NeverBlink

September 3, 2025

SemDev @ SEMANTiCS, Vienna, Austria

Jelly: a Fast and Convenient RDF Serialization Format

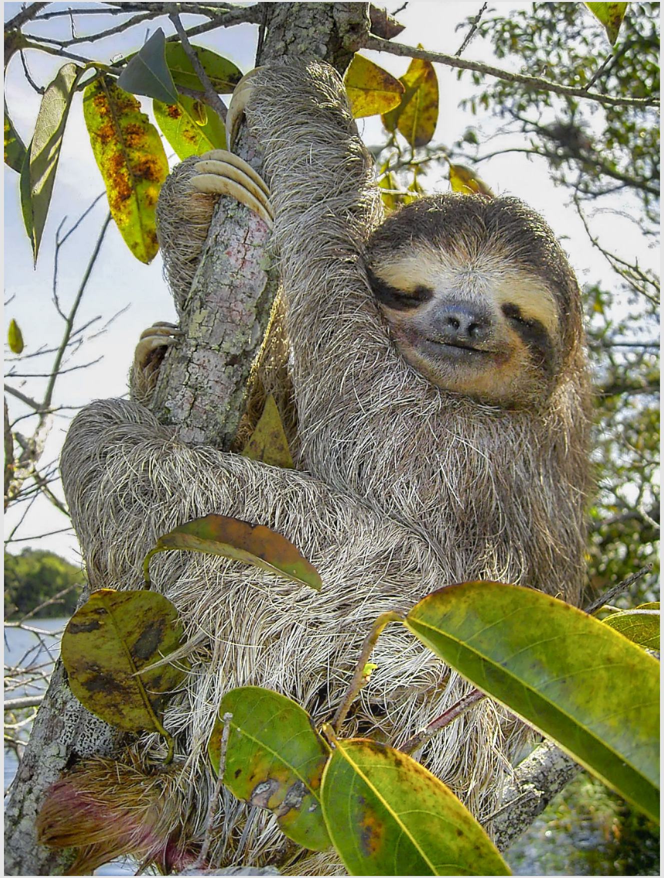
Piotr Sowiński[1,2], Karolina Bogacka[1,2], Anastasiya Danilenka[1,2], Nikita Kozlov[1,2]

[1] *NeverBlink*

[2] *Warsaw University of Technology, Poland*

We have RDF formats that do things well!

- **N-Triples** – simplicity
- **Turtle** – human readability
- **JSON-LD** – non-RDF experience
- **HDT** – compressed, queryable dumps
- **RDF/XML** – being annoying…

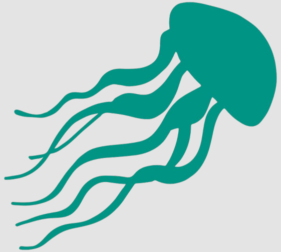


But, I desire *speed.*

- Client-server communication
- Inter-service communication
- DB dumps and bulk loads
- Streaming pipelines
- DB replication
- Change data capture

Stefan Laube, public domain

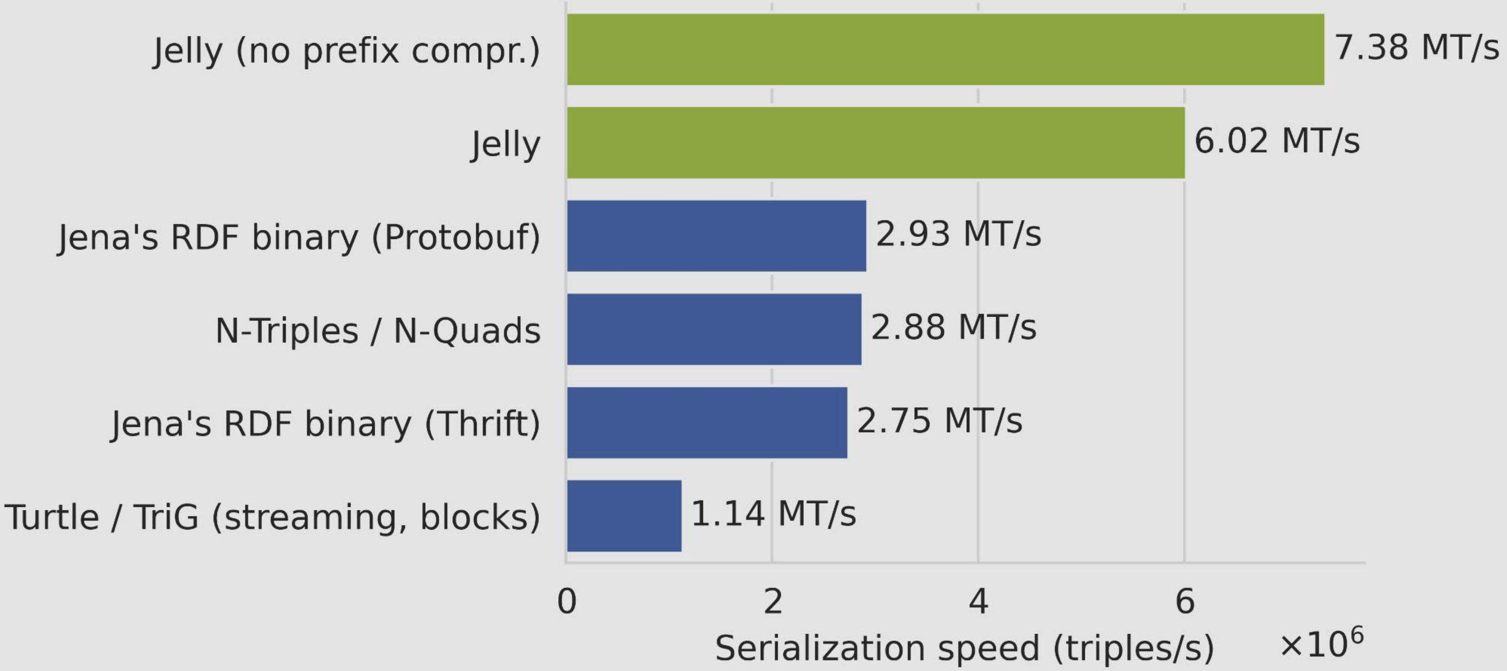
Jelly in a nutshell



- Binary RDF format based on Protobuf
- 100% open spec & open source (https://w3id.org/jelly)
- Implemented for: Java, Python, Rust *(experimental)*
- Convenient CLI tool
- ***Very fast***
- Reasonably well compressed

