



Co-financed by the Connecting Europe
Facility of the European Union

Introduction to CRAFTS: Configurable RESTful APIs For Triple Stores



Guillermo Vega Gorgojo

guiveg@tel.uva.es

10-02-2021



Linked Open Data

■ Linked Open Data (LOD) in a nutshell

- Set of best practices for publishing and interlinking structured data on the Web
- Based on the architecture of the Web
 - URIs (**IRIs**) for naming things
 - **HTTP** as the application protocol

■ Additional pieces

- **RDF**, an abstract syntax for representing data
- Ontologies (typically in **OWL**) for modeling data
- LOD published in triple stores (or endpoints) and queryable through **SPARQL**

Quick example

■ A basic ontology (one class)

```
http://crossforest.eu/ifn/ontology/Tree rdf:type owl:Class .
```

■ Some data (two entities)

```
http://crossforest.eu/ifn/data/tree/06-0035-A-1-5 rdf:type  
http://crossforest.eu/ifn/ontology/Tree .
```

```
http://crossforest.eu/ifn/data/tree/06-0035-A-1-6 rdf:type  
http://crossforest.eu/ifn/ontology/Tree.
```

■ A triple store

```
https://forestexplorer.gsic.uva.es/sparql
```

■ A SPARQL query

```
SELECT ?tree WHERE {  
  ?tree rdf:type <http://crossforest.eu/ifn/ontology/Tree> .  
} LIMIT 2
```

