CONNECTING CULTURAL HERITAGE COLLECTIONS DATA WITH LINKED ART

Hands-on Session with Linked Art Notebooks - Exercise Sheet

Introduction **3**
The Transformation process 3
How to Interact with the Linked Art Notebooks **4**
Notebook, Software and Data Requirements **4**
Linked Art Jupyter Notebooks 4
Trusted Notebooks 4
Software 4
Anaconda 4
Data 5
GitHub 5
Your own Data 5
Additional Collection Data 5
Related Documentation **6**
Linked Art Sessions Files 6
From Session 1 6
Example Notebook: Minimum Representation in Linked Art **7**
Exercise 1: Extract Collections Data **8**
Exercise 1a: Extract Collection Data in CSV Format 8
Extension Activities 8
Exercise 1b: Extract Collections Data to Linked Data - XML 8
Exercise 1c: Extract Collections Data to Linked Data - JSON 9
Exercise 1d: Extract Collections Data to Linked Data - HTML 9
Exercise 2: Map Collection Data to Linked Art **10**
Exercise 3: Transform Collections Data to Linked Art JSON-LD **11**
Exercise 3a - Transform Collection Data (CSV) to JSON 11
Exercise 3b - Transform JSON to LinkedArt JSON-LD 11
Exercise 4: Transform Cleveland Museum of Art Collections Data to Linked Art **12**
Exercise 5: Reconcile Collection Data using Linked Art **13**
Exercise 6: Visualise Collection Data using Linked Art **14**

