STTL
SPARQL Template Transformation Language for RDF Graphs

Olivier Corby
INRIA Sophia Antipolis

olivier.corby@inria.fr

STTL

STTL : transformation language for RDF

XSLT : transformation language for XML

- Input RDF graph
- Output Text format
- SPARQL based
- Declarative transformation rules

XSLT - STTL

```
<xsl:template  match="person">
        <xsl:apply-templates select="knows"/>
</xsl:template>
```

```
template { st:apply-templates(?y) }
where { ?in a foaf:Person ; foaf:knows ?y }
```

XSLT - STTL

	XSLT	STTL
Input	XML	RDF
Output	XML	Text
Syntax	XML	SPARQL extension
Template	xsl:template	template where
Named template	xsl:template name="test"	template ex:test
Apply templates	xsl:apply-templates	st:apply-templates
Apply named template	xsl:call-template	st:call-template
Parameters	xsl:with-param	(?x, ?y)
Numbering	xsl:number	st:number
Sorting	xsl:sort	order by
Grouping	xsl:for-each-group	group by
Condition	xsl:if	if (exp, then, else)

Differences XSLT - STTL

- XSLT :
  - XML Tree
  - Edges are ordered
- STTL :
  - RDF Graph
  - Edges are not ordered

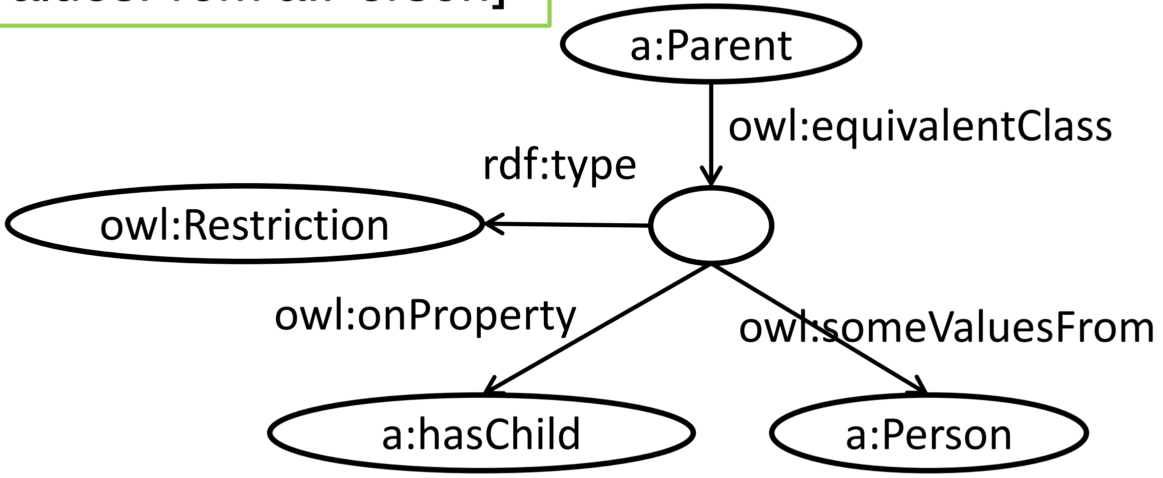
STTL motivating use cases

1. Transformation of RDF data from one RDF syntax to another:
   - Turtle
   - RDF/XML
   - JSON LD
2. Presentation of RDF data:
   - RDF to HTML
   - RDF to Latex
   - RDF to Natural Language
   - RDF to graphic format (GML)
3. Transformation of statements in a given language from RDF to another syntax:
   - OWL/RDF to OWL functional syntax
   - SPARQL/RDF (SPIN) to SPARQL syntax
   - AST of *L* in RDF to concrete syntax of *L*
4. Constraint checking
   - OWL Profile: OWL RL
   - SHACL

Example use case: OWL/RDF to OWL/FS

```
a:Parent owl:equivalentClass [ a owl:Restriction ;
                               owl:onProperty a:hasChild;
                               owl:someValuesFrom a:Person]
```

OWL/RDF (Turtle)



STTL transformation

```
EquivalentClasses (a:Parent ObjectSomeValuesFrom(a:hasChild, a:Person) )
```

OWL/Functional Syntax

SPARQL

Query forms

1. SELECT WHERE { GP }
2. CONSTRUCT { GP } WHERE { GP }
3. ASK { GP }
4. DESCRIBE WHERE { GP }

SPARQL Template

Query forms

1. SELECT WHERE { GP }
2. CONSTRUCT { GP } WHERE { GP }
3. ASK { GP }
4. DESCRIBE WHERE { GP }

5. TEMPLATE { Text Pattern } WHERE { GP }

SPARQL Template

An additional SPARQL query form:

```
TEMPLATE { Text Pattern } WHERE { GP }
```

with Text Pattern = (VARIABLE | EXP | TEXT)*

RDF to Turtle transformation

```
TEMPLATE { ?x  " "  rdfs:label  " "  ?name  "." }
WHERE { ?x a foaf:Person ; foaf:name ?name }
```

```
ns:olivier a foaf:Person ; foaf:name "Olivier".
ns:catherine a foaf:Person ; foaf:name "Catherine".
```

```
ns:olivier rdfs:label "Olivier".
ns:catherine rdfs:label "Catherine".
```

RDF to HTML transformation

```
TEMPLATE { format {"<a href='%s'>%s</a>"  str(?x) str(?name) } }
WHERE { ?x a foaf:Person ; foaf:name ?name }
```

```
ns:olivier a foaf:Person ; foaf:name "Olivier".
ns:catherine a foaf:Person ; foaf:name "Catherine".
```

```
<a href='http://ns.inria.fr/olivier'>Olivier</a>
<a href='http://ns.inria.fr/catherine'>Catherine</a>
```

STTL: Transformation

A set of templates

```
TEMPLATE { "EquivalentClasses (" ?in " " ?c ")" }
WHERE { ?in owl:equivalentClass ?c }

TEMPLATE { "SubClassOf (" ?in " " ?c ")" }
WHERE { ?in rdfs:subClassOf ?c }

TEMPLATE { "ObjectSomeValuesFrom (" ?p " " ?c ")" }
WHERE { ?in a owl:Restriction ;
          owl:onProperty ?p ;
          owl:someValuesFrom ?c }
```

Template recursive call

```
TEMPLATE { "EquivalentClasses ("
    ?in " " ?c ")" }
WHERE { ?in owl:equivalentClass ?c . }
```

Template recursive call

```
TEMPLATE { "EquivalentClasses ("
    st:apply-templates(?in) " " ?c ")" }
WHERE { ?in owl:equivalentClass ?c . }
```

Template recursive call

```
TEMPLATE { "EquivalentClasses ("
    st:apply-templates(?in) " " st:apply-templates(?c) ")" }
WHERE { ?in owl:equivalentClass ?c . }
```

STTL

1. **Template**: SPARQL Template Query form
2. **Transformation**: set of templates
3. **Extension function**: st:apply-templates, st:call-template

Focus Node

```
template {
        st:apply-templates(?y)
}
where { ?in foaf:knows ?y }
```

Focus Node

```
template {
        st:apply-templates(?y)
}
where { ?in foaf:knows ?y }

template {}
where {
        ?in a foaf:Person
}
```

Named Template

```
template {
        st:call-template(st:title)
}
where {}
```

Named Template

```
template {
        st:call-template(st:title)
}
where {}

template st:title {}
where {}
```

Named Template

```
template {
        st:call-template(st:title, ?y)
}
where {}

template st:title (?x) {}
where {}
```

STTL Features

STTL Extension Functions

prefix st: <http://ns.inria.fr/sparql-template/>

st:apply-templates(term)

st:apply-templates-with(transform-uri, term)

st:call-template(template-uri, term)

st:call-template-with(transform-uri, template-uri, term)

st:turtle(term)

