IIIF for Scholars: Sharing, Consuming, and Annotating the World's Images

Wednesday, May 8, 2019 - 11:00am to 1:00pm

Research Library (Charles E. Young)Research Commons (RC) Classroom

Slides:
https://docs.google.com/presentation/d/1Zwl4WrGEQybS_bq2oFpjHdfAuc66fNjbVxFlyjX8meM/edit?usp=sharing

Access to image-based resources (manuscripts, artworks, maps, etc.) is fundamental to research, scholarship and the transmission of cultural knowledge. Yet much of the Internet's image-based resources are locked up in silos, with access restricted to bespoke, locally built applications. IIIF (International Image Interoperability Framework), a framework for interoperable image delivery, gives an unprecedented level of uniform and rich access to image-based resources hosted around the world enabling scholars to view, annotate, and remix digital images (http://iiif.io).

In this workshop, participants will learn basic IIIF concepts and technologies, and will work with IIIF-hosted content to explore the benefits and scholar use-cases through discussion and hands-on exercises. No prior experience is assumed, but you will need to bring a laptop and have a GitHub account to participate in all the hands-on exercises. If you do not already have one, you can sign up for a GitHub account here: https://github.com.

Explore: URI Parameters

- Image API Playground: https://www.learniiif.org/image-api/playground
- Image Request URI Syntax (IIIF documentation): https://iiif.io/api/image/2.1/#image-request-uri-syntax
- {scheme}://{server}{/prefix}/{identifier}/{region}/{size}/{rotation}/{quality}.{format}
    - http :// www.e-codices.unifr.ch/loris / cma/cma-1955-74 / cma-1955-74_000b.jp2 / 650,2300,1000,750 / pct:50 / 0 / default.jpg

Explore: IIIF & Deep Zoom

- Pyramids - how IIIF enables Deep Zoom: http://tomcrane.github.io/presentations/tile-exploder.html

Explore: info.json

- Revisit the Image API Playground: https://www.learniiif.org/image-api/playground
- Essentially: what can we do with a particular image
    - how far we can zoom in
    - possible transformations
    - available services (e.g. auth, physical dimension info) https://iiif.io/api/image/2.1/#image-information

Hands-on: Leaflet-IIIF

- First we'll need to make sure GitHub Pages is activated on ur forked repository
    - In GitHub repo, go to the Settings tab, then scroll down to the GitHub Pages section and choose `Master branch` from the Source pull-down menu
    - Test GitHub Pages by copying the "Your site is published at" URL and pasting into a browser window. You should see a page that says "IIIF Workshop @ UCLA"
    - If you add `/leaflet` to the site URL, you should see the Leaflet-IIIF viewer in action.
- Our Leaflet-IIIF application is in our repo in the `leaflet` folder.
    - Open and inspect the `leaflet.js` file
    - How would you display another IIIF-hosted image?
    - Replace the existing IIIF image with a different one (hint: you can use our e-codices example)
    - Look at the javascript - what are we telling Leaflet-IIIF to do with the image?
    - Notice the section of the `leaflet.js` file where you can layer two different images. Why might we want to do that?

Explore: IIIF Manifests

- Let's first look at the manifest template in the `manifests` folder of the GitHub repo: `manifest-template.json`
    - Can you spot the components of a manifest that we just discussed? (I included the placeholder `[image url]` so you can see where the IIIF Image API URLs fit in a manifest)
- Explore the `toganoo-1.json` manifests
    - What other information can we find in this manifest?
    - What about "sequence"?

Hands-on: Building a Manifest

- Building a manifest with the Bodleian Manifest Editor: http://dmt.bodleian.ox.ac.uk/manifest-editor/
- Export your new manifest and add it to the `manifests` folder in our GitHub repo

Hands-on: Set up your own Mirador

- The Mirador app is setup in our GitHub repo (see the `mirador` folder?)
- Since we are using GitHub Pages, we can view the live Mirador app in our browser by adding `/mirador` to our site URL (like we did with Leaflet-IIIF above)
- Back in the GitHub repo, view the `mirador` folder. You should see another folder, also named `mirador` and and `index.html` file. The `index.html` file is what we'll work with to customize our Mirador instance.
- Open the `index.html` file.
    - Take a look at the different sections. What info are we presenting? How does this correlate to what you see on in the live Mirador view?
    - We have one manifest available for viewing right now. How might you add the manifest that you created with the Bodleian editor?
- Let's play with some configurations: https://github.com/ProjectMirador/mirador/wiki/Configuration-Guides
    - How would you configure kabuki to load on open?
    - …set the default view for kabuki to the Book Viewer?
    - …configure Mirador to open with two object windows?