JSON-LD Support

The primary driver of this development is the BrownDog project.

Current sprint is JSON-LD-2.

As of June 8th the following need to be implemented

1. When adding metadata check if context already exists and create new if it doesnt
2. Ability to edit context
3. Ability edit metadata
4. Create GUI to manipulate this new metadata
5. Store extractor metadata (on top of context metadata)
    a. store external services and libraries used
6. Create example extractor using this new representation
7. Add heartbeat for extractors that sends over extractor metadata and updates the heartbeat

key	summary	assignee	status	resolution

⚠ Jira project doesn't exist or you don't have permission to view it.

View these issues in Jira

Example json-ld document:

https://opensource.ncsa.illinois.edu/stash/projects/CATS/repos/clowder/browse/doc/src/json/metadata.js?at=refs%2Fheads%2Ffeature%2Fjsonld

Available API endpoints:

POST	/api/contexts	@api.ContextLD.addContext()	
GET	/api/contexts/:id	@api.ContextLD.getContextById(id: UUID)	
GET	/api/contexts/:name/context.json	@api.ContextLD.getContextByName(name: String)	
DELETE	/api/contexts/:id	@api.ContextLD.removeById(id: UUID)	
POST	/api/datasets/:id/metadataJsonLD	@api.Datasets.addMetadataJsonLD(id: UUID)	
GET	/api/datasets/:id/metadataJsonLD	@api.Datasets.getMetadataJsonLD(id: UUID)	
POST	/api/files/:id/metadataJsonLD	@api.Files.addMetadataJsonLD(id: UUID)	
GET	/api/files/:id/metadataJsonLD	@api.Files.getMetadataJsonLD(id: UUID)	

Comments:

1. Extractor context definitions are similar to user specified metadata, they both provide what kind of metadata will be added
    a. How much flexibility do extractors get when adding metadata?

Open questions:

1. Should ContextLD and Metadata services be combined into one?

Extractor info tentative model:

```
case class ExtractorInfo (
  id: UUID,
  name: String,
  description: String,
  creator: User,
  version: String,
  lastSeen: Date,
  contexts: List[UUID],
  external_services: List[URL],
  libraries: List[String],
  bibtex: List[String]
)
```

Example body of post from extractor when registering itself. This is a POST to http://localhost:9000/api/extractors. You can list extractors with GET http://localhost:9000/api/extractors and get info about a specific one with http://localhost:9000/api/extractors/559c39557d840f25a725e4be. If the entry already exists it will be updated.

```
{
  "name": "ncsa.dbpedia",
  "version": "0.1",
  "description": "Simple JSON-LD extractor to extract information from a text file using named-entity
recognition and dbpedia.",
  "author": "Luigi Marini <lmarini@illinois.edu",
  "contributors": [],
  "contexts": [],
  "repository": {
    "repType": "git",
    "repUrl": "https://opensource.ncsa.illinois.edu/stash/scm/cats/extractors-dbpedia.git"
  },
  "external_services": [
    "http://live.dbpedia.org/sparql"
  ],
  "libraries": [
    "nltk"
  ],
  "bibtex": [
    "book{BirdKleinLoper09, author = {Steven Bird and Ewan Klein and Edward Loper}, title = {{Natural Language
Processing with Python}}, publisher = {O'Reilly Media}, year = 2009}"
  ]
}
```

Research

- Implementing Transitive Credit with JSON-LD
  - http://arxiv.org/abs/1407.5117
    - https://www.researchgate.net/publication/279968805_Report_on_the_Second_Workshop_on_Sustainable_Software_for_Science_Practice_and_Experiences_%28WSSSPE2%29