

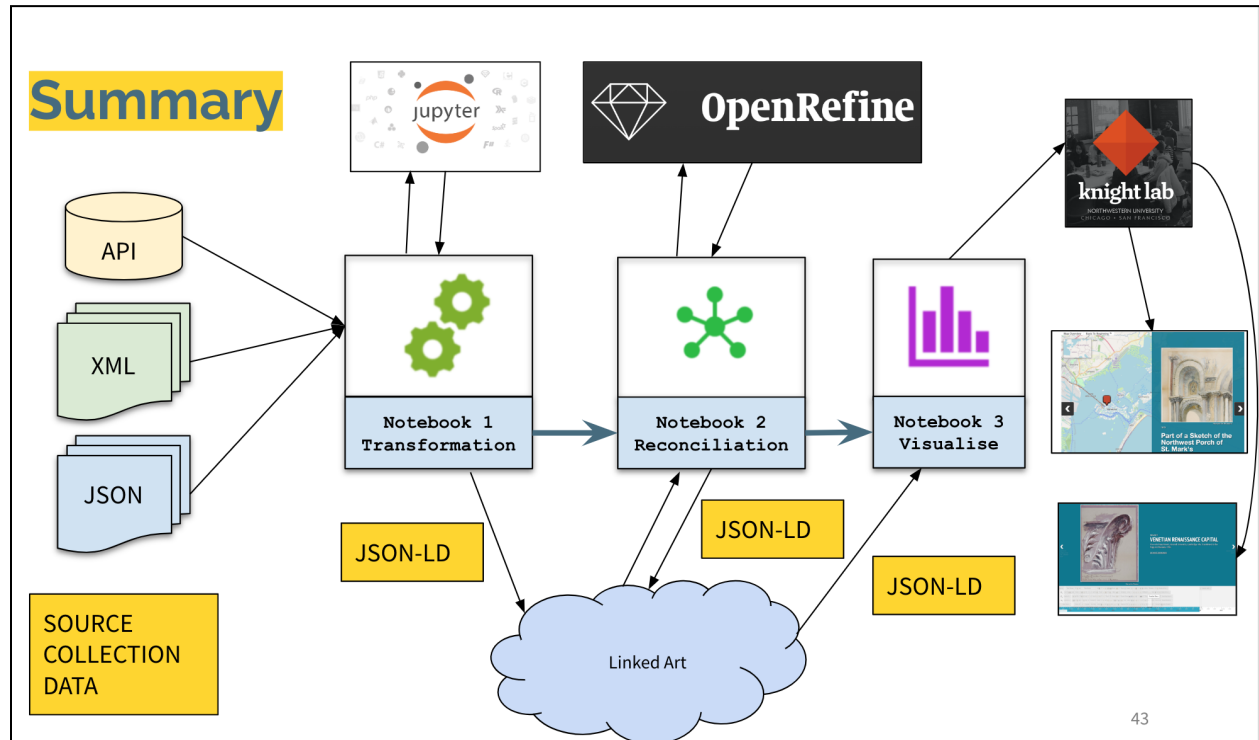
CONNECTING CULTURAL HERITAGE COLLECTIONS DATA WITH LINKED ART

Hands-on Session with Linked Art Notebooks - Exercise Sheet

Introduction	2
The Transformation process	2
How to navigate through the notebooks	3
Linked Art Jupyter Notebooks	3
Trusted Notebooks	3
Software	3
Anaconda	3
Related Documentation	4
Example Notebook: Minimum Representation in Linked Art	5
Exercise 1: Extract Collections Data	6
Exercise 1a: Extract Collection Data in CSV Format	6
Extension Activities	6
Exercise 1b: Extract Collections Data to Linked Data - XML	7
Exercise 1c: Extract Collections Data to Linked Data - JSON	7
Exercise 1d: Extract Collections Data to Linked Data - HTML	7
Exercise 2: Map Collection Data to Linked Art	8
Exercise 3: Transform Collections Data to Linked Art JSON-LD	9
Exercise 3a - Transform Collection Data (CSV) to JSON	9
Exercise 3b - Transform JSON to LinkedArt JSON-LD	9
Exercise 4: Transform Cleveland Museum of Art Collections Data to Linked Art	10
Exercise 5: Reconcile Collection Data using Linked Art	10
Exercise 6: Visualise Collection Data using Linked Art	11

