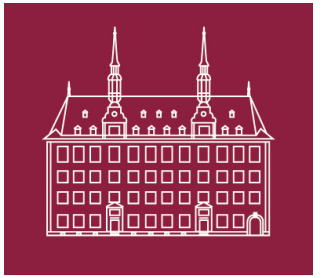


IEG

Leibniz-Institut für
Europäische Geschichte

Custom Data Fields in Zotero & Automatic Data Requests via API

IEG DH Brownbag Lunch

Dr. Markus Müller

23 September 2019

I Need More Data Fields!!

Sooner or later, you probably wish your bibliography manager had more data fields:

- You want to store more than one link to online ressources for this item, or you want to specify between different types of online ressources – but there exists only one field for one weblink.
- You work with Jewish literature and want to keep track of the year of publication according to the Hebrew calender – but there is only one field for “year”.
- You want to store informations about (different kinds of) relations between the items in your database, e.g. this book has a dedication from person A to person B, or book X is a translation of book Y, etc. – but your database has no data field for that or allows only one type of relations.
- etc.

One solution in Zotero is:

Fill the “extra” field with key-value pairs to store any kind of additional data in a machine readable way.

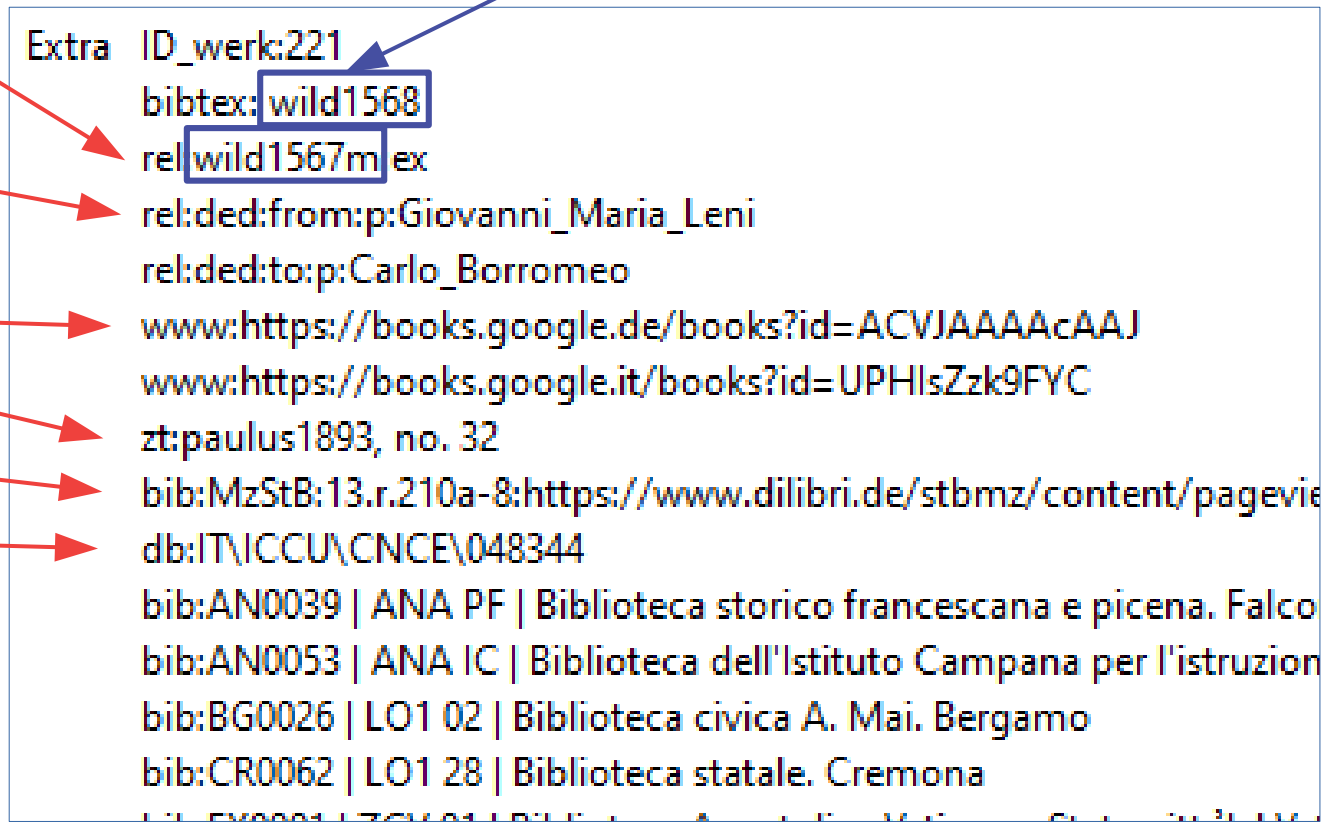
For example, instead of “See additional information on http://www.zotero.org”, just write “www:http://zotero.org”. Instead of “Year of publication according to Hebrew calendar: 5779”, just write “hebr:5779”. Instead of “Translation of the Köln edition 1555”, write “rel:wild1555:tr” (meaning: this item has a **rel**ation with the item “wild1555”, and the relation is of the type “**tr**anslation”).

