a Data Dictionary is relied upon repeatedly, this reuse of dictionary definitions accelerates adoption and improves quality of applications using the dictionary.

1.2 Headwinds and challenges

Moving existing XML standards over to APIs presents a number of challenges[v]; indeed, it is often difficult to motivate users to redeploy existing standards as APIs. Fortunately, the Open Retailing API development life cycle automatically creates data dictionary definitions as part of the process of standardization. These definitions are very *useful in their own right*, and help bridge the terminological gap between proprietary APIs, which may voluntarily adopt the data dictionary definitions, and standard APIs which always use those definitions. This reuse across proprietary and standard APIs provides commonality that produces significant benefits.

The multiplier effects associated with reuse of dictionary definitions are substantial, but creating a useful API Data Dictionary is not free: a large body of developers and users must communicate and collaborate, they must agree on definitions, and they must agree to reuse those definitions in their work as much as possible. Further, they may agree to suggest new metadata definitions where none currently exist. Failure to consider any of these factors properly – resulting in conflicting names, misaligned descriptions, etc., – can render the dictionary less useful, or even useless.

In addition, even a perfectly constructed dictionary requires that *developers be able find the right definitions* quickly and accurately. Tools for indexing and finding definitions play a crucial role in making the dictionary live up to its promise of productivity.

While the costs for constructing and maintaining a useful API Data Dictionary are not trivial, the costs of not having a common dictionary can be destructively high and substantially reduce the cumulative value of industry data. An effective API Data Dictionary must not only be accurate and complete, but it must provide some means to access the dictionary definitions quickly and efficiently, as well as provide a consensus-driven process for performing the work necessary to extend and revise the dictionary.

2 Benefits

A well-constructed data dictionary targeting Transportation, Energy, and Convenience retail provides tangible advantages, unlocking potential and sometimes hidden economies of scale and enhanced productivity in information technology projects. A common dictionary helps minimize friction between existing IT systems and suppliers of new systems. Basic benefits arising from consistent naming and content control enable organizations to collaborate more easily, and ultimately these basic benefits lead to recognizable industry-wide benefits.

2.1 Foundational benefits

Simply having better naming and management of data definitions improves the ability of both humans and machines to make use of commonalities in meaning and structure of data. These benefits improve the organization and comprehensibility of the data, regardless of the scope of adoption.

Enhanced clarity and understanding:

Entries in the data dictionary will conform to basic rules (or “conventions”) as defined in dictionary design guidelines, including naming of data definitions and how best to define useful content models for data definitions. This consistency makes it easier for users, developers, and machines to understand and interpret data definitions in various contexts. Such consistency is essential for the maturation of the data dictionary into a common business language.

Common language for both standardized and proprietary interfaces:

Because of development pressure, many organizations, both retailer and supplier, often must create programming interfaces quickly. In the best case, these interfaces would be standardized; but in many cases that outcome may not be possible. In those cases, use of standard data dictionary definitions in proprietary interfaces is the next best option. Using standard dictionary definitions allows those proprietary interfaces to integrate more easily with standardized software, providing a path for future standardization for the proprietary interfaces.

Better search capabilities:

The improved organization and centralization of the data definitions supports more advanced tools for understanding the definitions. Tools that support basic lexical and term searches will encourage developers to take the time needed to look for items that might be reused. As the dictionary matures and developer acceptance of the dictionary improves, additional searches based on data semantics will emerge and will further improve the proper use of the dictionary definitions.

Improved data quality:

Improvements in consistency in naming and content lead to improved data quality and integrity. Quality data is more likely to be useful without additional filtering and adjustment (intake processing), thus saving valuable development and processing cycles. Although advanced systems may be able to deduce some data meanings from relatively low-quality data, the need for such deduction is avoided with better quality data. Better data quality improves the results of even the most advanced processing, including processing with AI tools. Errors, inconsistencies, and redundancies become easier to find, reducing the need for AI “interpolation” leading to “AI hallucinations,” where a language model creates non-factual content.

Better Access to Data Analytics:

Standard data definitions make suppliers better able to serve the needs of all retailers. A given retailer can rely on having access to a library of filtering and searching functions to apply to its own specific data. Suppliers may provide customized analytics reports, but these, too, will benefit from common and well-known definitions describing the source data.

2.2 Inter-organization Adoption Benefits

As the basic benefits are being realized, additional benefits begin to emerge. These additional benefits help reduce friction in moving data between various systems and improve the reliability and accuracy of data across various retail entities and suppliers.

