Sparnatural deployment checklist

*This note lists the prerequisites and steps in order to achieve a successful deployment of Sparnatural in an industrial context.*

*MoSCoW notation (MUST, SHOULD, CAN) are used to express the checklist recommendations.*

# Data (Knowledge Graph) prerequisites

1. The Knowledge Graph model/structure/ontology MUST be specified and SHOULD documented.
   * Why ? because the Sparnatural configuration will be based on this data structure
   * Potential solution : An efficient way to document the structure of a Knowledge Graph is to use SHACL.
2. Knowledge Graph MUST be populated with data. All the data conversion processes must have been conducted or be in place.
   * Why ? because Sparnatural will query this knowledge graph.
3. The entities that will be presented to the user in Sparnatural SHOULD have a label property that will be used as a display label (typically rdfs:label, skos:prefLabel, foaf:name, etc.).
   * Why ? So that every entity involved can be presented with a label and the label does not have to be computed at query time.
   * Potential solution : this may require the concatenation of multiple other properties (e.g. “{code} - {human-readable label}”). Typically this may involve separating the “normal label” of an entity (typically rdfs:label or another name) from its “display label” (typically skos:prefLabel).

# Infrastructure prerequisites

1. The data MUST be loaded in a SPARQL-compatible triplestore (typically a GraphDB repository). The triplestore must expose an HTTP SPARQL endpoint that can be queried from the outside.
   * Why ? Because Sparnatural will send SPARQL queries to the repository.
2. The triplestore MUST be CORS-enabled, or a CORS-enabled proxy MUST be used, or Sparnatural must be deployed on the same domain name as the SPARQL endpoint.
   * Why ? Because it is the HTML page that includes Sparnatural that will send the SPARQL query to the triplestore, and, if not deployed on the same domain name, the query will be subject to the Same-Origin-Policy security constraint.
   * Potential solution: If enabling CORS directly on the triplestore is not possible, a solution is to use a SPARQL endpoint proxy.
   * Reference documentation:<http://docs.sparnatural.eu/SPARQL-proxy.html>
3. The triplestore CAN take advantage of full-text indexes that can be leveraged by Sparnatural for autocompletion fields (e.g. Virtuoso bif:contains, GraphDB Lucene connector). The indexes should be configured according to how the autocomplete fields in Sparnatural will be configured.
   * Why ? 1/ so that autocomplete search fields are faster 2/ so that full-text criteria on the complete query are interpreted faster
   * Reference documentation: <http://docs.sparnatural.eu/Integration-with-GraphDB-Lucene-Connector.html>
4. The triplestore MUST be sized appropriately (in terms of memory, scaling on multiple servers, etc.).
   * Why ? in order to answer SPARQL queries efficiently.
   * Potential solution: This could involve separating 2 synchronized servers : “ingestion server” vs. “dissemination server” (which would be read-only).
5. On the local machine where the Sparnatural configuration will be tested, browser security MUST be adjusted to enable the loading of local files.
   * Why ? because when working locally, the browser should be instructed that it is safe to load the Sparnatural configuration from a file in the same folder at the HTML page.
   * Reference documentation: <http://docs.sparnatural.eu/OWL-based-configuration.html#enable-cors-for-local-files-in-your-browser>
6. Fontawesome icon license CAN be purchased.
   * Why ? Icons are an important visual part of what the user sees in Sparnatural. Fontawesome offers a free set of icons, but spending a few dozens dollars in purchasing the complete icon set can bring immediate visual value to the query builder.
7. It MUST be possible to deploy the HTML page integrating Sparnatural on an HTTP server
   * Why ? so that the page with Sparnatural included is accessible for clients

# Competency prerequisites

1. The person configuring Sparnatural MUST understand the user needs in terms of querying.
   1. Why ? 1/ to ensure that the Sparnatural configuration can answer the competency questions 2/ to propose predefined sample queries to the user
   2. Potential solution: This can be typically expressed by the form of “competency questions” on the data (“which questions would you like to ask to your knowledge graph ?”). Note that this user need could have happened at very early stages of the projects (before the actual modeling work has happened).
2. The person responsible for configuring Sparnatural MUST master the model/structure/ontology of the knowledge graph. As such this should be properly documented.
   1. Why ? in order to be able to properly map the user needs (in terms of query) to the underlying data structure
   2. Potential solution : An efficient way to document the structure of a Knowledge Graph is to use SHACL. Focus should be the domain/ranges of the properties, cardinalities (which properties are mandatory/optional ? repeatable ?), multilingualism.
3. The person configuring Sparnatural MUST have read the Sparnatural configuration documentation and MUST be proficient in the SPARQL query language.
   1. Why ? in order to configure Sparnatural efficiently, and in order to write the datasources SPARQL queries.
4. The person configuring Sparnatural MUST have basic knowledge in HTML structure, JSON and Javascript.
   1. Why ? in order to configure Sparnatural efficiently.