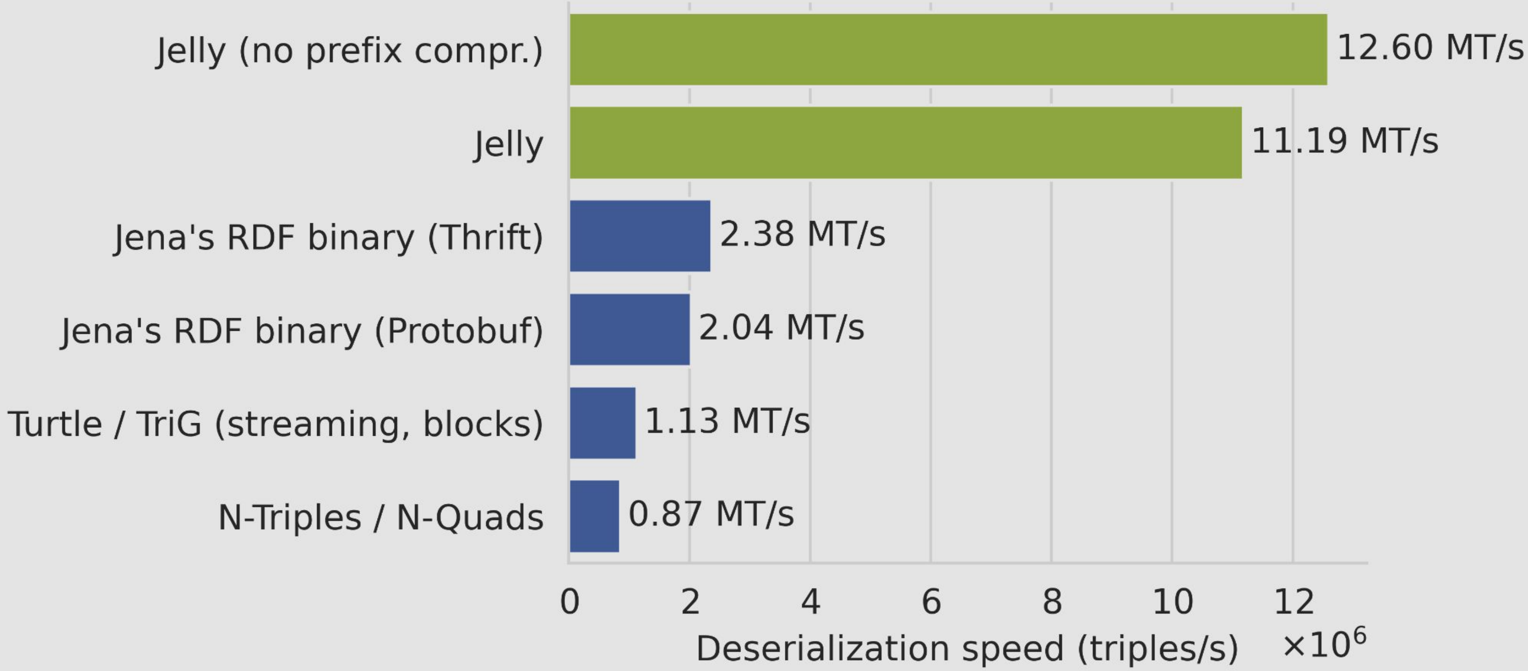
Serialization speed (Apache Jena)

RiverBench task: flat-serialization-throughput, profile: flat-mixed-rdfstar 2.1.0. Details: https://w3id.org/jelly/dev/performance



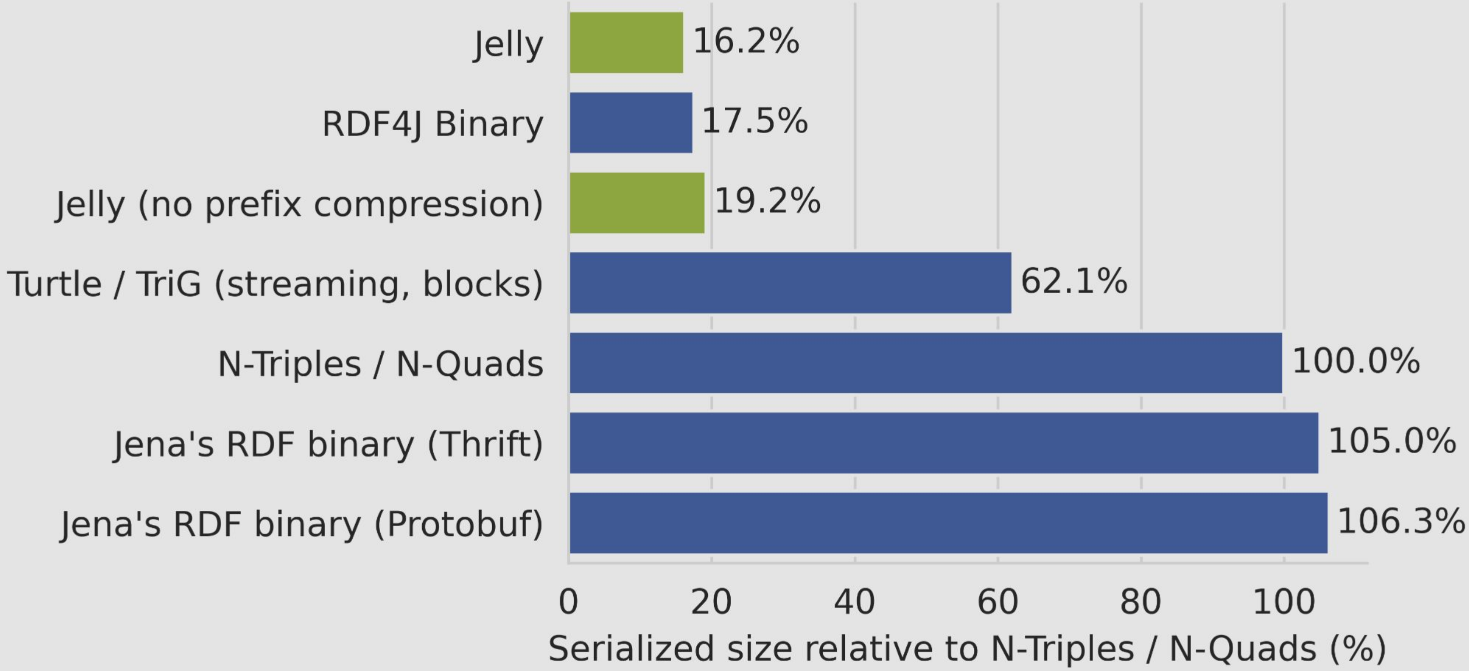
Parsing speed (Apache Jena)

RiverBench task:
flat-deserialization-throughput, profile:
flat-mixed-rdfstar 2.1.0. Details:
https://w3id.org/jelly/dev/performance