Priority

```
template { }
where { }
pragma { st:template st:priority 200 }
```

Start template

```
template st:start {
	st:apply-templates(?x)
}
where {
	?x a foaf:Person
}
```

Extension Functions

```
function us:display(?x) {
   if (isBlank(?x),
         concat("bnode: " , ?x),
         st:turtle(?x))
}
```

Profile: Define Extension Functions

```
template **st:profile** {}
where {}

function us:display(?x) {
   if (isBlank(?x),
         concat("bnode: " , ?x),
         st:turtle(?x))
}
```

Variable Processing

```
template { ?y }
where { ?in ?p ?y }

template { st:process(?y) }
where { ?in ?p ?y }

function st:process(?x) {
        st:turtle(?x)
}
```

Overloading Variable Processing

```
function st:process(?x) {
  if (isBlank(?x),
        st:apply-templates(?x),
        st:turtle(?x))
}
```

Template Statements

- Separator
- Format
- Group
- Box
- Numbering

Separator

```
template {
        ?y
        ;  separator = ", "
}
where {
        ?in foaf:knows ?y
}
```

Format

```
template {
 format  {
        "<h2>%1$s</h2><p>%2$s</p>"

        st:apply-templates(?x)
        st:apply-templates(?y)
 }
}
where {
}
```

External Format

```
template {
  format  {
          <http://example.org/format/test.html>

          st:apply-templates(?x)
          st:apply-templates(?y)
  }
}
where {
}
```

Format Function

st:format(format, exp+)

Group

group { E1 .. En }

::=

group_concat(concat(E1, .. En))

Group

```
template {
        ?in " : " group { ?y }
}
where {
        ?in foaf:knows ?y
}
```

Box

box { E1 .. En }

::=

concat(E1, .. En)

st:nl()

box | sbox | ibox

Box

box: nl(+1) exp nl(-1)

sbox: nl(+1) exp indent(-1)

ibox: indent(+1) exp indent(-1)

Numbering

```
template {
        st:number() " " st:apply-templates(?x)
}
where {
        ?in foaf:knows ?y
}
order by ?x
```

Compiling STTL

```
template { E1 .. En }
where {}

compiled as :

select (concat(cp(E1), .. cp(En)) as ?out)
where {}
+
aggregate(Ω, group_concat, ?out)
```

Compiling STTL

cp(Var(x)) = st:process(x)

Default:

```
st:process(?x) = st:turtle(?x)
```

Overloaded:

```
function st:process(?x) {
  st:apply-templates(?x)
}
```

STTL Transformations

1.	RDF to Turtle	st:turtle
2.	RDF to RDF/XML	st:rdfxml
3.	RDF to JSON-LD	st:jsonld
4.	OWL to Functional Syntax	st:owl
5.	SPIN to SPARQL	st:spin
6.	SPARQL Query Result	st:sparql
7.	SPARQL Tutorial	st:web
8.	DBpedia Navigator	st:navlab
9.	Wikipedia Edit History Navigator	st:dbedit
10.	Calendar	st:calendar
11.	History Timeline	
12.	Sudoku (1 template)	
13.	OWL Profile check	st:owlrl
14.	SHACL Validation	st:dsmain

Usage

Create a directory e.g. sttl

Write templates in separate files, with extension .rq

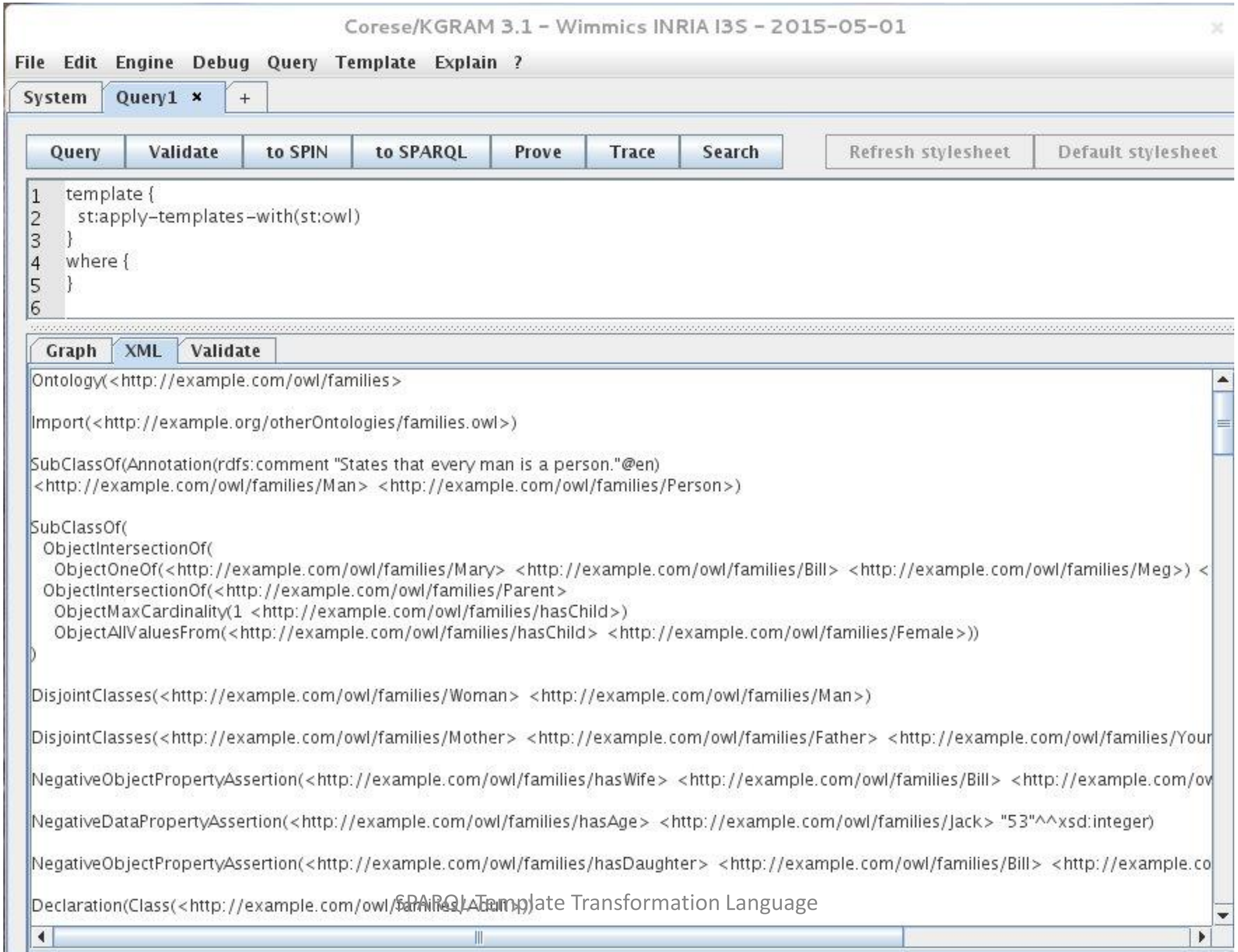
Use:

```
st:apply-templates-with("/home/myself/sttl/")
```

Use in Java:

```
Transformer t = Transformer.create(g, "/home/myself/sttl/");
String str = t.transform();
```

STTL development environment


Corese/KGRAM 3.1 - Wimmics INRIA I3S - 2015-05-01
File Edit Engine Debug Query Template Explain ?
System Query1 × +
Query Validate to SPIN to SPARQL Prove Trace Search
Refresh stylesheet Default stylesheet
Graph XML Validate

```
template {
  st:apply-templates-with(st:owl)
}
where {
}
```