Extension Activities **15**

Explore Further Linked Art Code Notebooks 15

Explore Linked Art website 15

Try Different Tools to Interact with Jupyter Notebooks 15

Visual Studio Code 15

Google Colaboratory 15

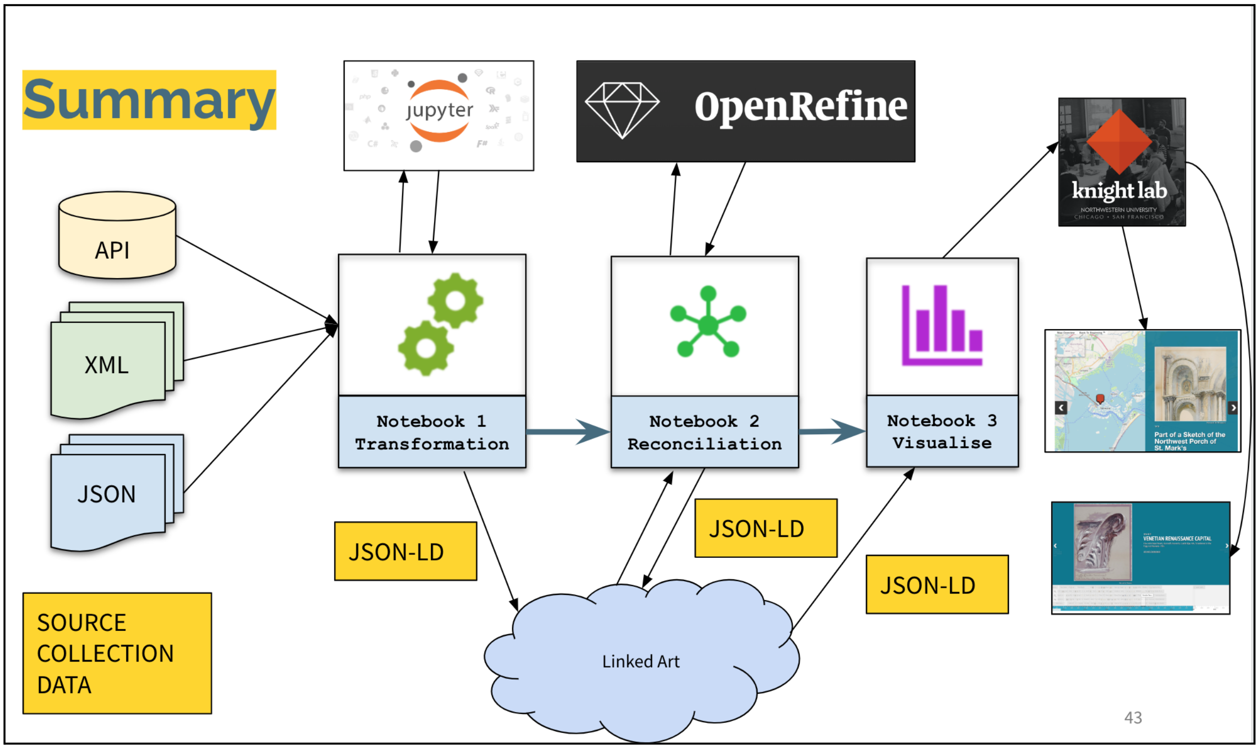
Voila 15

JSON-LD 15

JSON-LD Playground 15

Introduction

In this session, we will support you in a hands-on exercise looking at the processes involved in transforming cultural heritage data to Linked Data. We will use code notebooks to transform cultural heritage data using the Linked Art data model.



Summary

API

XML

JSON

SOURCE COLLECTION DATA

Notebook 1 Transformation

Notebook 2 Reconciliation

Notebook 3 Visualise

JSON-LD

JSON-LD

JSON-LD

Linked Art

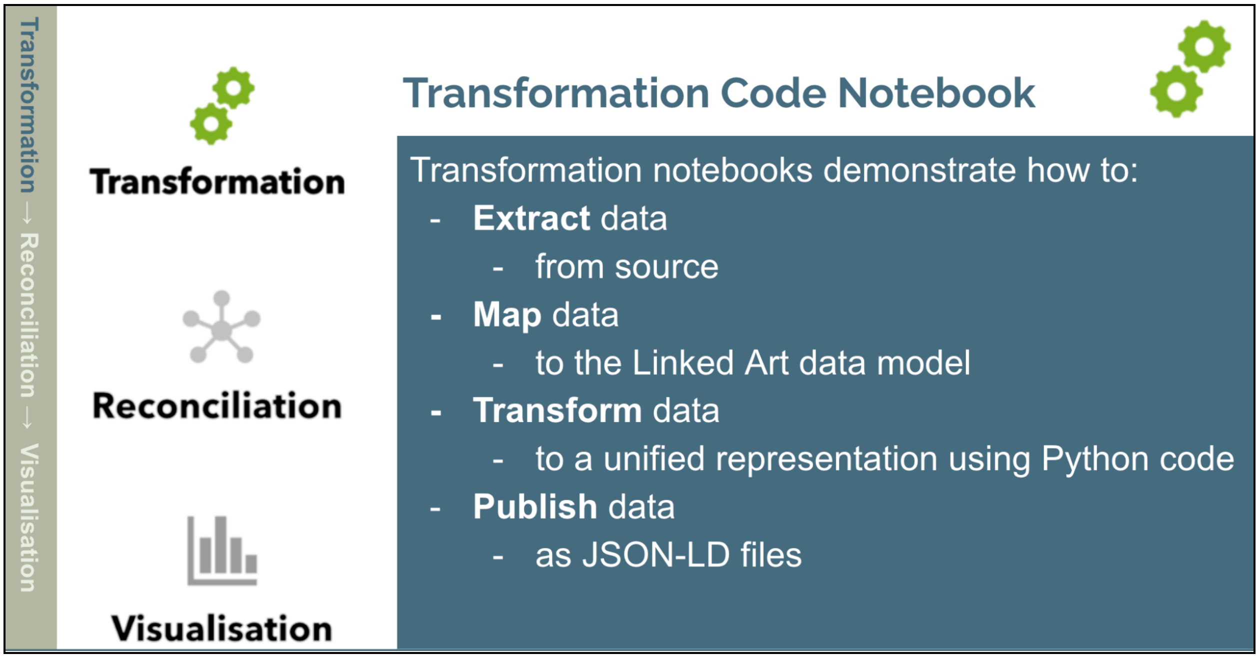
43

jupyter
OpenRefine
knight lab
NORTHWESTERN UNIVERSITY
CHICAGO + SAN FRANCISCO
Part of a Sketch of the Northwest Porch of St. Mark's
VENETIAN RENAISSANCE CAPITAL