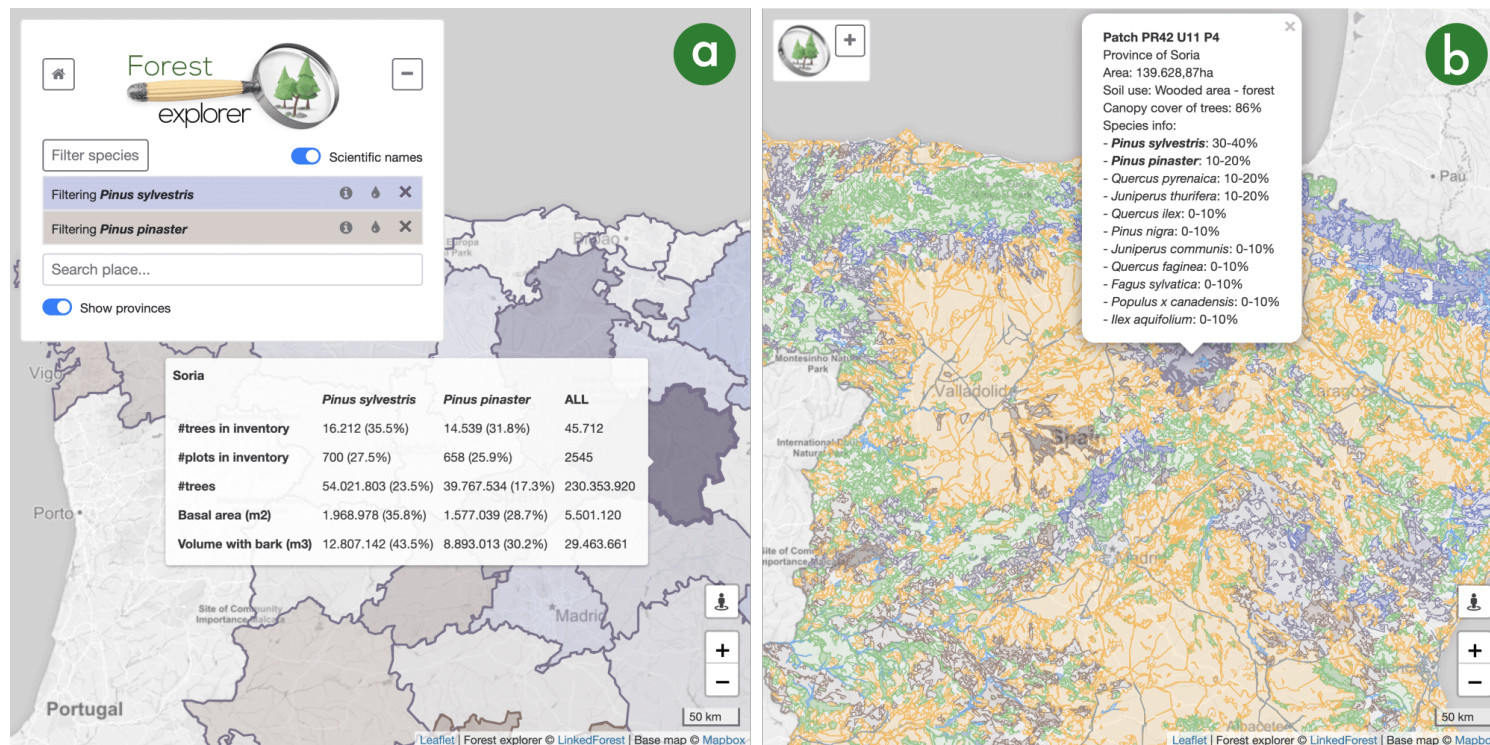
A web application that consumes LOD

■ Forest Explorer

– <https://forestexplorer.gsic.uva.es/>



- G. Vega-Gorgojo, J.M. Giménez-García, C. Ordóñez, F. Bravo. Pioneering Easy-to-Use Forestry Data with Forest Explorer. Semantic Web. 2021



LOD is great but...

- Publishing LOD is difficult

- (But not the topic of this talk)

- Consuming LOD is difficult

- Knowledge of RDF/OWL/SPARQL is not widespread
- Typically unknown to potential LOD consumers
 - Web developers
 - Domain experts
 - Lay users
- Would it be possible to access LOD through a regular RESTful API?

What is a RESTful API?

- Representation State Transfer (REST)
 - The architecture style of the Web
 - The *de-facto* standard for building web applications
- A RESTful API
 - Exposes resources identified with **URIs**
 - E.g. <http://forestexplorer.gsic.uva.es/crafts/apis>
 - Support a subset of standard **HTTP** operations
 - GET, POST, PUT, DELETE...
 - Similar to CRUD operations (create, read, update, delete)
 - Exchange resource representations
 - Typically in **JSON** or XML