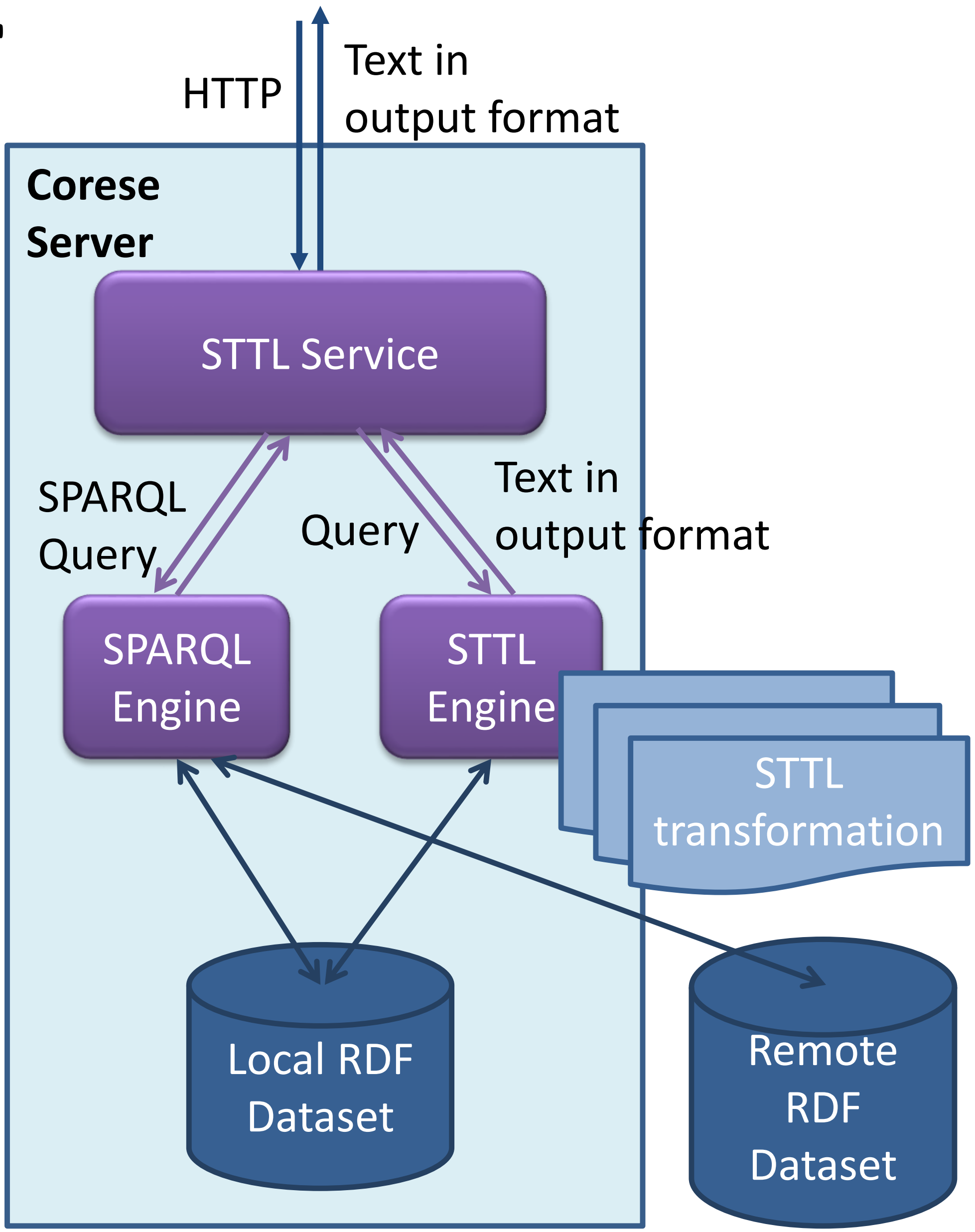
```
Ontology(<http://example.com/owl/families>

Import(<http://example.org/otherOntologies/families.owl>)

SubClassOf(Annotation(rdfs:comment "States that every man is a person."@en)
<http://example.com/owl/families/Man> <http://example.com/owl/families/Person>)

SubClassOf(
  ObjectIntersectionOf(
    ObjectOneOf(<http://example.com/owl/families/Mary> <http://example.com/owl/families/Bill> <http://example.com/owl/families/Meg>) <
  ObjectIntersectionOf(<http://example.com/owl/families/Parent>
    ObjectMaxCardinality(1 <http://example.com/owl/families/hasChild>)
    ObjectAllValuesFrom(<http://example.com/owl/families/hasChild> <http://example.com/owl/families/Female>))
)

DisjointClasses(<http://example.com/owl/families/Woman> <http://example.com/owl/families/Man>)

DisjointClasses(<http://example.com/owl/families/Mother> <http://example.com/owl/families/Father> <http://example.com/owl/families/Your

NegativeObjectPropertyAssertion(<http://example.com/owl/families/hasWife> <http://example.com/owl/families/Bill> <http://example.com/ov

NegativeDataPropertyAssertion(<http://example.com/owl/families/hasAge> <http://example.com/owl/families/Jack> "53"^^xsd:integer)

NegativeObjectPropertyAssertion(<http://example.com/owl/families/hasDaughter> <http://example.com/owl/families/Bill> <http://example.co

Declaration(Class(<http://example.com/owl/families/Adult>))
```

