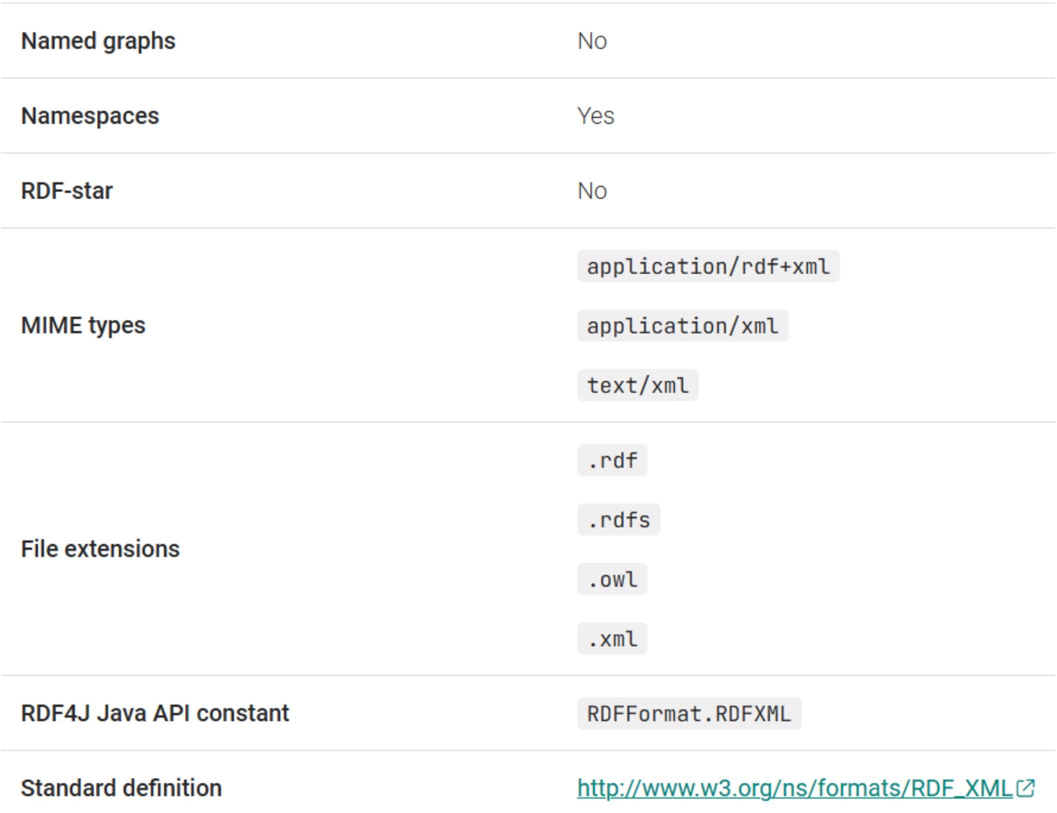
List of supported RDF formats and their use-cases

1. RDF/XML

- XML syntax for RDF. Verbose but historically widespread.
- Use Case: Legacy compatibility, archives, import/export to older tools.



Named graphs	No
Namespaces	Yes
RDF-star	No

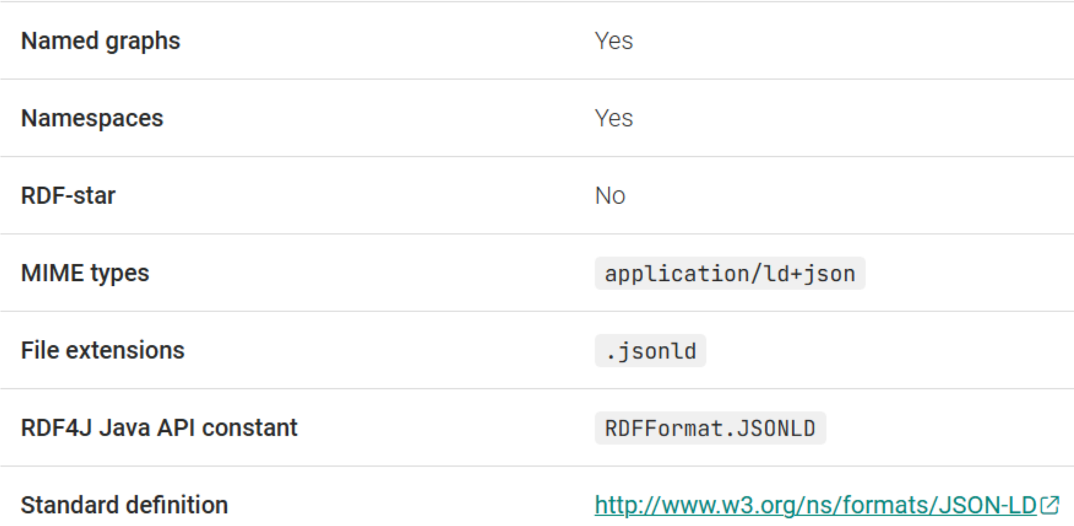
MIME types	`application/rdf+xml`
	`application/xml`
	`text/xml`

