

## *Digital Literary Map of Japan* 日本のデジタル文学地図: A Technical Perspective

Leo Born (Heidelberg)

### Abstract

The Digital Literary Map of Japan (DLM) is a project aimed at documenting and displaying geographical spots within Japan that convey literary associations or are found to have had significant impact on the literary-historic landscape of Japan over the centuries. In the context of the project, a web application with an online database has been developed in order to make the data easily accessible. This paper presents the technical infrastructure behind the web application and describes its core features as well as providing context for possible future directions of the application. The application is publicly hosted under <https://literarymaps.nijl.ac.jp> and can be accessed without a user account.

### 1 Motivation and Background

The present paper gives an overview of the technical infrastructure behind the "Digital Literary Map of Japan" project. Since its inception, the project was conceived for providing a way for the data to be accessed without technical hurdles. For this reason, we opted for a web-based application that can be accessed from anywhere, with any device and that would give its users an intuitive way to explore and visualize its data.

Literary spots by definition are spatially grounded, paving the way for an exploration of associated literature by way of a visualized spatial dimension. Each spot evokes certain images and might be connoted differently over time, so that tracing the importance and/or impact of spots across literary works and across the centuries can provide fruitful new insights for researchers. Providing a tool that allows this kind of spatial-first exploration was therefore one of the primary goals of this endeavor. A secondary goal was to create a curated, high-quality data set of such literary spots and associated references that was separate from its presentational layer. Thus, the database allows for the data to be analyzed regardless of the web application herein described, opening up the possibility of data sharing in the future.

The overall aim is thus to provide a research and educational resource that works both along the data and the presentational layer, allowing an exploration of literary spots embedded in an explicitly spatial context as well as the overarching cultural-historical

context of Japanese literature. We will describe the data and presentation layers separately, starting with the structure of the database.

## 2 Database

The database serves as the core of the project. Due to the fluid requirements regarding the structural schematics of the data, we opted for a document-based NoSQL MongoDB database. While some of the data is relational in nature, the core concept of data representation within the DLM project was that of *embedding* data – e.g. entries for spots containing literary references as sub-documents. More important was thus the fact that we were not constrained by any given schema that needed to be applied to every entry without exception. Even today, roughly six years after the project’s inception, there arise occasions where either some entries need to contain more or less information than others or some new structural components need to be accounted for without affecting previously saved data. This reinforces the decision to opt for a flexible non-schematic data structure for the project’s contents.

The data types we consider are simply centered around Strings as most of our data is text-based. Even sanitized HTML for the purposes of highlighting different parts (especially in translated texts) can thus be embedded for seamless display within the app (see Section 3 for more details). Data is either provided as Excel files, which are manually imported based on automatic Excel to CSV to JavaScript conversion, or are directly input/edited from within the web application by authorized users. Since MongoDB is document-oriented, import from and export to JSON files is also an option. This is especially handy in the context of the web application, which consumes JSON objects provided by the API, therefore providing a uniform representation of the data.

The database contains at the moment, spring 2022, 75 literary spots, 985 associated poems and 311 associated literary prose references based on 15 literary works. These literary works serve to ground the spots to concrete examples from the Japanese literary history. Therefore, each poem or prose text that is associated with a place is indexed with one of the literary works. Consequently, visual representations of these literary references by way of images (see Section 3 for more detail) are also those of digitized versions of the same literary works.

## 3 Web Application

The primary goal of the project is to let users freely explore the data, even without technical expertise. For this reason, one of the main requirements was to provide a user interface accessible from anywhere without the need to install any software. As this also meant that data could be uploaded or edited without the need to update anything on the user’s end,

the path of a web application was chosen as this only requires a browser and a functioning internet connection.

The implementation of DLM started in 2015 with a MongoDB backend and a Meteor frontend. Due to limited documentation at the time and the more active community of AngularJS, however, six months in, development was pivoted towards an Angular-based frontend application. Due to time constraints, the API was also defined within the same Angular project by using the Loopback framework as the overarching full-stack environment. This allowed for rapid prototyping by generating API endpoints instantly when simply writing a model's corresponding schema file.<sup>1</sup> The database itself has remained the same throughout.



Figure 1. Main map view of DLM.