Introduction

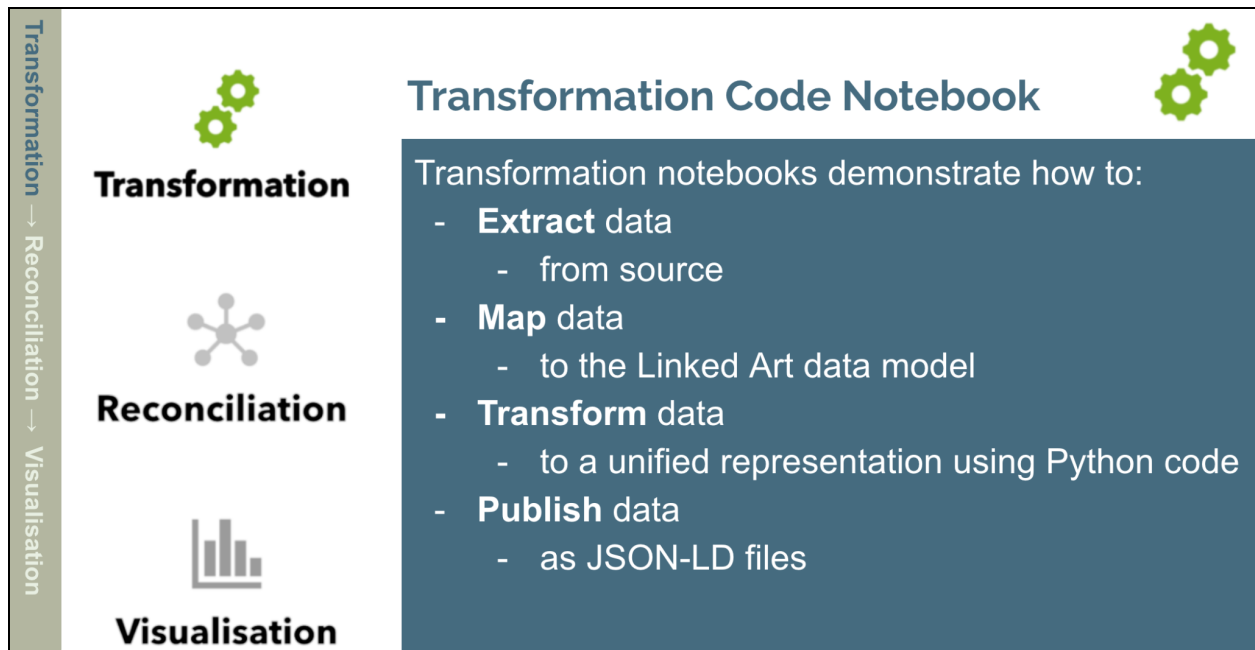
In this session, we will support you in a hands-on exercise looking at the processes involved in transforming cultural heritage data to Linked Data. We will use code notebooks to transform cultural heritage data using the Linked Art data model.



The Transformation process

The transformation process will be demonstrated across a number of notebooks:

- **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



How to navigate through the 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

Linked Art Jupyter Notebooks

You will also 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.

