

## DOCUMENTO DE CONSULTAS A ICMS

### 1. Objeto

El presente documento tiene como objetivo servir de guía sobre cómo se puede consultar la información disponible en un servidor iCMS.

Estas consultas se basan en el formato de uso del *framework* Apache [Solr](#), pero en un ámbito más reducido.

### 2. Documentación de referencia

La documentación de referencia para la realización del presente documento incluye:

- Documentación: iCMS - Interfaz para la interoperabilidad de contenidos web  
[http://www.juntadeandalucia.es/repositorio/usuario/listado/fichacompleta.jsf?linkDummyForm:\\_idcl=\\_id153&idProyecto=736&](http://www.juntadeandalucia.es/repositorio/usuario/listado/fichacompleta.jsf?linkDummyForm:_idcl=_id153&idProyecto=736&)
- Repositorio de software libre de la Junta de Andalucía. Proyecto iCMS:  
<http://www.juntadeandalucia.es/repositorio/usuario/listado/fichacompleta.jsf?idProyecto=736>

### 3. Definiciones

A continuación se incluye un breve glosario con la definición de una serie de palabras-clave para su correcto uso e interpretación.

- **ICMS.** Es una interfaz para la comunicación según un protocolo estándar de diversas aplicaciones utilizadas por la Junta de Andalucía y sus organismos asociados.
- **Sistema productor.** En iCMS, un productor es una fuente de información susceptible de ser compartida, reutilizada o explotada de alguna forma.
- **Sistema consumidor.** Es un sistema capaz y destinado a explotar la información que se origina en un sistema productor.
- **Sintaxis Lucene.** Lenguaje que permite interpretar una cadena de texto en una consulta sobre un índice de tipo Lucene, usado como motor en servicios de búsqueda.



## 4. Qué es iCMS

iCMS es un servidor que gestiona el intercambio de información entre productores y consumidores, posibilitando además un conjunto de operaciones común sobre los datos. Estas operaciones son:

- Búsqueda
- Suscripción de proveedores y notificación a consumidores
- Publicación

Este documento se centra en la funcionalidad de Búsqueda.

iCMS proporciona una interfaz común para realizar búsquedas en las diferentes colecciones, totalmente independiente del *backend* (sistema proveedor) en que se encuentren los contenidos de la colección.

iCMS utiliza el formato estándar de descripción y retorno de búsquedas [OpenSearch](#) que aumenta las posibilidades de integración con herramientas y aplicaciones externas.

Cada colección soportará una lista de capacidades de búsqueda. Se recomienda utilizar la [sintaxis de Apache Solr](#) para formular consultas, ya que el servidor iCMS se integra con el motor de búsqueda Apache Solr. No obstante, cada proveedor puede definir su propia sintaxis de consulta, siempre que el descriptor OpenSearch asociado sea correcto.

## 5. Colecciones de datos

Un servicio iCMS agrupa la información en colecciones de datos.

Una colección es un conjunto de información estructurada que refleja parte o toda la información de un sistema productor iCMS.

Un mismo sistema proveedor puede disponer de distintos tipos de contenidos que pueden provenir de gestores de contenidos, bases de datos, etc.

Así, el servidor iCMS acumula toda esa información de distintas fuentes y permite la realización de consultas sobre dichas fuentes en los siguientes formatos:

- XML
- JSON
- Atom



# JUNTA DE ANDALUCIA

- HTML
- RDF
- Turtle
- N3 (Notation3)

## 6. Consultas

En la documentación del servicio iCMS, el punto 5.5 Api Rest contiene todas las peticiones tipo REST que se pueden realizar sobre el servicio y que básicamente afectan a:

- Las propias colecciones
- Suscripciones
- Caché de datos
- Información de estado del servidor

El sistema cuenta con un control de cabeceras y autonegociación de contenidos que identifica por las cabeceras *Content-Type* y *Accept* cuáles son las representaciones y formatos deseados a través de los *mimetypes*. Además, usa los códigos HTTP para representar el estado de las respuestas indicando así errores en la conexión, operaciones no válidas, etc.

El sistema también posee un control de extensiones sobre las peticiones HTTP, de tal manera que devolverá las representaciones y formatos deseados a través de las extensiones asociadas a los *mimetypes* reflejados en la petición, por ejemplo .atom, .json, .n3, etc.

La petición de consulta sobre las colecciones de iCMS siempre será:

`<url_servidor_icms>/rest/datasets/{nombre_colección}`

De esta forma, si el servidor iCMS fuera: <http://www.juntadeandalucia.es/icms> y se desea consultar la colección: jda\_noticias, la URL de consulta debería ser:

[http://www.juntadeandalucia.es/icms/rest/datasets/jda\\_noticias.atom](http://www.juntadeandalucia.es/icms/rest/datasets/jda_noticias.atom)



# JUNTA DE ANDALUCIA

Esta URL debe devolver los 10 últimos contenidos publicados en la colección jda\_noticias.

