MITRE

Project Lantern Data Sources and Linking Mechanisms

Author(s):
Matt Mayer,
Emily Michaud

June 2020

The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation. Approved for Public Release; Distribution Unlimited, Case 20-1740

Table of Contents

1 Executive Summary

Lantern is an application that collects data about the availability and conformance of Fast Healthcare Interoperability Resources (FHIR) endpoints. Lantern sources most of its data from publicly available endpoint and organization lists. However, some of the data is generated from the Lantern application itself. This document details the publicly available data sources and explains the processes used to produce data by the Lantern application.

2 Endpoint Data

The Lantern project uses publicly available endpoint lists to generate an aggregated list of FHIR API endpoints:

- Epic (https://open.epic.com/MyApps/EndpointsJson),
- Cerner (https://github.com/cerner/ignite-endpoints/blob/master/dstu2-patient-endpoints.json)
- CareEvolution (https://fhir.docs.careevolution.com/overview/public_endpoints.html)
- 1upHealth (https://1up.health/fhir-endpoint-directory)
- Endpoints with type “FHIR” in the CMS National Plan and Provider Enumeration System (NPPES) National Provider Identifier (NPI) endpoint file.
- Endpoints from the endpoint lists included in CHPL

The minimum required information that needs to be included in vendor-provided endpoint lists is an organization name and the FHIR endpoint base URL. This is the only information that Lantern will parse from vendor-provided endpoint lists.

The FHIR Capability Statements retrieved from these endpoints have the capacity to list software names and versions. However, inclusion of this data is inconsistent and does not clearly map to the Certified Health IT Product List (CHPL). Furthermore, the FHIR Capability Statements do not have the capacity to link the FHIR endpoint to an organization, Lantern relies on the organization names reported by the FHIR endpoint list data sources to link a FHIR endpoint with an organization. Details regarding the methods used to link endpoints to organizations are included in Section 6.

2.1 NPPES Endpoint List

The NPPES NPI Endpoint File is available as a downloadable file in CSV format containing the following information:

Table 4. NPPES Endpoint List Fields

Field Name	Field Contents
NPI	NPI Number of the organization using this endpoint
Endpoint Type	Type of endpoint, we only parse rows with the value ‘FHIR’
Endpoint Type Description	Description of endpoint type, endpoint type fields populated with ‘FHIR’ contain the value ‘FHIR URL’ in this field
Endpoint	The base URL of the FHIR endpoint
Affiliation	Indicates if the endpoint is affiliated with another organization
Endpoint Description	Free text description of the endpoint

Affiliation Legal Business Name	Business name of affiliated organization
Use Code	Either Direct, HIE, OTHER, or <no_value>
Use Description	Either DIRECT, Health Information Exchange, Other, or <no_value>
Other Use Description	Free text description of other uses
Content Type	Content type of hosted content, current values are ‘CSV’ and ‘Other’
Content Description	Description of hosted content, current values are ‘CSV’ and ‘Other’
Other Content Description	Free text description of content
Affiliation Address Line One	Affiliated organization address line 1
Affiliation Address Line Two	Affiliated organization address line 2
Affiliation Address City	Affiliated organization address city
Affiliation Address State	Affiliated organization address state
Affiliation Address Country	Affiliated organization address country
Affiliation Address Postal Code	Affiliated organization address zip code

Further information about the contents of the NPPES NPI Endpoint file can be found here https://nppes.cms.hhs.gov/webhelp/nppeshelp/HEALTH%20INFORMATION%20EXCHANGE.html

3 Organization Data

Organization data is parsed from the National Plan and Provider Enumeration System (NPPES) National Provider Identifier (NPI) maintained by the Centers for Medicare and Medicaid services (CMS).

The Lantern project uses the fields listed below to associate organizations with their FHIR endpoints. Lantern only stores NPI entries that represent health organizations, denoted in the NPI files where the “Entity Type Code” field equal is to “2”.

Table 5. NPPES NPI List Fields

Field Name	Field Contents
NPI	NPI Number of the organization using this endpoint
Provider_Organization_Name_Legal_Business_Name	Provider Organization Name (Legal Business Name)
Provider_Other_Organization_Name	Provider Other Organization Name
Provider_First_Line_Business_Practice_Location_Address	Provider First Line Business Location Address
Provider_Second_Line_Business_Practice_Location_Address	Provider Second Line Business Location Address
Provider_Business_Practice_Location_Address_City_Name	Provider Business Location Address City Name
Provider_Business_Practice_Location_Address_State_Name	Provider Business Location Address State Name
Provider_Business_Practice_Location_Address_Postal_Code	Provider Business Location Address Postal Code
Healthcare_Provider_Taxonomy_Code_1	Healthcare Provider Taxonomy Code

4 Vendor Data

Vendor data is parsed from the Certified Health IT Product List (CHPL) API /developers route. Entries that appear in the CHPL developers represent developers of certified health IT software products.