Feel free to invent your own abbreviation system. The only important thing is, that it is consistent and follows the pattern **key:value**.

Use the “extra” Field to Add Custom Data Fields

These unique identifiers are generated by the BetterBib(La)TeX add-on

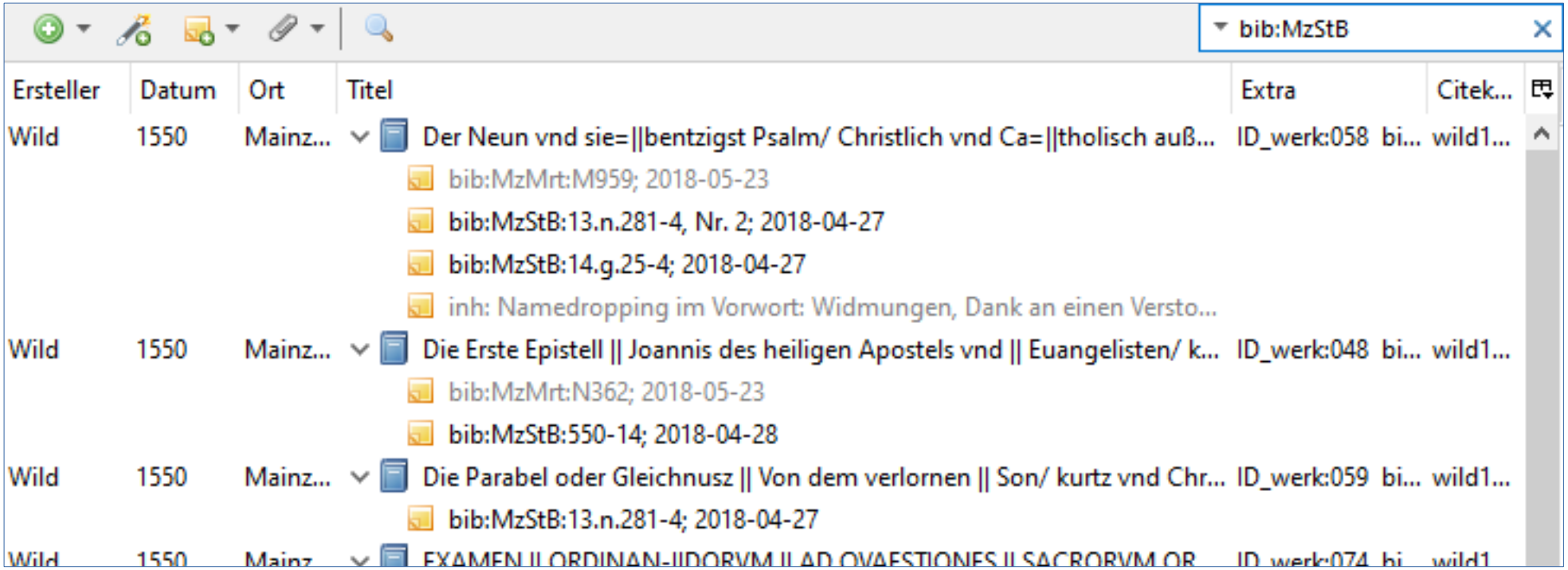
- **rel**ationships:
  - – this is an **ed**ition, **tr**anslation or **ex**purged edition of item X
  - – this book has a **ded**ication from **p**erson A to **p**erson B
- **www** ressources (e.g. scans)
- authors referencing this item
- libraries owning this edition
- ID in another **d**ata**b**ase



