



EBook Gratis

APRENDIZAJE marklogic

Free unaffiliated eBook created from
Stack Overflow contributors.

#marklogic

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con marklogic.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	2
Instalación o configuración.....	2
Capítulo 2: Buscar en MarkLogic.....	4
Observaciones.....	4
Examples.....	4
Recuperando todos los documentos con la palabra "marklogic".....	4
Recuperando todos los documentos con la palabra "marklogic", documento de la colección "pr.....	4
Obtención de todos los documentos con un valor particular de un elemento.....	4
Comprobación de presencia de elementos y atributos en documentos.....	4
Capítulo 3: Características de Marklogic 8.....	6
Observaciones.....	6
Examples.....	6
Nuevas características en MarkLogic 8.....	6
Capítulo 4: Ejemplos muy simples de CRUD para documentos XML.....	7
Examples.....	7
Crear un documento simple.....	7
Lea / acceda a nuestro documento de muestra.....	7
Actualizar el documento simple.....	8
Bonus: Ejemplo de búsqueda simple (Otra forma de leer).....	8
Eliminar - Por último, pero no menos importante.....	9
Capítulo 5: Trabajando con fn: count, xdmp: estimado y cts: frecuencia.....	11
Observaciones.....	11
Examples.....	11
Usando fn: count () para obtener el número de documentos coincidentes.....	11
Usando xdmp: estimar () para obtener el número de documentos coincidentes.....	11
Contando documentos cuando se definen fragmentos.....	11

Acerca de

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

It is an unofficial and free marklogic 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 marklogic.

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

Capítulo 1: Empezando con marklogic

Observaciones

Esta sección proporciona una descripción general de qué es marklogic y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de marklogic, y vincular a los temas relacionados. Dado que la Documentación para marklogic es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Versiones

Versión	URL para descargar	URL a las notas de la versión	Fecha de lanzamiento
8.0-5.5	Descargar	Notas de Rel	2016-07-15
7.0-6.4	Descargar	Notas de Rel	2015-07-15
9.0-1.1	Descargar	Notas de Rel	2017-05-10

Examples

Instalación o configuración

Las instrucciones detalladas sobre cómo configurar o instalar MarkLogic se pueden encontrar en la [Guía de instalación](#) . La oferta completa de documentación está disponible a través de docs.marklogic.com

El proceso de configuración general implica

1. Instalando el binario / rpm
2. Comenzando los servicios
3. Configurando los primeros y subsiguientes hosts

Un ejemplo de "Hello World" no es realmente necesario para verificar la configuración. El proceso de configuración incluye la verificación del éxito, ya que una vez instalado, el usuario navegará un navegador a la interfaz administrativa. Si algo se ha perdido, los servicios no se han iniciado o si algo salió mal con el proceso de instalación, la interfaz no se activará. Los procedimientos completos paso a paso se incluyen en el documento al que se hace referencia.

Una vez instalada y la configuración básica está en su lugar, una gran herramienta para comenzar a interactuar con el sistema se llama "Query Console" y está disponible a través del navegador, generalmente en el puerto 8000. La URL típica para una instancia instalada en la máquina local de un desarrollador es <http://localhost:8000/qconsole>

Esta poderosa herramienta tiene un documento separado disponible aquí .

Lea Empezando con marklogic en línea:

<https://riptutorial.com/es/marklogic/topic/5308/empezando-con-marklogic>

Capítulo 2: Buscar en MarkLogic

Observaciones

Esta sección proporciona una descripción general de la búsqueda en MarkLogic. La intención es cubrir `cts: buscar`, `buscar: buscar` y `qbe` con casos de uso y ejemplos

Examples

Recuperando todos los documentos con la palabra "marklogic"

```
cts:search(  
  fn:doc(),  
  cts:word-query("marklogic"))
```

Recuperando todos los documentos con la palabra "marklogic", documento de la colección "primera colección"

Esto se puede hacer de las siguientes dos maneras:

```
cts:search(  
  fn:collection("first-collection"),  
  cts:word-query("marklogic"))
```

En esto, el alcance se cambia de todos los documentos a los documentos de la colección "primera colección" solamente.