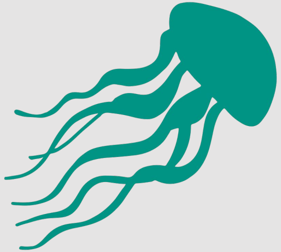


Compression ratio

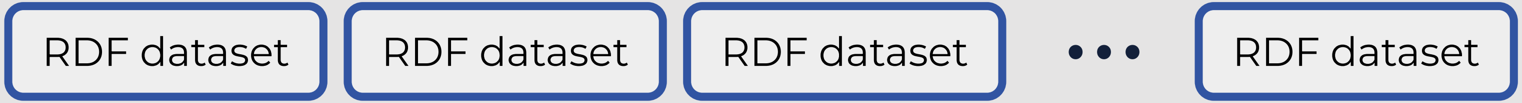
RiverBench task: flat-compression, profile: flat-mixed-rdfstar 2.1.0. Details: https://w3id.org/jelly/dev/performance



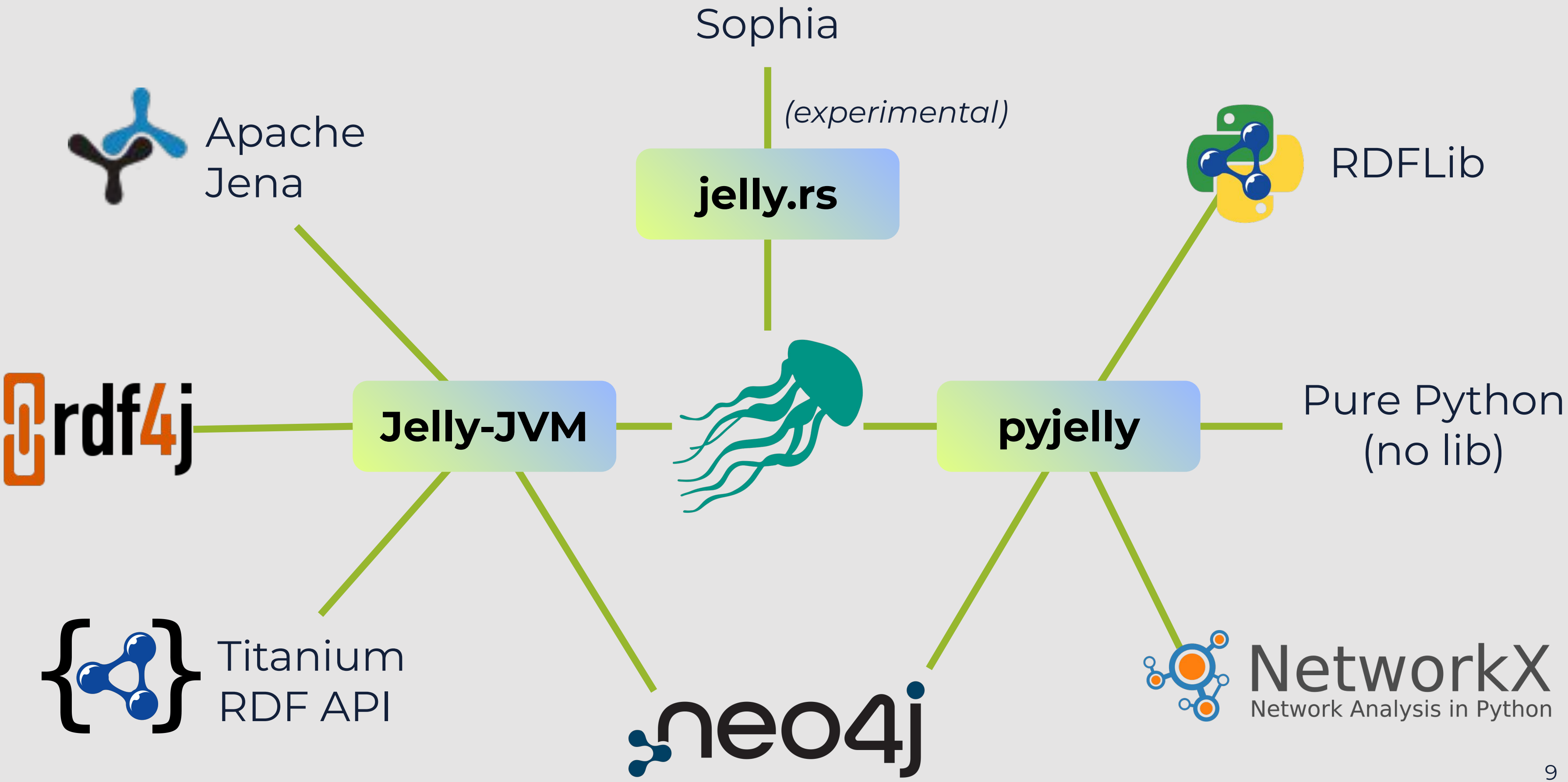
How does Jelly work?



- Lightweight streaming compression algorithm
- For ***n*** triples:
  - O(1) memory complexity
  - O(n) time complexity
- Max supported triple count = **∞**
- 1 file can contain 1 RDF document (graph or dataset)…
- …or 1 file can contain **many** RDF documents (**!**)



RDF dataset | RDF dataset | RDF dataset | • • • | RDF dataset



Sophia

(experimental)