key:value

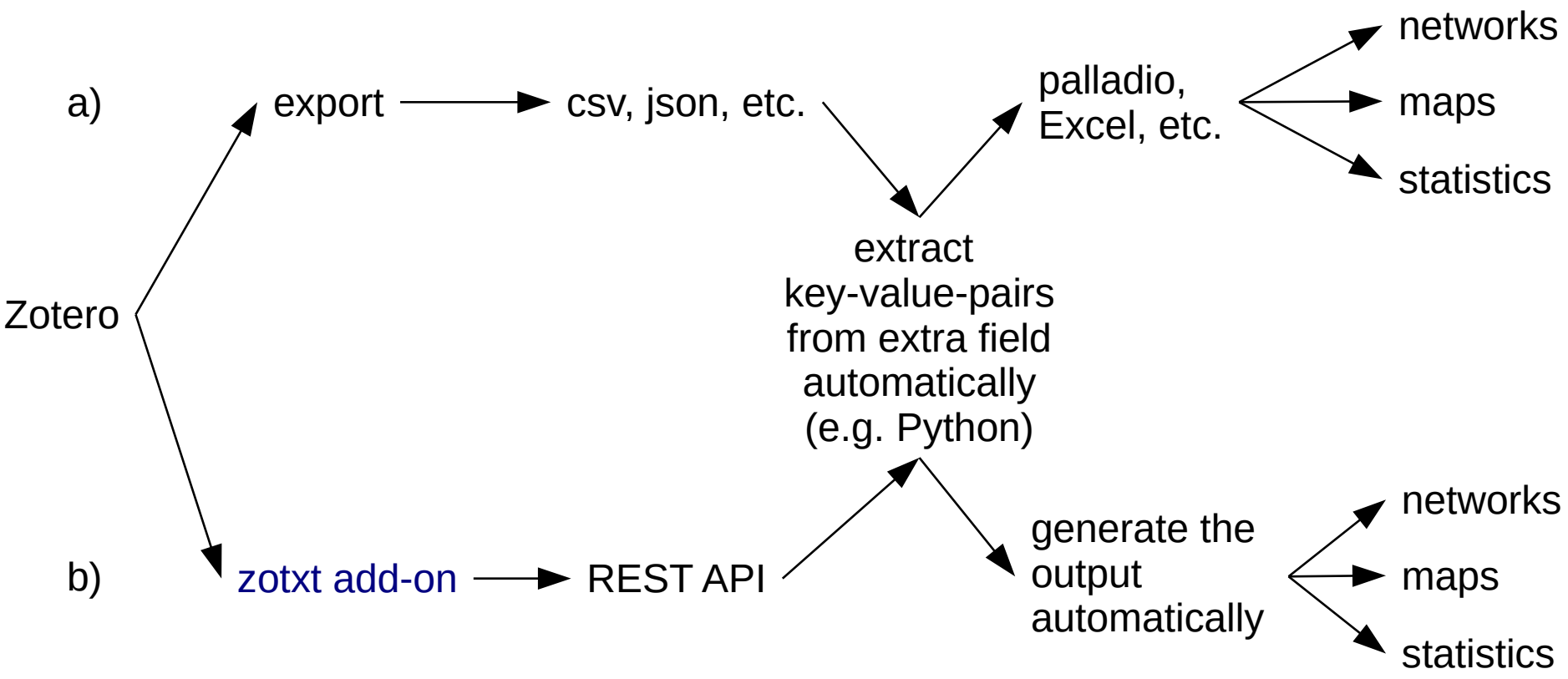
Searching Custom Data Fields

The big advantage of using this key:value pattern is that you can search your data very precisely: Searching for “bib:MzStB” in the standard search field returns all items with this specific key:value pair, i.e. all items that are available at the Wissenschaftliche Stadtbibliothek in Mainz. “bib:MzStB:14” would return all the items in that library whose shelf mark starts with “14”, and so on.



