fruschtique

by Norbert Luttenberger

Linked Data Format

for Ingredient Graphs

Overview

RDF for ingredient graphs

Fuseki triple store and SPARQL server

SPARQL sample queries

An OWL Ontology for ingredient graphs

RDF for ingredient graphs

What is RDF?

Resource **D**escription **F**ramework

- Resource is described by triples:
  subject – predicate – object

```
@prefix gndo: <http://d-nb.info/standards/elementset/gnd#> .

https://d-nb.info/gnd/118577166   gndo:variantNameForThePerson   "Mann, Paul Thomas";
                                  gndo:dateOfBirth               "1875-06-06"^^xsd:date;
                                  gndo:dateOfDeath               "1955-08-12"^^xsd:date;
                                  gndo:placeOfBirth              <https://d-nb.info/gnd/4036483-5>;
…
```

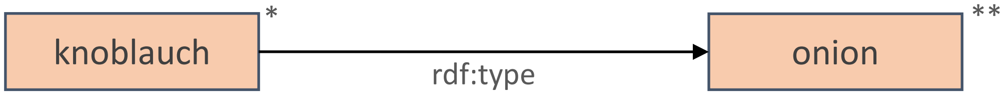
Subjects and predicates
identified by URI;
prefixes for abbreviated writing

Objects may be
- plain strings,
- typed data (XML Schema),

→ subjects of further triples.

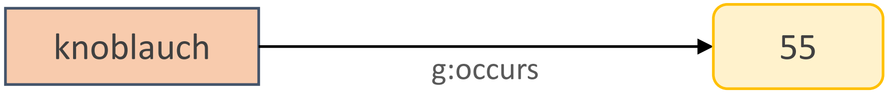
Modeling ingredients with RDF

1. Announce that ingredient is member of a certain ingredient class: RDF type



16 ingredient classes in total.

2. Announce that ingredient has an occurence value: g:occurs

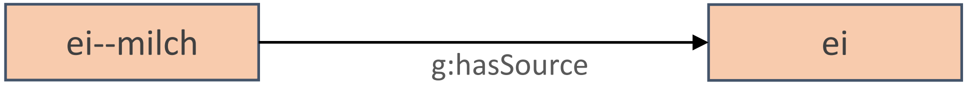


* ingredient IDs in lower case German

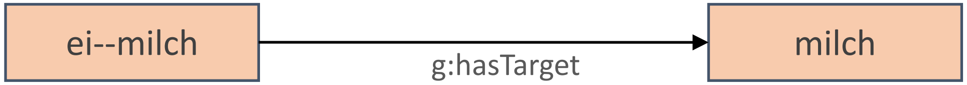
** ingredients classes in lower case English (disambiguate ingredient IDs and ingredient class IDs)

Modeling ingredient relations with RDF

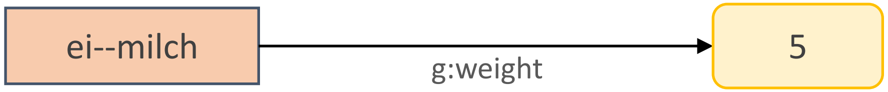
1. Announce that ingredient relation has a source: g:hasSource



