



LIVES Best Practices Guide

LIVES Phase 1: Data Extraction Guide

[Summary of Guide](#)

[About Open Data Standards & Leveraging them within Government](#)

[Specifications: Infrastructure & Ecosystem](#)

[LIVES](#)

[Export Options](#)

[Option 1: Use your existing tools](#)

[Option 2: Kettle](#)

[Installation](#)

[Launching Kettle](#)

[Configuring data extraction](#)

[Saving data to a file](#)

[Option 3: FME](#)

[Installation](#)

[Launching FME](#)

[Configuring data extraction and export to file](#)

[Sending Your Data to Socrata](#)

Summary

The LIVES Best Practices Guide walks those looking to launch LIVES restaurant inspection data through the process. While the guide is particularly useful for Socrata open data portals, much of the content could be scaled to other platforms. LIVES, as an open data standard, helps governments share their restaurant data with the confidence that it will be more accurately used by third parties. Benefits to opening up LIVES data include providing restaurant inspection data to third parties, who in turn surface this data to consumers. These consumers include high-risk for foodborne illness groups, like cancer patients, young children (and their parents who take them to restaurants), immune compromised individuals, and the elderly. By providing inspection information in popular third party websites and apps, governments extend their reach to these high-risk populations and ease informed decision-making.

The LIVES 2.0 schema, launched in August 2015 with coordination between Yelp and Socrata, is reflected in this document. Yelp currently ingests LIVES as a ZIP file, and LIVES can be shared via APIs or other file formats. While this document primarily focuses on the technical implementation aspects of LIVES from a Socrata open data portal use case, it contains recommendations which could be more broadly leveraged. We government recommended that



any government considering LIVES read over the “Non-Technical Best Practices Recommendations” attached at the end of this document.

About Open Data Standards & Leveraging them within Government, Content via Code for America

“BART is one of the oldest rail systems in the country and it was build before there was a standard size for rail tracks. As a result BART tracks are a different size than all other rail systems and so there is only one vendor for bart cars. The rest of the world has a standard size track. (...)”

Clearly documented, easily adopted, open-source protocols are the foundations for successful distributed systems. Conventions like HTTP and TCP/IP developed decades ago enabled the phenomenal growth of the internet. Web standards like HTML and CSS have enabled people around the world to build a kaleidoscope of once unfathomable interfaces for webpages and apps. Open data specifications like the General Transit Feed Specification (GTFS) too, have enabled entire ecosystems of apps, tools, and interfaces to bloom around specific subject matters.

In other words, widely-adopted specifications are tremendously valuable. They drive down the cost of communication, experimentation, and software development. They enable software to operate at scale.

Government services—and the data that underpins them—are prime candidates for standardization efforts. In the United States in particular, government services are distributed by an phenomenally large number of organizations. The authority of any one government is bound by geographic jurisdiction, domain-specific authority, and available funding mechanisms. The varied and nuanced contexts in which governments operate has led to idiosyncratic takes on delivering similar services. Things like solid waste collection happen a little different in the City of Chicago than they do in the City of Evanston, and a little bit more different than the City of Gary, across the Indiana border.

Open data specifications are the foundation for a different vision of government in the 21st century. A vision of networked governance, where standardized data and protocols enable great software—or great approaches to delivering public goods and services—to be used and contributed to by governments across the country.



The development of these specifications is not easy. Civic specifications are doubly difficult—for each domain, there is a mature ecosystem of talented, under-resourced people and institutions dedicated to serving their community.”

Specifications: Infrastructure & Ecosystem

Open Data standards reduce the barrier governments face when working with new technology

- Upfront benefit of pulling data into popular online platforms where people are already making decisions, which increases the reach and likelihood of government data to influence individual decision making.
- Standards are a scaffold of our future civic tech ecosystem. They reduce the cost of new startups to enter the market. They help governments experiment and adopt tech solutions at lower cost and risk.
- Standards allow smaller governments to participate and benefit from open data.
- There is more predictable value for open data initiatives.
- Standards accelerate adoption of new technology and the delivery of data to important end user products but also stabilize the civic technology ecosystem and give governments a clearer and stronger voice in the direction of the ecosystem.
- We (Code for America) think standards will enable an evaluative framework for identifying best practices in the provision of government services.
- Quality of documentation of government data improves when the format was developed transparently, as part of community. This allows community groups an easier path towards understanding how they can make use of the data.
- Standards accelerate the work of making open data meaningful



— *Via the Code for America Civic Specs Guide, written by Alan Joseph Williams & Jack Madans, used with permission of Code for America*

Who benefits from open data standards?