En el segundo enfoque, se hizo uso de `cts: collection-query` . Esto debería dar un mejor rendimiento que el primer enfoque.

```
cts:search(  
  fn:doc(),  
  cts:and-query((  
    cts:collection-query("first-collection"),  
    cts:word-query("marklogic"))))
```

Obtención de todos los documentos con un valor particular de un elemento.

Esta consulta devuelve todos los documentos con el elemento "empresa" y su valor como "marklogic"

```
cts:element-value-query(xs:QName('company'), 'marklogic')
```

Comprobación de presencia de elementos y atributos en documentos.

La siguiente consulta devuelve los documentos que tienen un elemento llamado "compañía":

```
cts:element-value-query(  
  xs:QName('company'), '*', ("wildcarded"))
```

La siguiente consulta devuelve los documentos que tienen un elemento llamado "compañía" con un atributo llamado "nombre":

```
cts:element-attribute-value-query(  
  xs:QName('company'), xs:QName('name'), '*', ("wildcarded"))
```

Lea **Buscar en MarkLogic en línea**: <https://riptutorial.com/es/marklogic/topic/5401/buscar-en-marklogic>

Capítulo 3: Características de Marklogic 8

Observaciones

Esta sección proporciona una descripción general de las novedades de MarkLogic 8 y sus otras versiones actualizadas. Caso de uso para cada característica también debe ser agregado.

Examples

Nuevas características en MarkLogic 8

- JavaScript del lado del servidor
- Nativo json
- Aplicación de muestra Samplestack
- Documentos temporales
- Gestión de descanso
- Mejoras API
- Más características de semántica, que incluyen SPARQL 1.1, Inferencia y ACTUALIZACIÓN DE SPARQL
- Node.js Client API
- Mejoras en la API de REST y Java Client
- Características mejoradas del servidor HTTP
- Mejoras de replicación flexibles
- Respaldo incremental
- Mejoras en los servicios de biblioteca de documentos (DLS)
- Mejoras de MLCP

Lea Características de Marklogic 8 en línea:

<https://riptutorial.com/es/marklogic/topic/5386/caracteristicas-de-marklogic-8>

Capítulo 4: Ejemplos muy simples de CRUD para documentos XML

Examples

Crear un documento simple

Este fragmento muy simple de XQuery se puede ejecutar en QueryConsole usando la base de datos integrada de "Documentos" como un sandbox. Cada fragmento del fragmento tiene un comentario para explicar qué significa la siguiente línea de código.

```
xquery version "1.0-ml";
(: Let's first insert a simple document to get started :)
(: You need a URI- the location where the document is found in the database :)
let $uri := "/stuff/mysimpledocument.xml"
(: Documents need content. This is a simple XML node :)
let $doc :=
  <my-document>
    <body>Very simple example</body>
  </my-document>
(: Permissions are a big topic. For now, we'll use the default permissions. :)
let $permissions := xdmp:default-permissions()
(: Document collections are optional. One or more can be specified :)
(: Adding a collection for further examples that will use it. :)
let $collections := "simple-example"
(: Now we're just going to insert this document in the database :)
let $insert := xdmp:document-insert($uri,$doc,$permissions,$collections)
return <message>Document saved to {$uri}</message>
```

Cuando esto se ejecuta, la consola vuelve.

```
<message>Document saved to /stuff/mysimpledocument.xml</message>
```

Ahora que el documento existe, podemos probar operaciones adicionales ...

Lea / acceda a nuestro documento de muestra

Si conocemos el URI deseado del documento que estamos buscando:

```
fn:doc("/stuff/mysimpledocument.xml")
```

Devuelve el documento completo de la base de datos, utilizando el URI para localizarlo.

Como esto es XQuery, podemos usar XPath para encontrar el documento cuando conocemos la estructura, pero no el URI:

```
/my-document
```

Devuelve el elemento "mi documento" y su contenido ...

