**Implementing the JSON-LD Format**

**Purpose**

The Pew Charitable Trusts' civil legal modernization project seeks to make civil court systems more open, efficient, and equitable by promoting policies, processes, and technologies that can improve outcomes for civil litigants. As part of this effort, a data standard and storage format is being developed in partnership with court administrators and information technology specialists to make courthouse information more accessible to the public via internet search engines.

**The JSON-LD Format**

JSON-LD is a lightweight format for storing linked information on the web. It is the preferred format for Google Rich Text and can be used by the search engine to automatically generate the information boxes that appear to the right of search results pages when enough information about a subject is available. This is the most effective way to make courthouse information to the public. In cases where search results are ambiguous due to courthouse naming or overly-broad user search terms and an information box is not generated, the format can still make information more visible by increasing the liklihood that a relevant page is returned in search results.

While appearing to be complicated at first look, JSON-LD is conceptually identical to creating a document or hierarchical outline and so should be easier than other formats to understand and implement. Once a JSON-LD file has been created, it is very easy to change or add information to it. More information on the format is available at <https://json-ld.org/>.

**Building JSON-LD files**

**Step One: Collection**

Collect the information intended to be made more accessible to civil litigants. This can be information already available on a website or information that is not currently recorded anywhere. Examples include courthouse address, phone number, maps, and services provided.

**Step Two: Mapping**

For each included data point, identify the property in the schema to which it most closely matches. Some properties map directly while others may require some interpretation. Potential maps have been standardized as much as possible, but please inform the maintainer of the court data standard if there is data that does not fit any of the provided properties so that it may be added to the schema if necessary.

[Schema.org](https://schema.org/) includes examples of its properties in several different formats that can be useful if you are unclear about syntax or implementation in code.

**Step Three: Building the File**

**JSON-LD Standards and Practices**

JSON-LD organizes data similarly to JSON. Familiarity with the latter is sufficient for understanding JSON-LD. For users unfamiliar with JSON, an example record indicating a courthouse's address will be created step-by-step with explanations provided for each implemented practice or rule.

When building an HTML document, it is necessary to add the following line to the HTML header to indicate that the JSON-LD file should be imported and where it is located.

<link href="<PATH\_TO\_JSON-LD\_FILE>" rel="alternate" type="application/ld+json" />

Alternatively, the JSON-LD data may be embedded inline by inserting it between the script tags in the following line. This is not recommended for readability reasons but can be done if the above method does not work. The script tags should be added to the end of the HTML document.

<script type="application/ld+json" src="<PATH\_TO\_JSON-LD\_FILE>"></script>

**General Standards**

JSON-LD data, whether stored in a separate file or included inline with HTML, must be surrounded by braces. Additionally, each indented level of data should also be surrounded by braces. Data is stored in property-value pairs in the following format: "property": "value". Note the quotation marks surrounding each string. Multiple properties in the same level are separated by a comma and a line break.

{

"top-level property": {

"sub-property": "some value",

"another sub-property": "another value"

}

}

**Nested Properties**

Consider a court system with multiple courthouses. Nesting properties, such as operating hours, makes it clearer with which courthouse the hours are associated and avoids confusion that may arise from identical properties being included in the same level.

A conceptual example of best practice for the organization of information follows.

Courthouse A

Operating Hours

Courthouse B

Operating Hours

**File Structure and Syntax**

The top level of data should begin with an indication of the source of the schema the file follows. A URL to the schema is typically used as the value rather than its name: "@context": "https://schema.org". Each level should also indicate the Type of Thing that is being described if it is different than the Type of the level above it: "@type": "ATypeOfThing". Information about Things, Types, properties, and other terms can be found in [the documentation provided on Schema.org](https://schema.org/docs/styleguide.html).

Types and properties are cased in a manner that is consistent with JavaScript. Types are written in PascalCase while properties are written in camelCase.

**Example Model**

The following model demonstrates how these rules come together to create a JSON-LD file. Note that it does not represent all of the data that may be included in a file. The information used is taken from <https://www.lacourt.org/courthouse> which can be referenced to compare the presentation of data on a webpage and in a JSON-LD file.

