

 eBook Gratuit

APPRENEZ

sparql

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#sparql

Table des matières

À propos	1
Chapitre 1: Démarrer avec sparql	2
Remarques.....	2
Versions.....	2
Exemples.....	2
Premiers pas avec un point de terminaison SPARQL.....	2
Chapitre 2: Travailler avec des balises de langue	3
Exemples.....	3
Obtenir une balise de langue à partir d'un littéral.....	3
Comparaison des balises de langue.....	3
Chapitre 3: Travailler avec des graphiques	4
Introduction.....	4
Exemples.....	4
Stockage de données dans un graphique nommé.....	4
Crédits	5

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [sparql](#)

It is an unofficial and free sparql ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official sparql.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec sparql

Remarques

Cette section fournit une vue d'ensemble de ce qu'est sparql et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets de sparql, et les relier aux sujets connexes. La documentation de sparql étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Versions

Version	Date de sortie
1.0	2008-01-15
1.1	2013-03-21

Exemples

Premiers pas avec un point de terminaison SPARQL

Un moteur de requête est requis pour exécuter des requêtes SPARQL sur un ensemble de données. Les moteurs de requêtes peuvent être intégrés dans des applications ou accessibles à distance. L'accès à distance peut se faire via une API spécifique au fournisseur ou via le [protocole SPARQL](#) si une implémentation le prend en charge. Cette documentation ne couvrira pas la manière de soumettre des requêtes via des API de fournisseur spécifiques.

Les implémentations SPARQL Endpoint fournissent généralement une interface Web conviviale pour la soumission de requêtes et l'affichage de leurs résultats. Les points de terminaison SPARQL publics (tels que [DBPedia](#)) peuvent servir de jeux de données utiles pour les exemples non mutants.

Si vous souhaitez configurer un point de terminaison SPARQL privé pour l'expérimentation, [Apache Fuseki](#) fournit une option gratuite et indépendante de la plate-forme.

Lire [Démarrer avec sparql en ligne](#): <https://riptutorial.com/fr/sparql/topic/1303/demarrer-avec-sparql>

Chapitre 2: Travailler avec des balises de langue

Exemples

Obtenir une balise de langue à partir d'un littéral

Les littéraux dans RDF peuvent être des chaînes marquées par une langue. Ces littéraux ont une partie texte ainsi qu'une étiquette de langue. Par exemple, le littéral **"color" @en** a la partie texte **"color"** et l'étiquette de langue **"en"**. Dans une requête SPARQL, utilisez la fonction **lang** pour obtenir le langage d'un littéral avec une balise de langue. Si un littéral n'a pas d'étiquette de langue, alors **lang** renvoie la chaîne vide **""**.

Comparaison des balises de langue

La fonction SPARQL **langMatches** peut être utilisée pour comparer les balises de langue des littéraux RDF dans les requêtes SPARQL. L'opérateur d'égalité simple, **=**, peut être utilisé pour comparer des correspondances de chaînes exactes, mais ne tiendra pas correctement compte des variantes régionales. Par exemple, les quatre valeurs possibles de **? Str** dans:

```
values ?str { "color"@en-US "color"@en "colour"@en "colour"@en-GB }
```

sont toutes les chaînes de langue anglaise, mais deux d'entre elles ont des spécifications régionales. Cela signifie que

```
select ?str {
  values ?str { "color"@en-US "color"@en "colour"@en "colour"@en-GB }
  filter (lang(?str) = 'en')
}
```

ne renverra que deux résultats, car seules deux des valeurs de **? str** ont **"en"** comme étiquette de langue. cependant,

```
select ?str {
  values ?str { "color"@en-US "color"@en "colour"@en "colour"@en-GB }
  filter langMatches(lang(?str), 'en')
}
```

renverra les quatre valeurs.

Lire [Travailler avec des balises de langue en ligne](https://riptutorial.com/fr/sparql/topic/5210/travailler-avec-des-balises-de-langue):

<https://riptutorial.com/fr/sparql/topic/5210/travailler-avec-des-balises-de-langue>

Chapitre 3: Travailler avec des graphiques

Introduction

Qu'est-ce qu'un graphique nommé? Un identificateur (nom) de document de base de données interne utilisé par un SGBDR compatible SQL pour partitionner le stockage des relations représentées sous forme de graphiques de phrase / instruction RDF. Pourquoi les graphiques nommés sont-ils importants? Un graphique nommé est comme une page dans un livre (la base de données) qui contient une collection de paragraphes (collections de phrases). Ainsi, il fournit un mécanisme puissant pour déterminer la portée des requêtes, ce qui élimine la nécessité d'étendre toutes les requêtes de base de données à la base de données entière.

Exemples

Stockage de données dans un graphique nommé

```
PREFIX      rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX      owl: <http://www.w3.org/2002/07/owl#>
PREFIX      schema: <http://schema.org/>
PREFIX      foaf: <http://xmlns.com/foaf/0.1/>

WITH <urn:this:database:doc>
INSERT { <#relatedTo> a                rdf:Property, owl:ObjectProperty ;
                    rdfs:label         "relatedTo" ;
                    rdfs:domain        rdfs:Resource ;
                    rdfs:range         xsd:anyURI .
    foaf:knows rdfs:subPropertyOf <#relatedTo> ;
    owl:equivalentProperty schema:knows .
    <#this> a foaf:Person ;
           schema:name "John Doe" .
    <#that> a foaf:Person ;
           schema:name "Jane Doe" .
}
```

Lire Travailler avec des graphiques en ligne: <https://riptutorial.com/fr/sparql/topic/8092/travailler-avec-des-graphiques>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec sparql	Community , Joshua Taylor , Rob Hall , unor
2	Travailler avec des balises de langue	Joshua Taylor
3	Travailler avec des graphiques	i-blis , Kingsley Uyi Idehen