Actualizar el documento simple

Ahora agregaremos algunos nodos XML adicionales al elemento "my-document" y actualizaremos el documento. El fragmento nuevamente contiene comentarios para explicar lo que está sucediendo.

```
xquery version "1.0-ml";
(: We are preserving the same URI as we used originally :)
let $uri := "/stuff/mysimpledocument.xml"
(: Need to get the existing contents so we can append to those :)
let $orig-content := fn:doc($uri)/my-document
(: Documents need content. This is a simple XML node :)
let $new-content :=
  <notes>
    <note>Anything can be changed</note>
    <note>New content is added in this example, but we could replace it all too</note>
  </notes>
(: Now to build the new xml. There's lots of ways to do this :)
(: line 17 inserts the original contents into this new node construct :)
(: line 18 inserts the $new-content after the original contents :)
let $new-doc-content :=
  <my-document>
    {$orig-content/node()}
    {$new-content}
  </my-document>
(: Leave permissions untouched... :)
let $permissions := xdmp:document-get-permissions($uri)
(: Leave collections untouched... :)
let $collections := xdmp:document-get-collections($uri)
(: Now we're just going to insert this document in the database :)
let $insert := xdmp:document-insert($uri,$new-doc-content,$permissions,$collections)
return <message>Document {$uri} updated</message>
```

Cuando se ejecuta, esto devuelve el mensaje.

```
<message>Document /stuff/mysimpledocument.xml updated</message>
```

Ahora ejecute los diferentes comandos de lectura anteriores para ver el contenido actualizado. Debe tener un aspecto como este:

```
<my-document>
  <body>Very simple example</body>
  <notes>
    <note>Anything can be changed</note>
    <note>New content is added in this example, but we could replace it all too</note>
  </notes>
</my-document>
```

Bonus: Ejemplo de búsqueda simple (Otra forma de leer)

MarkLogic es, ante todo, un motor de búsqueda, por lo que usaremos dos métodos diferentes para buscar este documento.

Utilizando search: search ()

Esto le da un vistazo al uso de `search: search ()` para desarrollar aplicaciones de búsqueda. Esta biblioteca proporciona resultados de búsqueda similares a los de Google y probablemente acelerará su desarrollo de herramientas de búsqueda simples. Más información y una inmersión más profunda [se pueden encontrar aquí](#) .

```
xquery version "1.0-ml";
import module namespace search = "http://marklogic.com/appservices/search"
  at "/MarkLogic/appservices/search/search.xqy";
(: What is search without a keyword? :)
let $term := "very simple"
return search:search($term)
```

El resultado parece un poco confuso, pero puede ver que devuelve un resultado, nuestro documento de ejemplo.

Utilizando cts: search ()

Las situaciones de búsqueda más avanzadas pueden requerir capacidades de búsqueda más granulares. Esto es solo para despertar su apetito por lo que está disponible en la búsqueda. Más información detallada [se encuentra aquí](#) .

```
xquery version "1.0-ml";
(: What is search without a keyword? :)
let $term := "very simple"
(: Complex queries can be made from individual cts queries. Here, we just have one simple query :)
let $query := cts:word-query($term,"case-insensitive")
(: Return the documents that match the query :)
return cts:search(fn:doc(),$query)
```

Este es un ejemplo increíblemente simple. Por cierto, si queremos recuperar el URI de los documentos coincidentes, en lugar de los documentos en sí, podemos cambiar la última línea de este fragmento a:

```
return
  for $result in cts:search(fn:doc(),$query)
  return fn:base-uri($result)
```

Eliminar - Por último, pero no menos importante

Para redondear ejemplos simples de operaciones CRUD, presentamos los siguientes ejemplos. Siempre tenga mucho cuidado en la eliminación de documentos.

```
(: When we know the URI, we can delete it very easily :)
let $uri := "/stuff/mysimpledocument.xml"
return xdmp:document-delete($uri)
```

o simplificado:

```
xdmp:document-delete("/stuff/mysimpledocument.xml")
```

Ciertamente, puede usar XPath para encontrar el documento, obtener el URI del mismo y luego eliminarlo con algo como esto, pero el peligro es que cualquier documento que sea devuelto por las expresiones XPath se eliminará. No siempre es bueno.

```
(: Use caution when using XPath to select target docs to delete :)  
for $doc in /my-document  
return xdmp:document-delete(fn:base-uri($doc))
```