Enhanced collaboration:

It is common for data definitions produced by one IT system to be shared with multiple other systems, often from different vendors. Sharing data often requires development of “glue code” to translate data between systems. This “glue code” is a significant source of technical debt, and it must be maintained across multiple systems and versions of those systems, potentially by many different vendors. A common data dictionary helps eliminate the need for such “glue code” and streamlines development and integration projects that are intended to span multiple organizations.

In today's world, retailers increasingly need to work with other business partners to deliver new mobility capabilities, such as bike and scooter rental services, food “click and collect” services, etc. Better collaboration also helps make joint ventures and common development projects easier. The data definitions in the dictionary can be used to validate large segments of machine-shared data, relieving developers of the burdensome work of having to examine that data using direct observation. Improving collaboration using the data dictionary saves time and energy for bigger business and management problems, leading to better business relationships.

Ease of entry into IT ecosystem for new suppliers:

This streamlining of development and integration of data helps remove barriers to entry for new products, either from existing industry suppliers, from retailers, or from those new to the industry. This effect is especially relevant given the emergence of new research and analytic tools needing access to existing data. “Best of breed” solutions to specific problems can thereby enter the industry without having to be entangled in existing software systems and “glue code” with its associated technical debt.

2.3 Industry-wide Benefits

As the data dictionary becomes more widely adopted across a broader number of participants in the industry data ecosystem, still more benefits emerge. Benefits at this level improve development efficiency and reduce friction between suppliers in the industry, making convenience retail more competitive with adjacent vertical segments, as well as more cost effective for retailers.

Better proof of definition compliance:

Software systems that use common data definitions need to be able to prove compliance with the data dictionary. This proof, sometimes called “certification,” helps retailers select new software systems to augment current capabilities, as well as to replace existing systems without disrupting their entire IT infrastructure. This proof also helps suppliers move quickly to provide solutions (so-called “time to market”).

Regulatory compliance and reporting:

As organizations attempt to adhere to regulations, guidelines, and laws relevant to business process, they are attempting “regulatory compliance.”

Some examples of such regulations, guidelines, and laws are “Payment Card Industry Data Security Standard” (PCI DSS), “Health Insurance Portability and Accountability Act” (HIPAA), the EU’s “General Data Protection Regulation” (GDPR), and the “California Consumer Privacy Act/California Privacy Rights Act” (CCPA/CPRA).

Continued use of the common data dictionary allows developers and analysts to recognize the association of data definitions and sets of data definitions as applying to statutory and regulatory scenarios. Without such commonality, important patterns may be difficult or impossible to recognize. A standard data dictionary makes it easier to identify and label definitions related to compliance, thus improving the ability to report on data usage patterns that may be governed by various regulations.

Scalability and interoperability:

Interoperability, as enabled by better data quality and consistency, speeds development and reduces costs. Interoperability promotes scalability of operations across data repositories. Achieving operational scalability will have minimal need for extensive data transformation or manual mapping, i.e., it minimizes intake processing and the associated cost of "glue code."

Future proofing:

Interoperability and scalability reduce the need for "rip-and-replace" scenarios which might otherwise be required to support new and unforeseen uses for data. Data defined by the data dictionary is much more "future proof" and is more likely to be valid over time, producing expanded benefits as time goes by. The applications and data stores based on the dictionary benefit from the sustainability provided by the dictionary, which helps reduce the risk of future data obsolescence or inconsistency.

2.4 Reducing Total Cost of Ownership

Together, these benefits yield substantial time and cost savings. Total cost of ownership is normally calculated as the purchase price of an asset (in this case a software component system) plus the cost of operating that asset over its life. These two categories can be further subdivided:

- Purchase price
    - Selection cost for the retailer
    - Integration costs borne by the retailer
    - Product design and redesign costs borne by the supplier
- Operating costs
    - Integration costs borne by the supplier
    - Continuing regulatory compliance costs borne by both the retailer and the supplier

All of these costs are diminished by the use of a standard data dictionary. The supplier does not have to invent new data and documentation for elements, and does not have to communicate with other suppliers as intensively, saving time and money. Likewise, retailers can focus on the initial purchase and integration, with some assurance that ongoing costs will be held in check by the reduced complexity of required rework and reintegration.

In the short term it may be cheaper to buy and install a proprietary system, but in the longer term it is usually cheaper to use a standards-based system.

If it turns out that the system must be proprietary after all, using the standard dictionary will make it easier to turn the proprietary definitions into a standard.

3 Definition Discovery: Indexing and Tools

Having a well-defined data dictionary is a basic requirement for realizing the benefits discussed above, but that is not enough: developers need to be able to find the definitions they should be using quickly and easily. In addition, they also need to be able to propose new definitions or modifications to definitions quickly and efficiently.

