CapiTainS

A flexible tool suite for serving and managing interoperable corpora

Capitains.github.io

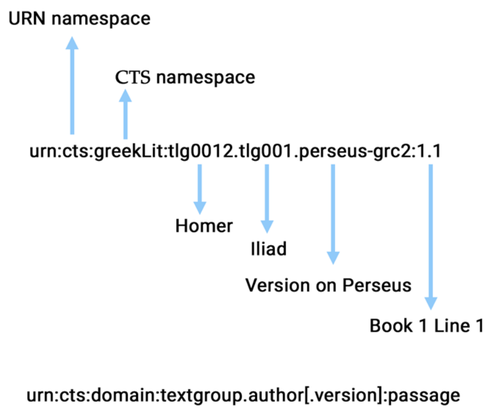
Canonical Text Services

The Canonical Text Services Protocol is a specification that "defines a network service for identifying texts and for retrieving fragments of texts by canonical reference expressed as CTS-URNs." CTS and CTS-URN provides an interoperable, open and persistent system for sharing text resources and parts of them on the web.

At the core of the CTS URN is the idea of representing texts as part of a graph, where nodes resolve to texts, objects or images, and the edges provide navigation between them. Text nodes themselves consist of citable nodes, with each node having the following properties:

- belongs to a specific version of a work in a FRBR-like hierarchy
- belongs to a citation hierarchy of 1 or more levels
- is ordered
- may have mixed content

Neel Smith and Chris Blackwell, Canonical Text Services ,
http://cite-architecture.github.io/



Ahab

Ahab provides new functionalities to CTS repositories, using CTS URNs as its basic structure. CTS has been designed for serving data flawlessly and in a persistent way. Ahab as a norm provides a new layer of functionalities :

- Curated passages : with curator defined display ranges for type of texts or for specific text, pagination using urns has never been so simple.
- Permalink : for structure using numerous inventories, retrieving a text and its inventory can be quite a challenge : permalink will retrieve automatically a version of a text and the inventory in which it is.
- Multiple passage request
- Full text search

Capitains Guidelines

Capitains Guidelines concern both the TEI encoding, the structure of directories in which those files are saved and the expression of text inventory metadata. They have been written so that the TEI files are self-containing their main metadata and the directories reflect the CTS data structure. All of those additions are TEI and Epidoc compliant.

For more informations :
http://capitains.github.io/pages/guidelines.html

URN Information

TEI	Epidoc
/TEI/text/@n	/TEI /text /body /div[@type] /@n

Data structure

```
data/
    |- textgroup
        |-__cts__.xml
        |- work
            |-__cts__.xml
            |- full-urn.xml
```

Citations

The way the text is divided should be expressed in the encodingDesc of the TEI file. In a refsDecl with an attribute @n="CTS", each level of CTS citation is represented by a cRefPattern with a @n attribute representing the type of the citation level (line, chapter, paragraph, etc.), a matchPattern representing the citation pattern, a replacementPattern with the xpath and value of n replaced by the regexp captured values in matchPattern.

```
<refsDecl n="CTS">
    <cRefPattern n="line" matchPattern="(.+).(.+)"
replacementPattern="#xpath(/tei:TEI/tei:text/tei:body/tei:div[@n='$1']//tei:l[@n='$2'])">
      <p>This pointer pattern extracts book and line</p>
    </cRefPattern>
    <cRefPattern n="book" matchPattern="(.+)"
replacementPattern="#xpath(/tei:TEI/tei:text/tei:body/tei:div[@n='$1'])">
      <p>This pointer pattern extracts book.</p>
    </cRefPattern>
</refsDecl>
```

CTS Inventory Metadata file

In Capitains, inventory files are cut into metadata files, which can be used then in InventoryMaker or others apps to create subinventory for the API. Each semantic level of a folder (ie phi1294, phi001…) must contain a file named __cts__.xml. as seen previously.

Group level

```
<ti:textgroup xmlns:ti="http://chs.harvard.edu/xmlns/cts" urn="urn:cts:latinLit:phi1294">
  <ti:groupname xml:lang="eng">Martial</ti:groupname>
</ti:textgroup>
```

Work level

```
<ti:work xmlns:ti="http://chs.harvard.edu/xmlns/cts" groupUrn="urn:cts:latinLit:phi1294"
urn="urn:cts:latinLit:phi1294.phi002">
  <ti:title xml:lang="eng">Epigrammata</ti:title>
  <!-- For each "text", either edition or translation, there should be a ti:edition or ti:translation node -->
  <ti:edition workUrn="urn:cts:latinLit:phi1294.phi002" urn="urn:cts:latinLit:phi1294.phi002.perseus-lat2">
    <ti:label xml:lang="eng">Epigrammata</ti:label>
    <ti:description xml:lang="eng">
      M. Valerii Martialis Epigrammaton libri / recognovit W. Heraeus
    </ti:description>
  </ti:edition>
</ti:work>
```

CAPITAINS TOOL SUITE

INTEGRATING

Using CTS based softwares or resources might be useful for most small projects, but building projects based software with project based problematics is the point of CTS implementation in diverses languages. Those implementations provide APIs, XML and Text level interactions.

Sparrow — My Capytain — Cavern

SERVING

Implementation of CTS using TEI-XML based resources comes from the Alpheios project (Alpheios.net). Being able to directly serve texts from their source file means an extremely enhanced ease of maintenance and curation for all types of DHers : scholar, engineer, archivist and librarian.

Inventory Maker — CTS API — Ahab

Nemo

Nemo is a UI centered software for serving CTS resources to your users. Using this either as a testing environment or as a small project production environment, you will be able to use your CTS resources easily and show them to your audience, be it restricted or complete.

TESTING

Drawing on software engineering best practices, we are building an architecture meant for continuous integrations: analogous to the way Travis integrates with Github, we are developing a customizable service that test individual files upon each contribution made to our public git repositories. The services can be configured to test and report status on a variety of checkpoints - from schema compliance to CTS-ready markup.

Hook

Hook is the application required to connect github activities with tests and reports. It provides information such as the number of CTS compliant texts found, the amount of unit tests covered and the full compliancy of one repository with Capitains guidelines and TEI or Epidoc schemes.

Hook Test

HookTest is both a library for Python and a commandline tool providing a test system making use of multithreading. It provides tests for texts and metadata files.

Institutions :

PERSEIDS — A Collaborative Editing Platform for Source Documents in Classics

Perseus Digital Library — Tufts University Medford, MA,USA — The University of Leipzig Leipzig, Germany

Digital Humanities — Universität Leipzig

Funders

The Andrew W. Mellon Foundation

Alexander von Humboldt Stiftung/Foundation

Credits

Author : Thibault Clerice
Designer : Rafael Lopes