jelly.rs

Apache Jena

Jelly-JVM

rdf4j

Titanium RDF API

neo4j

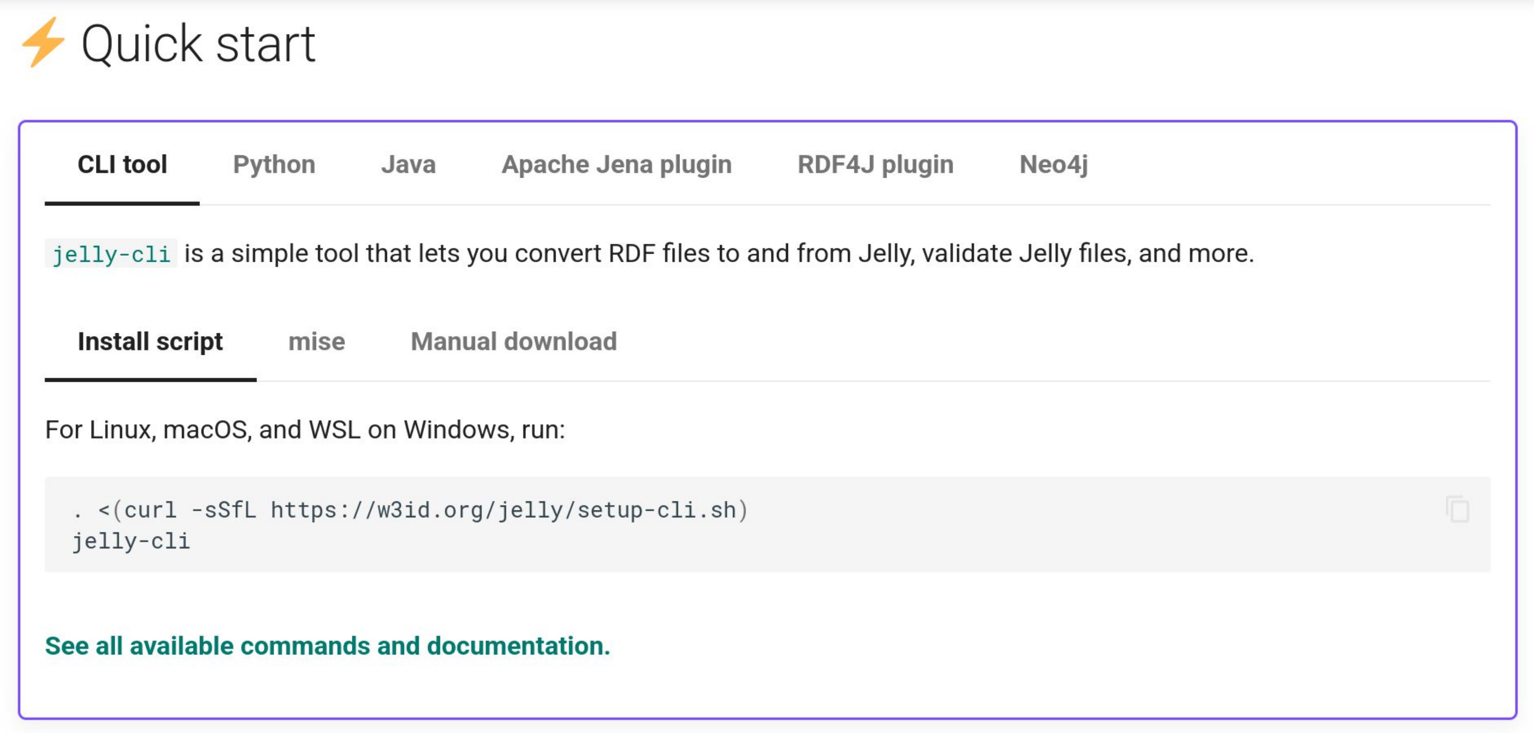
pyjelly

RDFLib

Pure Python (no lib)

NetworkX
Network Analysis in Python

Get started -> https://w3id.org/jelly



⚡ Quick start

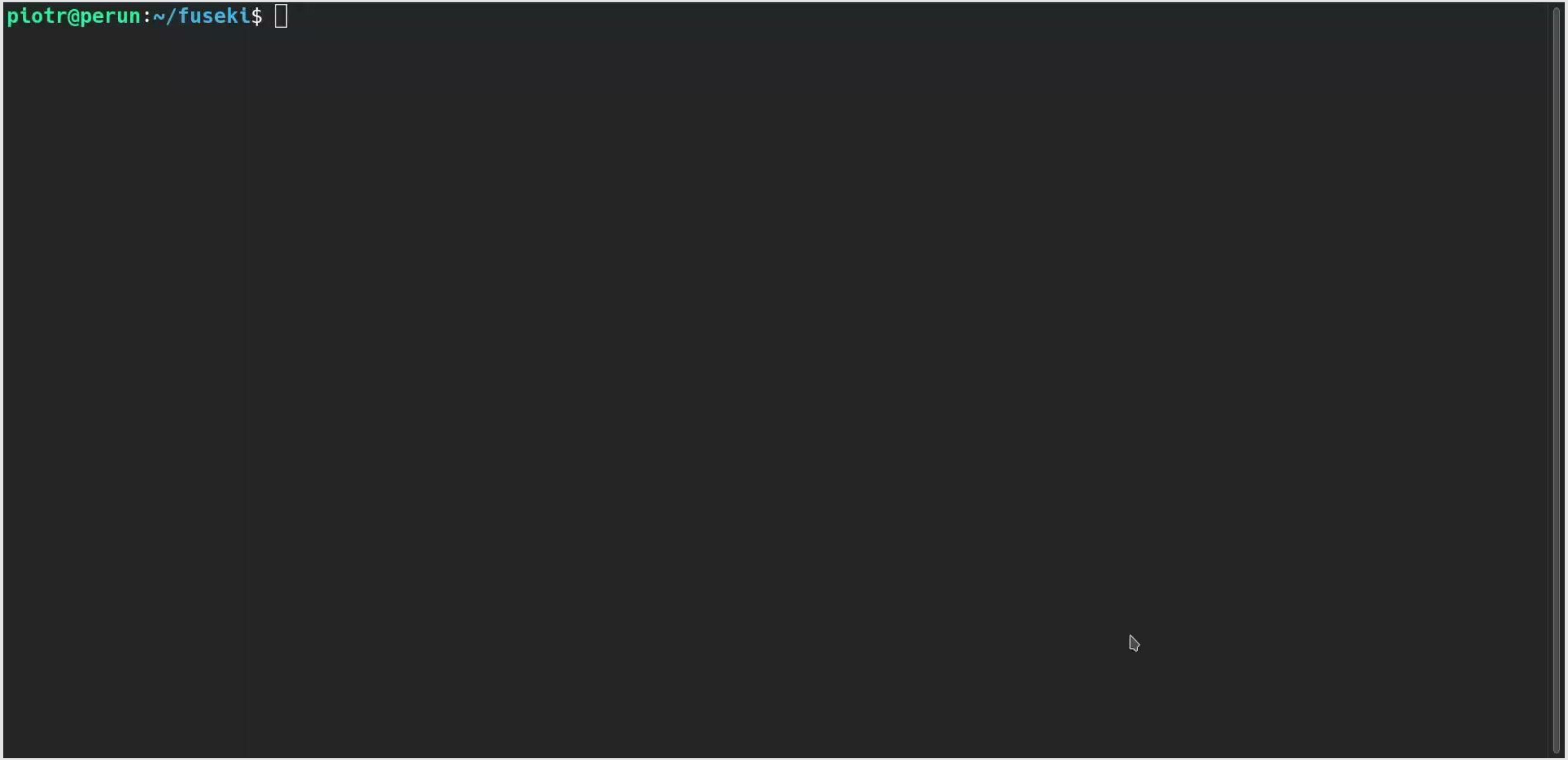
`jelly-cli` is a simple tool that lets you convert RDF files to and from Jelly, validate Jelly files, and more.

