This item is the archived peer-reviewed author-version of:

Scaling data streams with Catmandu and Linked Data Fragments

Patrick Hochstenbach and Ruben Verborgh

In: Proceedings of the European Libraries Automation Group Conference, 2015.

http://elag2015.org/program/scaling-data-streams-with-catmandu-and-linked-data-fragments/

Scaling data streams with Catmandu and Linked Data Fragments

Name of presenters: Patrick Hochstenbach and Ruben Verborgh

About the presenters

Patrick Hochstenbach is a digital library architect at Ghent University Library.

Ruben Verborgh is Postdoctoral Researcher in Semantic Hypermedia at iMinds in Ghent.

Intended audience

Programmers, Librarians and Lukas Koster

Abstract

Data streams are a prevalent source of information in a wide range of applications, e.g. indexing catalog data, statistical reporting, and data augmentation. Each library product has its own suite of tools to pre- and postprocess data connected to their knowledge bases when it be catalogs, search engines or library portals.

In this context, Catmandu has taken the task to explore technical ways to make data stream processing independent from their library product. In Catmandu, data streams are expressed as a JSON model that can be queried with the JSONPath query language and transformed with an ETL language called Fix. Library protocols and formats such as OAI-PMH, Z39.50, MARC and MAB can serve as inputs. Databases such as Solr, ElasticSearch, MongoDB or formats such as Atom and RDF can act an output. One of the main goals of Catmandu is to define a common, but extensible model for data stream processing. This model can serve as input for several types of library ETL (Extract-Transform-Load) engines that can exchange recipes, and best practices of library data processing.

As a special use case, Catmandu imports and produces Linked Data. The Web is full of high-quality Linked Data that can serve as input for our library processes. But in general we can't reliably query it. Public SPARQL endpoints are often unavailable because they need to answer many unique queries. The Linked Data Fragments conceptual framework allows to define more lightweight interfaces, which enable client-side execution of complex queries.

In this talk we'll explore Catmandu and Linked Data Fragments and how they can cooperate to build an environment for data stream processing at large.