File extensions	`.rdf`
	`.rdfs`
	`.owl`
	`.xml`

RDF4J Java API constant	`RDFFormat.RDFXML`
Standard definition	http://www.w3.org/ns/formats/RDF_XML

2. JSON-LD

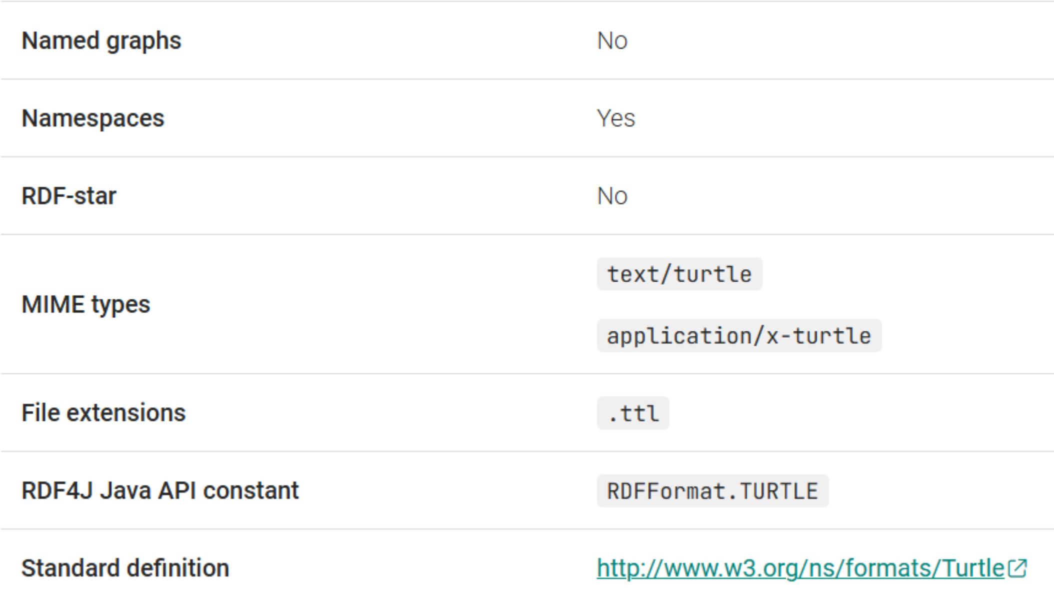
- JSON with an `@context` that maps keys to IRIs. Natural for web APIs and SEO.
- Use Case: Embed in web pages, return from REST endpoints, easy for frontend consumers.



Named graphs	Yes
Namespaces	Yes
RDF-star	No
MIME types	`application/ld+json`
File extensions	`.jsonld`
RDF4J Java API constant	`RDFFormat.JSONLD`
Standard definition	http://www.w3.org/ns/formats/JSON-LD