For Linux, macOS, and WSL on Windows, run:

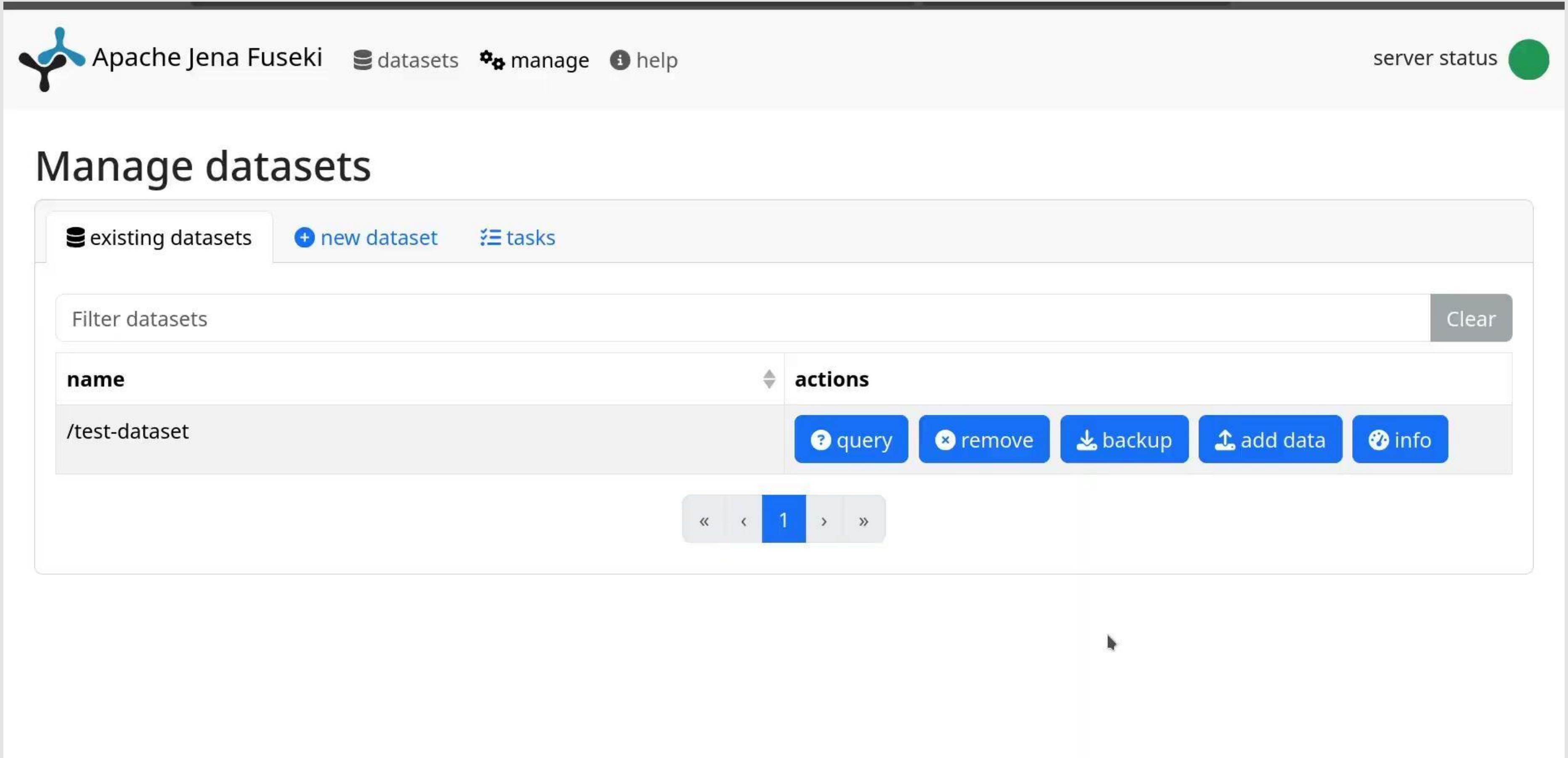
```
. <(curl -sSfL https://w3id.org/jelly/setup-cli.sh)
jelly-cli
```

