wac-ldp architecture (status: draft)

Michiel de Jong, Aaron Coburn - August 1, 2019

Purpose

This document describes the code structure of inrupt's wac-ldp component.

Legend

The architectural diagram follows standard UML notation.

For more specific symbols that are not part of UML, Node.js/JavaScript/TypeScript conventions were used as follows:

T? represents a value that is either not present or a value of type T.

Promise<T> represents a value that will asynchronously resolve to a value of type T.

Readable<T> represents an asynchronous one-time readable stream of values of type T.

Buffer is an in-memory buffer of bytes, possibly with a character encoding.

Diagram

StoreManager \ll interface \gg StoreManager + delete (url: URL): Promise < void> + exists (url: URL): Promise < boolean > + getResourceData (url: URL): Promise<ResourceData> + addQuad (quad: Quad): Promise<void> + deleteMatches (pattern: Pattern): Promise<void> + match (pattern: Pattern): Promise<Array<Quad» + subjectsMatching (pattern: Pattern): Promise<Array<RdfJsTerm» + predicatesMatching (pattern: Pattern): Promise<Array<RdfJsTerm» + objectsMatching (pattern: Pattern): Promise<Array<RdfJsTerm» + getRepresentation (url: URL, options?: any): Promise<ResourceData> + setRepresentation (url: URL, metaData: ResourceData): Promise<void> + load (url: URL): Promise<void> + save (url: URL): Promise<void> + patch (url: URL, sparqlQuery: string, appendOnly: boolean): Promise<void> + flushCache (url: URL): void