# Distributed Text Services (DTS)
# A Community-driven Text Interoperability API

Specs: w3id.org/dts

Demo: web + notebooks

FAQ page

## What is DTS?

**Goal**: enable machine-actionable operations across a wide variety of TEI texts.

As **DTS strives for genericity**, it does not dictate:

- how collections should be organized
- type of persistent identifiers to use
- what ontologies to use for metadata
- how the texts are structured.

---

## Why DTS?

DTS provides a **standard** way for clients to interact with collections of TEI documents. A standard API allows users to access many text collections using the **same client software**. It also allows editors to publish text collections in a **usable** way that existing clients can use.

---

## A Generic Text API

DTS supports:

- Multi-level/competing text hierarchies
- Any identifier scheme (*)
- Non-homogeneous text hierarchies

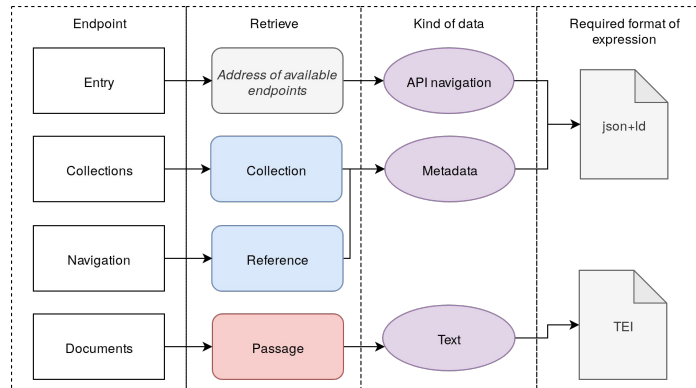## Endpoints & Operations

*Collections* endpoint:
- List available collections
- Support for pagination & virtual collections
- Navigate between nested collections

*Navigation* endpoint:
- List available text passages

*Documents* endpoint:
- Retrieve the text of a document



## DTS-compliant resources

*APIs*

ENC

Alpheios

Perseids

EDH

Beta masāḥǝft

*Implementations*

TEI publisher client/server

Perseids DTS API

Nautilus + MyCapytain

## Classes of objects in DTS

**Collections:**

- Hold metadata
- Serve as a catalog
- May be:
    - nested
    - readable
    - chunked into citable textual elements

**References:** hold identifiers & metadata for subsections of documents

**Passages:** hold textual content

---

## DTS Community

46 **Members**

**Communication** via Google Group list

**Technical exchanges**: GH issues

**Minutes**: meeting notes GH repo

Want to join? Subscribe to the Google Group