3. Turtle

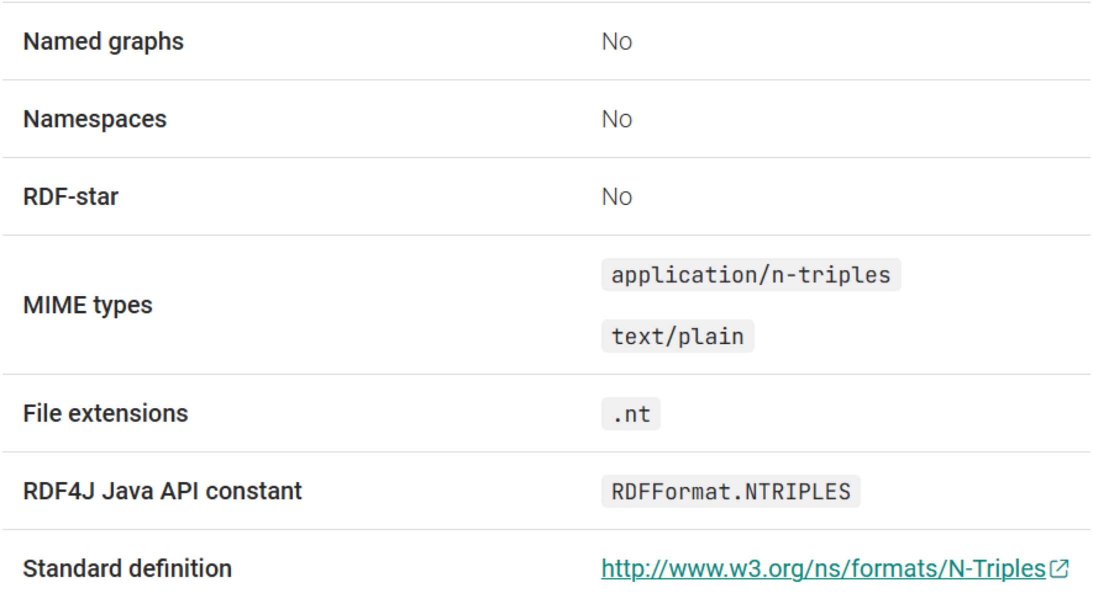
- Compact, human-readable syntax for RDF graphs.
- Use Case: Hand-authored files, ontology source files, code reviews, diffs.



Named graphs	No
Namespaces	Yes
RDF-star	No
MIME types	`text/turtle` `application/x-turtle`
File extensions	`.ttl`
RDF4J Java API constant	`RDFFormat.TURTLE`
Standard definition	http://www.w3.org/ns/formats/Turtle

4. N-Triples

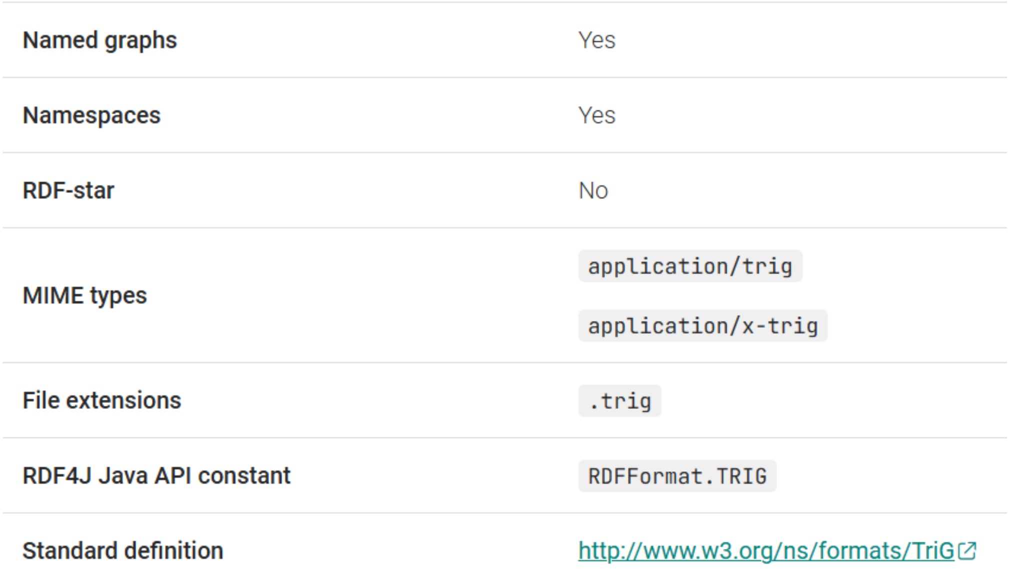
- One triple per line; very simple, stream-friendly.
- Use Case: Large dumps, ETL pipelines, debugging serialization.