Analyze Data in Custom Data Fields

The key:value patterns can be easly recognized and processed with any programming language. This enables you to automate your workflow. There are many different possibilities how to implement this, but three steps are always necessary: 1) Get the data from Zotero, 2) extract the key:value pairs and split them to get the individual bits of information, 3) process the data, e.g. generate a visualization. Two possible workflows:

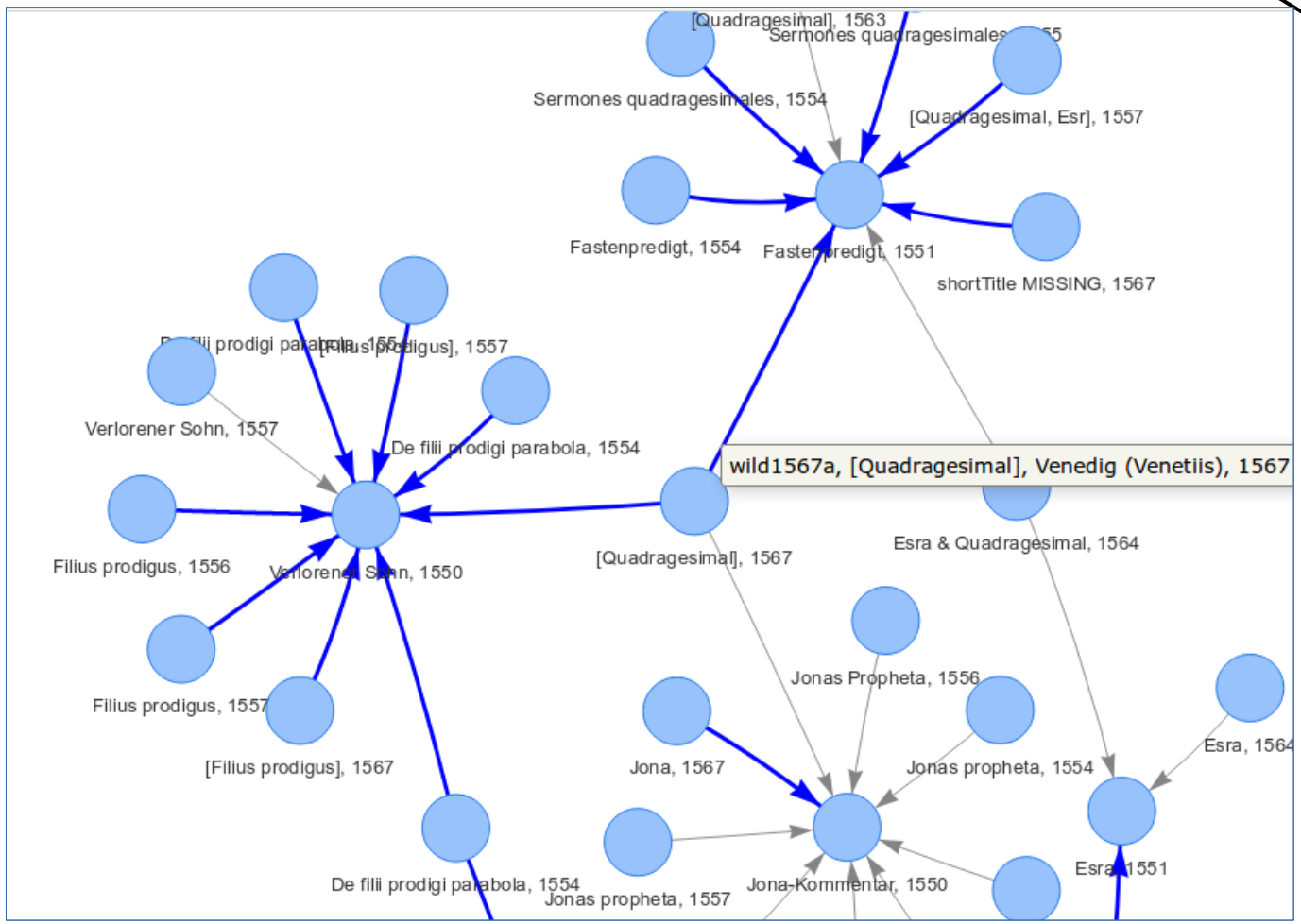


API = Application Programming Interface
REST = Representational State Transfer

Example: Relationships Between Books

One example: A Python script requests data from Zotero by using the API provided by the zotxt add-on. Then, it extracts informations about relations between books from the key:value pairs in the “extra” field. Finally, it generates a network graph to visualize the relations:

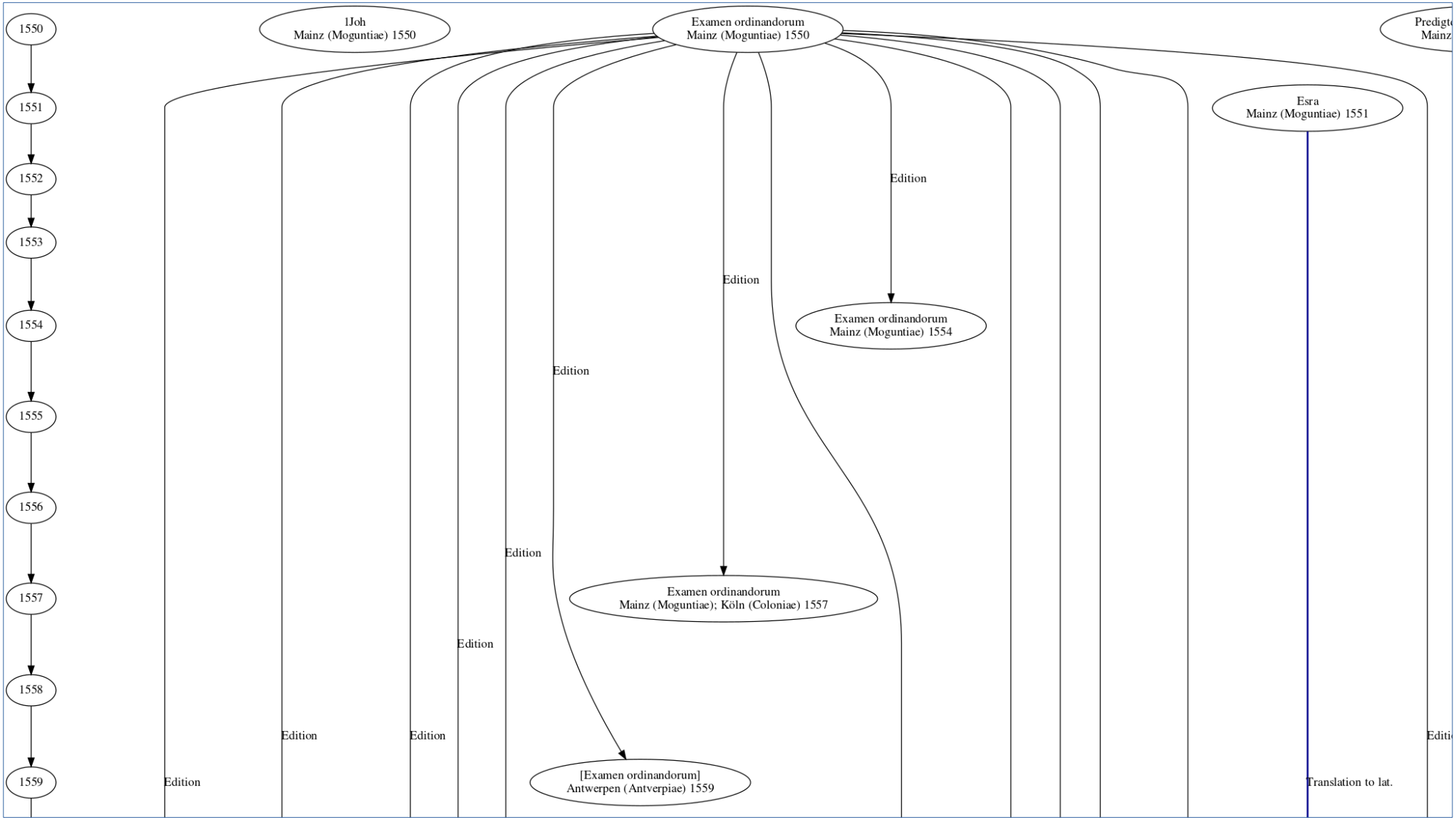
Zotero → zotxt add-on → REST API → Python script → network graph



