
Application Web Sémantique pour les Requêtes SPARQL

Ce guide présente comment nous avons procédé pour configurer et exécuter une application Web sémantique permettant de réaliser des requêtes SPARQL sur des endpoints locaux ou externes, comme Wikidata.

Aperçu du Projet

Cette application :

1. Permet aux utilisateurs d'entrer une requête SPARQL.
2. Affiche les résultats sous deux formes :
 - **Liste formatée** : Pour des résultats descriptifs standards.
 - **Visualisation en diagramme camembert** : Pour des données numériques, comme des comptages.

Pré-requis

1. Configuration de GraphDB :

- Installez GraphDB Free Edition et configurez-le pour votre projet.
- Ajoutez vos données dans un dépôt local (par exemple dans notre cas, **projet-Eglise-UNESCO** ou un dépôt personnalisé).

2. Environnement React :

- Installation Node.js et npm .
- Configuration d'une application React.

3. Endpoints SPARQL :

- Configuration de notre endpoint SPARQL local dans GraphDB (par exemple, **<http://localhost:7200/repositories/repo-teachers>**).
- Activez CORS dans les paramètres de GraphDB :
 - Ajoutez la propriété **`graphdb.workbench.cors.enable`** et définissez-la à **`True`**.

Données et Exemples de Requêtes

Données RDF

- Le fichier **`RDF_Eglises.ttl`** contient des données structurées sur les églises UNESCO et leurs styles architecturaux.
- Chargez ce fichier dans votre dépôt GraphDB.

Données CSV

- Utilisez [EgliseUnesco-2.csv](#) pour intégrer des données tabulaires à votre dataset RDF pour une analyse approfondie.

URL EndPoint de grapheDB

- http://desktop-1igae9a:7200/repositories/Projet_Eglise_Unesco (Repository Nathan)
- http://PC_AORUS_Axel:7200/repositories/Projet-DB-eglise (Repository Axel)

Exemples de Requêtes SPARQL

1. Requêter les Noms et Styles des Églises

```
PREFIX iut: <https://cours.iut-orsay.fr/npbd/projet/ratovo/>
SELECT ?nomLocal ?styleArchi ?localDescription
WHERE {
    ?localChurch a iut:Eglise ;
                iut:styleArchitectural ?styleArchi;
                iut:description ?localDescription;
                iut:nomEglise ?nomLocal.
}
```

2. Compter les Styles Architecturaux

```
PREFIX iut: <https://cours.iut-orsay.fr/npbd/projet/ratovo/>
SELECT ?architecturalStyle (COUNT(?church) AS ?styleCount)
WHERE {
    ?church a iut:Eglise ;
            iut:styleArchitectural ?architecturalStyle .
}
GROUP BY ?architecturalStyle
ORDER BY DESC(?styleCount)
```

3. Fusionner les Données Locales et Wikidata

```
PREFIX iut: <https://cours.iut-orsay.fr/npbd/projet/ratovo/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
INSERT {
    ?localChurch owl:sameAs ?wikidataChurch ;
                iut:styleArchitectural ?styleWiki .
}
WHERE {
    ?localChurch a iut:Eglise ;
                iut:nomEglise ?localName ;
                iut:description ?localDescription .
    SERVICE <https://query.wikidata.org/sparql> {
        ?wikidataChurch wdt:P31 wd:Q16970 ;
    }
}
```

```

        rdfs:label ?wikidataName ;
        wdt:P149 ?styleWiki .
    FILTER(LANG(?wikidataName) = "en")
}
FILTER(STR(?localName) = STR(?wikidataName))
}

```

Application React

Fonctionnalités Clés

1. Entrée Utilisateur pour les Requêtes :

- Saisir le nom de l'application.
- Spécifier l'URL de l'endpoint SPARQL.
- Écrire et exécuter des requêtes SPARQL.

2. Affichage Dynamique :

- Affiche les résultats sous forme de liste formatée ou de diagramme camembert, selon la structure des données.

Mise en œuvre

QueryInterface.js

Ce fichier gère l'interface utilisateur et la logique d'exécution des requêtes SPARQL, ainsi que le rendu des résultats.