**Nota:** Hay que tener en cuenta que se debe haber indicado los parámetros de la cabecera *Content-Type* y *Accept* para que la respuesta esté disponible en el formato requerido. Aunque también es posible indicar el formato en la extensión de la URL, por ejemplo si solicitamos XML:

[http://www.juntadeandalucia.es/icms/rest/datasets/jda\\_noticias.xml](http://www.juntadeandalucia.es/icms/rest/datasets/jda_noticias.xml)

## 6.1. Parámetros de la QueryString

La petición HTTP acepta los siguientes parámetros:

- **query.** Para filtrar por parámetros mediante sintaxis Lucene
- **page.** Para paginar la respuesta.
- **items.** Para representar un número de elementos por página en el resultado.
- **order.** Para ordenar los resultados según un metadato.
- **sort.** Para indicar el valor de la ordenación que admite asc y desc.

### 6.1.1. Parámetro: query

En este parámetro se incluye la consulta que pasará directamente al motor de Solr.

Siempre que esta consulta conste de varias premisas, es decir, se trate de una consulta con operadores lógicos que implican varios operandos, se deben utilizar paréntesis:

- Peticiones con una única premisa pueden realizarse de manera query=drupal\:nid:12345
- Peticiones con varias premisas deberán ir entre paréntesis query=(drupal\:nid:12345 OR drupal\:nid:54321)

### Búsquedas por un campo específico.

Se puede buscar por cualquier campo definido en la colección escribiendo el nombre del campo seguido por el carácter: (dos puntos) y el término que se esté buscando. Pero hay que tener en cuenta las siguientes condiciones:

- Si el nombre del campo contiene un prefijo de nombre (*namespace*), este debe ser introducido en la cadena búsqueda utilizando el carácter \. Por ejemplo: buscar el contenido cuyo campo drupal:nid es 94619 en la colección de jda\_noticias



# JUNTA DE ANDALUCIA

[http://www.juntadeandalucia.es/icms/rest/datasets/jda\\_noticias.atom?query=drupal\.:nid:94619](http://www.juntadeandalucia.es/icms/rest/datasets/jda_noticias.atom?query=drupal\.:nid:94619)

- Los posibles tipos de campos en Solr afectan al modo en que pueden ser consultados. Por ejemplo: un campo de tipo fecha puede ser consultado utilizando un rango, pero no se puede utilizar un rango para consultar un tipo texto. En este sentido el motor de Solr en iCMS tiene disponibles los siguientes tipos:
  - string (solr.StrField). Cadena de texto
  - boolean (solr.BoolField). Verdadero/Falso
  - tdate (solr.TrieDateField). Fecha
  - tinteger (solr.SortableIntField). Entero (ordenable)

**Nota:** Se debe tener en cuenta que Apache Solr trabaja para los campos de tipo fecha en formato UTC, por lo que las fechas deben ir en dicho formato. En cualquier caso, el servidor iCMS trabaja por defecto con el tipo String.

- Uso de comodines. Se permiten dos caracteres: interrogación y asterisco. El primero permite enmascarar un único carácter del término de búsqueda y el segundo un grupo de caracteres. Ninguno se puede usar como primer carácter de una búsqueda. Ejemplos de uso de ambos caracteres:
  - [http://www.juntadeandalucia.es/icms/rest/datasets/jda\\_noticias.atom?query=drupal\.:nid:25?](http://www.juntadeandalucia.es/icms/rest/datasets/jda_noticias.atom?query=drupal\.:nid:25?)
  - [http://www.juntadeandalucia.es/icms/rest/datasets/jda\\_noticias.atom?query=drupal\.:nid:2\\*](http://www.juntadeandalucia.es/icms/rest/datasets/jda_noticias.atom?query=drupal\.:nid:2*)
- Búsquedas por rangos. Permite localizar contenidos donde el valor del campo se encuentra entre dos valores que marcan un límite inferior y un límite superior. Ejemplo de uso:  
[http://www.juntadeandalucia.es/icms/rest/datasets/jda\\_noticias.atom?query=drupal\.:nid:\[94619 TO 94700\]](http://www.juntadeandalucia.es/icms/rest/datasets/jda_noticias.atom?query=drupal\.:nid:[94619 TO 94700])



# JUNTA DE ANDALUCÍA

**Nota:** En los rangos se puede usar el carácter especial asterisco (\*) para indicar que uno de los límites (o ambos) son infinitos. Por ejemplo: si en la consulta anterior se desean obtener los contenidos del nid 94619 en adelante, la URL debería ser:

[http://www.juntadeandalucia.es/icms/rest/datasets/jda\\_noticias.atom?query=drupal\.:nid:94619 TO \\*](http://www.juntadeandalucia.es/icms/rest/datasets/jda_noticias.atom?query=drupal\.:nid:94619 TO *)

Para más información acerca del funcionamiento de la sintaxis de Lucene/Solr, se puede consultar:

- Apache Lucene: [https://lucene.apache.org/core/2\\_9\\_4/queryparsersyntax.html](https://lucene.apache.org/core/2_9_4/queryparsersyntax.html)
- Apache Solr: <https://wiki.apache.org/solr/SolrQuerySyntax>

## Búsquedas usando varios campos. Operadores lógicos

Cuando en una búsqueda deban entrar en juego varios campos, es posible utilizar operadores *booleanos* que permiten unir los diferentes filtros que se desean realizar.