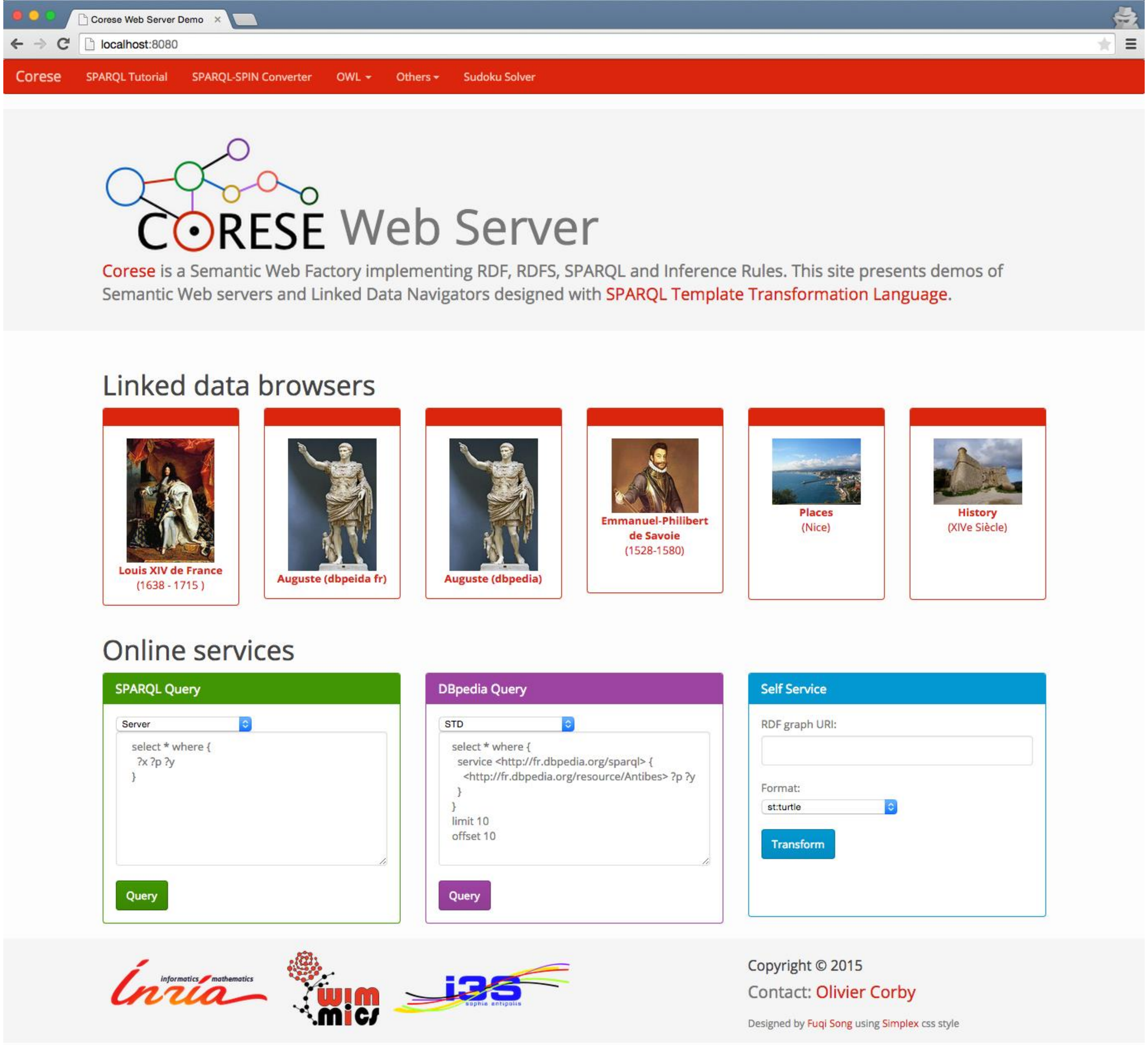
STTL engine

available in the Corese Semantic Web Factory

- Free download: http://wimmics.inria.fr/corese
  - SPARQL engine
  - STTL engine
  - Standalone environment to develop transformation
  - SPARQL endpoint
  - STTL server
- Web Server: http://corese.inria.fr

STTL Server


HTTP
Text in output format
Corese Server
STTL Service
SPARQL Query
Query
Text in output format
SPARQL Engine
STTL Engine
STTL transformation
Local RDF Dataset
Remote RDF Dataset



Web Server

Corese is a Semantic Web Factory implementing RDF, RDFS, SPARQL and Inference Rules. This site presents demos of Semantic Web servers and Linked Data Navigators designed with SPARQL Template Transformation Language.

Linked data browsers

Online services

SPARQL Query

Server

```
select * where {
  ?x ?p ?y
}
```

Query

DBpedia Query

STD

```
select * where {
 service <http://fr.dbpedia.org/sparql> {
   <http://fr.dbpedia.org/resource/Antibes> ?p ?y
  }
}
limit 10
offset 10
```

Query

Self Service

RDF graph URI:

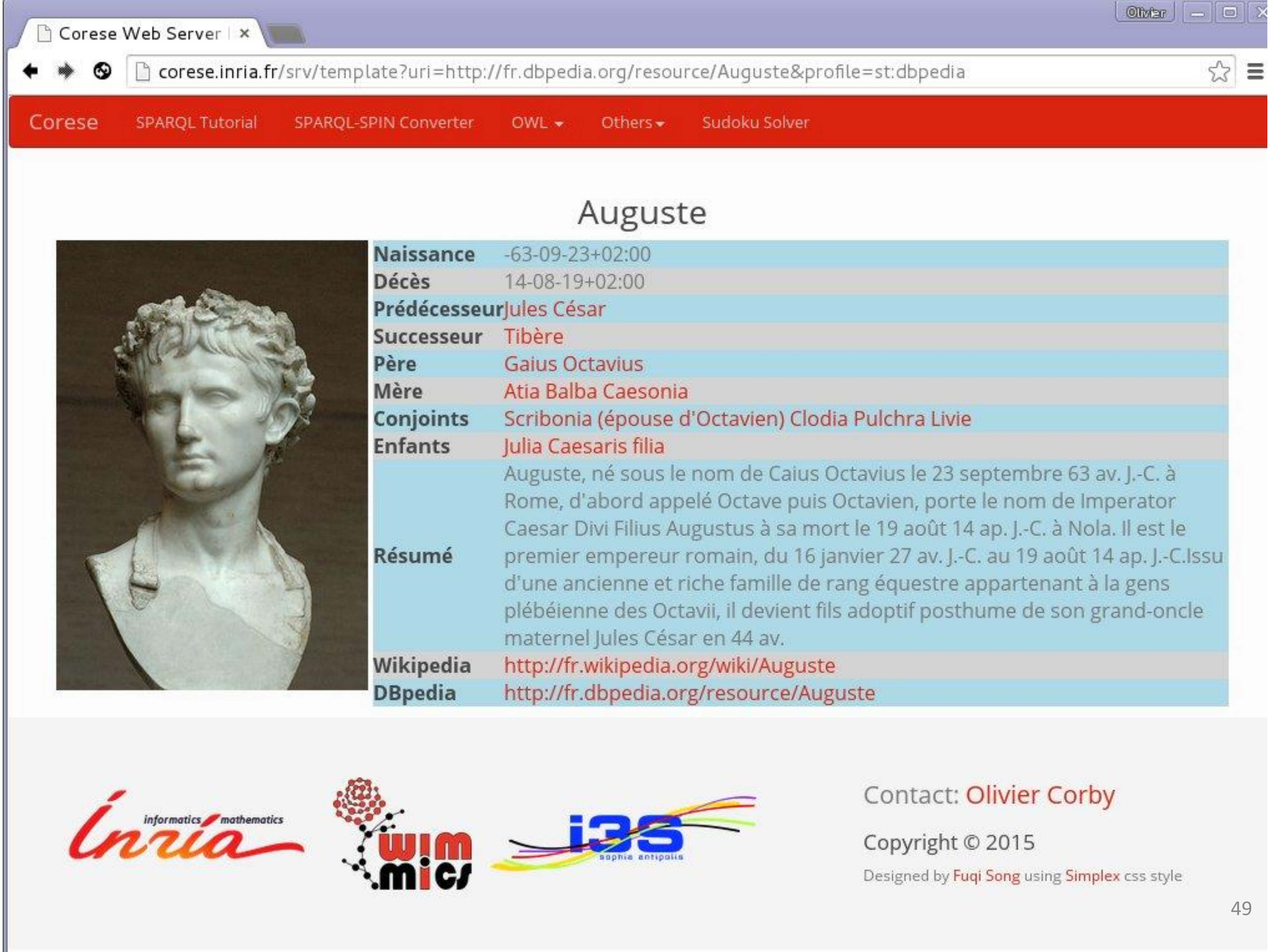
Format: st:turtle

Transform

Copyright © 2015

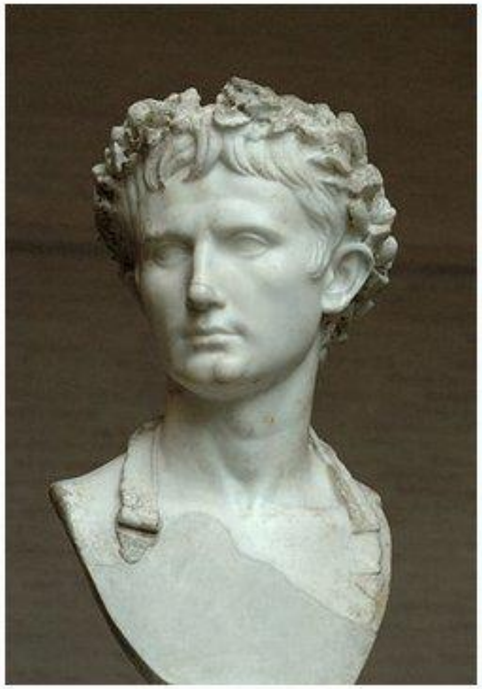
Contact: Olivier Corby

Designed by Fuqi Song using Simplex css style



Corese Web Server
corese.inria.fr/srv/template?uri=http://fr.dbpedia.org/resource/Auguste&profile=st:dbpedia
Corese
SPARQL Tutorial
SPARQL-SPIN Converter
OWL
Others
Sudoku Solver