2. Announce that ingredient relation has a target: g:hasTarget

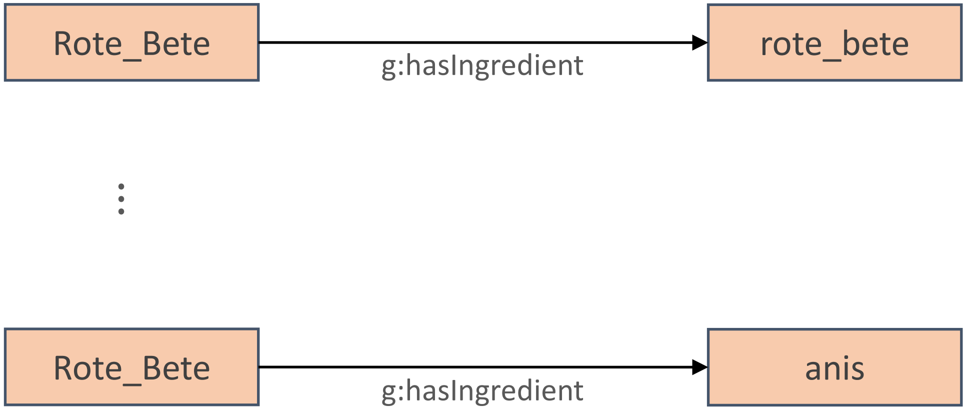


3. Announce that ingredient relation has a weight: g:weight



Modeling recipes with RDF

1. Announce that recipe has ingredients: g:hasIngredient



RDF for ingredient graphs

RDF types	ingredients	z.B.: `g:zwiebel rdf:type g:onion ;`

Properties for ingredients	g:hasClass	value from `g:i-class`
	g:occurs	value from `xsd:integer`

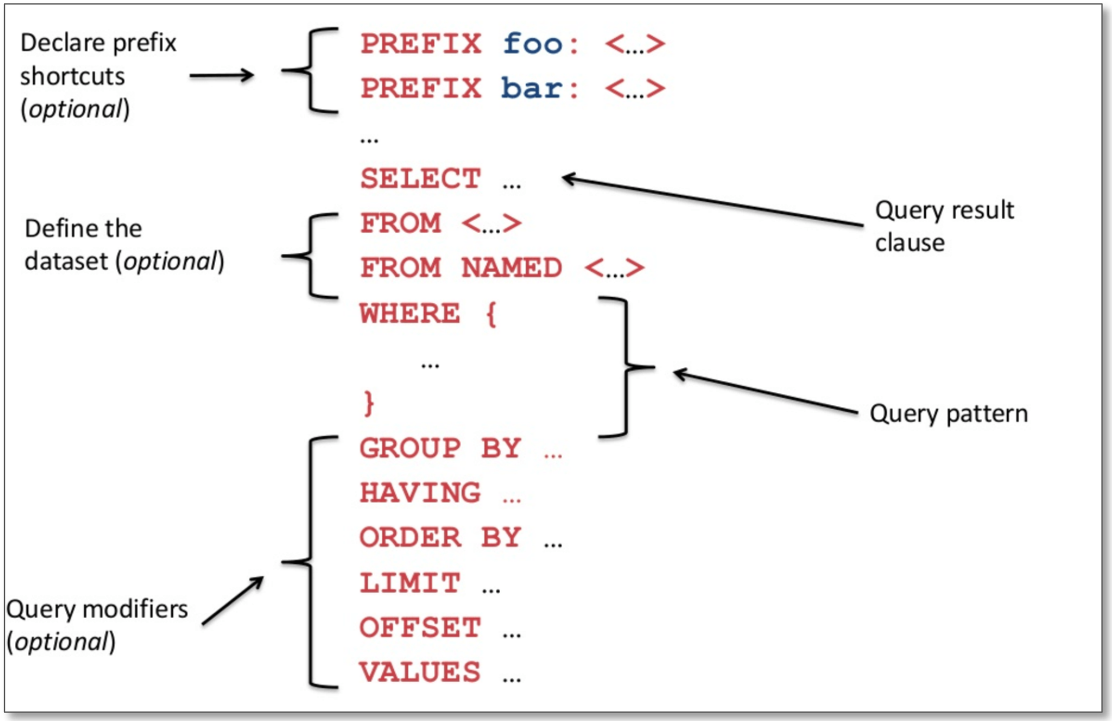
Properties for ingredient relations	g:hasSource	value from `g:ingredients`
	g:hasTarget	value from `g:ingredients`
	g:weight	value from `xsd:integer`

Properties for recipes	g:hasIngredient	value from `g:ingredients`

What is SPARQL?