Named graphs	No
Namespaces	No
RDF-star	No
MIME types	`application/n-triples` `text/plain`
File extensions	`.nt`
RDF4J Java API constant	`RDFFormat.NTRIPLES`
Standard definition	http://www.w3.org/ns/formats/N-Triples

5. TriG

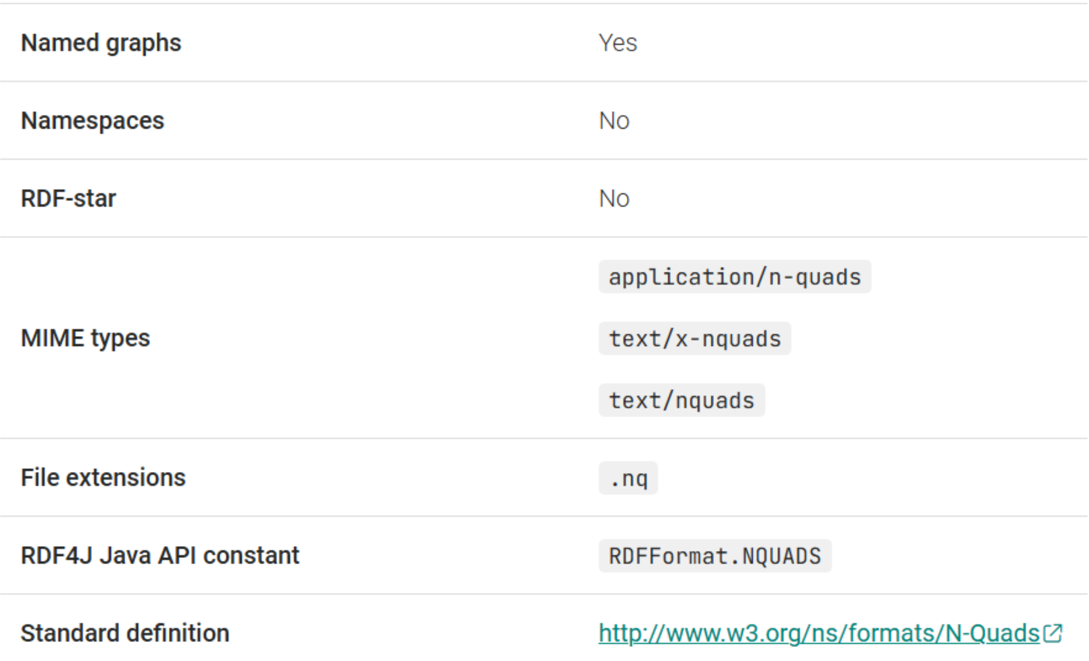
- “Turtle for datasets” - supports named graphs.
- Use Case: When you need multiple graphs (e.g., one graph per day/version).



Named graphs	Yes
Namespaces	Yes
RDF-star	No
MIME types	`application/trig` `application/x-trig`
File extensions	`.trig`
RDF4J Java API constant	`RDFFormat.TRIG`
Standard definition	http://www.w3.org/ns/formats/TriG

6. N-Quads

- N-Triples + graph name (quad).
- Use Case: High-volume dataset dumps and fast imports in data pipelines.



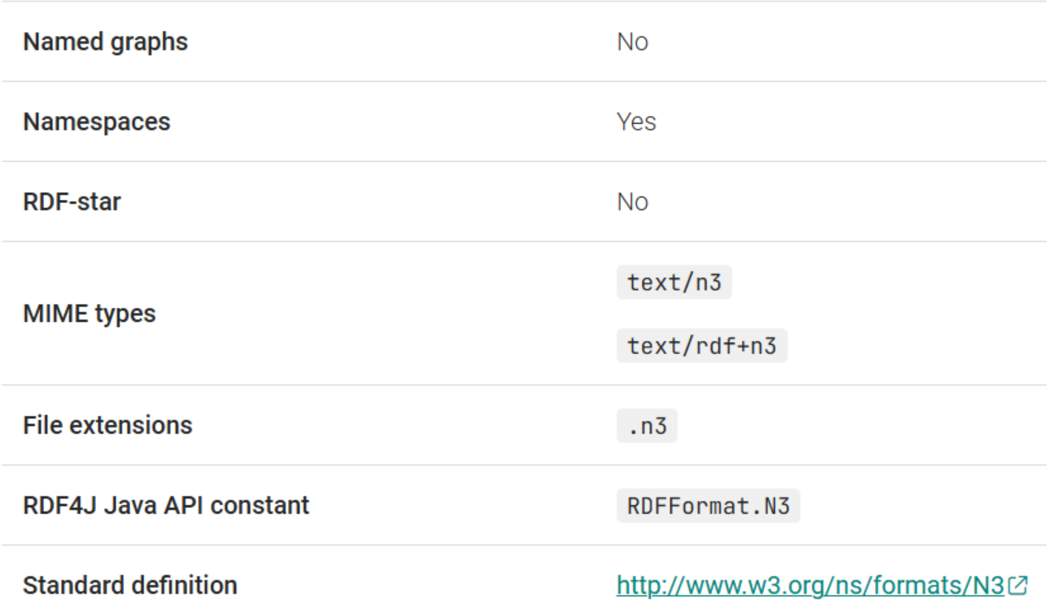
Named graphs	Yes
Namespaces	No
RDF-star	No