Los operadores *booleanos* permiten combinar términos mediante operadores lógicos. Lucene soporta los siguientes:

- **AND.** Coincide con documentos donde existan ambos términos en cualquier parte del texto de un documento único. Equivale a una intersección de conjuntos.
- **OR.** Es el operador de conjunción predeterminado. Es decir, que si no hay operador *booleano* entre dos términos se utiliza OR. El operador OR enlaza dos términos y encuentra un documento coincidente si cualquiera de los términos existen en un documento. Equivale a una unión de conjuntos
- **NOT.** Excluye documentos que contengan el término después de NOT. Equivale a una diferencia de conjuntos.

**Nota:** Los operadores booleanos deben escribirse en mayúsculas.

Por ejemplo: con la siguiente consulta el usuario obtendría contenidos donde el campo: drupal:nid tenga el valor 94619 o bien el 94623.

[http://www.juntadeandalucia.es/icms/rest/datasets/jda\\_noticias.atom?query=\(drupal\.:nid:94623 OR drupal\.:nid:94619\)](http://www.juntadeandalucia.es/icms/rest/datasets/jda_noticias.atom?query=(drupal\.:nid:94623 OR drupal\.:nid:94619))



## 6.1.2. Parámetros: order y sort

El parámetro *Order* permite especificar un campo para la ordenación de la consulta, teniendo en cuenta que el servidor iCMS solamente permite ordenar por campos que no pueden almacenar más de un valor.

Este parámetro suele ir en conjunto con *Sort* que permite indicar el sentido de la ordenación de la consulta. *Sort* únicamente establece dos valores: *asc*, y *desc*. El primero ordena los datos de forma ascendente y el segundo de forma descendente. Se trata de un parámetro opcional de tal forma que, si no se indica, la ordenación se realiza de forma ascendente.

Por ejemplo: si se desea ordenar los contenidos de la colección *jda\_noticias* por el campo *dc:modified* de forma descendente, la URL debería ser:

[http://www.juntadeandalucia.es/icms/rest/datasets/jda\\_noticias.atom?order=dc:modified&sort=desc](http://www.juntadeandalucia.es/icms/rest/datasets/jda_noticias.atom?order=dc:modified&sort=desc)

## 6.1.3. Parámetros: page e items

Estos parámetros permiten configurar el número de la página a consultar así como el número de elementos por página, respectivamente.

El parámetro *page* solo admite números positivos a partir del uno. Por defecto, si no se especifica en la consulta, toma el valor 1.

El parámetro *items*, a diferencia del anterior, sí admite el valor cero, pero en este caso indica al servidor iCMS que la consulta debe devolver todos los contenidos encontrados hasta un límite que se encuentra definido en el servicio y que actualmente es de 10000. Si se especifica otro valor formará las páginas con el número de elementos indicados. Si no se indica, toma por defecto el valor 10.

## 6.2. Parámetros de la cabecera (Header)

*Content-type* y *Accept* son los dos parámetros que permiten especificar el formato de salida de la consulta.

Las representaciones soportadas por la interfaz Rest son:

- XML: `application/xml`



# JUNTA DE ANDALUCIA

- Atom: application/atom+xml
- RDF/XML: application/rdf+xml
- RDF/JSON: application/json
- RDF/TURTLE: text/turtle
- RDF/N3: text/n3
- Texto: text/plain
- HTML: text/html

Las peticiones realizadas a la interfaz Rest son respondidas en la representación que admite cada una de las operaciones y que deben ser definidas en la cabecera.

Una petición a la operación /rest/info, será respondida en formato JSON si en la cabecera se ha establecido el *Content-Type* del *Accept* como application/json.

La extensión de la URL para cada formato son:

- XML: xml
- Atom: atom
- RDF/XML: rdf
- RDF/JSON: json
- RDF/TURTLE: turtle
- RDF/N3: n3
- Texto: text
- HTML: html





# JUNTA DE ANDALUCIA

De esta forma, una consulta típica en formato Atom puede realizarse de dos formas:

a) Directamente en el navegador:

[http://www.juntadeandalucia.es/icms/rest/datasets/jda\\_ayudas.atom](http://www.juntadeandalucia.es/icms/rest/datasets/jda_ayudas.atom)

b) Petición rest:

[http://www.juntadeandalucia.es/icms/rest/datasets/jda\\_ayudas](http://www.juntadeandalucia.es/icms/rest/datasets/jda_ayudas)

Standard HTTP Headers:

- Content-Type = application/atom+xml
- Accept = application/atom+xml