SPARQL **P**rotocol **A**nd **R**DF **Q**uery **L**anguage

- Related RDF triples are collected in a dataset
- A dataset can be queried by SPARQL queries

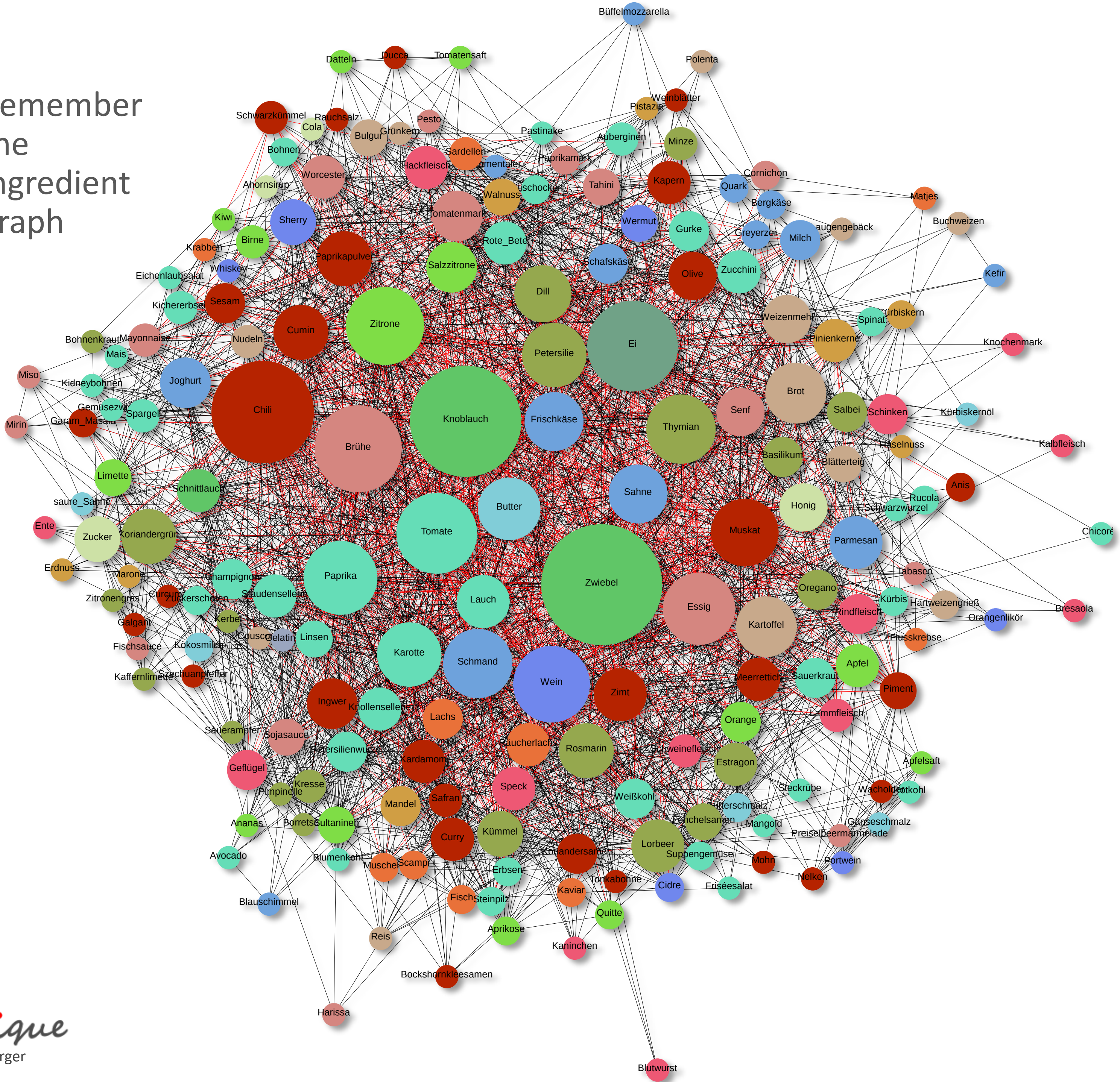


```
PREFIX foo: <…>
PREFIX bar: <…>
…
SELECT …
FROM <…>
FROM NAMED <…>
WHERE {
      …
}
GROUP BY …
HAVING …
ORDER BY …
LIMIT …
OFFSET …
VALUES …
```

Declare prefix shortcuts (*optional*)
Query result clause
Define the dataset (*optional*)
Query pattern
Query modifiers (*optional*)

https://wordlift.io/blog/en/entity/sparql/

Remember the ingredient graph



fruschtique
by Norbert Luttenberger

Fuseki as SPARQL server

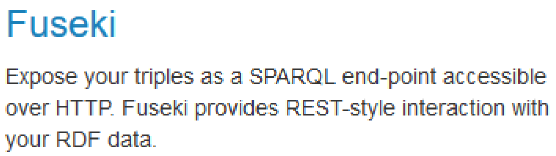
Apache Jena

Apache Jena

A free and open source Java framework for building Semantic Web and Linked Data applications.