Local Government

Open standards make the public data governments already collect more valuable. Standardized data can travel to citizens where and when they need it most. Adopting standards also gives governments access to the world of open-source civic apps—often available a little to no cost—that enable their public agencies to improve service delivery and do more with less.

Technologists

When cities across the country format their data in the same way, technologists can build tools that scale across cities at a fraction of the effort and cost. Open data formats streamline the process of finding and using data across city agencies. That means that civic apps can come from technologists with talent and drive, not just big companies with cash on hand.

Citizens

Data standards help citizens get the information they need to make the best decisions possible for their families, businesses, and communities. Every day countless citizens make hard decisions about where to live, how to commute, and where to spend their money. With standardized public data, citizens anywhere can receive unprecedented guidance on the decisions they face, helping our cities become fair, efficient, and sustainable.



Why use LIVES for restaurant inspection data?

"We're just at a point for food safety, where nothing has changed in the last 50 years. Foodborne illness has remained static or increasing. We're at a point where we need to use data and see ... what else we can do to bring down foodborne illness." -Cyndy Comerford, Planning and Fiscal Policy Manager Environmental Health San Francisco Department of Public Health (SFDPH)

According to the CDC one in six Americans experience foodborne illness every year. Government reports and studies suggest access to restaurant inspection scores impact foodborne illness rates, like the case of Toronto, Canada, which reduced foodborne illnesses within two years of posting restaurant inspection scores. What if your restaurant inspection scores could be accurately presented to the public where they're already going for restaurant information? Could that help your team meet its mission with a new method?

Over the last two years, a handful of US counties and cities have piloted the LIVES open data standard, making their restaurant inspection scores widely available to the public and available on restaurant information services like Yelp. Their work received positive media coverage and support from citizens while informing their public health programs in new and innovative ways. LIVES open data standard for restaurant inspection information represents an opportunity to scale the standard across governments and facilitate citizen education on food handling standards. LIVES has met success with 12 major metropolitan areas using LIVES over the last two years.



Beginning LIVES implementation through data export

The first step to exporting your data requires you to identify and gain access to the source system or database that houses the data you will be exporting and sending to Socrata. This guide describes three options for exporting your LIVES data to a file programmatically which is a prerequisite to publishing the data automatically.

Export Options

Option 1: Use your existing tools

If you have the resources to configure a process to generate exports of the data to a file in CSV or any other machine-readable format, then you are all set. Ideally the export tool or feature you are using to export should not require using a user interface. If it requires a user interface then the data publishing process will not be fully automateable and will require manual intervention for each data update, which must be performed at least once per month.

If you do not have an existing tool or feature to perform the export we have included guides for two recommended Extract-Transform-Load (ETL) tools that enable exporting your data to a CSV file programmatically. Both tools do not require any programming and include capabilities not just to extract data but also to transform and publishing it automatically.

Option 2: Kettle

Kettle is a completely free and open source tool from Pentaho that can be used to extract data from lots of different source systems and file formats (databases, Excel files, and many more formats/systems). If needed, it also allows you to perform transformations on your data and merge multiple data sources together. You can then easily output your data as a CSV file ready to publish to Socrata.

Installation

Prerequisites

- Runs on Windows, Mac, or Linux
- Kettle requires the Sun Java Runtime Environment (JRE) version 1.5 (also called 5.0 in some naming schemes) or newer. You can obtain a JRE for free from <http://java.sun.com>

Download the latest version from <http://community.pentaho.com/projects/data-integration>. Note that the download will start automatically and it is not necessary to click on anything on the page that you are redirected to. Once it has completed downloading, go to your **Downloads** folder and extract the application from the zip file. The **data-integration** folder can then be moved into any directory on your computer (see images below).



More information on installing Kettle can be found at <http://wiki.pentaho.com/display/EAI/01.+Installing+Kettle>.

Launching Kettle

Open the data-integration folder and depending on your operating system click on either the Spoon.bat file in Windows or Data Integration.app file on Mac. For Linux execute spoon.sh on Unix-like operating systems. More detailed documentation on [launching Kettle here](#). (See examples below.)

Configuring data extraction

Once Kettle opens, you will be able to create a new transformation by selecting **File > New >**



Socrata LIVES Access Program materials (c) Socrata, Inc. 2015. All Rights Reserved.

Transformation. You will then be able to connect to your data source by expanding the **Input** tab under the **Design Window**. You will want to use the [Table Input](#) if you are reading from a database. Other options for Inputs include [Microsoft Access Input](#) and [Excel Input](#). All Kettle input types are [listed here](#) (only those labeled with 'Input' as the category).