See all available commands and documentation.

```
piotr@perun:~/jelly$ 
```

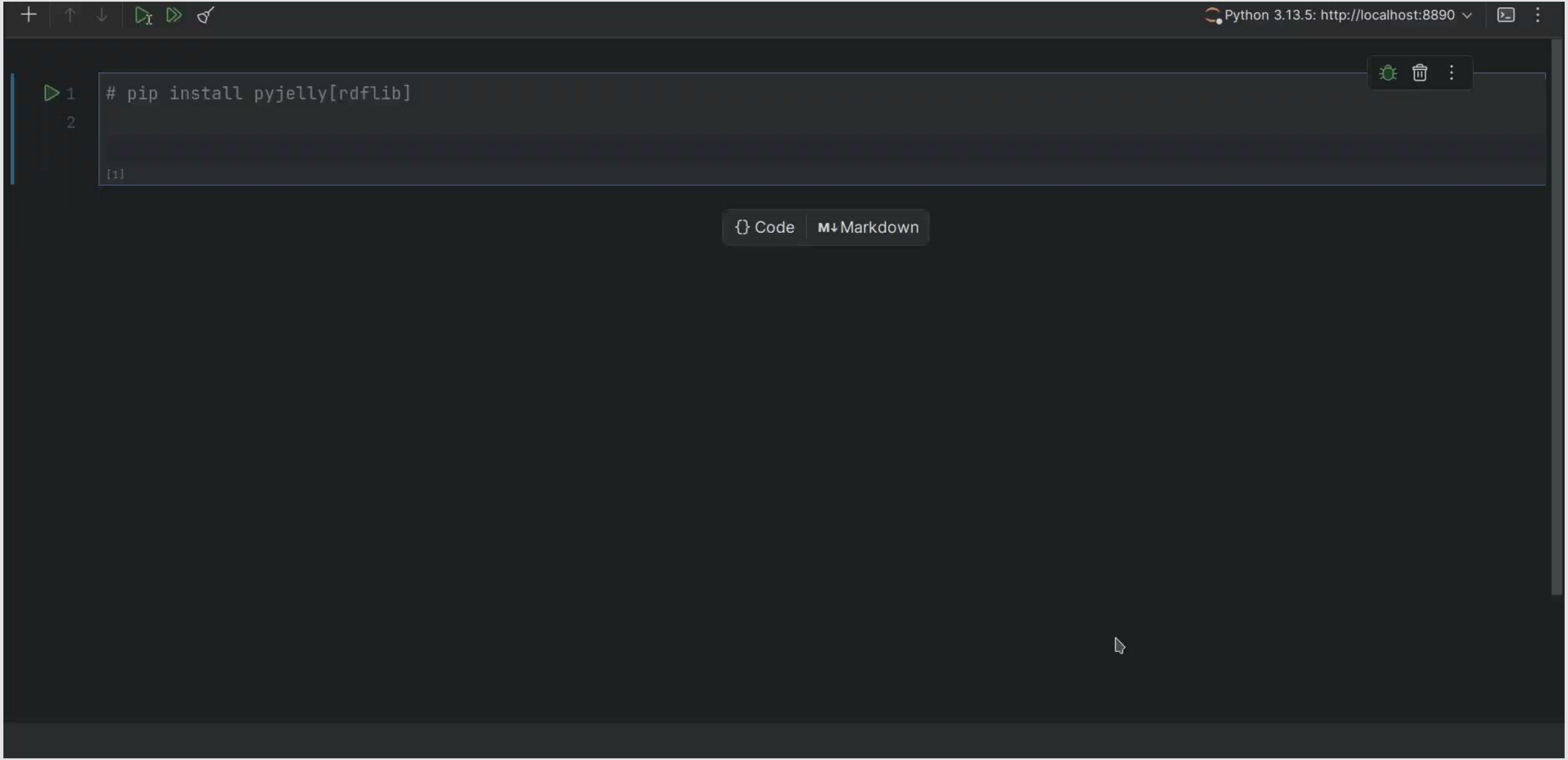


```
piotr@perun:~/fuseki$ 
```

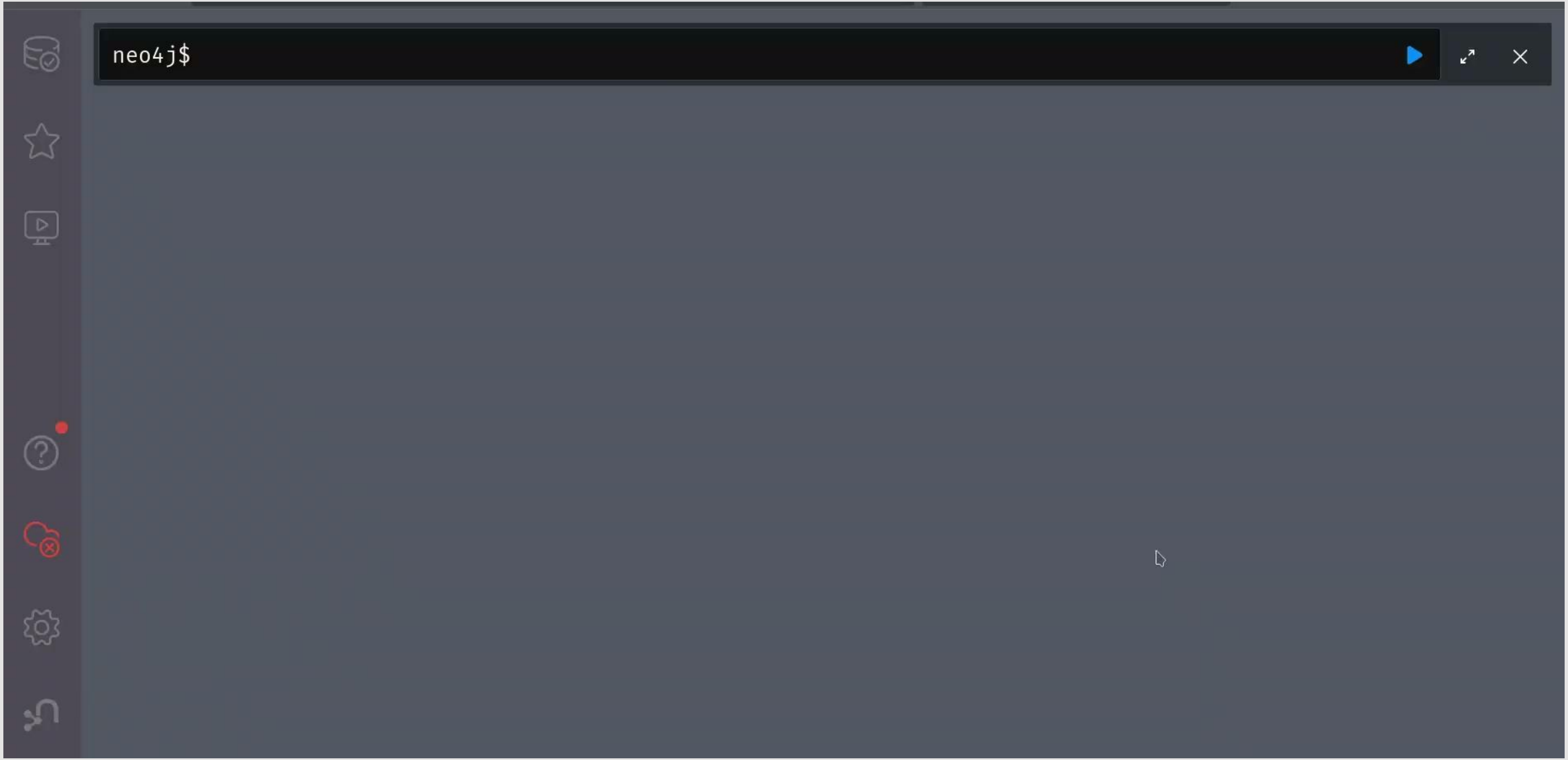


Manage datasets

name	actions
/test-dataset	query remove backup add data info



```
# pip install pyjelly[rdflib]
```



```
neo4j$
```

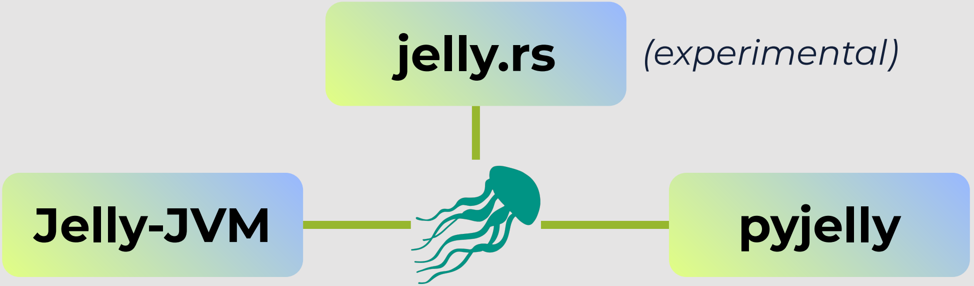
Other cool things you can do with Jelly

- Streaming over Kafka, MQTT, etc.
- gRPC
- Merging & recompressing RDF streams on the fly
- Live database replication (with Jelly-Patch)
- Change data capture (with Jelly-Patch)

Real-life use cases? □ **See my Industry Track talk on Friday!**

Interoperability (and reuse)

- 100% open specification
- Conformance test kit
- Guide for new implementations
- Full coverage of RDF 1.1 and RDF-star



jelly.rs *(experimental)*

Jelly-JVM

pyjelly

Reuse (and interoperability)

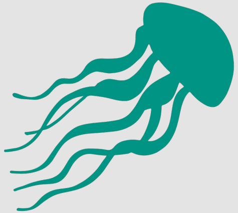
- Generic implementations + library integrations pattern
  - **Less work in maintaining implementations!**
- Focus on supporting popular RDF libraries
- Simple dependencies (only Protobuf)
- Comprehensive docs with user stories
- Quick start guide

Conclusion

https://w3id.org/jelly

- **Fast and convenient RDF format**
- Works with Java, Python, Rust *(experimental)*
- Integrations for Jena, RDF4J, RDFLib, Neo4j & more
- Community
  - 100% open source
  - Community Discord server – **come chat with us!**
  - Community-led implementation for Rust *(big thanks to Arthur Vercruysse)*
- Real-life gains □ **Friday, Industry Track talk**

⭐ **Star us on GitHub!**



Acknowledgements

Big thanks to all Jelly developers and the community who make this project possible!