Table 5. Fields Parsed From The CHPL Developers List

Field Name	Field Contents

developerId	Unique ID used within CHPL to identify developer
developerCode	Additional developer identification number
name	Name of the developer
website	URL of developer's website
lastModifiedDate	Date which the developer's entry was last modified
status	Indicates the active status of the developer
addressId	Unique ID of the address entry within CHPL
line1	Developer address line 1
line2	Developer address line 2
city	Developer address city
state	Developer address state
zipcode	Developer address zip code
country	Developer address country

5 Software Product Data

Software product data is parsed from the Certified health IT Product List (CHPL) API "/collections/certified_products" route. Software products returned at this route represent certified health IT products that have been registered in the CHPL.

Table 5. Fields Parsed from the CHPL Developers List

Field Name	Field Contents
id	The CHPL ID of the vendor who makes this software product
edition	The certification edition of this software product
product	Name of the software product
version	Version of the software product
chplProductNumber	Unique string used by CHPL to identify this software product
certificationStatus	Indicates if the software product is currently active
criteriaMet	List of CHPL criteria IDs which this software product meets
certificationDate	Date that the software product was certified

6 Data Validations

The Lantern system runs validations on the CapabilityStatement data returned by endpoints and stores the results of the validations in the validations column of the fhir_endpoints_info database table alongside the CapabilityStatement data. The results of the validations are stored as an array of JSON objects in the following format. Lantern will run the set of base validations against all returned CapabilityStatements and will run FHIR version specific validations depending on the version of FHIR advertised in the CapabilityStatement.

Table 6. Validation Result Object Format

Field Name	Field Contents
valid	Indicates whether the actual value matched the expected value
actual	The actual value as reported by the endpoint
comment	Narrative explaining the validation

expected	Value(s) that will result in a passed validation
ruleName	Name of the validation
implGuide	Reference to an implementation guide (if any) relevant to the validation
reference	Link to relevant rule or standard that defines the expected value of the validation

Table 7. Base Validations

Validation Name	Validation Description
capStatExist	Asserts that a CapabilityStatement was returned by the endpoint
generalMimeType	Asserts that the MIME type "application/json+fhir" is supported
httpResponse	Asserts that the CapabilityStatement request resulted in an HTTP 200
fhirVersion	Asserts that the reported FHIR version is “4.0.1” in accordance with the ONC certification criteria
kindRule	Asserts that the CapabilityStatement.kind field has the value “instance”

Table 8. FHIR R4 Validations

Validation Name	Validation Description
tlsVersion	Asserts that TLS version 1.2 or higher is used during transmission
patResourceExists	Asserts that the CapabilityStatement advertises support of the Patient resource
otherResourceExists	Asserts that the CapabilityStatement advertises support for a resource in addition to the Patient resource
smartHttpResponse	Asserts that the endpoint responded to a request to the well-known endpoint with an HTTP 200
instanceRule	Asserts that if the CapabilityStatement.kind field has the value ‘instance’ then the CapabilityStatement.instance field should be available
messagingEndptrule	Asserts that if the CapabilityStatement.kind field has the value ‘instance’ then the CapabilityStatment.messaging field should not be available
endpointFunctionRule	Asserts that the CapabilityStatement includes at least one CapabilityStatement.rest, CapabilityStatement.messaging or CapabilityStatement.document element
describeEndpointRule	Asserts that CapabilityStatment includes a value for either CapabilityStatement.description, CapabilityStatement.software or CapabilityStatement.implementation fields
documentValidRule	Asserts that if elements exist in the CapabilityStatement.document field, that the documents listed are unique when keyed by the document.profile and document.mode fields
uniqueResourcesRule	Asserts that the list of resources advertised in the CapabilityStatement.rest does not contain duplicate resources
searchParamsRule	Asserts that the names of search parameters within a resource are unique to said resource

7 Linking Mechanisms

7.1 Linking Endpoints to Organizations

Endpoints can be linked to organizations in two ways, either by the NPI ID (preferred), or by the organization name. Lantern links endpoints and organizations by associating endpoint entries with organization entries, and storing the association in the database along with a confidence metric representing the confidence that the association is correct. Details about how the confidence metric is calculated will be discussed further in this section.

7.1.1 Matching by NPI ID

As of the writing of this document, the NPPES Endpoint file is the only endpoint list source that provides both an endpoint FHIR base URL along with a mechanism (an NPI ID) to link the endpoint to a unique organization. Links made between organizations and endpoints using an NPI ID are given a match confidence value of 1, which is higher than any possible confidence value for matches made using the organization name.