The Transformation process

The transformation process will be demonstrated across a number of notebooks looking at the following sub-processes:

- **Extract** collection data
- **Map** collection data to the Linked Art data model
- **Transform** collection data to a unified representation
- **Publish** collection data as Linked Art JSON-LD



Transformation → Reconciliation → Visualisation

Transformation

Reconciliation

Visualisation

Transformation Code Notebook

Transformation notebooks demonstrate how to:

- **Extract** data
  - from source
- **Map** data
  - to the Linked Art data model
- **Transform** data
  - to a unified representation using Python code
- **Publish** data
  - as JSON-LD files

How to Interact with the Linked Art Notebooks

The notebooks can be approached in different ways

- Step-by-step guided code process
- Interactive coded process where you can:
  - Run the code with your own collection data file and/or
  - edit the existing code

Notebook, Software and Data Requirements

Linked Art Jupyter Notebooks

You will need to download the Linked Art Jupyter notebooks from https://github.com/tgra/Linked-Art

Trusted Notebooks

When viewing a notebook, ensure that the notebook is in the ‘trusted’ state.

Software

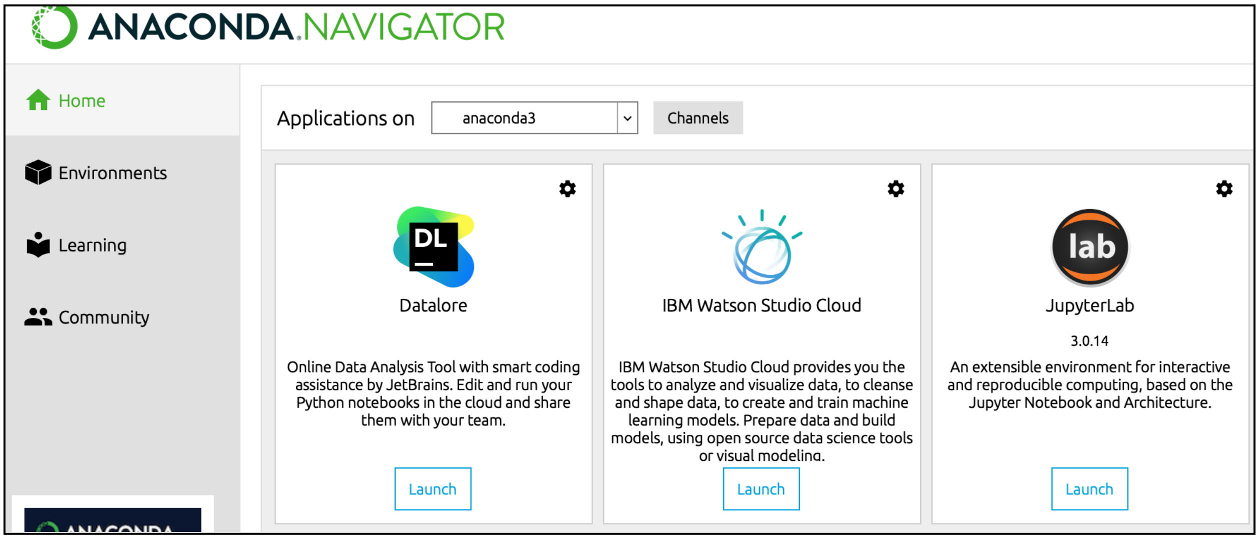
You will need to have installed the required software to view the Jupyter Notebooks - refer to the Software Installation Instructions shared in Session 1.

