Distributed Text Services

H. Cayless [1] M. Romanello [2]

[1]Duke University Libraries
[2]SARI, University of Zurich, Switzerland

25.09.2025

Credits

The slides are a mix of previous presentations, given at DH2024 and Digital Classicist London (2025), written by H. Cayless, T. Clérice, M. Romanello, and I. Scott.

`https://cayless.net/presentations/DTS_2025.pdf`

Table of Contents

Canonical Text Services

- First mooted at a workshop at the Harvard Center for Hellenic Studies in 2003 by Neel Smith and Chris Blackwell.
- Took the TLG and PHI digital publications of the Greek and Latin classical canon as a model.
- Developed (along with the subsequent CITE architecture) as a core support technology for the Homer Multitext Project.
- URN-based referencing system for authors, works, editions, and edition parts down to the character level.
- Provides ability to discover collection content, find out how to reference components of a text, retrieve those components.

Why DTS ?

- CTS protocol is an XML-RPC-style web service.
- The text inventory is an unwieldy structure when you have very large collections.
- The “canonical” frame makes an assumption that you’re dealing with texts with a standard referencing system.
- The FRBR-influenced URN scheme doesn’t necessarily fit all the scenarios we find with texts.
- More of a project than a standard: no room for building on top of CTS and feedback.

DTS Requirements

- Support CTS URNs (at least the text identifier part)
- Provide commensurate:
  - Discovery capability for collections
  - Referencing / navigation facility
  - Text / text fragment retrieval
- Be more flexible and democratic, update technologies, and try to be an implementable standard.

The birth of DTS

2015	"Distributed" Text Services first floated at the 2015 Triangle Scholarly Communications Institute. Chris Blackwell and Bridget Almas were organizers. One of their group dropped out at the last minute, so they asked Hugh to join. ▶ A group formed to discuss the idea and monthly meetings began. Bridget Almas agreed to chair. ▶ Lots of ideas, lots of discussion, some quite passionate arguments.
2017	Jonathan Robie takes over as chair.
2018	funded by a mini grant from the Pelagios Network, we held an in person meeting at Duke where we made a lot of progress on the public draft standard, which was finalized in July.
2018–2024	Periodic online meetings, slow but mostly steady progress.

DTS History

2019	Workshop in Hamburg
2021	Citation Structures for TEI released as part of Guidelines v4.2.0.
2021	Online Hackathon.
2022–2023	Things are stalling a bit. We decide we need another in-person sprint, but it takes a long time to come together.
2024	(March) in-person meeting at Duke. Massive progress; 1.0-alpha proposal released.
2025	July 11 release candidate; aiming for 1.0 in October.

The Survivors

After three days in a basement conference room writing and correcting specifications...

Members of the committee

Some major contributors on this journey include:

- Bridget Almas
- Pietro Liuzzo
- Matteo Romanello
- Jeffrey Witt

Current Technical Committee:

- Hugh Cayless
- Thibault Clérice
- Jonathan Robie
- Ian Scott

Table of Contents

Table of Contents

What is DTS then?

- `https://www.w3id.org/dts`
- Structured text collections
  - Hierarchical structuring of the collection
  - Structuring of each member text with a citation scheme
- Access
  - Discovering the collection's structure
  - Discovering a member's citation scheme structure
  - Retrieving metadata about a specific collection, subcollection, text, or passage
  - Retrieving the textual content of a text or passage
- Standardized
  - Allowing clients to access any collection/text without retooling
  - Allowing publishers/editors to make collections/texts FAIR

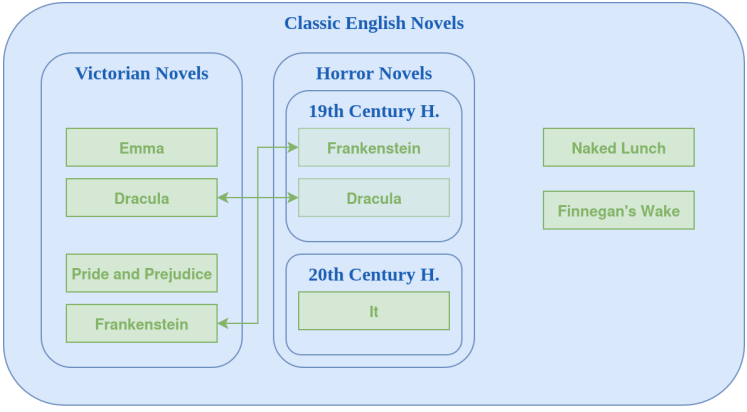
Use cases

- Use a standard reader client to explore any text or collection
- Combine or aggregate diverse collections
- Ingest text for data processing