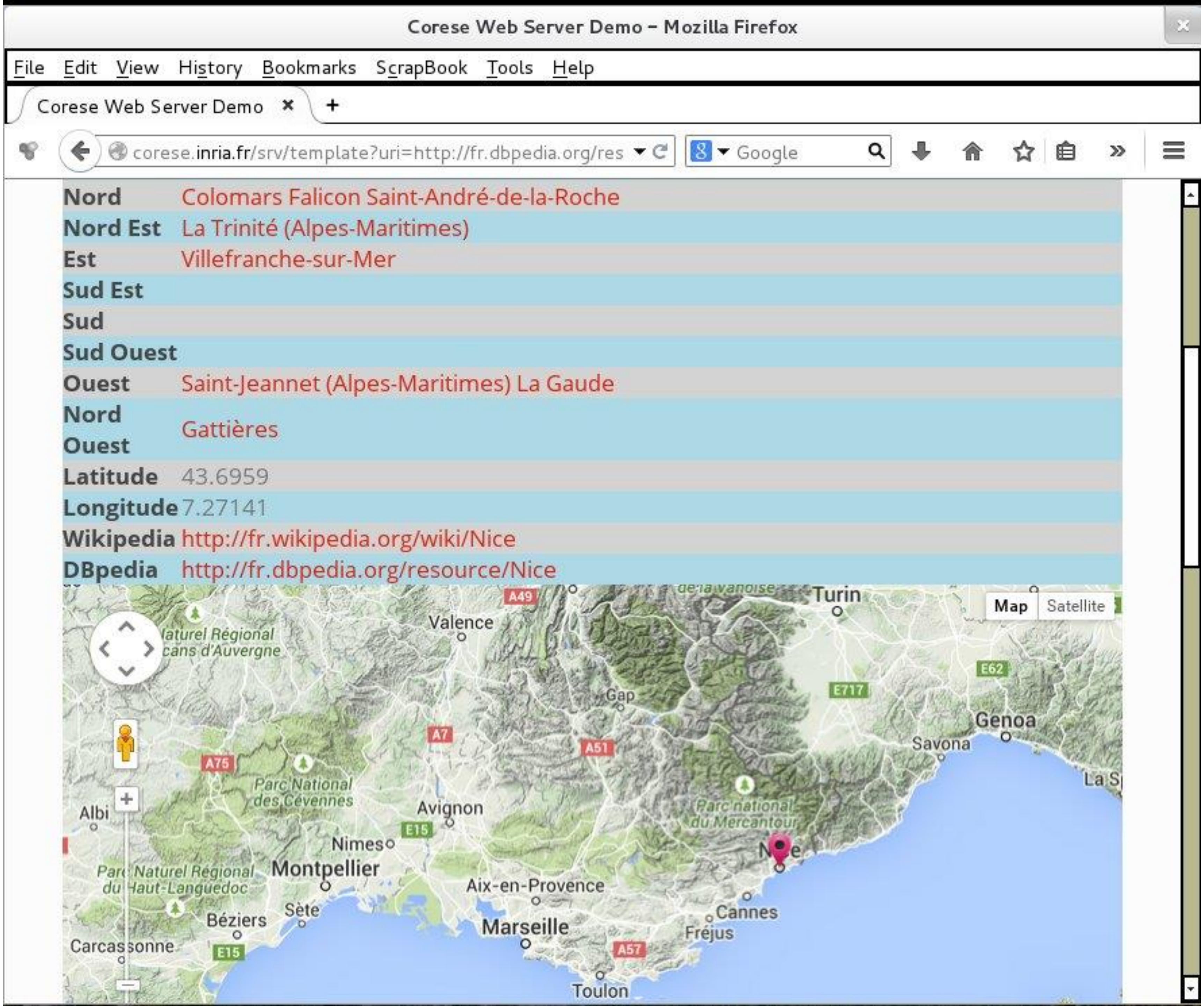
Auguste



Naissance	-63-09-23+02:00
Décès	14-08-19+02:00
Prédécesseur	Jules César
Successeur	Tibère
Père	Gaius Octavius
Mère	Atia Balba Caesonia
Conjoints	Scribonia (épouse d'Octavien) Clodia Pulchra Livie
Enfants	Julia Caesaris filia
Résumé	Auguste, né sous le nom de Caius Octavius le 23 septembre 63 av. J.-C. à Rome, d'abord appelé Octave puis Octavien, porte le nom de Imperator Caesar Divi Filius Augustus à sa mort le 19 août 14 ap. J.-C. à Nola. Il est le premier empereur romain, du 16 janvier 27 av. J.-C. au 19 août 14 ap. J.-C.Issu d'une ancienne et riche famille de rang équestre appartenant à la gens plébéienne des Octavii, il devient fils adoptif posthume de son grand-oncle maternel Jules César en 44 av.
Wikipedia	http://fr.wikipedia.org/wiki/Auguste
DBpedia	http://fr.dbpedia.org/resource/Auguste

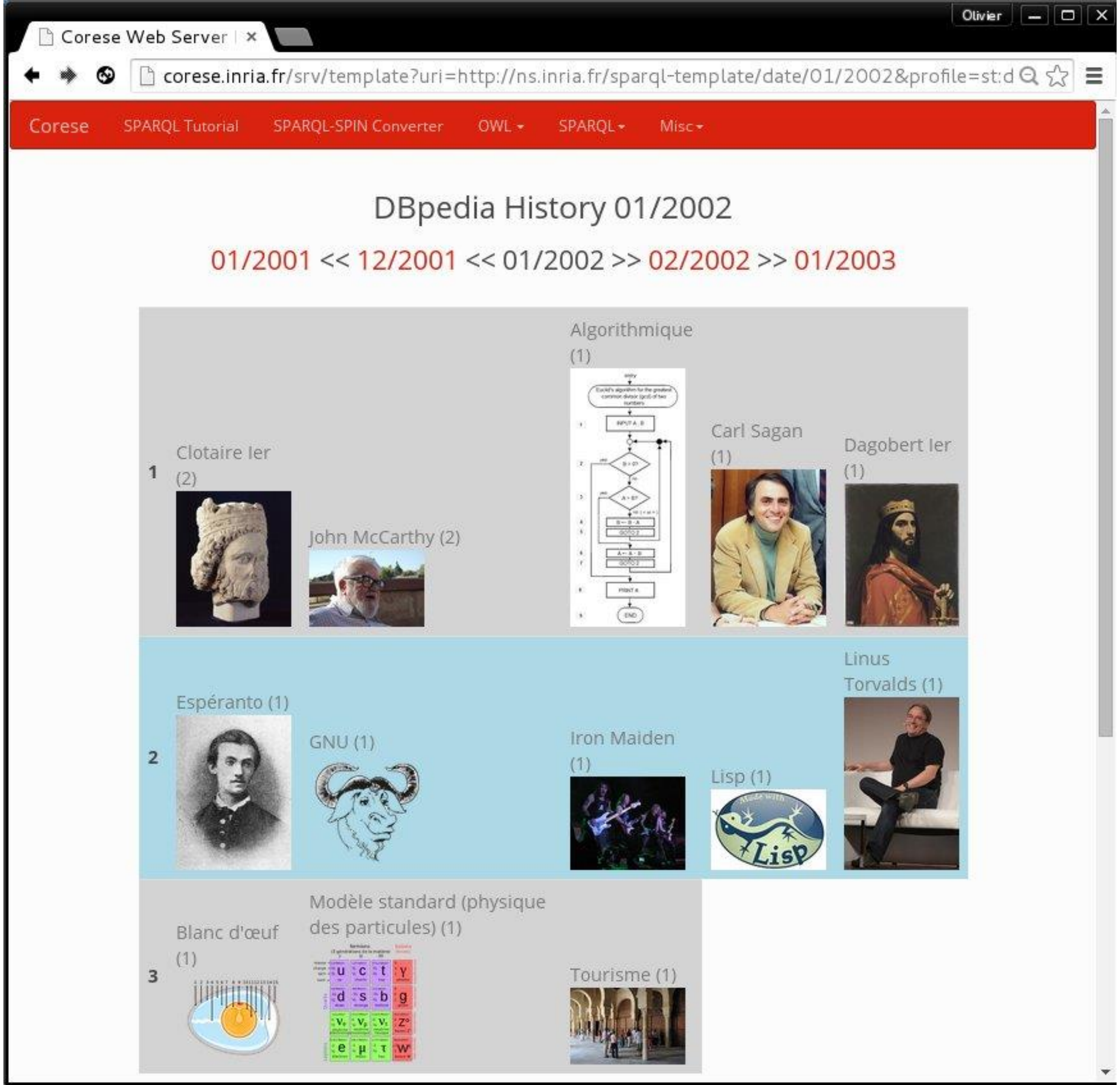
Contact: Olivier Corby

Designed by Fuqi Song using Simplex css style


Corese Web Server Demo – Mozilla Firefox
File Edit View History Bookmarks ScrapBook Tools Help
Corese Web Server Demo
corese.inria.fr/srv/template?uri=http://fr.dbpedia.org/res
Google