Anaconda

If are using Anaconda to interact with the notebooks:

- Open the Anaconda application

- In the Anaconda.navigator interface, locate JupyterLab and click on the ‘launch’ button



Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

- JupyterLab will open in your browser at localhost:8888
  - - if you lose the browser window containing JupyterLab use this address to open JupyterLab again
- In JupyterLab, navigate to the downloaded folder containing the Linked Art Jupyter notebooks

Data

GitHub

Example data will be included in the Linked Art notebook GitHub repository mentioned above, in the ./workshop/data/example folder.

Your own Data

There is an opportunity to upload and create your own data files. These will be saved in the ./workshop/data/working folder.

Additional Collection Data

There are additional collection data files available in the root of the Linked Art notebook repository, in the ./data folder. The source of these data files are described in the Jupyter notebooks, also in the root folder.

Related Documentation

Linked Art Sessions Files

Google Drive folder
https://drive.google.com/drive/folders/1CDUESuflSX-rkUHTxKUduy4I-2oxQz0L?usp=sharing

Example Notebook: Minimum Representation in Linked Art

The ***Linked Art - Core Properties*** can be used to help demonstrate some of the features of Jupyter notebooks. The notebook describes the minimum representation for an entity in a Linked Art JSON-LD file, and includes the following steps:

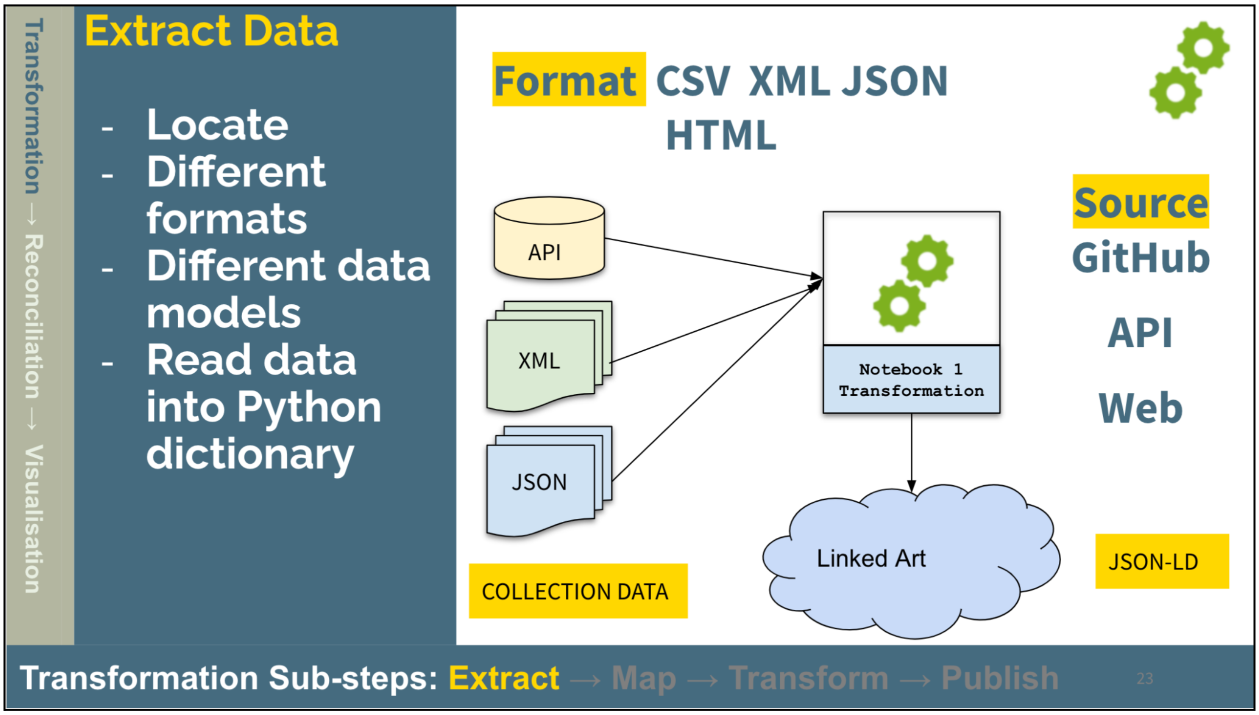
1. import the *json* module to create JSON.
2. If *json* isn't already installed, the code will install it using *pip*
3. Python dictionary *jsonld* will be created to hold the properties and values
4. The *jsonld* dictionary is transformed to JSON using the *json.dumps* method of the *json* module

