Handling JSON data with JMESPath and jq

The goal of this exercise is to produce, for every scholarly journal matching a given keyword query, the title of this journal and the average number of articles published every year in this journal. The final result should by a JSON array of objects with title and average number of publications, sorted by the latter.

1. Crossref is a nonprofit which collects information from scholarly publishers about every scholarly publication and makes the result avaialble as structured metadata. It has a freely usable REST API. Study and experiment with the `/journals` endpoint to determine how to obtain the required information.
2. Follow the JMESPath tutorial to understand how to express basic navigation patterns in JSON with this query language.
3. Write a Python program that uses the `jmespath` package, a Python implementation of the JMESPath query language, in order to navigate the output of the CrossRef API to obtain the desired output; it is fine to mix regular Python code and JMESPath expressions, as JMESPath alone is not powerful enough to express the entire transformation.
4. We now want to do the same with `jq`, a very powerful command-line JSON processor. Install it, follow the tutorial, and check the full documentation. Once familiar with `jq`, write a single `jq` program that produces the same output as the Python program you developed in the previous question.

Handling XML data with XQuery

Write an XQuery program that returns all current news from the newspaper Le Monde containing a specific keyword (with their publication date, title, and description), relying on its RSS feed.

Modifié le: mardi 21 novembre 2023, 15:01