Apart from core dependencies for the project to run, external libraries are used for styling (Bootstrap), text input (TinyMCE), IIF-compatible image display (Universal Viewer), and, most importantly, the digital map itself. For mapping purposes, we employ Leaflet using the open-source OpenStreetMap-backed cartographic material provided by Mapbox.<sup>2</sup>

---

<sup>1</sup> While the database works without a schema, the model definitions can guarantee type-safe data processing on the frontend. However, modifying the model definitions at any time does not break compatibility with database entries created before the modification, therefore still allowing for dynamic adaptations to changes to the data requirements.

<sup>2</sup> We chose Mapbox as an overlay provider due to their minimalistic "light" theme which does not overly highlight territorial boundaries. This was important as any cartographic material we would use was necessarily modern due to the nature of standard GIS libraries and since the historic realities

Our central view (see Figure 1) shows the map centered on Japan indicating all literary spots with blue pins. Clicking on a spot highlights the pin and reveals an information overlay, showing the number of associated literary references as well as an excerpt of its history. Apart from the map view, there is a text-based table view showing the number of literary references concurrently. In either mode, spots can be filtered by name using a simple search function. The map view provides additional filters that leverage the associated literary references and the literary works they stem from.

Concretely, these are implemented as a text search filter, where literary references are queried, and as a work-based filter. The text-based filter returns matching literary references and can highlight their associated spots. The work-based filter highlights spots based on whether one of the selected literary works contains references to a spot (see Figure 2). This allows to visually display the spatial reach of a single work, further allowing the comparison of it between multiple works. In addition, multiple works can be selected at any time, thus showing the cumulative spatial dimension of multiple literary works.



Figure 2. Document-based filtering highlighting all spots associated with the *Kokin waka shū* 古今和歌集.

Another visual dimension we consider is the one rooted in images. Embedding of images is currently implemented exclusively for literary references, by means of including links to

---

of counties/territories which provide the context for our literary spots does not correspond to the modern ones. Thus, we favored a very simple overlay, which the standard OpenStreetMap material does not provide.

digitized versions of the literary works as well as directly previewing their contents so long as the material is provided in an IIIF-compatible format.<sup>3</sup> Since the National Institute of Japanese Literature (国文学研究資料館) provides a comprehensive resource of digitized literary works in their Database of Pre-Modern Japanese Works (新日本古典籍総合データベース), their data can be seamlessly referenced from within our web application. The International Image Interoperability (IIIF) framework guarantees a uniform standard of encoding and thus we can use any IIIF-compatible viewer to embed images into our web application. For this purpose, we use the Universal Viewer which is illustrated in Figure 3 for a poetic reference of the spot Akashi 明石.

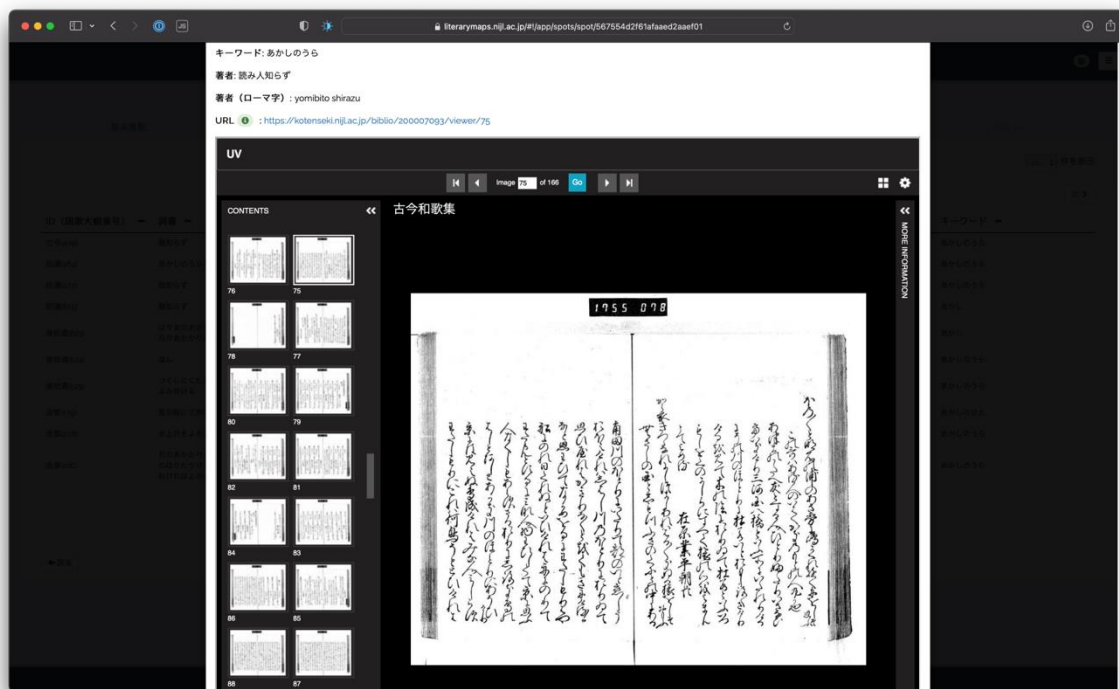


Figure 3. IIIF-compatible image and viewer showing the digitized version of a poem.