Open the ***Linked Art - Core Properties*** Jupyter notebook:
./workshop/elinkedart-core-properties.ipynb GitHub Binder

Exercise 1: Extract Collections Data

In this exercise you will learn how to read in collection data in various formats and convert it to a Python dictionary, ready for transformation to Linked Art.

After stepping you through the process, the notebooks allow you to upload your own data and step through the process again.



Transformation → Reconciliation → Visualisation

Extract Data

- Locate
- Different formats
- Different data models
- Read data into Python dictionary

Format CSV XML JSON HTML

API

XML

JSON

Notebook 1 Transformation

Source GitHub API Web

Linked Art

COLLECTION DATA

JSON-LD

Transformation Sub-steps: Extract → Map → Transform → Publish

23

Exercise 1a: Extract Collection Data in CSV Format

In this exercise you will learn how to read in collection data in CSV format and convert it to a Python dictionary, ready for transformation to Linked Art. You will work through the ***Extract Collection Data - CSV*** notebook.

> Open the ***Extract Collection Data - CSV*** Jupyter notebook:
> ./workshop/1-extract-collection-data-csv.ipynb GitHub Binder

Extension Activities

There are additional notebooks available that take different collection data formats as their starting point, e.g. XML, JSON and HTML. As an extension activity, you can work through one or more of these notebooks.

Exercise 1b: Extract Collections Data to Linked Data - XML

In this exercise you will learn how to read in collection data in XML format and convert it to a Python dictionary, ready for transformation to Linked Art. You will work through the ***Extract Collection Data - XML* notebook**.

Open the ***Extract Collection Data - XML*** Jupyter notebook:
./workshop/1-extract-collection-data-xml.ipynb GitHub Binder

Exercise 1c: Extract Collections Data to Linked Data - JSON

In this exercise you will learn how to read in collection data in JSON format and convert it to a Python dictionary, ready for transformation to Linked Art. You will work through the ***Extract Collection Data - JSON*** notebook.

Open the ***Extract Collection Data - JSON*** Jupyter notebook:
./workshop/1-extract-collection-data-json.ipynb GitHub Binder