Saving data to a file

Expand the **Output** tab under the **Design Window** and drag a CSV onto the main window and click on it. (See next page.)

First, you will need to change the extension to CSV and provide a name for the Filename

Then click on the Content tab and change the Separator to a comma.



The final step is to go to the Fields tab and click on the **Get Fields** button for it populate all of your data fields. You can right-click on any of the fields to remove any that you do not wish to send to Socrata. (An alternate way to choose and rename fields is to add a **Select Values** transformer before the creating the CSV file)

Option 3: FME

FME (Desktop version) is another ETL tool similar to Pentaho that will enable extracting and transforming data from an even greater number of source systems and file formats. However it does require purchasing a license after 30 day trial period has expired ([FME desktop pricing](#)).

Installation

You can get a free 30-day trial license at <http://www.safe.com/solutions/for-applications/socrata/trial> and then download the latest version of FME Desktop for your operating system at <http://www.safe.com/support/support-resources/fme-downloads>. Double click on the downloaded file and follow the installation wizard instructions. You may need to check the Spam folder in your e-mail program to retrieve the Safe FME license.

Launching FME

FME will be found under programs/applications. Launch it by double-clicking on FME Workbench.

Configuring data extraction and export to file

Once FME Workbench opens, you will be able to create a new transformation by selecting **File > New > Workspace**. Choose the default **Generate workspace** option to get started.



First you will select a **Reader** that can access your raw data. FME supports hundreds of data sources, <http://www.safe.com/fme/format-search>.



Next you will select **CSV** for your **Writer** format and then click on **OK** button.



FME will then add the **Reader** and **Writer** to the workflow where you are able to add **Transformers** in case there are any changes you need to make to the data before sending Socrata your CSV file. Once you are ready to run the workflow, just click on the green Play button.



If you're sending Your Data to Socrata...

Once you have your data saved as a file email it to Socrata at lives@socrata.com. If the data file is too large to send us over e-mail, you can send us a partial file.





LIVES Best Practices Guide, Phase 2

What is the purpose of this document? The Simplified LIVES Schema document walks you through how the LIVES schema works—both the required and non-required components. Double-check if something is non-required. Use this schema document in partnership with the “flattened schema,” to map your restaurant inspection terminology to the LIVES Schema. This document will help you interpret and approve elements of your LIVES data.

Simplified LIVES Schema

This document represents a simplified and optional alternative to the original multi-table LIVES schema. It is backwards-compatible with the original LIVES schema.

Why an additional schema?

In many cases, it may be simpler to provide a single data file for LIVES instead of the more complicated multi-file zip bundle specified by the original specification. While the multi-table schema more closely replicates the normalized database tables that would represent the data, it is:

- More difficult for non-experts to analyze using tools like Excel, since it requires doing joins or lookups between different tables to relate the data
- More difficult for developers to make use of via download, since they must either load the data into a database to perform joins on, or perform them in memory
- More difficult for developers to use via an API, since it requires multiple API calls to simulate joins
- More difficult to maintain, since it requires the creation of multiple data files and bundling them together into a zip file

Links

- [Original LIVES standard](#)
- [Sample data based on San Francisco's LIVES feed](#)
- [Mapping template](#)

Differences from the original LIVES Schema

- The three key data files - Businesses, Inspections, and Violations, are combined into one dataset
- Dates are formatted as [ISO8601](#) dates, instead of YYYYMMDD dates
- Field names have been prepended with their original table name. For example “name” from the “Businesses” table is now “business_name”



- The “business_id” fields from the Inspections and Violations tables are repetitive and removed
- The [“Score Legend” table](#) has been omitted for simplicity. Instead of providing a legend, specify the result via the “inspection_result” field
- Scores can now fall into one of three scoring methods - “graded”, “cumulative”, or “unscored”

Important Notes

- All of the columns associated with the listing of violations are optional. If a data provider does not collect or does not wish to provide details of violations, those columns can be blank or omitted
 - If violations are not included then the dataset will have one record for each inspection and the associated business information will be duplicated for each inspection
 - If the data provider includes violations then the dataset will have one record for each violation and the associated inspection and business information will be duplicated for each violation, but can be differentiated by their business_ids and inspection_ids. Any inspection that resulted in no violations can be represented as a single row for the inspection with blank data in the violations fields.
- Additional fields beyond those required by the schema may be included in your data file, but they must not override field names that are part of the schema

Feed Information

Along with your data file, the following metadata must be provided. It should be provided as a single line CSV as long as it has these fields. This is the same as the “Feed Information” file from the original spec, minus the feed_date, and with some additional metadata:

Field Name	Data Type	Required	Description
feed_version	string	Yes	Version
municipality_name	string	Yes	Name of the municipality providing this feed. Examples: ‘San Francisco’ or ‘Multnomah County’
municipality_url	string	Yes	URL of the publishing municipality’s website.

contact_email	string	Yes	Email address to contact regarding invalid data in this feed. (We suggest an email address for an IT or data manager with your public health department, not a general "webmaster" email address.)
scoring_method	string	Yes	<p>The scoring method used by this city. Can be one of the following:</p> <ul style="list-style-type: none"> - graded - Within a given range, like 0-5, 0-100, with the highest value being a perfect score - cumulative - Points from violations are summed, and a score of zero is a perfect score. - unscored - No score is published, simply the results of the inspection

Data File Schema

Field Name	Data Type	Required	Description
business_id	string	Yes	Unique identifier for the business. For many cities, this may be the license number.



business_name	string	Yes	Common name of the business.
business_address	string	Yes	Street address of the business. Example: 706 Mission St.
business_city	string	Yes	City of the business. This field must be included if the file contains businesses from multiple cities.
business_state	string	Yes	State or province for the business. In the U.S. this should be the two-letter code for the state.
business_postal_code	string	No	Zip code or other postal code.
business_latitude	number	No	Latitude of the business. This field must be a valid WGS 84 latitude. Example: 37.7859547
business_longitude	number	No	Longitude of the business. This field must be a valid WGS 84 longitude. Example: -122.4024658
business_location	Location	No	If you wish to have geospatial API capabilities or wish to geocode addresses you should include this Location datatype column. Examples: <ul style="list-style-type: none"> - (37.7859547, -122.4024658) - 600 Fourth Ave, Seattle, WA 98104



business_phone_number	string	No	Phone number for a business including country specific dialing information. Example: +14159083801
inspection_id	string	Yes	A unique identifier for a given inspection
inspection_date	date	Yes	Date of the inspection in YYYY-MM-DD format. Example: 2015-08-22
inspection_score	number	No	Calculated inspection score, may be either graded (0-5, 0-100), or cumulative, and this should be defined in your feed metadata.
inspection_result	string	No *	For jurisdictions that do not capture a score, this string represents the non-numeric result of the inspection, for example "Pass" or "Fail". The original LIVES standard requires this field to contain 4 characters or fewer. For broader use of LIVES data, we suggest shortened terms for this field. * If inspections are unscored, this value must be provided.
inspection_description	string	No	Single line description containing details on the outcome of an inspection.



inspection_type	string	No	String representing the type of inspection. Must be one of: initial, routine, follow-up, complaint
violation_id	string	No	A unique identifier for a given violation
violation_description	string	No *	One line description of the violation. * If violation data is provided then this field is required
violation_code	string	No	Code for the violation. It is recommended that this be based on the FDA Food Code. However, municipalities can decide to use pre-existing codes for this field.
violation_critical	boolean	No	Describes whether the violation is critical (i.e., if it would cause the restaurant to fail their inspection) Must be one of: true, false



Important Notes

- All of the columns associated with the listing of violations are optional. If a data provider does not collect or does not wish to provide details of violations, those columns can be blank or omitted
 - If violations are not included then the dataset will have one record for each inspection and the associated business information will be duplicated for each inspection
 - If the data provider includes violations then the dataset will have one record for each violation and the associated inspection and business information will be duplicated for each violation, but can be differentiated by their `business_ids` and `inspection_ids`. Any inspection that resulted in no violations can be represented as a single row for the inspection with blank data in the violations fields.
- Additional fields beyond those required by the schema may be included in your data file, but they must not override field names that are part of the schema

Additional Field Options

While not a part of the LIVES 2.0 schema currently, we suggest a couple additional fields for those who'd like to provide more connection to the agency managing the data and/or use their LIVES data to increase likelihood of foodborne illness reporting.