MIME types	`application/n-quads`
	`text/x-nquads`
	`text/nquads`

File extensions	`.nq`
RDF4J Java API constant	`RDFFormat.NQUADS`
Standard definition	http://www.w3.org/ns/formats/N-Quads

7. N3

- Turtle-like with additional features (rules, shortcuts).
- Use Case: Prototyping; some tools support it, but Turtle/JSON-LD are more standard today.



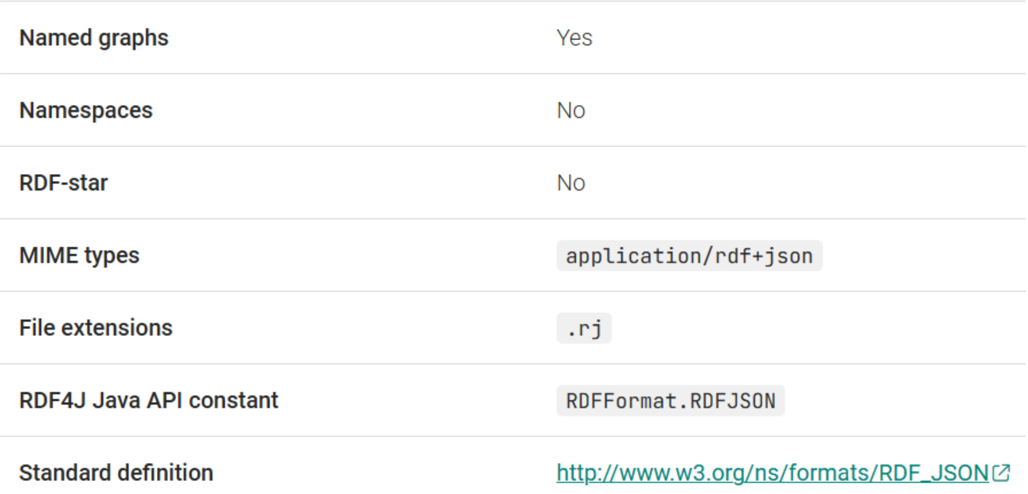
Named graphs	No
Namespaces	Yes
RDF-star	No
MIME types	`text/n3` `text/rdf+n3`
File extensions	`.n3`
RDF4J Java API constant	`RDFFormat.N3`
Standard definition	http://www.w3.org/ns/formats/N3

8. RDFa

- Attributes in HTML/XML that express RDF.
- Use Case: Enrich existing HTML without separate files; crawlers can extract data.

9. RDF/JSON

- Older “RDF in JSON” (not JSON-LD).
- Use: Interop with older tooling that expects this exact shape; otherwise prefer JSON-LD.