CRAFTS: Configurable RESTful APIs For Triple Stores

■ Key idea

- Simplify access to LOD by providing configurable RESTful APIs

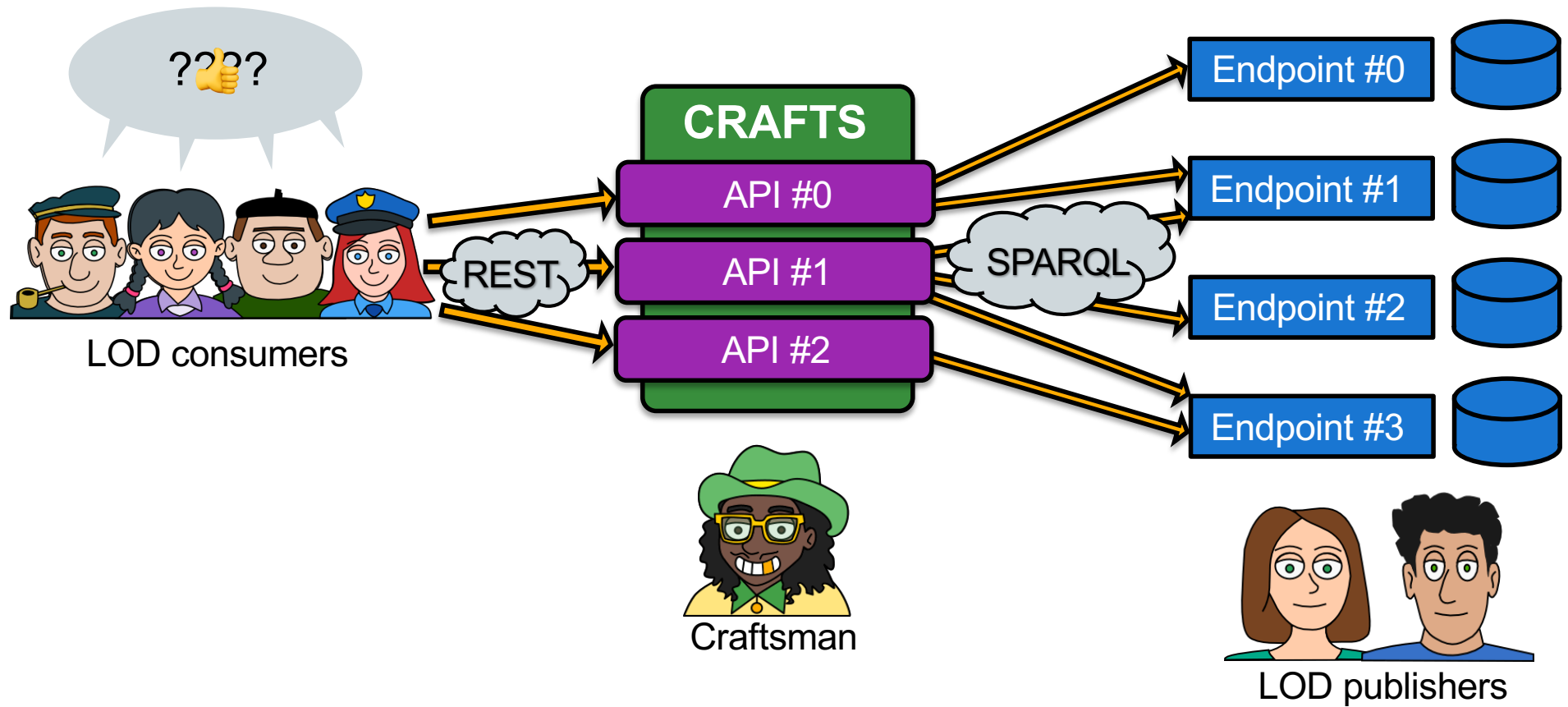
■ A CRAFTS API

- Translates API requests into SPARQL queries
- A configuration file defines the translation

■ Two types of users

- Craftsmen
 - In charge of authoring configuration files
- LOD consumers
 - Transparent access to data through regular web requests
 - No need to know RDF/OWL/SPARQL

Overview of CRAFTS



The API of CRAFTS

■ Functionalities supported

- User management
 - API management
 - Resource retrieval
 - Query templates
 - Resource update
 - Dump generation
- } *for another talk...*