7.1.2 Matching by Organization Name

In instances where a unique identifier to match an organization to an endpoint is not provided, Lantern uses the organization name which each endpoint list provides, and the primary and secondary organization names provided by the NPPES NPI data set to associate an endpoint with an organization from the NPPES NPI data set. We use a modified version of the Jaccard Index equation to match organizations based on their names and assign a match confidence score. The modified Jaccard Index algorithm assigns weights to each token in the organization names, the weights are inversely proportional to the number of times the token appears across all organization names.

7.2.1.1 Jaccard Index

The un-weighted Jaccard index is calculated by dividing the number of words that the two names being compared have in common by the total number of words across both names. In the un-weighted Jaccard Index calculation, every token is given the same weight of 1. Mathematically, this is defined as the intersection of two sets, divided by the size of the union of the sets. In this case, each set is the list of words (tokens) that make up an organizations name. For example, if one was comparing the names “Foo Bar” and “Foo Bar Baz”, the sets being compared would be [“Foo”, “Bar”] and [“Foo”, “Bar”, “Baz”] and the resulting Jaccard Index would be 2/3=.666 since the two names have tokens “Foo” and “Bar” in common and there are a total of three unique tokens between both names, “Foo”, “Bar” and “Baz”. The higher the Jaccard Index value, the more similar the two names being compared are.

7.2.1.2 Weighted Jaccard Index

The Lantern application uses a modified version of the Jaccard Index to measure the similarity of organization names. In this modified version, tokens are weighted proportional to their uniqueness within the set of tokens that appear in all organization names. The consequence of assigning a lower weight to common words is that a higher degree of certainty is assumed when a unique word appears in both sets of organization name tokens. Token values are calculated as follows:

1. Build a list of all tokens that appear in all organization names existing in the NPI data source in addition to every organization name reported by the FHIR endpoints lists. Names will be normalized by converting them to uppercase and removing of any special characters.

2. Build a dictionary of the words that provide the least matching value. This dictionary is composed of the top N most frequent words, combined with words found to have little matching value through trial and error. First, sort the token count list by frequency and determine the top number of words which appear so frequently and do not provide any uniqueness to help aid in matching that their weight in the Jaccard Index calculation should be especially low. Next, determine the top words that hinder match outcomes rather than improve match outcomes by repeatedly running the algorithm and evaluating the difference in the output produced by the matching script to check the validity of match scores of endpoints that were matched to organizations, as well as the validity of endpoints that were *not* matched to organizations. The endpoints that were successfully matched but had lower scores than expected were compared with their NPPES NPI organization match to see if there are any unnecessary tokens causing a lower matching score, such as abbreviations. Those endpoints that were unsuccessful in matching were compared with the NPPES NPI data set to see if there were any organizations they should be matching with, and if there was, the endpoint organization name was compared with the NPPES NPI organization name to see if there were any unnecessary tokens causing the organizations not to match. These words, along with the most frequent words that do not aid in matching, are added to a dictionary of least valuable words, these words represent the tokens that have the lowest weight. Words to be added to this dictionary are determined by iteratively running the algorithm and increasing the number of included words until the number of quality matches decreases. Determining the number of quality matches is done by examining the difference in output of the matching script between runs and is not yet an optimized process. As of the writing of this paper, we have found the number of these words to be equal to 31, with a fluff dictionary containing the following words:

"EMS","DR","PA","MD","LLC","LTD","PC","DPM","LLP","AND","OF","IN","THE","MCC", "MMC","TO","PLC","PLLC","SYSTEM","SERVICES","DPMPC","MDSC","CORP","HSHS", "ST","CARE","INC","CLINIC","GROUP","CENTERS","CENTER"

3. Calculate the max, the mean and standard deviation of the token-frequency list, use the chart below, which is ordered by precedence, to assign the value that should be used in the Jaccard Index calculation for each token.

Table 9. Token Frequency to Token Value Mapping Ordered by Precedence

Token Frequency	Resulting Token Value
Tokens Included in Dictionary of Least Valuable Tokens	1 - (Token Frequency/Max Frequency) * .2
0 < Frequency < Mean	1 - (Token Frequency/Max Frequency) * 2.5
Mean < Frequency < (Mean + Standard Deviation / 3)	1 - (Token Frequency/Max Frequency) * 1.6
(Mean + Standard Deviation / 3) < Frequency < Mean + Standard Deviation	1 - (Token Frequency/Max Frequency) * 1.3
Mean + Standard Deviation < Frequency< Mean + Standard Deviation * 3	1 - (Token Frequency/Max Frequency)
Mean + Standard Deviation * 3 < Frequency< Mean + Standard Deviation * 6	1 - (Token Frequency/Max Frequency) * 0.8

Mean + Standard Deviation * 6 < Frequency< Mean + Standard Deviation * 9	1 - (Token Frequency/Max Frequency) * 0.6
Mean + Standard Deviation * 9 < Frequency	1 - (Token Frequency/Max Frequency) * .4

4. After a value is assigned to a token using the chart above, check to see if the token appears in either the list of organization names gathered from the endpoints or the list or organization names gathered from the NPPES NPI data set, but not the both. A token that may appear frequently in one source, but never in the other, source may have a high frequency weight but will never be valuable in matches. Tokens that meet this criterion will have their weight multiplied by .3.