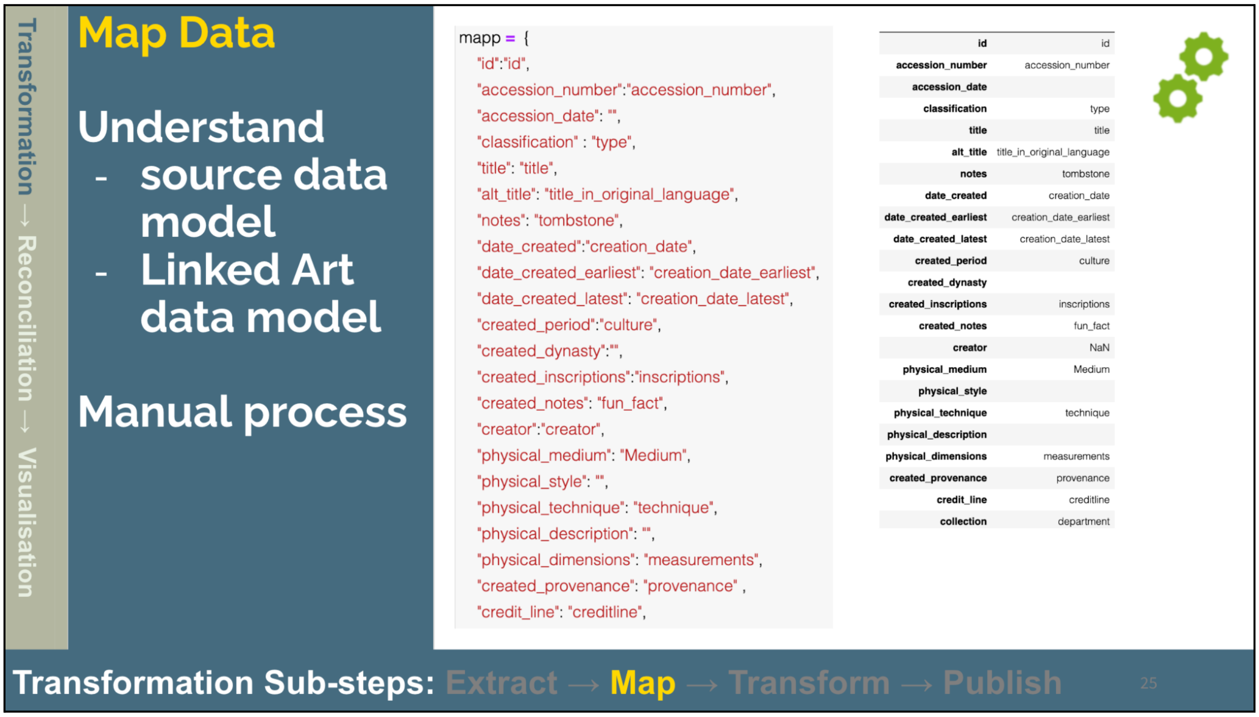
Exercise 1d: Extract Collections Data to Linked Data - HTML

In this exercise you will learn how to read in collection data in HTML format and convert it to a Python dictionary, ready for transformation to Linked Art. You will work through the ***Extract Collection Data - HTML*** notebook.

Open the ***Extract Collection Data - HTML*** Jupyter notebook:
./workshop/1-extract-collection-data-html.ipynb GitHub Binder

Exercise 2: Map Collection Data to Linked Art

In this exercise you will learn how to map collection data to the Linked Art data model. You will work through the ***Map Collection Data - CSV*** notebook.



Transformation → Reconciliation → Visualisation

Map Data

Understand

- source data model
- Linked Art data model

Manual process

```
mapp = {
  "id":"id",
  "accession_number":"accession_number",
  "accession_date": "",
  "classification" : "type",
  "title": "title",
  "alt_title": "title_in_original_language",
  "notes": "tombstone",
  "date_created":"creation_date",
  "date_created_earliest": "creation_date_earliest",
  "date_created_latest": "creation_date_latest",
  "created_period":"culture",
  "created_dynasty":"",
  "created_inscriptions":"inscriptions",
  "created_notes": "fun_fact",
  "creator":"creator",
  "physical_medium": "Medium",
  "physical_style": "",
  "physical_technique": "technique",
  "physical_description": "",
  "physical_dimensions": "measurements",
  "created_provenance": "provenance" ,
  "credit_line": "creditline",
```

id	id
accession_number	accession_number
accession_date	
classification	type
title	title
alt_title	title_in_original_language
notes	tombstone
date_created	creation_date
date_created_earliest	creation_date_earliest
date_created_latest	creation_date_latest
created_period	culture
created_dynasty	
created_inscriptions	inscriptions
created_notes	fun_fact
creator	NaN
physical_medium	Medium
physical_style	
physical_technique	technique
physical_description	
physical_dimensions	measurements
created_provenance	provenance
credit_line	creditline
collection	department