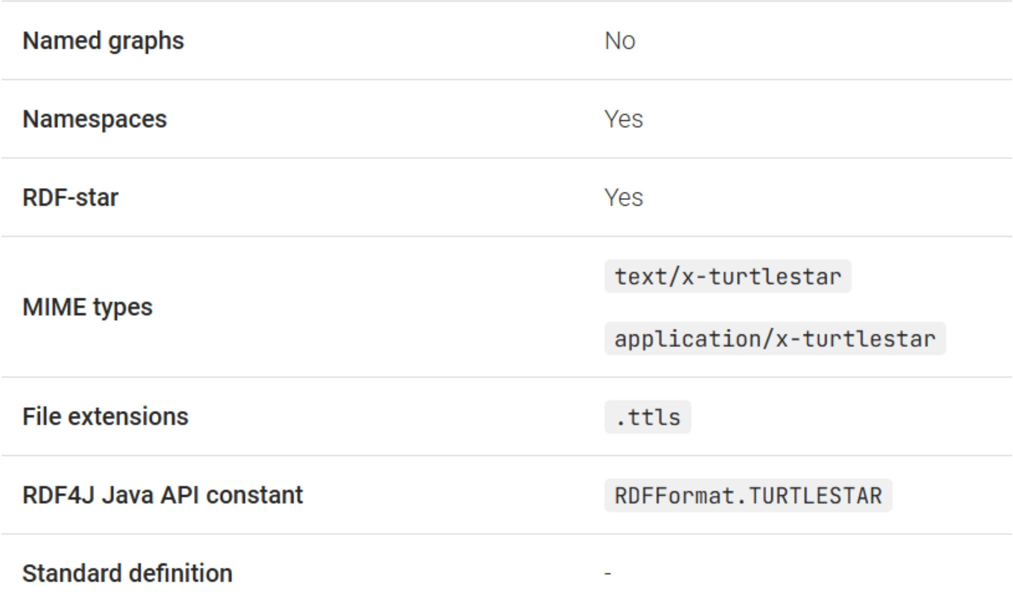


Named graphs	Yes
Namespaces	No
RDF-star	No
MIME types	`application/rdf+json`
File extensions	`.rj`
RDF4J Java API constant	`RDFFormat.RDFJSON`
Standard definition	http://www.w3.org/ns/formats/RDF_JSON

some additional ones, but maybe we don’t need that many 🙂🎉

---

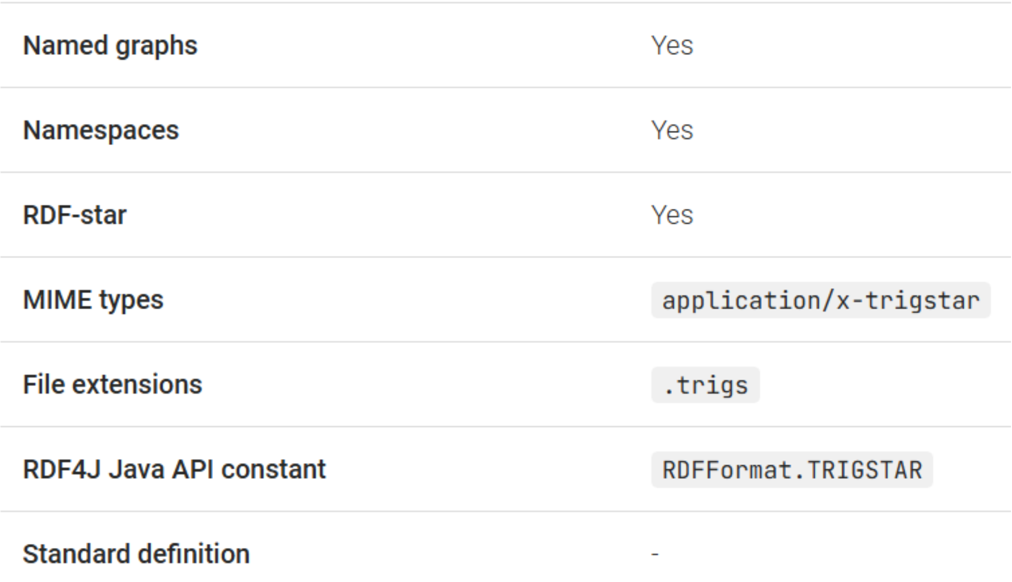
10. Turtle-star



Named graphs	No
Namespaces	Yes
RDF-star	Yes
MIME types	`text/x-turtlestar` `application/x-turtlestar`
File extensions	`.ttls`
RDF4J Java API constant	`RDFFormat.TURTLESTAR`
Standard definition	-