5. Compute the Jaccard Index between organization names provided by FHIR endpoint lists and organization names provided by the NPPES NPI database substituting the computed token values for the token counts that were used in the unweighted Jaccard Index detailed in Section 4.2.1.1. The computed match confidence is equal to the Jaccard Index multiplied by .99, since we can only have a match confidence of 1 when organization matching is performed via a unique identifier provided by an endpoint list.

6. Any calculated match confidences at or above the match confidence threshold of .85 will be considered as a match.

An additional modification to the Jaccard Index formula in the Jaccard Index calculation used by Lantern is that Lantern considers each individual token in an organization name to be unique, meaning that repeated words will be included when an organization name is converted into a set of tokens. For example, the organization name “Foo Bar Foo” would result in the set [“Foo”, “Bar”, “Foo”].

7.2 Linking Endpoints to Vendors

Lantern links FHIR endpoints to vendors using vendor names reported both in the publisher field of the capability statement and the CHPL developers list.

When a capability statement is received, the following matching steps are performed:

1. Normalize both the reported publisher from the capability statement and all of the CHPL vendor names by converting all names to lowercase, removing any of the following words "inc.","inc","llc","corp.","corp","corporation","lmt","lmt.","limited","corporation.". Finish the normalization process by removing any trailing punctuation.

2. Iterate over the entire list of normalized vendor names from the CHPL developers list, if the normalized developer name is a substring of the publisher name or vice versa, then the vendor is considered to be a match.

This approach to name mapping is simpler than our methodology that links endpoints to organizations by name, because there are far fewer vendors than organizations to match. In

addition, the publisher field of capability statements is less variable than the organization names in endpoint lists published by vendors.

8 Query Intervals

Lantern queries its list of known FHIR endpoints once every 23 hours. Setting the query interval to once every 23 hours means that over time Lantern will have queried each endpoint at every different hour of the day. During each query Lantern records data from each endpoints' capability statement in addition to the HTTP response code and response time associated with the request made to the endpoint.

9 Endpoint Info History Pruning

After every query interval, once the capability querier has finished querying all endpoints and updating both the fhir_endpoint_info table and subsequently the fhir_endpoint_info_history table, the history pruning algorithm is run. The pruning algorithm will iterate over all of the fhir_endpoint_info_history entries for each distinct FHIR endpoint URL that have entered_at dates that are older than the time determined by subtracting the LANTERN_PRUNING_THRESHOLD from the current time, and also have entered_at dates that are newer than the current time minus the LANTERN_PRUNING_THRESHOLD plus three times the query interval. Having a lower limit of the LANTERN_PRUNING_THRESHOLD time plus three times the query interval ensures that the algorithm does not repeat pruning checks on the same entries after every query interval, but that it also does not miss any entries that have not yet been pruned. The LANTERN_PRUNING_THRESHOLD, which set to one month by default, ensures that there is always data newer than the LANTERN_PRUNING_THRESHOLD that is not pruned, since an entry has to be older than the threshold in order to be considered for pruning.

The pruning algorithm will remove any consecutive duplicate entries in the fhir_endpoint_info_history table. A fhir_endpoint_info_history entry is considered a duplicate if there is an older consecutive entry that that has the same stored information for the endpoint's TLS version, MIME types, and SMART response, and if the newer entry's stored capability statement only differs by fields included in a list of ignored fields, such as the CapabilityStatement.date field. If a fhir_endpoint_info_history entry is found to be a duplicate of an older consecutive entry, it is deleted from the table, and this continues until only the oldest of the consecutive duplicated entries remains. This pruning strategy is advantageous in that there will always be a duration of at least LANTERN_PRUNING_THRESHOLD minutes worth of queries in the history table for each endpoint, therefore Lantern can inspect LANTERN_PRUNING_THRESHOLD minutes worth of data to see how every endpoint responded within each query interval while still saving storage space by removing duplicate data or data which only differs in the values reported for fields in the ignored fields set. Keeping all entries containing any unique data allows Lantern to keep track of how each endpoint has changed over long periods of time.