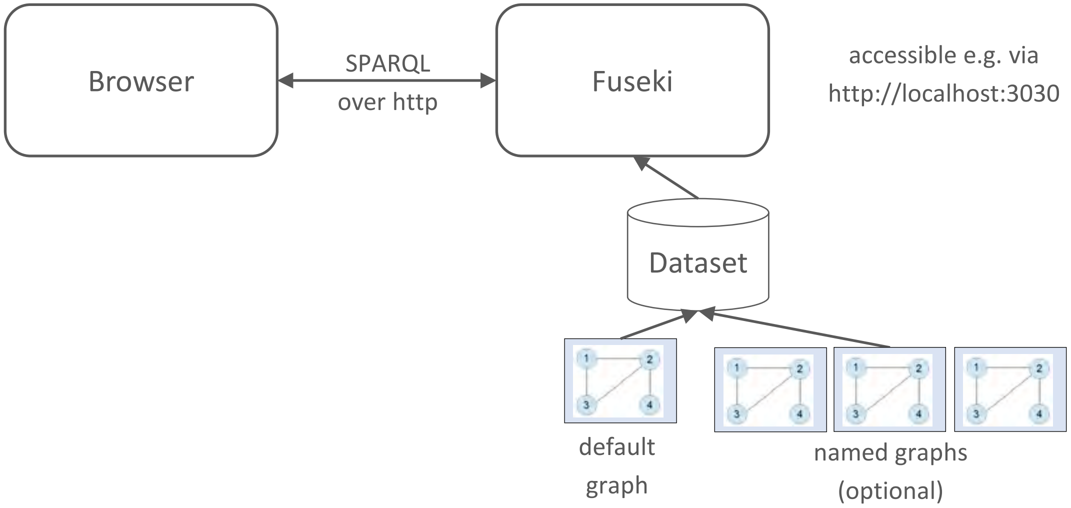
https://jena.apache.org/index.html

Fuseki



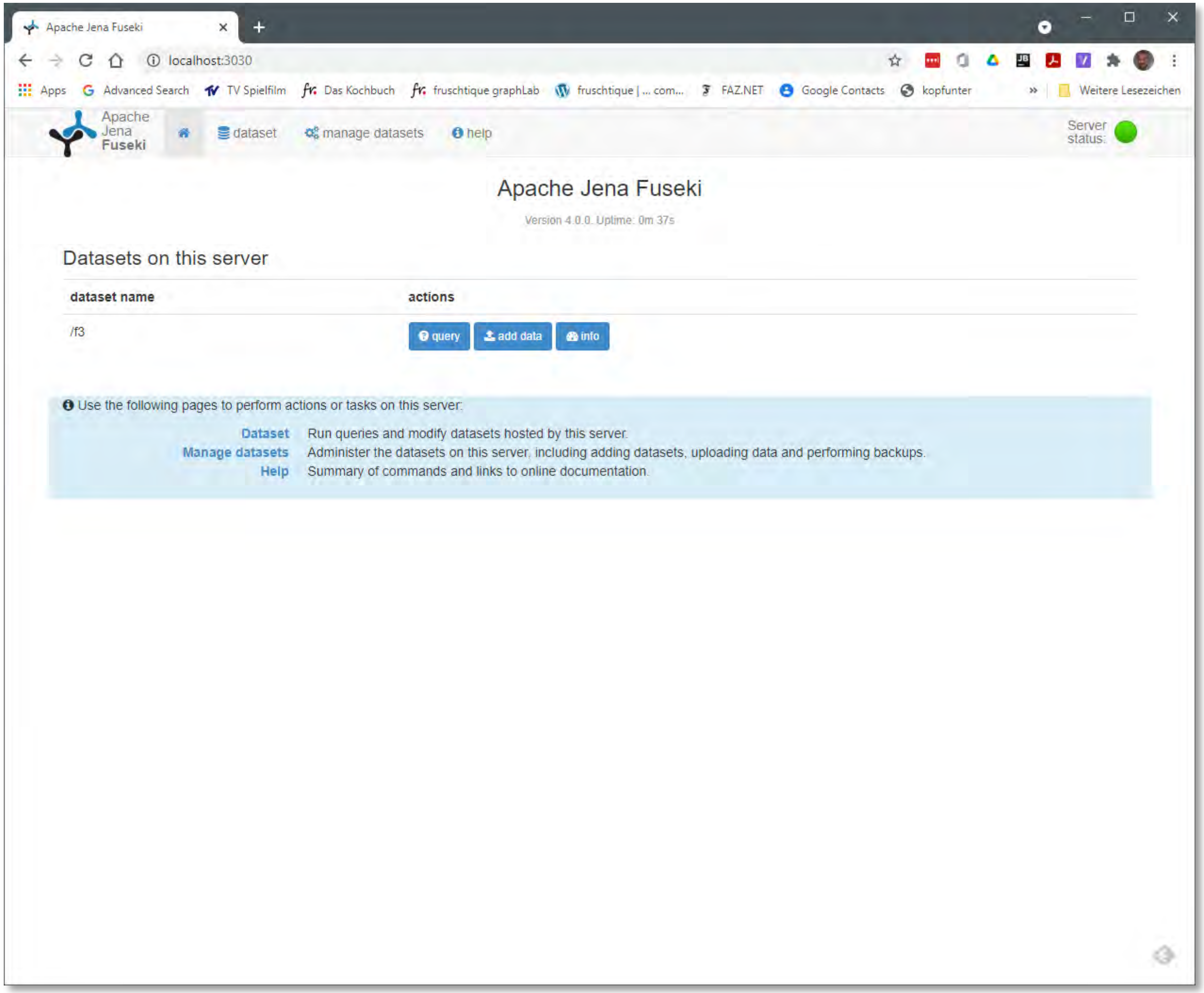
Fuseki

Expose your triples as a SPARQL end-point accessible over HTTP. Fuseki provides REST-style interaction with your RDF data.