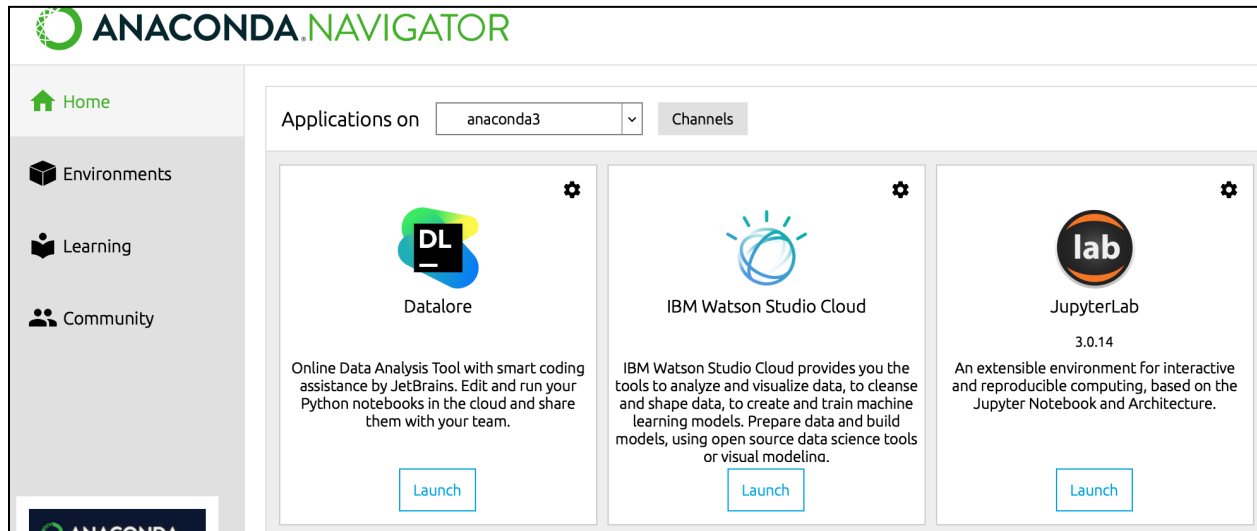
Anaconda

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



- 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

Related Documentation

From Session 1

- [Linked Art notebook slide deck](#)
- [Software installation instructions](#)

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

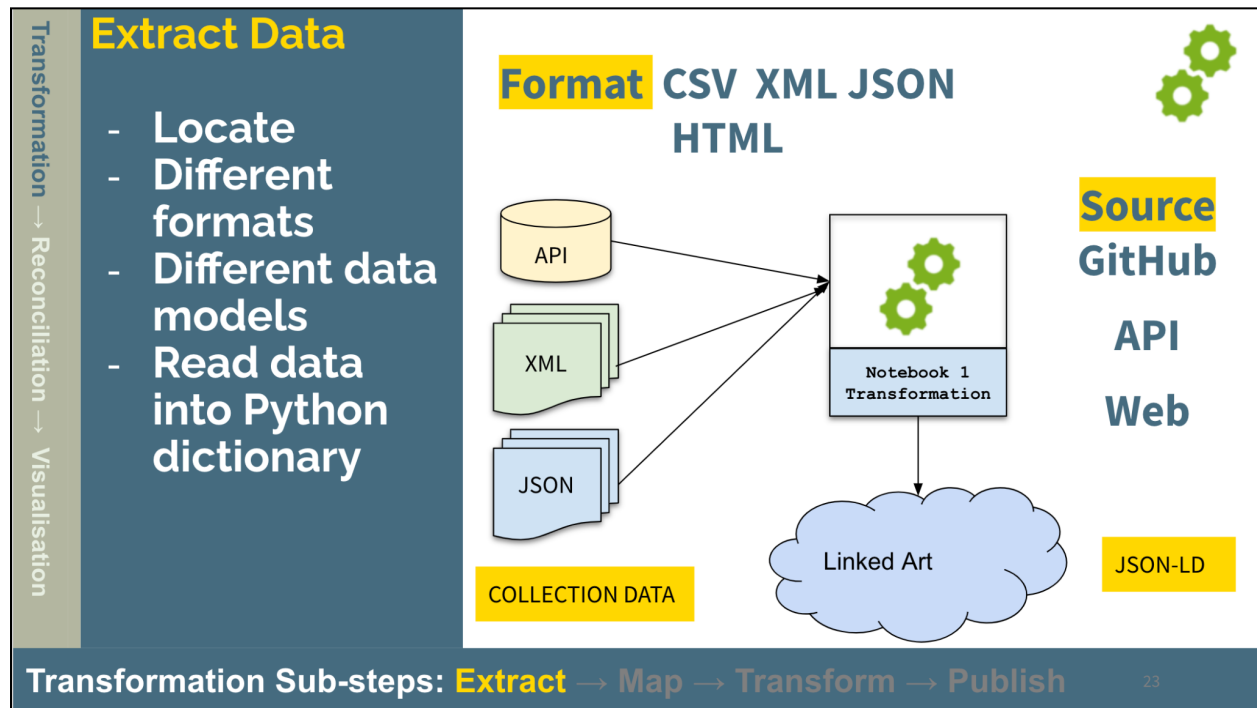
Open the ***Linked Art - Core Properties*** Jupyter notebook:

[./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.



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

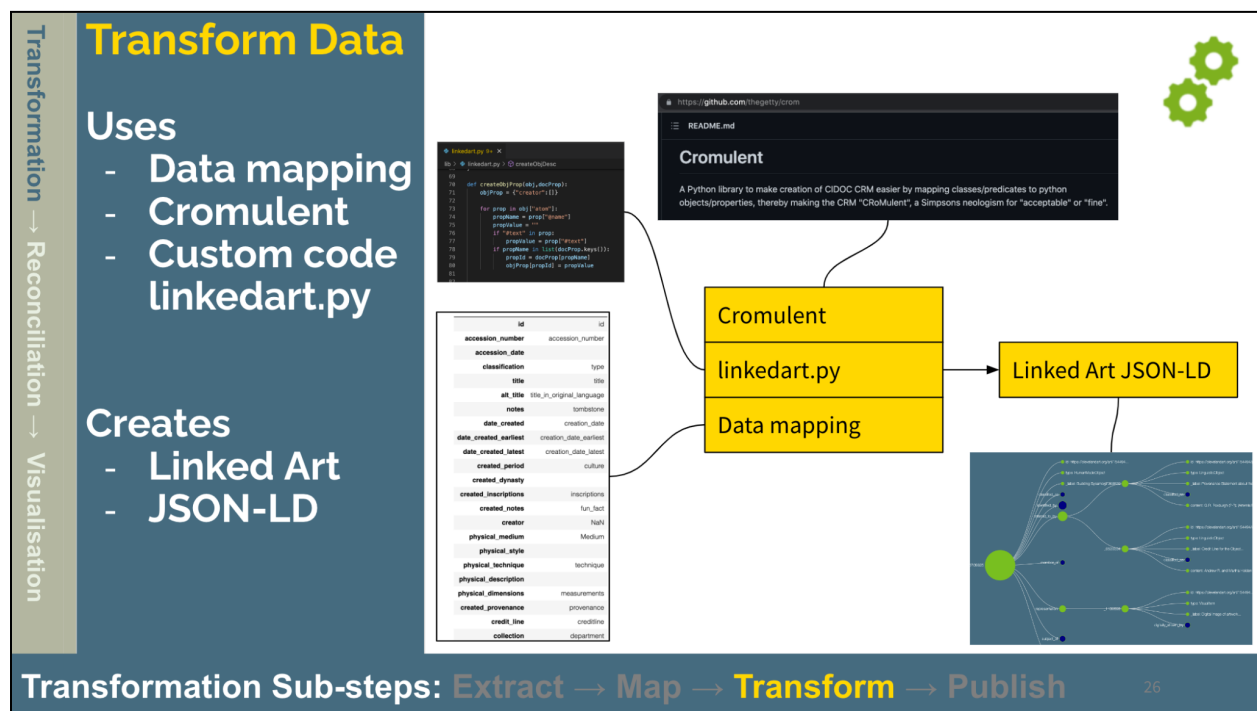
Open the **Map Collection Data - CSV** Jupyter notebook:

[./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



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** 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



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

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 **Reconcile Linked Art** Jupyter notebook:

[./workshop/5-reconcile-linked-art.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](#)