The development of the Jelly protocol, its implementations, and supporting tooling was co-funded by the European Union. Project no. 0021/2025, funding program FENG.02.28-IP.02-0006/23 (Startup Booster Poland – HugeThing Sector Agnostic). The views expressed are of its authors and do not necessarily reflect the views of the European Union.

Total cost of project: 149 941,44 PLN
Contribution from European Funds: 149 941,44 PLN

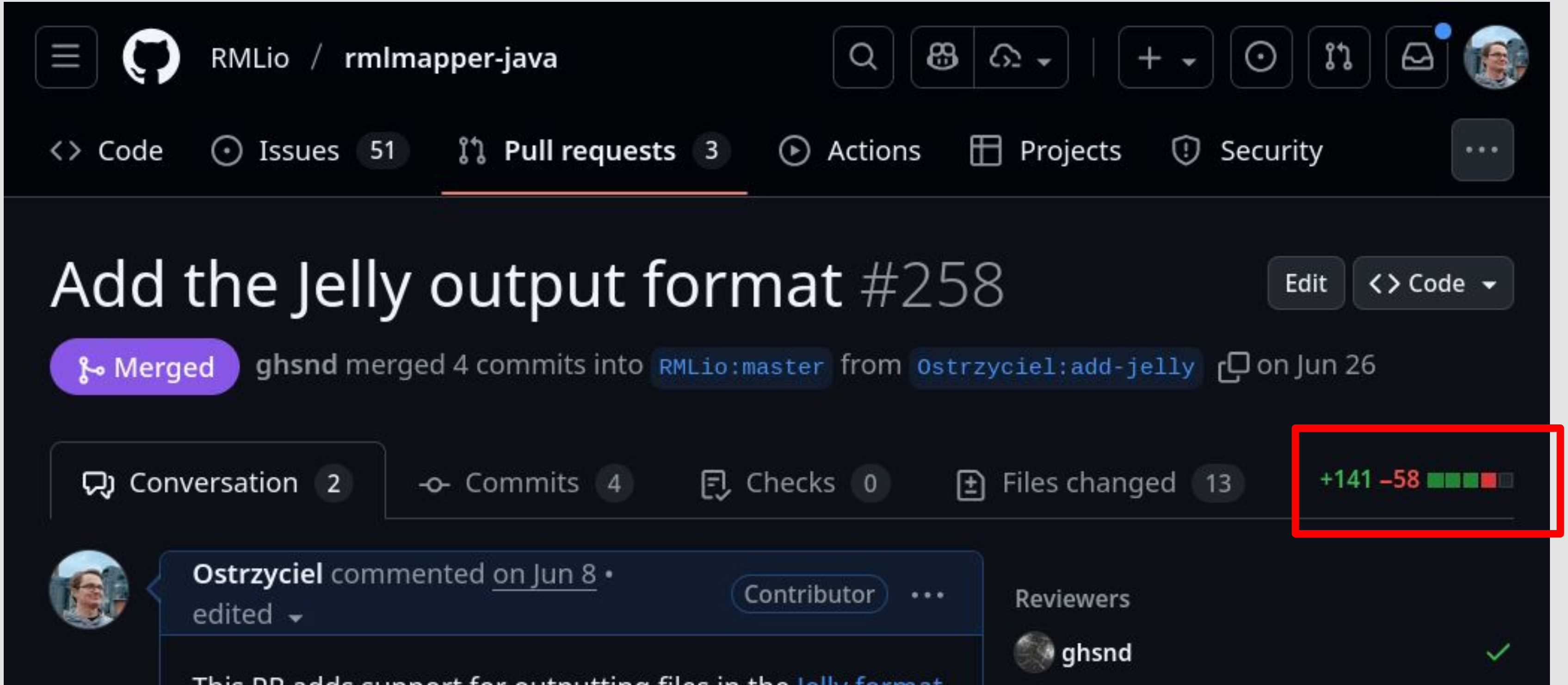
European Funds
for Smart Economy

Republic
of Poland

Co-funded by the
European Union

Backup slides

Interoperability & reuse in action

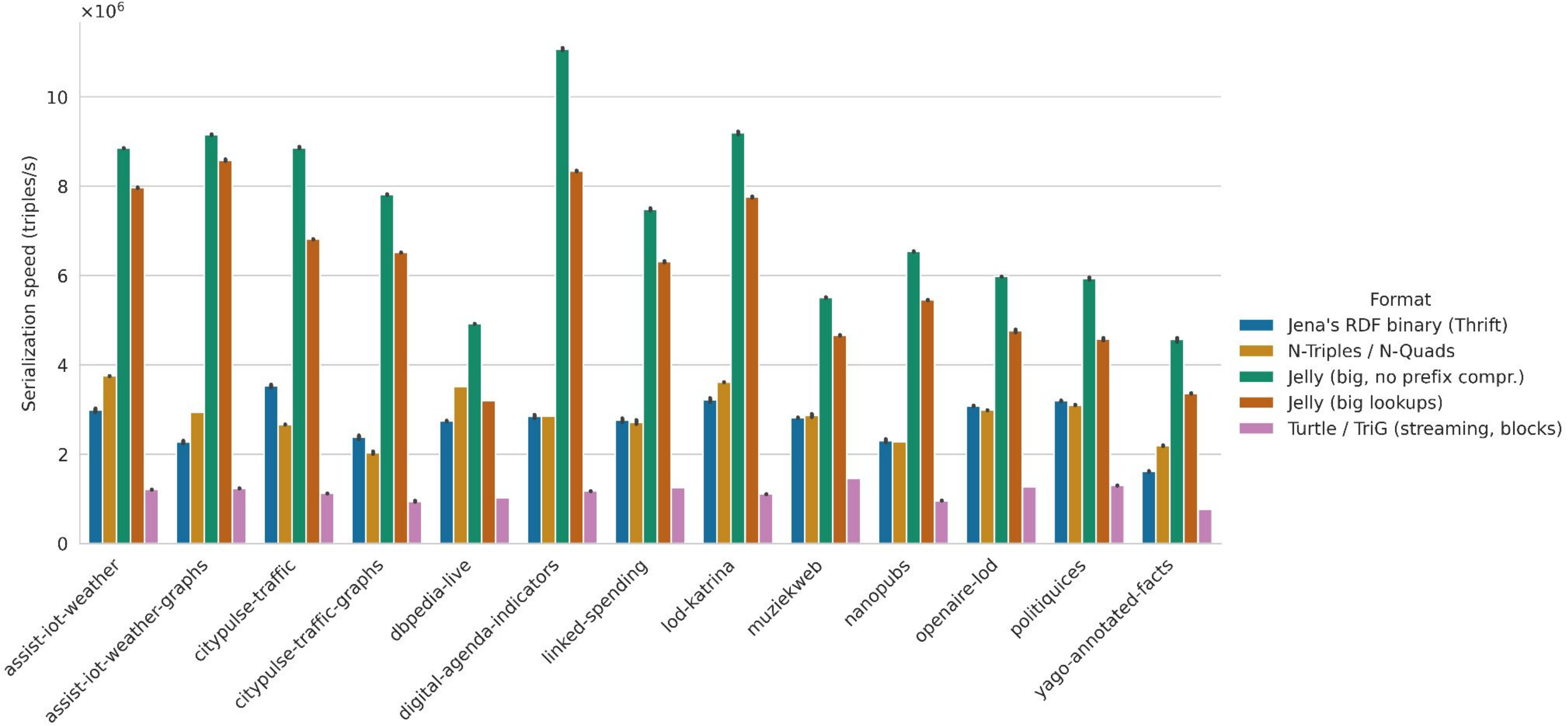


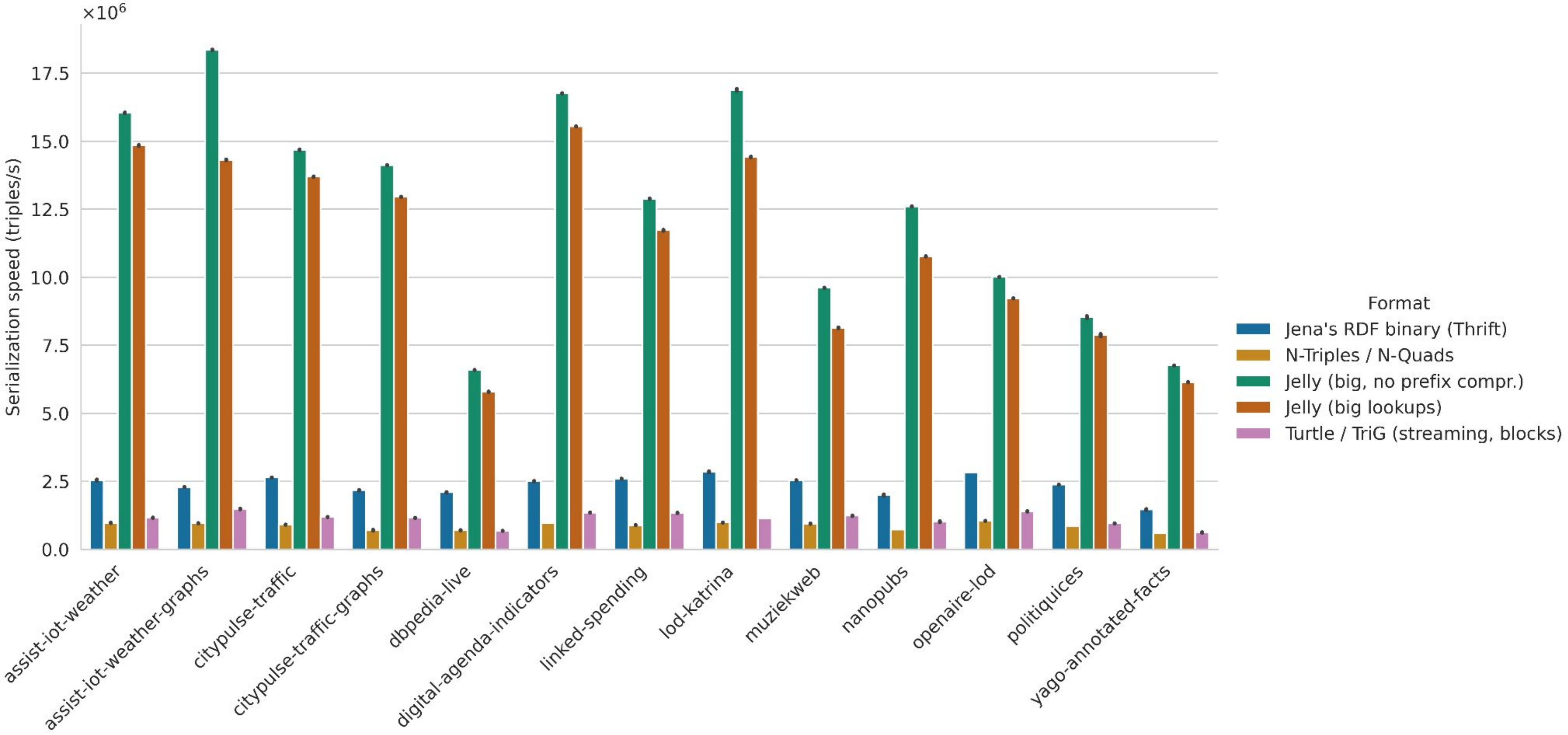
Add the Jelly output format #258

ghsnd merged 4 commits into `RMLio:master` from `Ostrzyciel:add-jelly` on Jun 26

Ostrzyciel commented on Jun 8 • edited

This PR adds support for outputting files in the Jelly format


×10^6
Serialization speed (triples/s)
0
2
4
6
8
10
assist-iot-weather
assist-iot-weather-graphs
citypulse-traffic
citypulse-traffic-graphs
dbpedia-live
digital-agenda-indicators
linked-spending
lod-katrina
muziekweb
nanopubs
openaire-lod
politiquices
yago-annotated-facts
Format
Jena's RDF binary (Thrift)
N-Triples / N-Quads
Jelly (big, no prefix compr.)
Jelly (big lookups)
Turtle / TriG (streaming, blocks)


×10^6
Serialization speed (triples/s)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
assist-iot-weather
assist-iot-weather-graphs
citypulse-traffic
citypulse-traffic-graphs
dbpedia-live
digital-agenda-indicators
linked-spending
lod-katrina
muziekweb
nanopubs
openaire-lod
politiquices
yago-annotated-facts
Format
Jena's RDF binary (Thrift)
N-Triples / N-Quads
Jelly (big, no prefix compr.)
Jelly (big lookups)
Turtle / TriG (streaming, blocks)