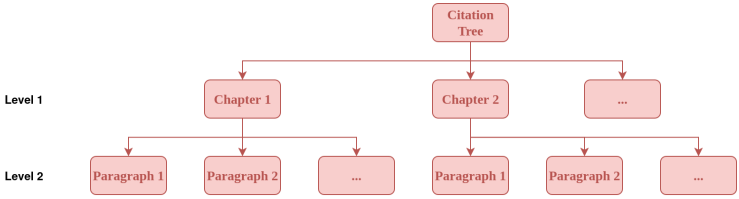
Vocabulary

Resource	A text document.
Collection	A named aggregation of digital Resources. Collections may contain other Collections or Resources.
Citable Unit	A portion of a Resource identified by a reference label (string).
Citation Tree	A tree of reference labels corresponding to the hierarchical structure of a Resource.

Collection and Resource



Citation Trees and Citable Unit



Structure: the DTS Endpoints

Entry

- JSON-LD
- URL templates for the other three endpoints.
- DTS version supported

Collection

- JSON-LD
- Collection structure
- Metadata about collection, subcollections, resources

Navigation

- JSON-LD
- Resource's internal structure

Document

- Textual content of a Resource or citable unit

Table of Contents

Entrypoint

- Key information about how to navigate the other API endpoints
  - URI Templates for the other three endpoints
  - Supported DTS version
- DTS does not restrict the URLs used by implementations
  - Flexible endpoint names
  - Flexible path and parameter arrangement
  - Allows for static sites
- Request example
  - https://example.org/api/

Entrypoint example

```
{
  "@context": "https://distributed-text-services.github.io/specifications/context/1.0rc1".json",
  "dtsVersion": "1.0rc1"",
  "@id": "/api/dts/",
  "@type": "EntryPoint",
  "collection": "/api/dts/collection/{?id,page,nav}",
  "navigation" : "/api/dts/navigation/{?resource,ref,start,end,down,tree,page}",
  "document": "/api/dts/document/{?resource,ref,start,end,tree,mediaType}"
}
```

Table of Contents

Collection endpoint

- Role
  - Expose the structure of the collection
  - Expose metadata about the collection and its subcollections
- Collections can have nearly any structure
  - Tree of nested Collection and Resource objects
  - Branches can have varying "depths"
  - Resources can appear in more than one Collection

URI Examples

Top level `https://example.org/api/collection/`

Child Collection `https://example.org/api/collection/?id=horror-novels`

Child Resource `https://example.org/api/collection/?id=Dracula`

Response

- Returns a Collection or Resource object (JSON-LD)
- Returned Collection or Resource must include
  - @id, @type, dtsVersion, @context
  - title
  - collection (URL template)
  - totalParents, totalChildren
- May include a "member" list of children
  - Either Collection (subcollections) or Resource (documents)
  - Can carry parent instead of children
- If the object is a Resource
  - URLs for the Navigation and Document endpoints

Example Response

```
{
  "@context":
  "https://distributed-text-services.github.io/specifications/context/1.0rc1".json",
  "@id": "general",
  "@type": "Collection",
  "collection": "https://example.org/api/collection/{?id,page,nav}",
  "dtsVersion": "1.0rc1"",
  "totalParents": 0,
  "totalChildren": 8,
  "title": "Classic English Novels",
  "member": [
    {
      "@id" : "horror-novels",
      "title" : "Horror Novels",
      "@type" : "Collection",
      "collection": "https://example.org/api/collection/?id=horror-novels{&page,nav}",
      "totalParents": 1,
    }, /* … */
  ]
}
```

Collection Response Can Also Include...