It may be hard or impossible for a developer to devote time to searching for appropriate definitions. The phrase "move fast and break things"[vi] has long been the IT industry developers' mantra and curse. Even though this view of development is now held in increasing disdain, it is, in fact, how a large number of developers must continue to approach their jobs. The pressure to show immediate payoffs for development efforts forces the hurried work developing the initial proof of concept, and may result in "scattered, broken pieces" that remain in a project long afterward. As a result, there is little motivation during the initial development to conform to the data dictionary, despite the clear longer-term advantages of doing so.

In the rush to complete a project, developers may skip over examination of the dictionary entirely, even when viable definitions are available; having to search for useful definitions in a dictionary introduces a drag on producing a fast proof of concept. Thus, any effective dictionary implementation must provide easy to use and effective semantic searches[vii] of its contents; semantic searches are distinguished from lexical searches in that they focus on meaning as opposed to finding literal matches in documentation text.

3.1 Basic searches

3.1.1 Lexical searches

Tools for performing lexical searches have been around for decades; for example, "grep"[viii] in Linux, and before that in Unix. Lexical searches use the full text of definitions, regardless of their position in the definition file. As a result, the usefulness of the search is diminished by the very large number of false-positive results that can be found and accumulate.

3.1.2 Cross References

Cross references focus on two relationships fundamental to API development:

1. Starting with an API, find all of the dictionary definitions it uses, and
2. Starting with a dictionary definition, find all the APIs that use it.

Thus, if a developer starts work on a new API, they might look at APIs in the same application "domain" to see what useful data definitions exist in that space. On the other hand, when considering a specific data definition for inclusion, it can be useful to see exactly where else that definition is used in existing APIs.

3.1.3 Topical searches

Unlike lexical and cross reference searches, supporting topical searches requires additional work to apply classifications to data dictionary definitions. Any definition can have numerous different classifications. For instance, a "digital offer" object might be classified both as "payment" (reducing the overall charge for a purchase) and as a "discount." These orthogonal classifications, if applied carefully, may provide valuable information in finding useful data definitions.

3.2 Semantic searches

Semantic search[ix] tools have become more widely available over the past decade. These tools are most often accessed through search engines such as "Google" and "Baidu." The defining characteristic of semantic search is that it uses meanings of words in the requested search as opposed to literal text matches. The emergence of artificial intelligence LLMs (large language models) promise to support semantic search capabilities even more widely, although there are some legal issues that may impede some uses. While basic searches are extremely powerful, they may miss many subtle associations between items or categories of definitions that would be obvious to either a trained human or a trained machine. A trained machine will have an index resulting from interaction between an LLM and a set of target context data. This index will return readable answer explanations as well as references of where to find the definitions used in formulating the answer.

The index may be used in different ways. A developer might query against key words, or a key phrase that has meaning within defined context of the index. The query would return a brief explanation as well as a list of definitions used in the explanation.

An even more advanced system may be able to assist a developer while working in the context of a new API definition. In this case, a continual matching between the work context and potential "solutions" provides the developer with options for including definitions on the fly. Advanced tools such as GitHub Copilot[x] provide a first generation of this kind of "AI pair programming."

4 The Data Dictionary in Open Retailing

The Open Retailing Data Dictionary provides purpose-built definitions for the Transportation, Energy, and Convenience retailing industry. The definitions cover many specific needs that often don't appear in other segments of retail.

4.1 Topics in the Data Dictionary

The data definitions in the dictionary can be grouped by function. It is important to remember that any definition may appear in many different functions. That flexibility is by design, and the dictionary provides many of the benefits enumerated earlier. Following is an example list of topics covered by definitions in the dictionary:

- Reporting – includes groups of properties and common currency or quantity formats.
- Sales associate messaging – important messaging for store personnel, provides a common event stream for tracing events in a retail location or across locations.
- Dry goods product definitions – includes pricing and nutritional information.
- Wet stock (fuel) product definitions – includes pricing and delivery information.
- Money values – currency is always indicated, so programming from one currency to another is possible.
- Counting values – common ways to express either how many or how much of a thing was sold or delivered.
- Units of measure – provides addition information to counting values, allowing conversion.
- Payment data – ranges from private cardholder personal information to more general information useful for analytics.
- Common product codes for items or fuel sold – often required for regulatory purposes.
- Energy sales information – these are used to determine and relate deliveries, sales, and prepay requests, allowing that information to be examined by data analytics software.
    - Wet stock (fuel) sales information
    - EV charging sales information
    - Hydrogen sales information

4.2 Structure of the Data Dictionary

Definitions are all written in the Open API Specification[xi] using the YAML[xii] data serialization language. The three basic types of definition are:

- **Files ending in "Type.yaml"** – these definitions govern the content of property fields. Simple examples are decimal numeric amounts and dates or times. Types are designed to be specific enough to prevent errors in interpretation (semantic errors) as well as to prevent certain security exploits, such as buffer overflow conditions.