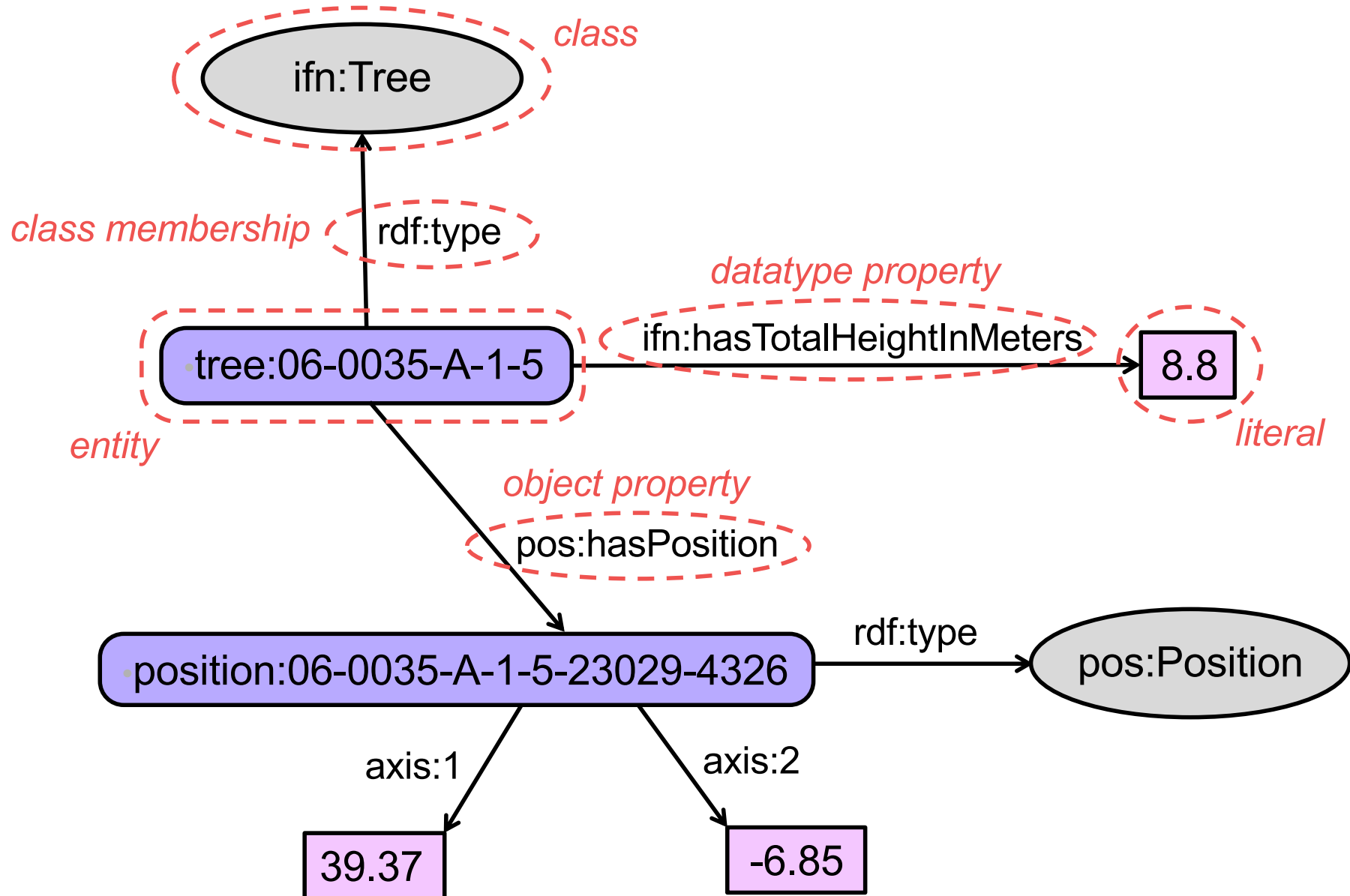
■ Documented with OpenApi

- Resource paths, operations, parameters, data schemas
- Not only useful for documentation, but for **validation**
- Available at
<https://forestexplorer.gsic.uva.es/crafts/docs/>

DEMO TIME!

<https://forestexplorer.gsic.uva.es/crafts/>

Excerpt of RDF data



Authentication and user management

- All CRAFTS operations (except user creation 😊) require authentication
- Any registered user can
 - View other APIs
 - Carry out read operations with other APIs
 - Create new APIs and update their own APIs
 - Carry out read and write operations with their own APIs
 - Share read and write tokens of their own APIs
 - Enables non-registered users to employ CRAFTS APIs
 - Bearer authentication
 - Regenerate read and write tokens of their own APIs

Summary (1/2)

- CRAFTS aims to simplify access to LOD
 - Not tied to a specific domain, ontology, dataset...
- Accessing a configured CRAFTS API is easy
 - As simple as using a well-documented RESTful API
 - No need to know RDF/OWL/SPARQL
- Some cool features of CRAFTS
 - Effective federation of triple stores
 - Data caching reduces the workload of triple stores

Summary (2/2)

- **Configuring a CRAFTS API is the critical step**
 - CRAFTS includes a very picky built-in validation process to facilitate the creation of valid configurations
 - The configuration step only needs to be carried out once
 - A good configuration enables meaningful data access to end users
 - Filtering out unnecessary classes and properties of the sources
 - Exploiting embedding to include an RDF subgraph into a single JSON object
 - Note that the goal is not to expose RDF triples as JSON (JSON-LD is perfect for this purpose)

Additional features of CRAFTS

■ Not only for reading data

- CRAFTS can be used to **update triple stores**
- PUT, PATCH and DELETE operations to create/modify/delete exposed resources
 - CRAFTS transparently updates the triple stores through SPARQL Update
- Requires SPARQL Update credentials in the configuration

■ Generating dumps in batch mode

- Daisy-chaining of requests

Recommended resources

- W3C specifications

- RDF, OWL, SPARQL

- RESTful APIs

- URIs and IRIs: RFCs 3986, 3987, 8820
- HTTP 1.1: RFCs 2616, 7230-7235
- JSON: RFC 8259
- HTTP PATCH: RFCs 5789, 6902

- OpenApi specification (V3.0.3)

- <https://swagger.io/specification/>

- Other competing Web APIs for Linked Open Data

- OBA, RAMOSE, Basil, GRLC, r4r



Co-financed by the Connecting Europe
Facility of the European Union

Many thanks

Give it a try!

<https://forestexplorer.gsic.uva.es/crafts/>