- Description
- Metadata objects
  - dublinCore
  - extensions (any scheme)
- Additional information on a Resource object
  - download (URL)
  - citeStructure (list of citationTree objects)
  - mediaTypes

Response with enriched metadata

```
{
  "@type": "Collection",
  /* … */
  "description": "A collection of horror novels"
  "dublinCore": {
    "publisher": ["École Nationale des Chartes"],
    "title": [
      {"lang": "en", "value": "Horror Novels"}
    ]
  },
  "extensions": [{
      /* metadata in other schemes */
  }]
}
```

Response with Resource

```
{
  "@type": "Resource",
  /* … */
  "navigation":
    "https://example.org/api/navigation/{?resource,ref,start,end,down,tree,page}",
  "document":
    "https://example.org/api/document/{?resource,ref,start,end,tree,mediaType}"
  "description": "A collection of horror novels",
  "dublinCore": { /* DCT metadata */ },
  "extensions": [ /* metadata in other schemes */ ],
  "download": "https://example.org/downloads/?id=dracula",
  "mediaTypes": ["application/tei+xml", "text/html"],
  "citationTrees": [
  /* … */
  ]
}
```

Citation Tree

```
"citationTrees": [
  {
    "@type": "CitationTree",
    "citeStructure" : [
      {
        "@type": "CiteStructure",
        "citeType": "poem",
        "citeStructure": [
          {
            "@type": "CiteStructure",
            "citeType": "line"
          }
        ]
      }
    ]
  }
]
```

Citation Trees

```
"citationTrees": [
  {
    "@type": "CitationTree",
    "identifier": "poems"
    "citeStructure" : [
        /* … */
    ]
  },
  {
    "@type": "CitationTree",
    "identifier": "pages"
    "citeStructure" : [
        /* … */
    ]
  }
]
```

Table of Contents

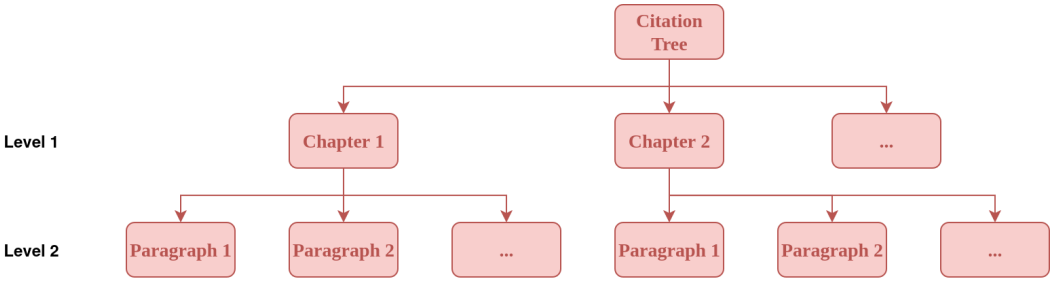
Overview

- Most confusing endpoint!
  - Remember our CTS origins—need to know structure if you want alignment.
- Role
  - Expose the internal structure of a Resource (one member of collection).
  - Expose metadata about citable units in the Resource.
  - Returns a Navigation object.
- Why not just give the TEI XML text?
  - Navigation endpoint tells you what you're dealing with.
  - Tells us where to find citable units in the XML.
  - Tells us the citation structure without having to parse the XML.

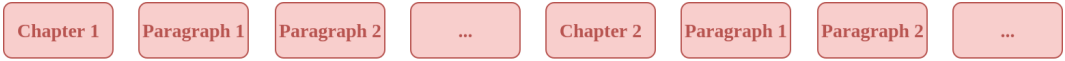
Navigation Response Object

- member
  - A section of the citation tree relative to ref/start/end.
  - A flat array of citable units in document order.
  - Can specify how deep.
  - Can specify alternate citation trees, e.g. sections / pages.
- basic contents
  - @id, @type, dtsVersion, @context
- passage
- navigation
- resource
- ref or start/end

Response in Document Reading Order



Response in Document Reading Order



Navigation Request Examples

- `https://example.org/api/navigation/?resource=dracula&down=1`
  - All the top-level citableUnits in the document's citation tree (`down=1`)
  - Use cases
    - Table of contents
    - Next/Previous navigation links

Result

```
{
  /* ... */
  "@type": "Navigation",
  "@id": "https://example.org/api/dts/navigation/?resource=https://en.wikisource.org/wiki/
  Dracula&down=1",
  "resource": {
    "@id": "https://en.wikisource.org/wiki/Dracula", 7
    "@type": "Resource", /* ... */
    "citationTrees": [ /* ... */ ]
  },
  "member": [
    {
      "identifier": "C1",
      "@type": "CitableUnit",
      "level": 1,
      "parent": null,
      "citeType": "Chapter",
    }, /* ... */
  ]
}
```

Navigation Request Examples

- `https://example.org/api/navigation/?resource=dracula&down=1`
- `https://example.org/api/navigation/?resource=dracula&down=2`
  - The document's whole citation tree down to the second level)
  - Use cases
    - Table of contents
    - Building a navigation menu
    - Determine structure prior to requesting text for data processing