Sample Fuseki access interface

comes with
Fuseki
download



Apache Jena Fuseki

Version 4.0.0. Uptime: 0m 37s

Datasets on this server

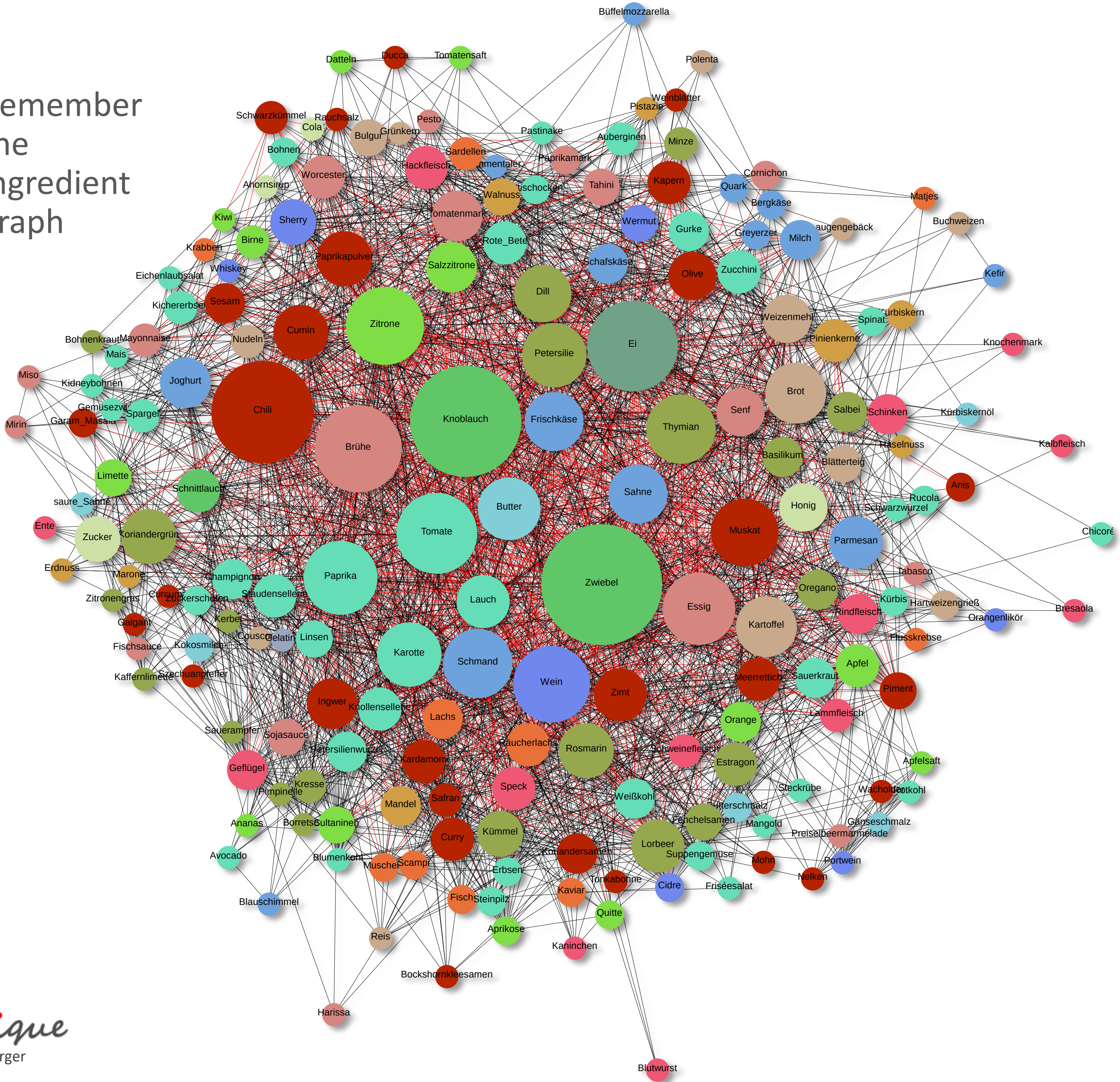
dataset name	actions
/f3	query · add data · info

Use the following pages to perform actions or tasks on this server:

Dataset	Run queries and modify datasets hosted by this server.
Manage datasets	Administer the datasets on this server, including adding datasets, uploading data and performing backups.
Help	Summary of commands and links to online documentation.

