

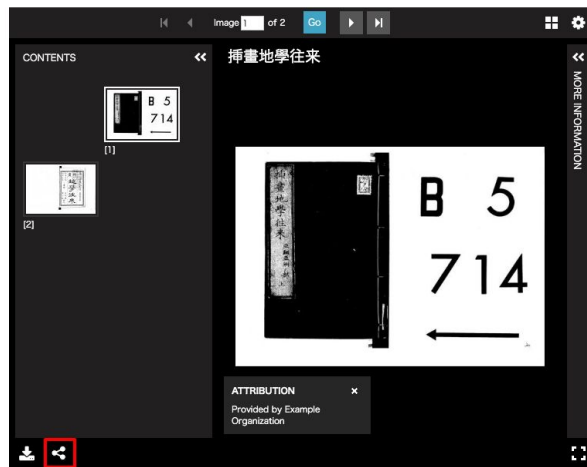
IIIF Manifestを書く

2019年4月12日 中村覚作成 (CC0)

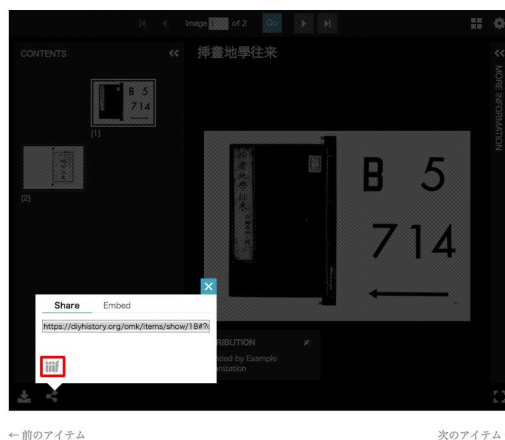
Omekaを使ってIIIF Manifestを作成する	1
The Bodleian's IIIF Manifest Editorを使ってManifestを作成する	2
準備	2
Manifestの保存	2
参考情報	2
Manifestの記述	3
1. マニフェストであることを示す情報	3
2. 資料オブジェクト全体に関する情報	3
メタデータの記述	3
3. 資料オブジェクトを構成する部分についての情報	3
様々なViewerで閲覧する	4
Curationする	4
アノテーションを付与する	4
JSON閲覧環境	5

Omekaを使ってIIIF Manifestを作成する

- 以下のURLからOmekaにログインしてください。
 - <https://diyhistory.org/omk/admin/>
 - ユーザ名 : [user01, user02, ..., user99]
 - パスワード : demodemo
 - (登録したデータは勉強会終了後に削除します。)
- 画面左部の「アイテム」をクリック
- 黄緑色の「アイテムを追加」をクリック
- 「Dublin Core」タブで「タイトル」などの情報を入力してください。
- 「ファイル」タブで適当な画像ファイルを選択してください。
 - PCが画像が入っていない場合にはウェブからダウンロードしてください。
- 画面右部で「公開」にチェックを入れて、「アイテムを追加」ボタンをクリック
- 登録したアイテムのタイトルをクリック
- 画面右部の「公開ページを見る」をクリック
- 画面下部のビューアで「Share」アイコン (下図の赤四角参照) をクリック



- 「IIIF Manifest」アイコン（下図の赤四角参照）をクリックをクリック



- IIIF Manifest（以下、Manifest）が表示されます
- この作業を通じて作成した画像リソースを後ほど使います。

このManifestの内容を理解し、ご自身で作成できるようになることを目指します。

The Bodleian's IIIF Manifest Editorを使ってManifestを作成する

- 以下のURLからThe Bodleian's IIIF Manifest Editor（以下、Manifest Editor）にアクセス
 - <https://digital.bodleian.ox.ac.uk/manifest-editor/>
- 「New Manifest」を選択

準備

- (何をやっているかは後ほど説明します。)
- 画面左下の「Add Canvas」をクリック
- 「Empty canvas」をクリック
- 画面右部の「Canvas Metadata」で、「Add Image to Canvas」をクリック
- 「From info.json URI」を選択し、以下のURIを入力
 - https://iiif.dl.itc.u-tokyo.ac.jp/iiif/kunshujou/A00_6010/001/001_0003.tif/info.json
- 「Submit」ボタンをクリック、画像が登録されます。

Manifestの保存

- 画面右上の「Save Manifest」ボタンをクリック
- 「Store manifest remotely」を選択
- 「Store manifest on Server」をクリック
- 表示されるURIにアクセス、Manifestが表示されます
- 以下のURLの末尾にManifestのURIを付与し、「Mirador」Viewerで閲覧してみる
 - <http://da.dl.itc.u-tokyo.ac.jp/mirador/?manifest=> (IIIF ManifestのURI)