- **Files ending in "Object.yaml"** – these definitions consist of a set of property names, each with a data definition. The most essential feature is that names of things (not just the contents of things) are defined.
- **Files ending in "Element.yaml"** – these definitions are designed to mimic definitions from older XML definitions, called "elements." An element applies BOTH a name AND contents for a name that should be reused when possible.

While every attempt has been made to make the names of the definition files self-explanatory, using the filenames is not a completely satisfactory way to find definitions in the data dictionary, and is the primary reason for implementing the search techniques described earlier.

4.3 Process for maintaining the Data Dictionary

Enticing developers and suppliers to use the data dictionary requires that it have a well-defined and understandable process for maintaining its definitions. Developers and suppliers need to feel that they have a "seat at the table" in establishing the definitions. Participating in the ratifying of data definitions helps ensure that retailers will understand those definitions and that suppliers have a stake in making sure they are used.

Open Retailing makes maintaining the data dictionary an integral part of the creation of standards. Once a working group or committee has determined how the specification will work and has defined data definitions for the targeted standard API, a review of the data definitions used in the API will determine the following:

- Are any of the definitions in the proposed specification duplicates of those already found in the dictionary? If so, use the dictionary version.
- Are there dictionary definitions with different *names* that could be used instead by the specification? If so, change the specification to use the dictionary name and definition.
- Could the contents (e.g., the "state" and "city" listed in an address) of any definitions be adjusted to align with already existing definitions? If so, refer to the dictionary content definition.
- Are any of the definitions in the specification candidates to become part of the dictionary? If so, recommend them for promotion.
- Is it possible that part of a proposed content definition might reuse an existing dictionary definition? If so, create a dictionary content definition that extends the existing definition and use that.

The specific process events follow:

1. Definitions proposed for promotion to the dictionary by
    a. Individual WGs
    b. External organizations
2. Joint API WG reviews and approves, or not
3. If approved
    a. Conexxus Technical Advisory Committee and IFSF Technical Committee review
    b. If approved, new items are moved to standards

This review process keeps the dictionary up to date and relevant for the industry.

5 Summary

Using a standardized data dictionary brings many benefits to the development and maintenance of software. It shortens development time, improves implementation of data analytics, eases friction between retail IT systems, and improves the value of information storing and future processing. These benefits reduce the Total Cost of Ownership (TC), promote scalability and interoperability, facilitate regulatory compliance, and help to make Open Retailing IT Systems future-proof.

6 Additional Information

6.1 Progress of the Work

As of the writing off this paper, the Open Retailing API Data Dictionary is a work in progress. Many of the definitions from older IFSF and Conexxus specifications have been approved and included in the dictionary, using the process cited. Lexical searching is fully operational, and topical searches and cross references are under development, expected to be finished in the near term. Semantic searching is a research item at this point, as is developer context implementations; these features are expected later.

The process for maintaining the dictionary is fully operational, and the dictionary review and promotion rules are documented. Conexxus and IFSF expect this portion of the project to continue to be revised and strengthened over time.

6.2 References/Notes

[i] Taxonomy - https://en.wikipedia.org/wiki/Taxonomy
[ii] Plato Theory of Forms - https://en.wikipedia.org/wiki/Theory_of_forms
[iii] Data Dictionary - https://en.wikipedia.org/wiki/Data_dictionary
[iv] Application Programming Interface (API) - https://en.wikipedia.org/wiki/API
[v] Gartner Hype Cycle - https://en.wikipedia.org/wiki/Gartner_hype_cycle
[vi] Move Fast and Break Things - https://en.wikipedia.org/wiki/Move_fast_and_break_things

[vii] Semantic Search - https://en.wikipedia.org/wiki/Semantic_search
[viii] grep - https://en.wikipedia.org/wiki/Grep
[ix] Semantic search - https://en.wikipedia.org/wiki/Semantic_search
[x] GitHub Copilot - https://github.com/features/copilot
[xi] OpenAPI Specification - https://swagger.io/specification/
[xii] YAML - https://en.wikipedia.org/wiki/YAML