Result

```
"member": [ {
    "identifier": "C1",
    "@type": "CitableUnit",
    "level": 1,
    "parent": null,
    "citeType": "Chapter",
  },{
    "identifier": "C1.E1",
    "@type": "CitableUnit",
    "level": 2,
    "parent": "C1",
    "citeType": "Journal Entry",
  },{
    "identifier": "C1.E2",
    "@type": "CitableUnit",
    "level": 2,
    "parent": "C1",
    "citeType": "Journal Entry",
  }, /* ... */ ],
```

Navigation Request Examples

- …/navigation/?resource=dracula&down=1
- …/navigation/?resource=dracula&down=2
- …/navigation/?resource=dracula&ref=C2&down=-1
  - All the descendants below one citableUnit (C2)
  - Use cases
    - Build a dynamic menu of subsections when navigating a particular chapter
- …/navigation/?resource=dracula&start=C1.E1&end=C2.E2&down=2

Table of Contents

Overview

- Exposes the textual content of a Resource or citable unit
- Should return TEI/XML as the default (but this is not a "must")
  - May expose other formats and media types as well
- Should also provide a link header referencing the collection endpoint
  - Link: `https://example.org/api/collection/?id=Dracula`; rel="collection"
  - (obviously can't be done if this is a static site, hence optional)

Example requests

- Type of requests

  An entire document …/document/?resource=Dracula

  One citable unit …/document/?resource=Dracula&ref=C1

  A range …/document/?resource=Dracula&start=C1.E1,P1&end=C1.E1,P2

- Requests can specify an alternate citation scheme
  - https://example.org/api/document/?resource=Dracula&tree=pages&ref=5
  - If the implementation exposes one for the Resource
  - Alternate citation trees exposed at Navigation and Collection
- Requests can specify an alternate media type
  - https://example.org/api/document/?resource=Dracula&mediaType=text/html

Table of Contents

DTS API Backend implementations

Known Client Side implementations:

- DTS Aggregators
- DoTS Vue

Known API implementations that are up to date:

- Dapytains (Python)
- DoTS (BaseX, PSL-École nationale des Chartes)
- py-dts by Renato Diaz (UNIL/FTSR)
- Implementation of KDL (Javascript [Doubt on version])
- Implementation of Heidelberg
- TEI Publisher implementation (eXistDB, no navigation implementation)

Dapytains

`https://github.com/distributed-text-services/MyDapytains`

- Complete implementation Distributed Text Services, up to date.
- Can be used as a webapp OR as a way to generate a static DTS API as it includes a way to parse data easily.
- Allows for mapping to at least one level of “milestone-type“ nodes, which enables epigraphic and papyrological corpora, and in general document-oriented citation trees.
- Python, but the algorithm is portable to other languages, as it mostly uses XPATH, and some xQuery (albeit mostly to deal with issues in Saxon CHE Python implementation)
- *Lightweight*, unlike MyCapytains (we learned from our mistakes): only focus on getting data in and out. For the webapp, metadata are stored in a sqlite database.

- Written in XQuery + some bash scripts for project management
- Supports BaseXML as the backend
- Input data must be in XML/TEI format
- Uses `<citeStructure/>` tags to declare retrievable passages for the Navigation endpoint.
- Documentation: `https://chartes.github.io/dots_documentation/`
- Code: `https://github.com/chartes/dots`

DraCor: Drama Corpora

`https://dracor.org/`

- 21 corpora of theater plays in 14 languages/dialects (total 3.7k plays)
- It covers a period of about 2500 years (472 BC – 2017 AC)

DTS API – `https://staging.dracor.org/api/v1/dts`

- Builds upon an eXistDB backend
- Implementation is (almost) up to date with the standard (version `unstable`).