1. To start, provide the source of the schema being applied at the top of the file.

{

"@context": "https://schema.org"

}

1. Next, identify the Type of Thing being described. A review the Types listed on [Schema.org](https://schema.org/) suggests that "CivicStructure" is the best option.

{

"@context": "https://schema.org",

"@type": "CivicStructure"

}

1. Now information specific to the courthouse can be added to the file. Google recommends including, at minimum, the name, an image, operating hours, telephone number, and address of a building offering services.

* *name* is the official name of the building and will be displayed in search results.
* *image* is optional and needs to include a link to the image file to be displayed.
* *openingHours* are the business hours of a building. Days of the week are recorded with two-letter abbreviations. Hours that are the same Monday through Friday can either be recorded by listing each day of the week or by separating the start and end days with a dash. Time is separated from the days by a space and should be recorded in 24-hour format.
* *telephone* follows no special convention and will be displayed as recorded.
* *address* is the mailing address of the building. It contains sub-properties and will be demonstrated in its own example below.

Excluding the address field, the JSON-LD looks like the following at this point. Note the commas between lines.

{

"@context": "https://schema.org",

"@type": "CivicStructure",

"name": "Metropolitan Courthouse",

"image": "https://www.lacourt.org/courthouse/images/courthouses/Metropolitan\_loc\_1.jpg",

"openingHours": "Mo,Tu,We,Th,Fr 08:00-16:30",

"telephone": "(213) 745-3200"

}

The address property contains several sub-properties. These values are indented one level ("nested") and surrounded by braces. By nesting them, their relationship to both the address field and the courthouse being described is made more explicit.

As the Type to which these properties belong is different than that of the outer level, it must be specified in the first line of the nested data.

{

"address": {

"@type": "PostalAddress",

"streetAddress": "1945 South Hill Street",

"addressLocality": "Los Angeles",

"addressRegion": "CA",

"postalCode": "90007"

}

}

Combining all of the data, the final JSON-LD file looks like the following.

{

"@context": "https://schema.org",

"@type": "CivicStructure",

"name": "Metropolitan Courthouse",

"image": "https://www.lacourt.org/courthouse/images/courthouses/Metropolitan\_loc\_1.jpg",

"openingHours": "Mo,Tu,We,Th,Fr 08:00-16:30",

"telephone": "(213) 745-3200",

"address": {

"@type": "PostalAddress",

"streetAddress": "1945 South Hill Street",

"addressLocality": "Los Angeles",

"addressRegion": "CA",

"postalCode": "90007"

}

}

**Validating the JSON-LD File**

Several tools can be used to check a JSON-LD file to ensure that the format and information included is correct. The [JSON Playground](https://json-ld.org/playground/)provides detailed error information in a copy-and-paste window. Google also [offers a tool](https://search.google.com/test/rich-results) that validates file information and displays the data it can read and make available to its search engine.

It should be noted that JSON-LD is a flexible format. Existing validators, such as those mentioned above, are opinionated and may report conflicting error information. When data passes one validator but not the other, it is good practice to defer to Google's tool.

For the purposes of the Pew Charitable Trusts' civil legal modernization project, the properties available on Schema.org have been extended to include information relevant to courthouses and may not pass validation from any source. Be aware that these properties will be flagged as errors and may be ignored in validation results.

**Questions and Comments**

If you have any questions, concerns, or comments about the JSON-LD format or any of the information included in this document, please reach out to [John Dziurlaj](file:///Users/home/The%20Turnout%20LLC/Pew%20Civil%20Legal%20System%20Modernization%20-%20Court%20data%20standard%20project/Data%20Standard/john@turnout.rocks) or [Brian Guayante](file:///Users/home/The%20Turnout%20LLC/Pew%20Civil%20Legal%20System%20Modernization%20-%20Court%20data%20standard%20project/Data%20Standard/brian@turnout.rocks).