Field Name	Example Content	Usecase
report_to	<p>Report a potential food poisoning (foodborne) illness event to your local public health department. Their staff will get in touch with you to follow up, see if this was a foodborne illness event, and investigate potential sources of your foodborne illness. Click the link here to report:</p> <p>Use links like mailto: or http:/ https:</p>	Those who are reviewing restaurant inspection scores on third party applications may also be reporting a foodborne illness via that third party application. By providing a report_to field in your LIVES data, your government gains an ability to get the proper reporting resource out into the hands of residents who are unaware that they should (or how to) report the illness. This also eases the process for your public health investigators to follow up on complaints as they emerge, instead of discovering a potential outbreak through solely lab test confirmations reported through local hospitals and testing labs.
report_to_link	<p>http://ourgovernmentwebsite.gov/publichealtagency/report</p> <p>Whatever that reporting link is, post it. Or, if you only take reports by phone, link to the reporting information page on your government's website, which contains contact information and instructions for contacting. Placing a "mailto:" is another option, but since email could be checked infrequently or not transferred to a new staff member, a general phone line for the agency may be a better option. Or, if your team monitors a social media account regularly, link to your most active public health social media account.</p>	Post a link that is maintained by your agency/office and has clear instructions for how to report a potential foodborne illness. Do not link to a "webmaster@government.gov" email address or to the homepage of your government. By linking to your reporting information page, you cut the work for a resident or traveler who may not know your government structure, public health procedures, or proper reporting process. Test out this link on non-government employees who are unfamiliar with your office, and see if they find the reporting resources on your link clear and actionable. Can they report a foodborne illness via the link in five minutes? Good. You've got a winner.

score_result_context	<p>Example Text</p> <p>Scored: In County of Evergreen, Washington, inspection scores can fall from a 0 to 100, with 0 lowest, 100 highest. A score which requires the restaurant's closure is a 70. Scores 71-85 are unsatisfactory, but not grounds for closure and will receive an educational (retraining) visit. Scores of 86 to 100 are considered passing and within the range of best practices. Link to more information.</p> <p>Non-Scored: In the City of Emerald, Texas, restaurants inspected and, based on the number of total violations they receive (and how severe certain violations are), are either given a "pass," "conditional pass," or "closed" result. Passing restaurants met expectations, with fewer than 10 violations. Conditional Pass results have 11 to 25 violations and must be reinspected within 48 hours and pass, or risk closure. Closed restaurants have either failed re-inspection or more than 26 violations on inspection, and were closed. Link to more information.</p> <p>Violations Only: The State of Oceania records violations on restaurant inspections. These violations are in two categories: "Critical violation," which is more likely to cause a foodborne illness, or "non-critical violation" which is a poor business or food-handling practice which is not as likely to result in foodborne illness." Any restaurant receiving more than 3 Critical violations or 10 non-critical violations will be closed. Link to more information.</p>	<p>As there is no American or international standard format for presenting restaurant inspection results, the variability in reporting formats can be confusing for consumers, especially when traveling or new to an area. To assist those using your LIVES data on a third party, consider adding the score_result_context field to provide LIVES users context on how your LIVES score, result, or violations documentation influences closures of restaurants. This provides a baseline that can be understood across jurisdictions.</p> <p>Everyone can understand that a closure is an undesired outcome and letting citizens know what constitutes grounds for closure or other inspection results, helps citizens interpret LIVES data in line with how your restaurant inspectors would interpret an inspection result.</p> <p>Keep the content short (one small paragraph at most). Provide a link within the content for more information on scoring systems within your jurisdiction.</p>
----------------------	---	---



A note to LIVES data third party users

LIVES can be leveraged by companies and organizations to assist consumers in making informed choices when dining out. As this data source comes from government inspections, we have a few recommendations for when you are surfacing this data:

- Resist the urge to normalize scores and results. Restaurant inspection data is **not** comparable to data outside the jurisdiction it was developed. A 0-100 scoring system in one city may close restaurants at a score of 70, and an 80 would be an “unsatisfactory” score. While another 0-100 city might have grounds for closure at 50 and an unsatisfactory score would be 70, with “excellent” scores ranging from 80-100. We suggest posting (or asking for) the additional fields listed below from a government you’re pulling LIVES data from, providing users (and your team) context for a jurisdiction’s inspection data. Not only is this more accurate, you also avoid potential legal issues, as many non-score producing governments have laws prohibiting the development of scores based on inspection information. A possibility worth checking if you intend on creating scores from which there were only results or violations.
- Learn the difference between types of inspection violations. Most common are critical and non-critical violations. These are within the [CDC Food Code](#), but are a recommendation, not consistent across inspection bodies.
 - “Critical items are violations “which are more likely to contribute to food contamination, illness, or environmental degradation and represent substantial public health hazards and [are] most closely associated with potential foodborne disease transmission.” This term seems commonly used.
 - For more violation type information, see the updates within the [CDC’s Food Code](#).
- Post “report a potential food poisoning here,” links adjacent to restaurant inspection scores, to help your users productively and quickly report a potential foodborne illness to the relevant public health agency. Most users do not know that they can report these and if they do “report,” it’s usually just via social media or to the wrong government agency. Your company could help truncate an outbreak through enabling timely reporting of potential incidents.