参考情報

- Manifestの説明は、神崎正英氏によるインフォコムブログが詳しい
 - <https://service.infocom.co.jp/das/loddiary/iiif/>
- こちらのブログを参照しながら進める

Manifestの記述

- 参考
 - <https://service.infocom.co.jp/das/loddiary/2017/01/20170127001583.html>

1. マニフェストであることを示す情報

2. 資料オブジェクト全体に関する情報

- Manifest Editorにおいて、画面右部の「Manifest Metadata」をクリック
- 「Label」の値を追記する
- 一旦保存する。画面右上の「Save Manifest」をクリックし、「Update Manifest on Server」をクリック
- 先に閲覧したManifestやViewerでのlabelの値が変わることを確認する
- Manifest Editorに戻り、同様に「Description」「Attribution」「License」を入力する

- また「Add metadata field」ボタンを押し、「Logo」を選択し、LOGO画像のURLを入力する。また「Viewing Direction」を選択し、「right-to-left」を入力する。これにより、画像を右から左に並べることができます。

メタデータの記述

- 参考
 - <https://service.infocom.co.jp/das/loddiary/2017/03/20170314001601.html>
- Manifest Editorにおいて、画面右部の「Manifest Metadataの「Custom Fileds」タブをクリック
- まず「Add metadata field」をクリック
- 「Label」と「Value」を入力する
- 次に「Add language metadata field」をクリック
- 「Label」「Value」「Language」（ja, enなど）を入力する
- Manifestを保存し、確認してみる

3. 資料オブジェクトを構成する部分についての情報

- シーケンス
 - Manifest Editorにおいて、画面右部の「Sequence Metadata」をクリック
 - 「Sequence Label」を編集する
- キャンバス
 - 画面右下の「Add Canvas」ボタンをクリック
 - 「Empty canvas」をクリック
 - 画面右部の「Canvas Metadata」で、「Add Image to Canvas」をクリック
 - 「From info.json URI」を選択し、先にOmekaに登録した画像のinfo.json URIを入力する
 - 先にOmekaに登録した画像のManifestから、「service」という項目の「@id」のURIにアクセスする
 - 自動的にURLの末尾に「info.json」が付与されます。
 - info.jsonの読み方は以下を参照
 - <http://www.asahi-net.or.jp/~ax2s-kmtm/ref/iiif/apiimage2.1.html#image-information-request>
 - このURIをManifest Editorで入力してください。
 - 「Submit」ボタンをクリック、画像が登録されます。

様々なViewerで閲覧する

- Mirador
 - <http://projectmirador.org/demo/>
- Universal Viewer

- <https://universalviewer.io/uv.html?manifest=>
- Image Annotator
 - <https://www.kanzaki.com/works/2016/pub/image-annotator?u=>
- IIIF Curation Viewer
 - <http://codh.rois.ac.jp/software/iiif-curation-viewer/demo/?manifest=>
- TIFY
 - <http://tify.sub.uni-goettingen.de/demo.html?manifest=>

Curationする

- 鈴木親彦氏（CODH）によるマニュアル
 - <http://www.ch-suzuki.com/icpt/>
 - 第2回 CODHサイトからICPを使ってみる（IIIF Curation Viewer）
- Curation API
 - <http://codh.rois.ac.jp/iiif/curation/>

アノテーションを付与する

- 永崎研宣氏によるマニュアル
 - <https://jinmoncom2017.omeka.net/items/show/1>
- 先に紹介したOmekaのURLに適宜読み換えてください。

JSONの閲覧環境

- JSON Viewer - Google Chromeの拡張機能
 - <https://chrome.google.com/webstore/detail/json-viewer/gbmdgpbipfallnflgaipaljiibnhdgobh?hl=ja>
- Firefoxの利用
 - Firefoxであれば特に設定なしでJSONがインデント表示

議論があったこと

- URLとかIDの永続性。今回のデモではシステムが付与する部分が多い。
 - かわってしまうとみんなこまる。
- DOIで付与する箇所とローカルに依存する箇所。

- recommended URIにDOIがあるのが理想
- validatorはある？
 - <https://iiif.io/api/presentation/validator/service/> など。Omekaは問題ない。
 - 「IIIFで」としか仕様がないと納品時に誤っていることがあるのでvalidatorを通すこととするとよいのでは。
- メタデータに seeAlsoがある。詳細なメタデータは他に投げていると言っている。