extracts:
rel:wild1550:ed
rel:wild1550:tr
etc.
from the “extra” field

html page with vis.js graph

The Same Data Visualized as a Timeline



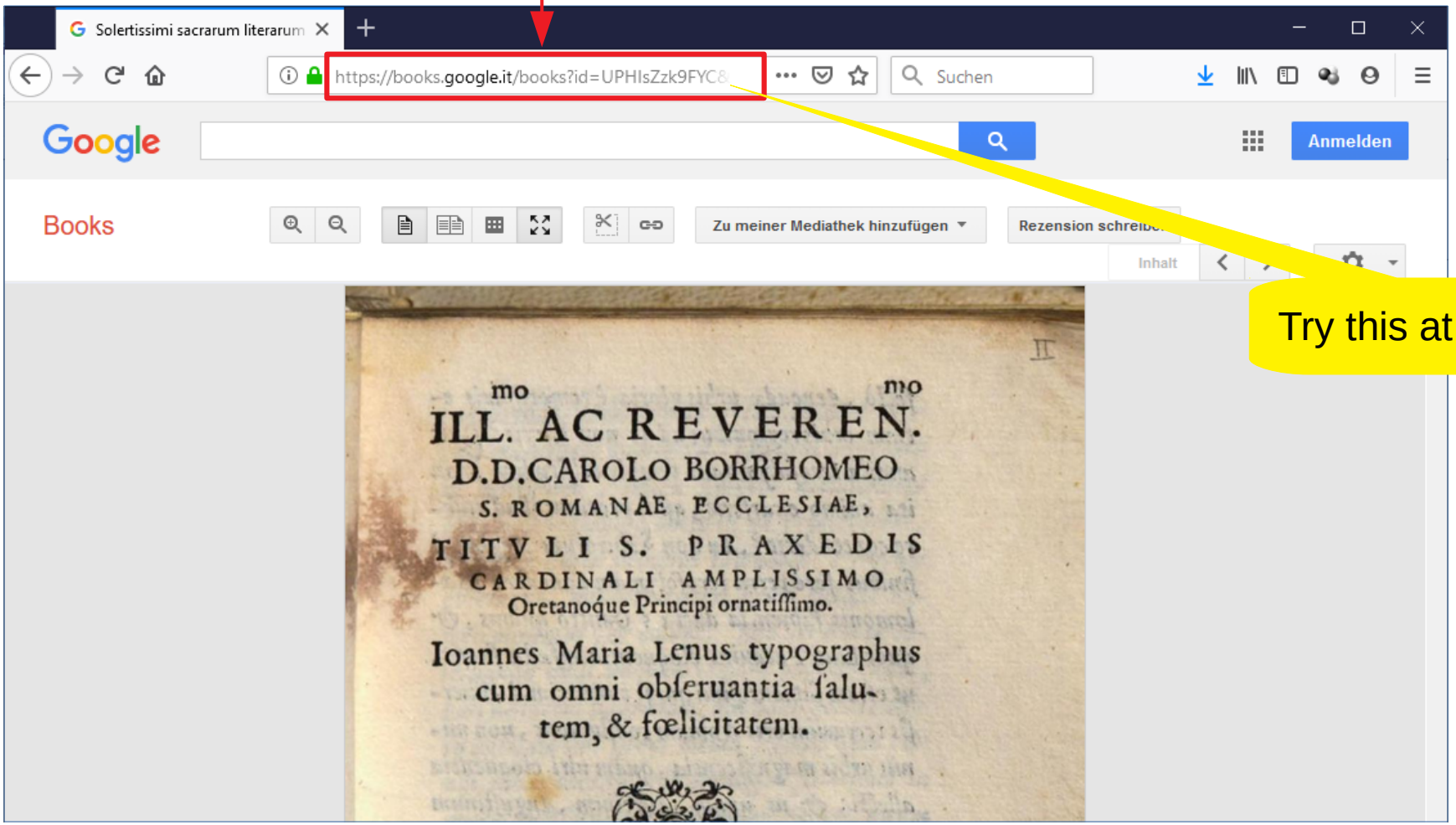
svg graph generated with graphviz

What's a REST API ?!