Transformation Sub-steps: Extract → Map → Transform → Publish

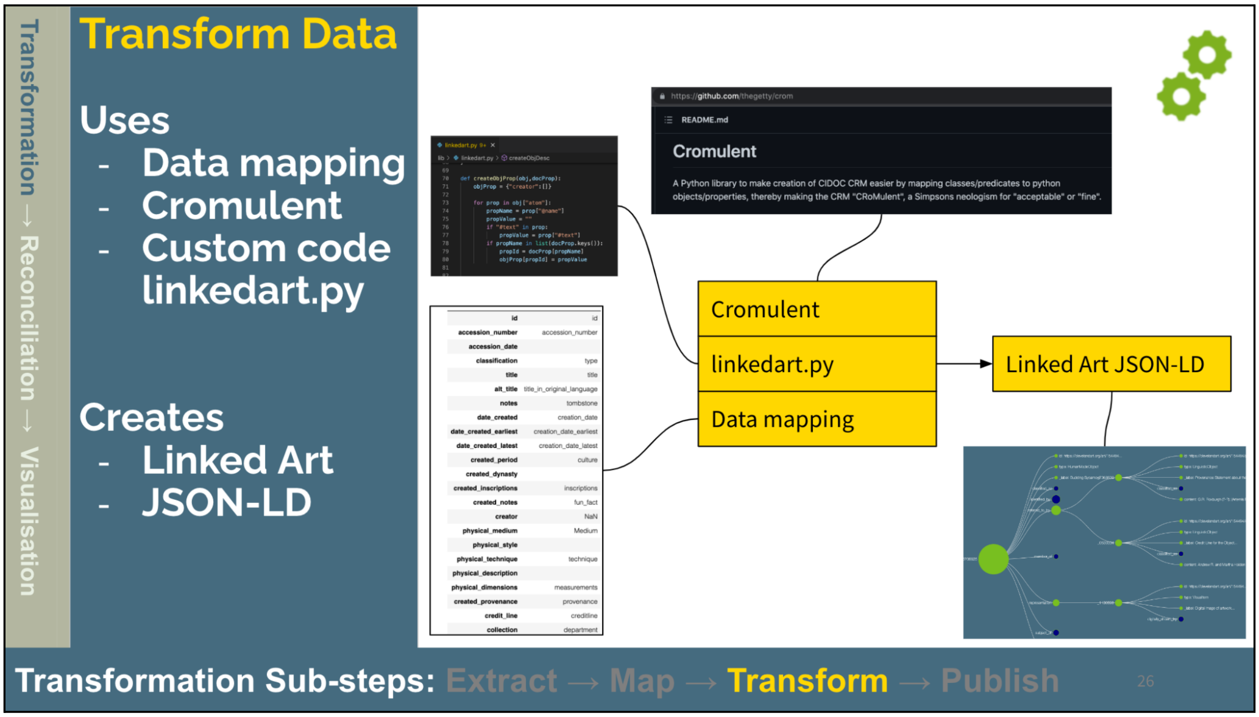
25

Open the ***Map Collection Data - CSV*** Jupyter notebook:
./workshop/2-map-collection-data-csv.ipynb GitHub Binder

Exercise 3: Transform Collections Data to Linked Art JSON-LD

In this exercise you will learn how to use the mapping created earlier to transform the collection data to Linked Art JSON-LD. You will work through the following notebooks:

- ***Transform Collection Data (CSV)***
  - Use the data mapping to map collection data to entities in Linked Art, producing Python dictionary
- ***Transform LinkedArt JSON-LD***
  - Use mapped data and script to transform mapped collection data to Linked Art JSON-LD



Transformation → Reconciliation → Visualisation

Transform Data

Uses
- Data mapping
- Cromulent
- Custom code linkedart.py

Creates
- Linked Art
- JSON-LD

linkedart.py 9+

lib > linkedart.py > createObjDesc

```python
def createObjProp(obj,docProp):
    objProp = {"creator":[]}

    for prop in obj["atom"]:
        propName = prop["@name"]
        propValue = ""
        if "#text" in prop:
            propValue = prop["#text"]
        if propName in list(docProp.keys()):
            propId = docProp[propName]
            objProp[propId] = propValue

```

id	id
accession_number	accession_number
accession_date	
classification	type
title	title
alt_title	title_in_original_language
notes	tombstone
date_created	creation_date
date_created_earliest	creation_date_earliest
date_created_latest	creation_date_latest
created_period	culture
created_dynasty	
created_inscriptions	inscriptions
created_notes	fun_fact
creator	NaN
physical_medium	Medium
physical_style	
physical_technique	technique
physical_description	
physical_dimensions	measurements
created_provenance	provenance
credit_line	creditline
collection	department

https://github.com/thegetty/crom

README.md

Cromulent

A Python library to make creation of CIDOC CRM easier by mapping classes/predicates to python objects/properties, thereby making the CRM "CRoMulent", a Simpsons neologism for "acceptable" or "fine".