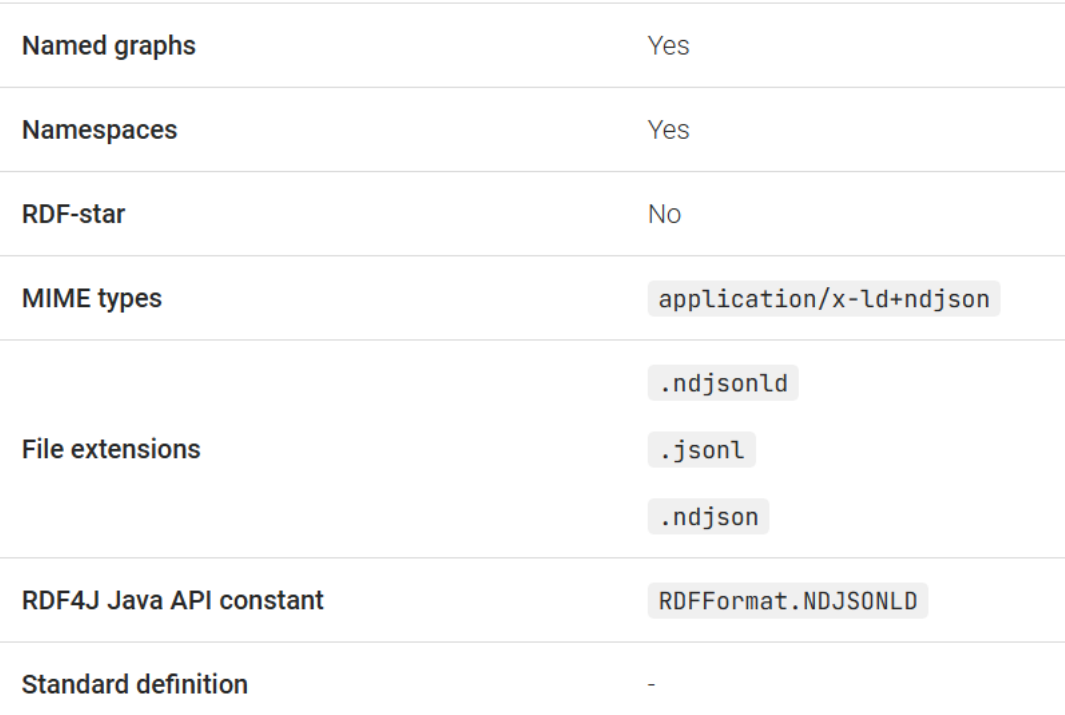
11. TriG-star

(text, human readable, standard based)



Named graphs	Yes
Namespaces	Yes
RDF-star	Yes
MIME types	`application/x-trigstar`
File extensions	`.trigs`
RDF4J Java API constant	`RDFFormat.TRIGSTAR`
Standard definition	-