A REST API provides a very simple interface that enables users to ask a server for specific data by sending a simple HTTP request. The server processes the request and returns the requested data. Example:

Hey, Google: Open the book with the **id** "UPHIsZzk9FYC" on **page** "PP11"

https://books.google.it/books?**id**=UPHIsZzk9FYC&**pg**=PP11



Try this at home!

Zotxt REST API

The zotxt add-on provides a similar REST API locally on your computer:

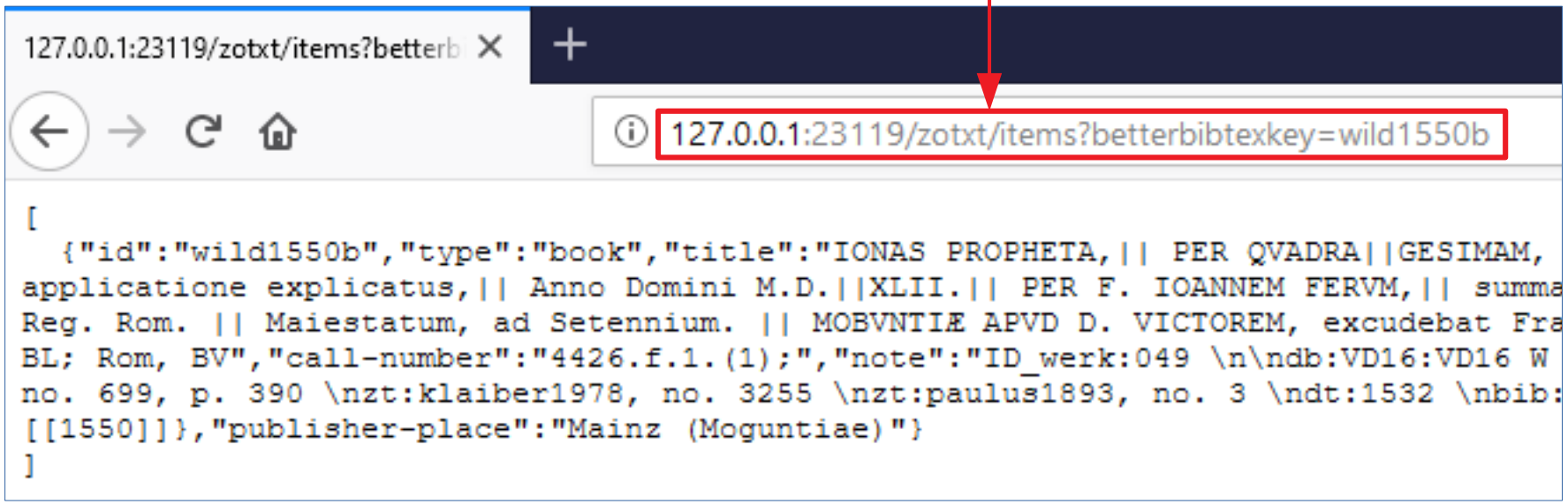
Hey, Zotero, give me all data of the item with the **bibtex key** “wild1550b”

http://127.0.0.1:23119/zotxt/items?**betterbibtexkey**=wild1550b

This IP adress points to your local machine on port 23119

This is the endpoint you send your request to.

This is your request: a key:value pair!!



Zotero answers by sending a text structured by key:value pairs (we know this concept already!). Every data field you find in Zotero is represented here by a key:value pair. In this case the structure follows the JSON standard (the response is a list of JSON objects). JSON can be processed *very* easily by any programming language.

References

Zotero Add-ons:
– Zotxt: https://github.com/egh/zotxt
– BetterBib(La)TeX: https://github.com/retorquere/zotero-better-bibtex

Visualization tools (network graphs etc.):
– JavaScript library vis.js: https://visjs.org/
– Command line tool Graphviz: http://www.graphviz.org/
(cf. https://de.wikipedia.org/wiki/Graphviz)
– Web app Palladio: https://hdlab.stanford.edu/palladio/