Sample SPARQL queries

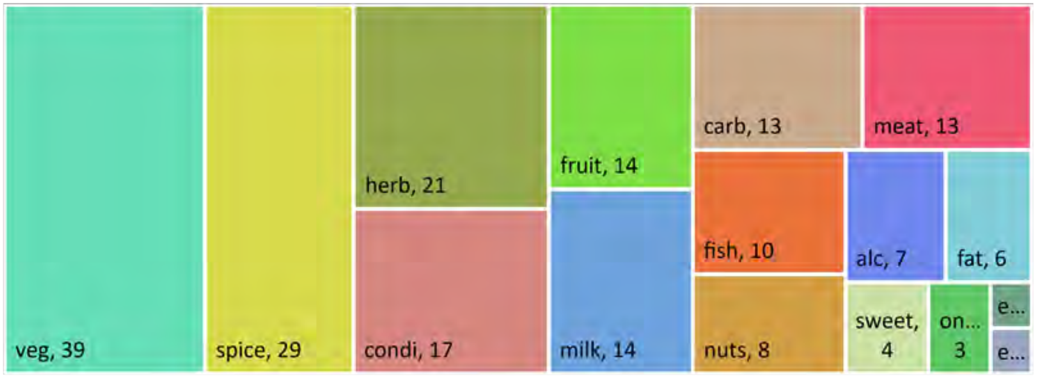
Remember the ingredient graph



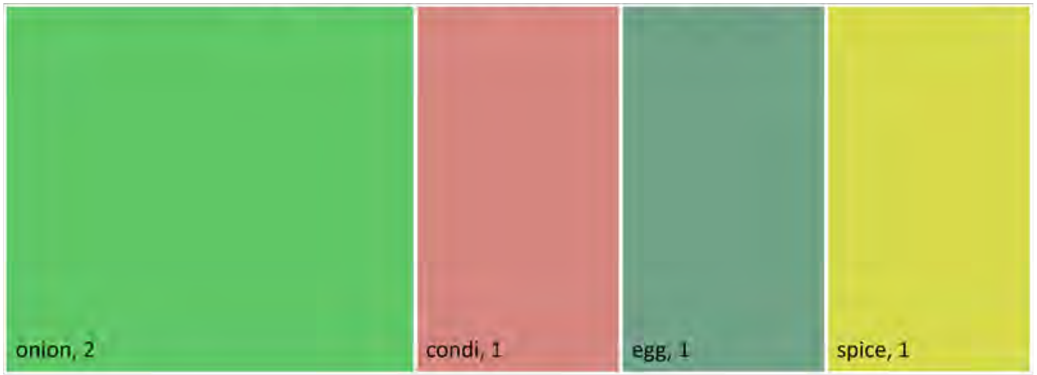
fruschtique
by Norbert Luttenberger

Population of ingredient classes

```
SELECT ?class (count (*) AS ?num)
WHERE {
  ?i g:hasClass ?class.
}
GROUP BY ?class ?num
ORDER BY ASC(?class)
```


Ingredient Class Population

```
SELECT ?class (count (*) AS ?num)
WHERE {
  ?i g:hasClass ?class.
  ?i g:occurs ?o.
  FILTER (?o > "30"^^xsd:integer)
}
GROUP BY ?class ?num
ORDER BY ASC(?class)
```