Nord	Colomars Falicon Saint-André-de-la-Roche
Nord Est	La Trinité (Alpes-Maritimes)
Est	Villefranche-sur-Mer
Sud Est	
Sud	
Sud Ouest	
Ouest	Saint-Jeannet (Alpes-Maritimes) La Gaude
Nord Ouest	Gattières
Latitude	43.6959
Longitude	7.27141
Wikipedia	http://fr.wikipedia.org/wiki/Nice
DBpedia	http://fr.dbpedia.org/resource/Nice

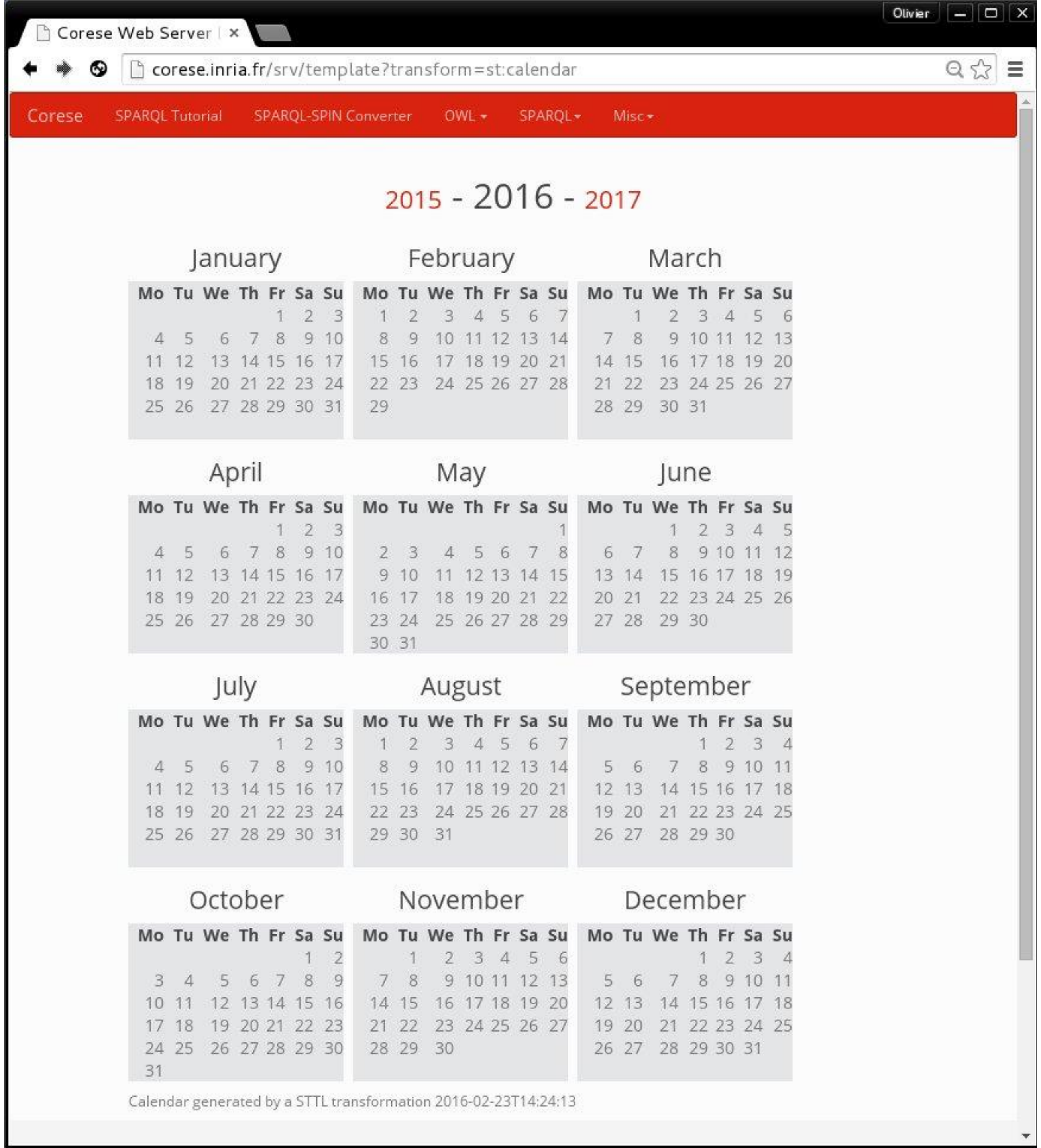
Map
Satellite
Turin
Valence
Gap
Genoa
Savona
Avignon
Nimes
Montpellier
Aix-en-Provence
Marseille
Toulon
Cannes
Fréjus
Béziers
Sète
Carcassonne
Albi
Parc National des Cévennes
Parc national du Mercantour


Olivier
Corese Web Server
corese.inria.fr/srv/template?uri=http://ns.inria.fr/sparql-template/date/01/2002&profile=st:d
Corese
SPARQL Tutorial
SPARQL-SPIN Converter
OWL
SPARQL
Misc

DBpedia History 01/2002

01/2001 << 12/2001 << 01/2002 >> 02/2002 >> 01/2003

1	Clotaire Ier (2)	John McCarthy (2)	Algorithmique (1)	Carl Sagan (1)	Dagobert Ier (1)
2	Espéranto (1)	GNU (1)	Iron Maiden (1)	Lisp (1)	Linus Torvalds (1)
3	Blanc d'œuf (1)	Modèle standard (physique des particules) (1)	Tourisme (1)		