Transformation Sub-steps: Extract → Map → **Transform** → Publish

26

Exercise 3a - Transform Collection Data (CSV) to JSON

Open the ***Transform Collection Data (CSV)*** Jupyter notebook: ./workshop/3a-transform-collection-data-csv.ipynb GitHub Binder

Exercise 3b - Transform JSON to LinkedArt JSON-LD

Open the ***Transform LinkedArt JSON-LD*** notebook Jupyter notebook: ./workshop/3b-transform-linkedart-json-ld.ipynb GitHub Binder

If you try the exercises with your own data, try exploring the data in the JSON-LD Playground

Exercise 4: Transform Cleveland Museum of Art Collections Data to Linked Art

THE CLEVELAND MUSEUM OF ART

In this exercise, you will use a notebook that contains all the steps described above to transform collections data from the Cleveland Museum of Art to Linked Data conforming to the Linked Art data model.

The exercise will demonstrate how to:

1. Extract data from source
2. Map data to the Linked Art data model
3. Transform data to a unified representation using Python code
4. Publish data as JSON-LD files

You will work through the Cleveland Museum of Art transformation code notebook.

> Open the ***CMA Simplified*** Jupyter notebook:
> ./workshop/4-transform-collection-data-cma.ipynb GitHub Binder

Exercise 5: Reconcile Collection Data using Linked Art

The Linked Art reconciliation notebook provides a step through the process of reconciling geographical place names that occur in the title of artworks by the artist, John Ruskin, using the tool OpenRefine.

The data reconciliation will enrich the collections data with an additional representation of the geographical coordinates of the place depicted in the artworks.

> Open the ***Reconcile Linked Art*** Jupyter notebook:
> ./workshop/5-reconcile-linked-art.ipynb.ipynb GitHub Binder

Exercise 6: Visualise Collection Data using Linked Art

In this exercise you will see how collection data transformed to a unified representation using the Linked Art data model, can be used as input to a data visualisation. We will demonstrate how to transform Linked Art JSON-LD to an input file for a timeline visualisation provided by KnightLab.

> Open the ***Visualise Linked Art*** Jupyter notebook:
> ./workshop/5-visualise-linkedart-timeline.ipynb GitHub Binder

Extension Activities

Explore Further Linked Art Code Notebooks

There are further Linked Art Code notebooks available that take collection data as input, and also demonstrate how to create a storymap visualisation.
https://github.com/tgra/Linked-Art

Explore Linked Art website

https://linked.art/
The Linked Art website contains information on the Linked Art data model and details of community activities that you may like to join.

Try Different Tools to Interact with Jupyter Notebooks

Various tools exist to interact with Jupyter Notebooks that you may like to try:

Visual Studio Code

https://code.visualstudio.com/docs/datascience/jupyter-notebooks
You can work with Jupyter Notebooks in Visual Studio Code

Google Colaboratory

https://colab.research.google.com/
Colab, or ‘Colaboratory’, allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Voila

https://voila.readthedocs.io/en/stable/index.html
Voilà allows you to convert a Jupyter Notebook into an interactive dashboard that allows you to share your work with others.

JSON-LD

https://json-ld.org/
Documentation on JSON-LD including information about code relevance are available.

JSON-LD Playground

Explore JSON-LD visually in the JSON-LD Playground e.g.
https://json-ld.org/playground/#startTab=tab-expanded&copyContext=true&json-ld=https%3A%2F%2Flinked.art%2Fexample%2Fobject%2F12