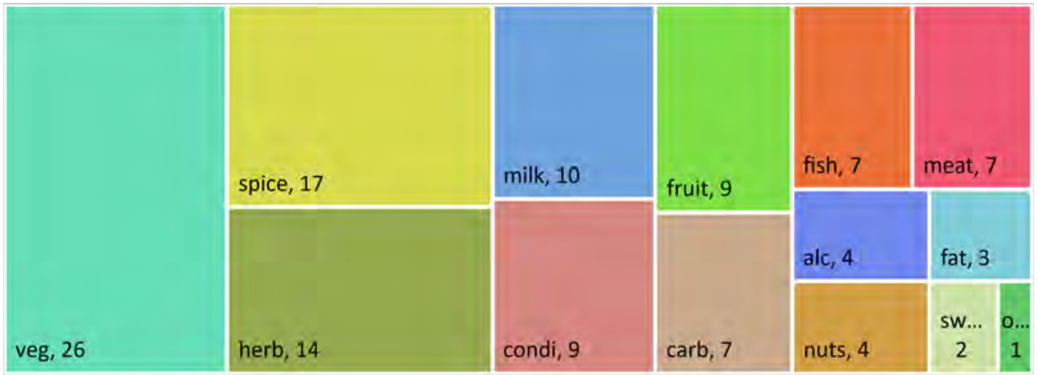

Ingredient Class Population "Big Five"

```
PREFIX g: <http://www.semanticweb.org/nlutt/ontologies/2021/1/graDL.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

Population of ingredient classes

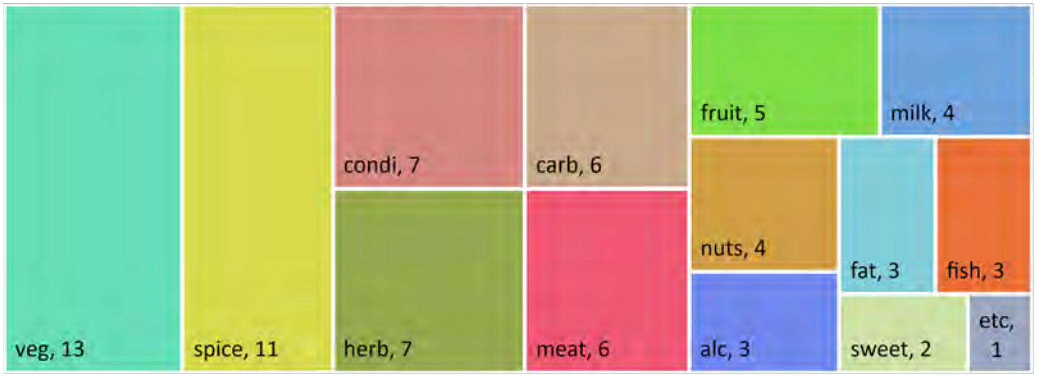
```
SELECT ?class (count (*) AS ?num)
WHERE {
  ?i g:hasClass ?class.
  ?i g:occurs ?o.
  FILTER (?o > "1"^^xsd:integer)
  FILTER (?o < "30"^^xsd:integer)
}
GROUP BY ?class ?num
ORDER BY ASC(?class)
```

Ingredient Class Population "Middle-earth"



```
SELECT ?class (count (*) AS ?num)
WHERE {
  ?i g:hasClass ?class.
  ?i g:occurs ?o.
  FILTER (?o = "1"^^xsd:integer)
}
GROUP BY ?class ?num
ORDER BY ASC(?class)
```

Ingredient Class Population "Zero-nodes"



```
PREFIX g: <http://www.semanticweb.org/nlutt/ontologies/2021/1/graDL.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

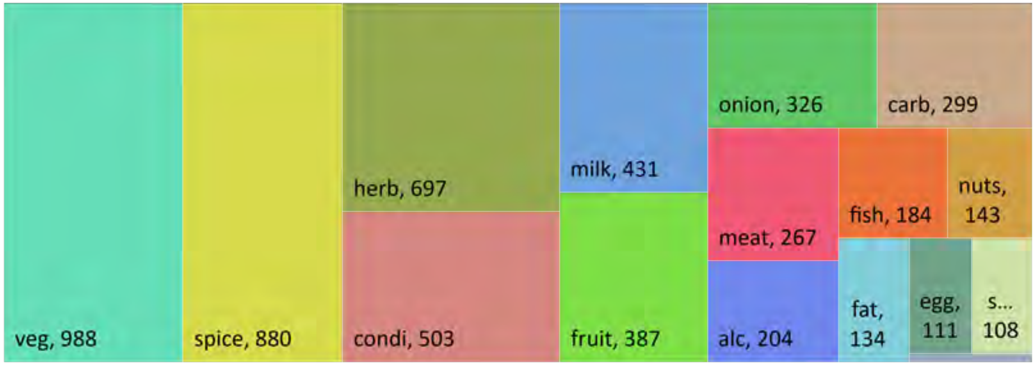
fruschtique
by Norbert Luttenberger

Relations classified

```
SELECT   ?class (COUNT (*) AS ?num)
WHERE {
  {?rel g:hasSource/g:hasClass ?class.}
  UNION
  {?rel g:hasTarget/g:hasClass ?class.}
}
GROUP BY ?class ?class
ORDER BY DESC (?num)
```

Mind that relations are counted double by this query!

Relations classified by source and target ingredient classes



```
PREFIX g: <http://www.semanticweb.org/nlutt/ontologies/2021/1/graDL.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

An OWL ontology for ingredient graphs

Graph-theoretic interlude

An ingredient graph $IG = (V, E, w, c, o)$ is an undirected, loop-free, weighted, and node-labeled graph where

- V is a finite set of ingredient nodes identified by their ID and
- $E \subseteq \binom{V}{2}$ is a set of edges.
- w is a function that assigns a weight value from $\mathbb{N} \backslash \{0\}$ to each edge,
- c is a function that assigns a class value to each node, where this value comes from $class = \{alc, carb, condi, egg, etc, fat, fish, fruit, herb, meat, milk, nuts, onion, spice, sweet, veg\}$
- o is a function that assigns an occurence value ω to each node with $\omega \in \mathbb{N} \backslash \{0\}$.
- The occurence value can easily be transformed into a prevalence value by dividing it by the number of recipes in the collection under scrutiny.

Thank you!