Extrait de Code Pertinent

La logique détermine comment afficher les résultats :

```

{results && chartData ? (
  <div>
    <h2>Diagramme des Styles Architecturaux</h2>
    <Pie data={chartData} />
  </div>
) : results && results.length > 0 ? (
  <div>
    <h2>Résultats de la Requête</h2>
    {formatResults(results)}
  </div>
) : (
  <p>Aucun Résultat Trouvé.</p>
)}

```

Étapes d'Installation

1. Installez les dépendances nécessaires :

```
npm install chart.js react-chartjs-2
```

2. Lancez l'application React :

```
#-- Se positionner à la racine du projet  
npm start
```

Résultats Visuels

Exemple de Diagramme Camembert

Requêtes Simplifiées

Requête 1: Styles architecturaux

Cette requête retourne les styles architecturaux des églises et leur nombre.

Exécuter

Interface pour requête SPARQL

Le nom de l'application :

test1

SPARQL Endpoint :

http://localhost:7200/repositories/Projet-DB-eglise

Requête SPARQL :

```
PREFIX iut: <https://cours.iut-orsay.fr/npbd/projet/ratovo/>
SELECT ?architecturalStyle (COUNT(?church) AS ?styleCount)
WHERE {
  ?church a iut:Eglise ;
    iut:styleArchitectural ?architecturalStyle .
}
GROUP BY ?architecturalStyle
ORDER BY DESC(?styleCount)
```

Exécuter la requête

Style Architectural des églises

http://www.wikidata.org/entity/Q176483

http://www.wikidata.org/entity/Q46261

http://www.wikidata.org/entity/Q849829

http://www.wikidata.org/entity/Q236122

http://www.wikidata.org/entity/Q2860299

http://www.wikidata.org/entity/Q186363

http://www.wikidata.org/entity/Q54111

http://www.wikidata.org/entity/Q9159129

http://www.wikidata.org/entity/Q122969

http://www.wikidata.org/entity/Q669863

http://www.wikidata.org/entity/Q46625

http://www.wikidata.org/entity/Q37853

http://www.wikidata.org/entity/Q176292

http://www.wikidata.org/entity/Q2864695

http://www.wikidata.org/entity/Q1786396

http://www.wikidata.org/entity/Q47591

http://www.wikidata.org/entity/Q744373

http://www.wikidata.org/entity/Q751878

Résultats Formatés

5 / 7

Ce projet a pour but d'interroger une base SPARQL de manière locale avec notre site. Mais aussi permettre d'interroger Wikidata à travers lui.

Pour cela, vous n'avez que trois choses à faire :

1. Choisissez un nom pour votre application web ayant un rapport avec votre requête.

2. Entrez le lien de votre Endpoint SPARQL où figure votre base de données.

3. Tapez votre requête SPARQL pour interroger votre base ou Wikidata et obtenez un résultat.

Requêtes Simplifiées

Requête 1: Styles architecturaux

Cette requête retourne les styles architecturaux des églises et leur nombre.

Exécuter

Interface pour requête SPARQL

Le nom de l'application :

Test2

SPARQL Endpoint :

http://localhost:7200/repositories/Projet-DB-eglise

Requête SPARQL :

```
PREFIX iut: <https://cours.iut-orsay.fr/npbd/projet/ratovo/>
SELECT ?localChurch ?styleWiki ?localDescription
WHERE {
  # Données locales
  ?localChurch a iut:Eglise ;
               iut:styleArchitectural ?styleWiki ;
               iut:description ?localDescription .
}
```

Executer la requête

Résultat de la requête

Local Church: https://cours.iut-orsay.fr/npbd/projet/ratovo/7

Local Style:

Description: cathedral in Croatia

Local Church: https://cours.iut-orsay.fr/npbd/projet/ratovo/11

Local Style:

Description: church building in Petäjävesi, Finland

Local Church: https://cours.iut-orsay.fr/npbd/projet/ratovo/16

Local Style:

Description: Roman-Catholic cathedral in Aachen, Germany

Local Church: https://cours.iut-orsay.fr/npbd/projet/ratovo/16

Local Style:

Utilisation

6 / 7

1. Lancez le serveur GraphDB et assurez-vous que votre dépôt fonctionne.
2. Exécutez l'application React.
3. Entrez l'URL de l'endpoint SPARQL et la requête dans l'application.
4. Visualisez les résultats dans le format approprié.

Notes Supplémentaires

- Si votre requête inclut un comptage ou des données numériques, un diagramme camembert sera affiché.
 - Pour des résultats textuels, une liste formatée sera affichée.
-