2015 - 2016 - 2017

January

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

February

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29						

March

Mo	Tu	We	Th	Fr	Sa	Su
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

April

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

May

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

June

Mo	Tu	We	Th	Fr	Sa	Su
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

July

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

August

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

September

Mo	Tu	We	Th	Fr	Sa	Su
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

October

Mo	Tu	We	Th	Fr	Sa	Su
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

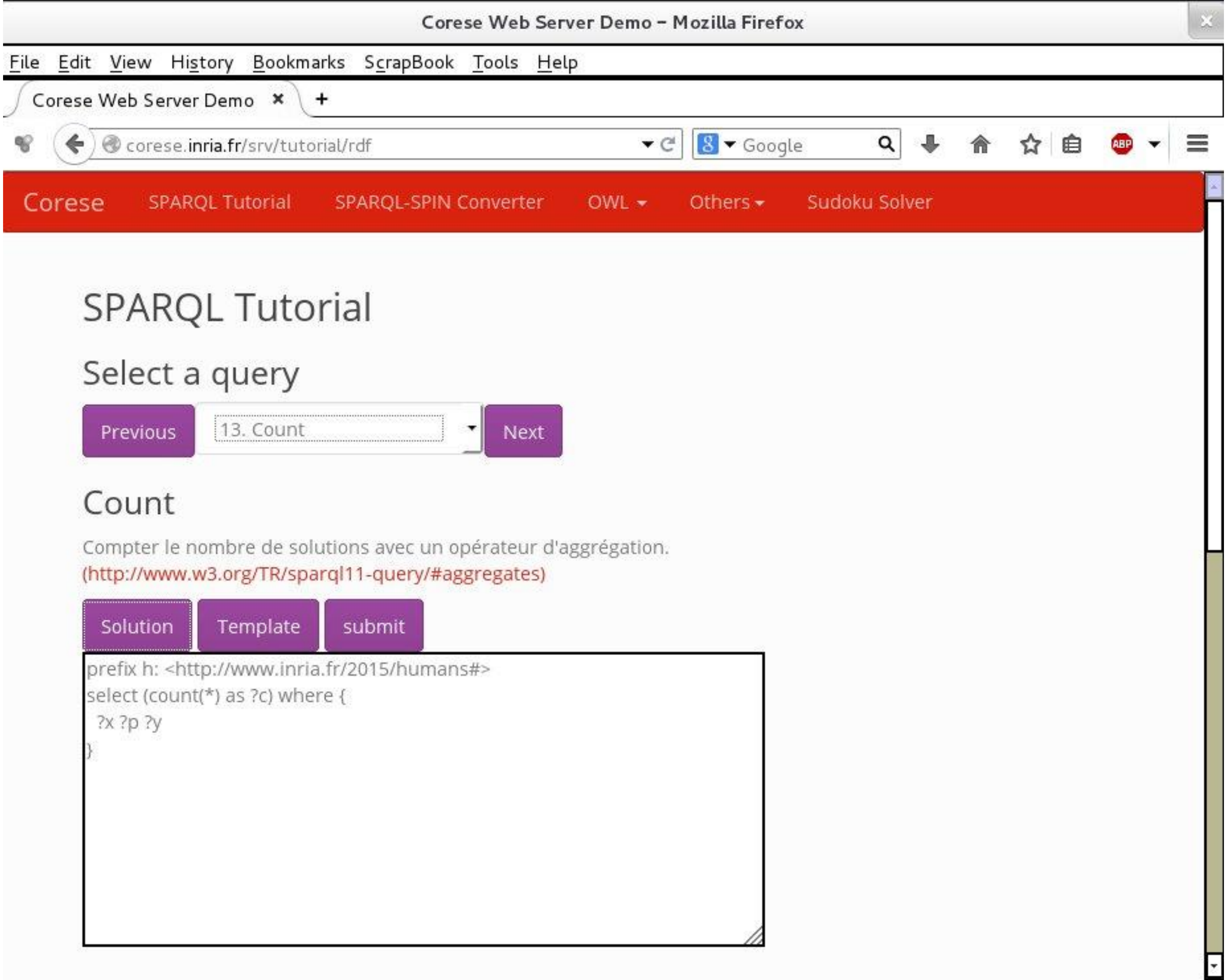
November

Mo	Tu	We	Th	Fr	Sa	Su
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

December

Mo	Tu	We	Th	Fr	Sa	Su
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Calendar generated by a STTL transformation 2016-02-23T14:24:13



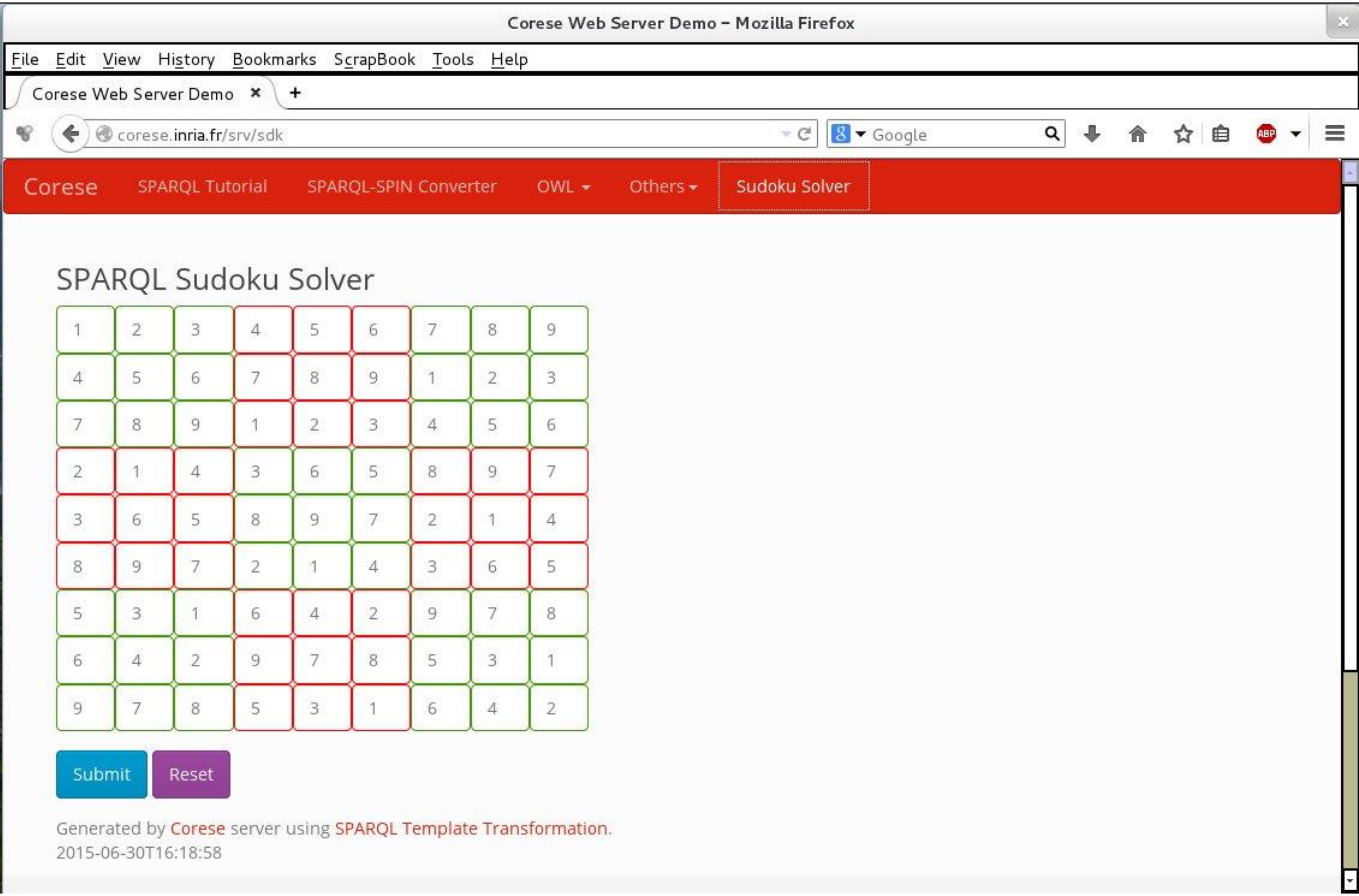
SPARQL Tutorial

Select a query

Count

Compter le nombre de solutions avec un opérateur d'aggrégation.
(http://www.w3.org/TR/sparql11-query/#aggregates)

```
prefix h: <http://www.inria.fr/2015/humans#>
select (count(*) as ?c) where {
  ?x ?p ?y
}
```



SPARQL Sudoku Solver

1	2	3	4	5	6	7	8	9
4	5	6	7	8	9	1	2	3
7	8	9	1	2	3	4	5	6
2	1	4	3	6	5	8	9	7
3	6	5	8	9	7	2	1	4
8	9	7	2	1	4	3	6	5
5	3	1	6	4	2	9	7	8
6	4	2	9	7	8	5	3	1
9	7	8	5	3	1	6	4	2