12. NDJSON-LD

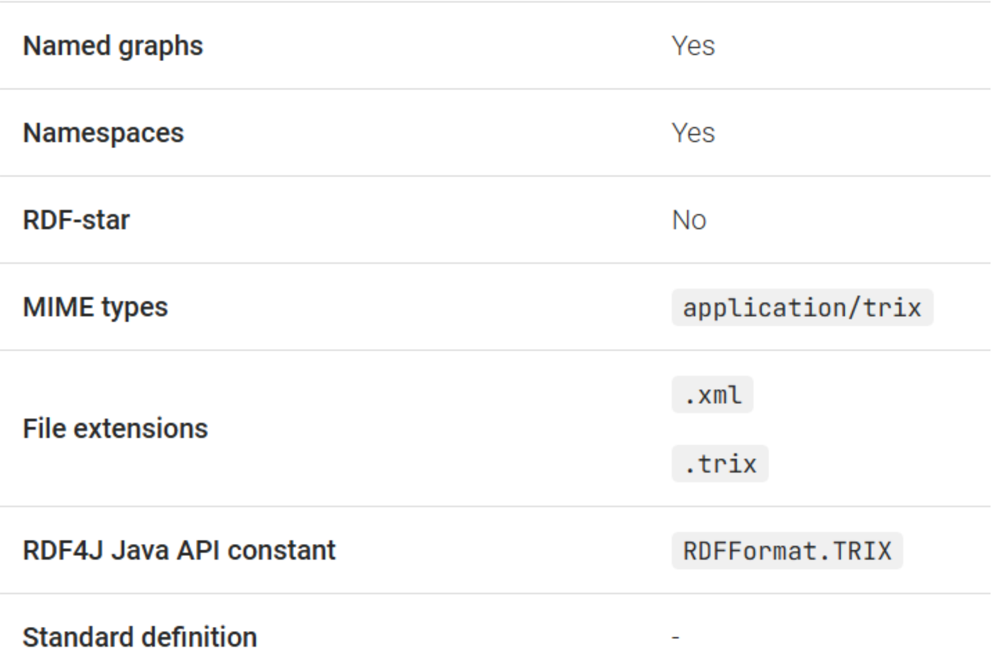


Named graphs	Yes
Namespaces	Yes
RDF-star	No
MIME types	`application/x-ld+ndjson`

File extensions	`.ndjsonld`
	`.jsonl`
	`.ndjson`

RDF4J Java API constant	`RDFFormat.NDJSONLD`
Standard definition	-

13. TriX



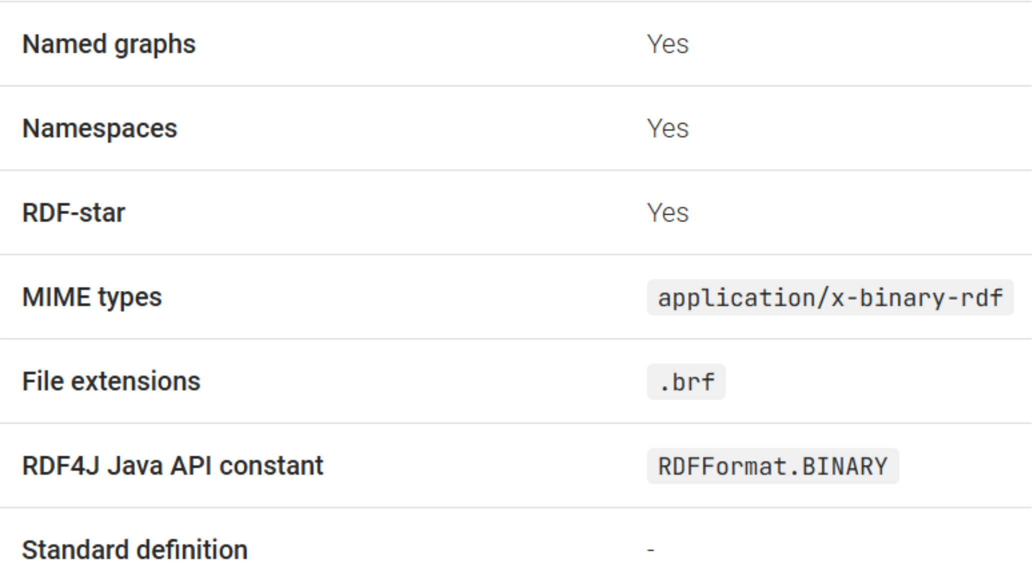
Named graphs	Yes
Namespaces	Yes
RDF-star	No
MIME types	`application/trix`

File extensions	`.xml`
	`.trix`

RDF4J Java API constant	`RDFFormat.TRIX`
Standard definition	-

14. BinaryRDF

(binary, compact representation, RDF4J-specific)



Named graphs	Yes
Namespaces	Yes
RDF-star	Yes
MIME types	`application/x-binary-rdf`
File extensions	`.brf`
RDF4J Java API constant	`RDFFormat.BINARY`
Standard definition	-

Side-by-side vs your current files

Udostępnij

Format	What it is	How it complements `feinschmecker.rdf` (RDF/XML)	When it's useful	Typical place in your repo
RDF/XML (already have)	XML syntax for RDF	Keep as **canonical** source (current)	Compatibility, many tools	`data/feinschmecker.rdf`
Turtle (.ttl)	Human-readable RDF	A **review-friendly** mirror of the KG	Code reviews, small curated samples, diffs	`data/snapshots/feinschmecker.ttl` and `data/samples/*.ttl`
JSON-LD (.jsonld)	JSON with `@context`	A **web/API-friendly** view (can be embedded in HTML)	REST responses, SEO, frontend use	`data/exports/feinschmecker.jsonld`
N-Triples (.nt)	One triple per line	**Simple, streamable** triple dump	ETL, grepping, quick debug	`data/dumps/feinschmecker.nt`
TriG (.trig)	Turtle for **datasets** (named graphs)	Human-readable **multi-graph** snapshot	Human-readable versioning/provenance	`data/snapshots/feinschmecker.trig`
N-Quads (.nq)	Line-oriented **quads**	Your **fastest archival/export** (one graph per day/source)	Nightly snapshots, bulk loads	`data/dumps/feinschmecker-YYYY-MM-DD.nq`
N3 (.n3)	Turtle-like + simple rules	Optional for **rules** (e.g., vegetarian inference)	Prototyping constraints/rules	`data/rules/*.n3`
RDFa (HTML)	RDF in HTML attrs	Output option if you publish recipe pages	Web extraction w/o JSON-LD	Inside your HTML templates
RDF/JSON (.rj)	Older JSON shape (not JSON-LD)	Only if a partner tool insists	Legacy interop	`data/exports/feinschmecker.rj`
recipes.json (already have)	Raw scraped JSON	Source for **conversion** → RDF (any syntax)	Scraping/ETL	`data/recipes.json`