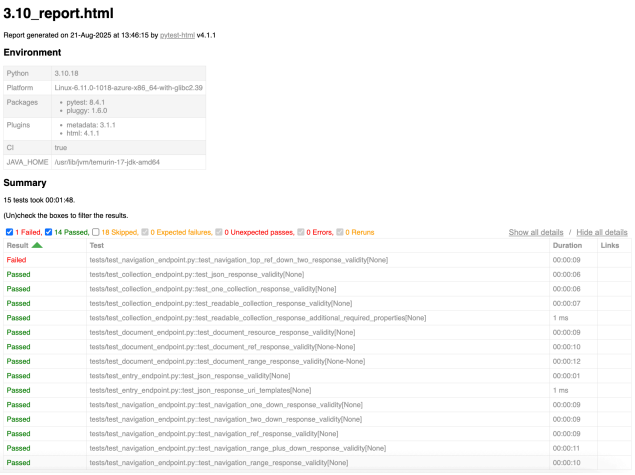
DTS Aggregator

- Client-side project demonstrating the **distributed side** of DTS
- Fed by a JSON registry of known DTS APIs
- Allows users to browse seamlessly the whole network of DTS APIs
- Ultimately, it could provide search capacities (once defined in DTS' specs)

DTS Validator

DTS validator – `https://github.com/mromanello/dts-validator/`

- Suite of tests (`pytest`) to validate DTS implementations
- Work-in-progress:
  - Ready to use, up to date with unstable.
  - Ready for CI/CD (already used by Dapytains)
- Features:
  - Validation of local/remote DTS API or examples from documentation
  - Validates JSON responses against JSON schemas
  - Ensures appropriate errors are raised (e.g. for invalid parameter combinations)
  - Generates HTML validation report
  - Can be used as a barebone Python DTS client
- Planned features:
  - Validation of JSON-LD/RDF via SHACL shapes



3.10_report.html

Report generated on 21-Aug-2025 at 13:46:15 by pytest-html v4.1.1

Environment

Python	3.10.18
Platform	Linux-6.11.0-1018-azure-x86_64-with-glibc2.39
Packages	• pytest: 8.4.1 • pluggy: 1.6.0
Plugins	• metadata: 3.1.1 • html: 4.1.1
CI	true
JAVA_HOME	/usr/lib/jvm/temurin-17-jdk-amd64

Summary

15 tests took 00:01:48.

(Un)check the boxes to filter the results.

1 Failed, 14 Passed, 18 Skipped, 0 Expected failures, 0 Unexpected passes, 0 Errors, 0 Reruns — Show all details / Hide all details

Result	Test	Duration	Links
Failed	tests/test_navigation_endpoint.py::test_navigation_top_ref_down_two_response_validity[None]	00:00:09	
Passed	tests/test_collection_endpoint.py::test_json_response_validity[None]	00:00:06	
Passed	tests/test_collection_endpoint.py::test_one_collection_response_validity[None]	00:00:06	
Passed	tests/test_collection_endpoint.py::test_readable_collection_response_validity[None]	00:00:07	
Passed	tests/test_collection_endpoint.py::test_readable_collection_response_additional_required_properties[None]	1 ms	
Passed	tests/test_document_endpoint.py::test_document_resource_response_validity[None]	00:00:09	
Passed	tests/test_document_endpoint.py::test_document_ref_response_validity[None-None]	00:00:10	
Passed	tests/test_document_endpoint.py::test_document_range_response_validity[None-None]	00:00:12	
Passed	tests/test_entry_endpoint.py::test_json_response_validity[None]	00:00:01	
Passed	tests/test_entry_endpoint.py::test_json_response_uri_templates[None]	1 ms	
Passed	tests/test_navigation_endpoint.py::test_navigation_one_down_response_validity[None]	00:00:09	
Passed	tests/test_navigation_endpoint.py::test_navigation_two_down_response_validity[None]	00:00:09	
Passed	tests/test_navigation_endpoint.py::test_navigation_ref_response_validity[None]	00:00:09	
Passed	tests/test_navigation_endpoint.py::test_navigation_range_plus_down_response_validity[None]	00:00:11	
Passed	tests/test_navigation_endpoint.py::test_navigation_range_response_validity[None]	00:00:10	

HookTest(V2)

`https://github.com/cllg-project/hooktest`

- To some extent, independent from Distributed Text Services.
- Ability to deal with CiteStructure TEI Elements and test them beyond what schema languages can do.
- Ability to evaluate both repositories (with catalog metadata) and TEI files (with CiteStructure) and provide a clear report.

Table of Contents

Distributed Text Services 1.0 Release Candidate Published – July 11, 2025
After years of development, independent implementations, interoperability testing, and a public comment period, the Distributed Text Services Technical Committee is pleased to announce the publication of the Distributed Text Services 1.0 Release Candidate (RC1).
We strongly encourage you to implement DTS now and share your feedback. The specification is feature-complete and stable, and we expect only minor clarifications before the final 1.0 release (planned for October 10th, 2025). Early adoption will ensure your projects are ready and help the community validate the standard in real-world use.