Generated by Corese server using SPARQL Template Transformation.
2015-06-30T16:18:58

Conclusion

- STTL Transformation Language for RDF
- Based on SPARQL
- XSLT like

Exercise

- https://eswc2018-sparql-ext.github.io/tutorial/
- Download corese-server-4.0.2.jar
- Download eswc.tar.gz
- Extract the archive -> eswc
- Move corese-server-4.0.2.jar in directory that contains eswc
- java "-Dfile.encoding=UTF-8" -jar corese-server-4.0.2.jar -lh -debug -pp eswc/profile.ttl
- URL: http://localhost:8080
- Select Demo/Demo (top right)

Exercise

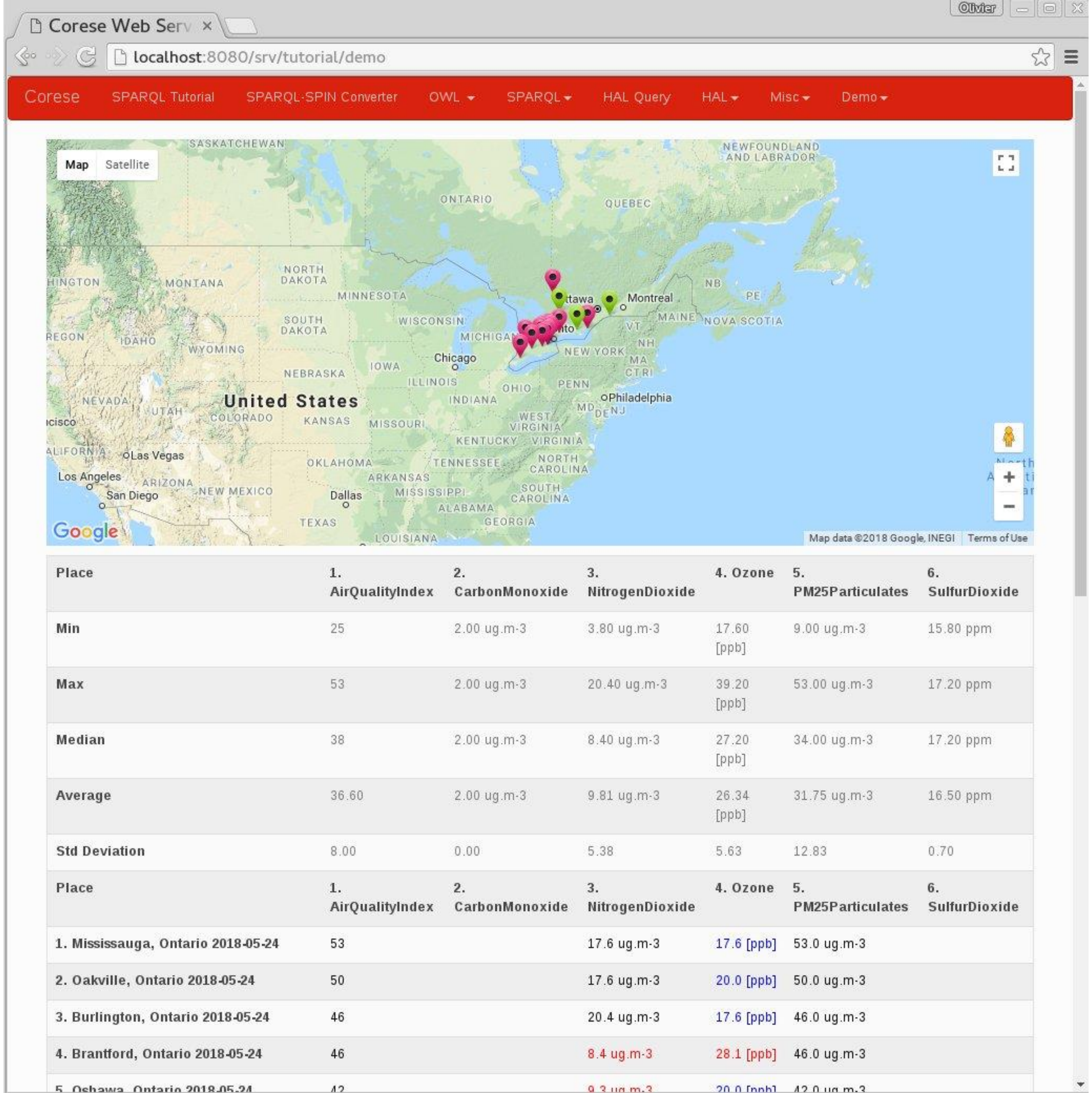
eswc archive content:

- profile.ttl specifies the demo service
- sttl contains the STTL transformation
- sttl/format contains HTML formats
- sttl/query contains SPARQL queries

Exercise

- Load RDF data produced by SPARQL Generate
- Generate HTML format using STTL
- Display sensor locations on a map
- Display sensor values in a table
- Compute aggregates: min, max, avg, etc.

```
<https://ci.mines-stetienne.fr/aqi/data/point?loc=44.150528,-77.3955>
    a   sosa:FeatureOfInterest , geo:Point ;
    rdfs:label    "Belleville, Ontario" ;
    rdfs:seeAlso  <http://aqicn.org/city/canada/ontario/belleville/> ;
    geo:lat       44.150528 ;
    geo:long       -77.3955 .

<https://ci.mines-stetienne.fr/aqi/station/1/observations/1527156000#no2>
    a   sosa:Observation ;
    sosa:hasFeatureOfInterest
<https://ci.mines-stetienne.fr/aqi/data/point?loc=44.150528,-77.3955> ;
    sosa:hasSimpleResult      "3.8 ug.m-3"^^cdt:ucum ;
    sosa:observedProperty
        <https://ci.mines-stetienne.fr/aqi/data/point?loc=44.150528,-77.3955#no2>;
    sosa:resultTime          "2018-05-24T10:00:00-05:00"^^xsd:dateTime .

<https://ci.mines-stetienne.fr/aqi/data/point?loc=44.150528,-77.3955#no2>
    a   aqio:NitrogenDioxideProperty ;
    ssn:isPropertyOf
        <https://ci.mines-stetienne.fr/aqi/data/point?loc=44.150528,-77.3955> .
```



Place	1. AirQualityIndex	2. CarbonMonoxide	3. NitrogenDioxide	4. Ozone	5. PM25Particulates	6. SulfurDioxide
Min	25	2.00 ug.m-3	3.80 ug.m-3	17.60 [ppb]	9.00 ug.m-3	15.80 ppm
Max	53	2.00 ug.m-3	20.40 ug.m-3	39.20 [ppb]	53.00 ug.m-3	17.20 ppm
Median	38	2.00 ug.m-3	8.40 ug.m-3	27.20 [ppb]	34.00 ug.m-3	17.20 ppm
Average	36.60	2.00 ug.m-3	9.81 ug.m-3	26.34 [ppb]	31.75 ug.m-3	16.50 ppm
Std Deviation	8.00	0.00	5.38	5.63	12.83	0.70

Place	1. AirQualityIndex	2. CarbonMonoxide	3. NitrogenDioxide	4. Ozone	5. PM25Particulates	6. SulfurDioxide
1. Mississauga, Ontario 2018-05-24	53		17.6 ug.m-3	17.6 [ppb]	53.0 ug.m-3	
2. Oakville, Ontario 2018-05-24	50		17.6 ug.m-3	20.0 [ppb]	50.0 ug.m-3	
3. Burlington, Ontario 2018-05-24	46		20.4 ug.m-3	17.6 [ppb]	46.0 ug.m-3	
4. Brantford, Ontario 2018-05-24	46		8.4 ug.m-3	28.1 [ppb]	46.0 ug.m-3	
5. Oshawa, Ontario 2018-05-24	42		9.3 ug.m-3	20.0 [ppb]	42.0 ug.m-3	