The remainder of the web application is structured into detailed display and editing views for the spots. This means that each spot can be further inspected by opening a separate display view of it. This view is divided into multiple tabs, where basic information (map, coordinates, modern location information, literary associations), a spot's history and topography, and various literary references (poems, prose, *yōkyoku* 謡曲) are found. Literary references are shown in a tabular view and contain more detailed information that can be accessed via a modal window as well. This serves to inform interested users more fully on a singular literary spot.

<sup>3</sup> Inclusion of more general images, e.g. showing the spot itself in various historical depictions, is a future feature.

A separate mode of interaction is reserved for authorized users – researchers or assistants of the project at the moment – who can add or edit any data from within the web application as well. The corresponding editor views mirror the normal display views with the only exception being that all fields are editable in-line. More complex text fields are handled by TinyMCE, which provides a visual text editor, similar to Microsoft Word. The map view in the edit mode was also enhanced by some custom hooks, allowing the editing person to simply double-click the map to automatically get the spot name and coordinates based on a reverse Nominatim search as an approximation of the spot to be added/edited. Furthermore, editor access also allows adding/modifying the document information regarding the literary works we use in this project.

Lastly, an open-source user guide was made accessible directly through the web application (by clicking on the green book symbol in the header) or by visiting the URL <https://digital-literary-maps.github.io>. The guide is provided in Japanese and English (like the web application) and serves to briefly describe the application's usage.

#### **4 Conclusion and outlook**

The Digital Literary Map of Japan project consists of a database storing information on literary spots as well as literary reference to these spots, and a web application used to access, edit, and visualize this information. Our goal is to provide a research and education tool that can be used intuitively by a variety of users to explore literary spots, their connotations and their relation to each other. We implemented this by tightly integrating API and UI to fulfill specific visualization requirements at hand. However, since the data itself can also be explored without this specific web application, data sharing or the usage of a different API are in principle possible as well.

One important point to keep in mind is the continuation of development with features that are incompatible with the current framework of the application. While the underlying application will work on modern browsers for the foreseeable future, code maintenance should not be made dependent on the future of a singular framework. When frameworks enter long-term support (LTS) or end-of-life (EOL), their active development becomes stalled and only critical security fixes are provided. Thus, LTS and especially EOL releases cannot accommodate any future feature requests that might become relevant for the project's direction ahead. Thus, compounding this, the design choice to integrate API and UI into the same codebase can become problematic with regards to changing requirements in the future. The project has reached a level of maturity now that separation of concerns should become a developmental concern as well. Fortunately, however, one avenue to continue development is thus to keep the current framework only for the API layer and develop a new web application on top of this, which can be more readily adapted to emerging feature sets.

For the time being, though, the DLM web application provides ample functionality to add new spots and their literary references and to visualize them. Further integration of images is also feasible since the foundation for this were already laid with the current inclusion of IIIF-compatible elements for literary references.

## Reference list

AngularJS: <https://angularjs.org> (accessed: 28.02.2021).  
Bootstrap: <https://getbootstrap.com> (accessed: 28.02.2021).  
Database of Pre-Modern Japanese Works: <https://kotenseki.nijl.ac.jp> (accessed: 28.02.2021).  
Digital Literary Map of Japan: <https://literarymaps.nijl.ac.jp> (accessed: 28.02.2021).  
Leaflet: <https://leafletjs.com> (accessed: 28.02.2021).  
Loopback: <https://loopback.io> (accessed: 28.02.2021).  
Mapbox: <https://www.mapbox.com> (accessed: 28.02.2021).  
Meteor: <https://www.meteor.com> (accessed: 28.02.2021).  
MongoDB: <https://www.mongodb.com> (accessed: 28.02.2021).  
Nominatim: <https://nominatim.org> (accessed: 28.02.2021).  
TinyMCE: <https://www.tiny.cloud> (accessed: 28.02.2021).  
Universal Viewer: <https://universalviewer.io> (accessed: 28.02.2021).