¿Quieres borrar todos los documentos? Esto lo hará, pero tenga mucho cuidado de saber en qué base de datos se ejecutará su código.

```
for $doc in fn:doc()  
return xdmp:document-delete(fn:base-uri($doc))
```

Lea Ejemplos muy simples de CRUD para documentos XML en línea:

<https://riptutorial.com/es/marklogic/topic/5498/ejemplos-muy-simples-de-crud-para-documentos-xml>

Capítulo 5: Trabajando con fn: count, xdmp: estimado y cts: frecuencia

Observaciones

Esta sección proporciona una descripción general de [fn: count](#) , [xdmp: estimación](#) y [cts: frecuencia](#) junto con ejemplos y casos de uso

Examples

Usando fn: count () para obtener el número de documentos coincidentes

El documento XML que usaré a lo largo de los ejemplos es:

```
<a>
  <b>test-value</b>
  <d>fragment-d</d>
  <c-root>
    <d>fragment-d</d>
    <e>fragment-e</e>
  </c-root>
</a>
```

Las siguientes consultas devuelven el número de documentos con el `fragment-d` valor `fragment-d` para el elemento `d` -

- Usando un [cts: buscar](#)

```
fn:count(cts:search(fn:doc(), cts:element-value-query(xs:QName("d"), "fragment-d")))
```

- Utilizando XPath

```
fn:count(fn:doc()[//d="fragment-d"])
```

Usando xdmp: estimar () para obtener el número de documentos coincidentes

```
xdmp:estimate(cts:search(fn:doc(), cts:element-value-query(xs:QName("d"), "fragment-d")))
```

[xdmp: la estimación](#) no se puede usar en XPaths a diferencia de que [fn: count](#) se usa en el ejemplo anterior

[xdmp: el estimado](#) realmente da el número de fragmentos coincidentes

Contando documentos cuando se definen fragmentos.

El documento XML a considerar en este ejemplo -

```
<a>
  <b>test-value</b>
  <d>fragment-d</d>
  <c-root>
    <d>fragment-d</d>
    <e>fragment-e</e>
  </c-root>
</a>
```

Se declara una raíz de fragmento en `<c-root>`

Si este es el único documento en la base de datos, `xdmp: estimación` y `fn: cuenta` se comportarán de manera diferente -

```
xdmp:estimate(cts:search(fn:doc(), cts:element-value-query(xs:QName("d"), "fragment-d")))
```

El resultado de la consulta anterior será 2 (Número de fragmentos)

```
fn:count(cts:search(fn:doc(), cts:element-value-query(xs:QName("d"), "fragment-d")))
```

El resultado de la consulta anterior será 1 (Número de documentos)

En términos de rendimiento `xdmp: la estimación` es mucho mejor que `fn: count`, ya que aprovecha las ventajas de los índices y resuelve los resultados de búsqueda

Lea [Trabajando con fn: count, xdmp: estimado y cts: frecuencia en línea](https://riptutorial.com/es/marklogic/topic/6300/trabajando-con-fn--count--xdmp--estimado-y-cts--frecuencia):

<https://riptutorial.com/es/marklogic/topic/6300/trabajando-con-fn--count--xdmp--estimado-y-cts--frecuencia>

Creditos

S. No	Capítulos	Contributors
1	Empezando con marklogic	Community , Harry Bakken , Tamas
2	Buscar en MarkLogic	Ankit Bhardwaj
3	Características de Marklogic 8	Ankit Bhardwaj
4	Ejemplos muy simples de CRUD para documentos XML	Harry Bakken
5	Trabajando con fn: count, xdmp: estimado y cts: frecuencia	Ankit Bhardwaj