

Anexo II

Manual de usuario

Webbot para Datos Bibliométricos (WB).

Integrantes :

De Domingo Gonzalo, Guastadisegni Federico, Rodríguez Nahuel.

Cátedra :

Proyecto de Software
Licenciatura en Sistemas.

2015

Requisitos

- Conexión a Internet.
- La carpeta “proyecto_soft” tiene que estar en el directorio /home para poder funcionar correctamente.
- El core “Index” que se encuentra dentro de la carpeta “proyecto_soft” debe estar posicionada en /var/solr/data.
- El Webbot ya tiene que haber sido ejecutado, y el core ya tiene que haber sido Indexado para que las URLs funcionen correctamente.

API Solr

Información general

La API de nuestro Webbot es un servicio REST que proporciona métodos para acceder a recursos específicos en URLs y para la búsqueda y filtrado en base a distintos parámetros. Todas las representaciones se codifican como objetos JSON.

Dominio del servicio

El dominio sobre el cual se basan todas las query posibles es: <http://localhost:8983/solr/Index>

Peticiones

Actualmente, el usuario solamente podrá hacer peticiones HTTP de tipo GET.

Estructura de la respuesta

La mejor manera de entender los resultados devueltos por la API es probar las peticiones que presentaremos mas adelante en este documento. Aún así, ofreceremos una inmersión más profunda en la estructura de resultado API y su organización.

En el caso del “ResponseHeader”, tenemos:

Atributo	Tipo	Descripción
Status	Int	Devuelve 0 en el caso de que la petición haya sido exitosa.
QTime	int	Devuelve el tiempo que tardo en realizarse la petición.
Indent	Boolean	En el caso de que este en TRUE, la petición permite espacios en blanco.
Start	Int	Valor desde el cual debe comenzar la petición.
q	String	Query que se va a enviar junto con la petición.
wt	String	Tipo que se necesita para el objeto respuesta.
rows	Int	Cantidad de resultados esperados.

En cuanto a “Response”, tenemos la siguiente estructura:

Atributo	Tipo	Descripción
id	int	Campo identificatorio de cada una de las publicaciones.
Autor	String	Campo multivaluado que muestra cada uno de los autores de cada publicación.
Titulo	String	Campo que indica el nombre de la publicación.
Año	String	Año en que fue publicado el artículo.
URL	String	Dirección hacia el PDF de la publicación.
ISBN	String	Campo que muestra, dependiendo de la publicación, el ISBN o el ISSN de la misma.
Version	Long	Campo necesario de Solr para su funcionamiento.
numFound	Int	Número de documentos encontrados.

Juntándolos, un resultado típico quedaría de la siguiente manera:

```
{
  "responseHeader":{
    "status":0,
    "QTime":1,
    "params":{
      "indent":"on",
      "start":0,
      "q":"id:1",
      "wt":"json",
      "rows":"1000"}},
  "response":{"numFound":1,"start":0,"docs":[
    {
      "id":1,
      "Año":"2013",
      "Titulo":"Acute Myocardial Infarction and Cocaine: Three Pathophysiological Models",
      "ISBN":"2314-2286",
      "URL":"http://www.sac.org.ar/wp-content/uploads/2014/04/3495.pdf",
      "Autor":["Gabriel Pérez Baztarrica",
        "Flavio Salvaggio",
        "Sandra Zamar",
        "Rafael Porcile"],
      "_version_":1517373211636924416}]
  ]}
```

Errores

Si la petición falla, el Status mostrará un 1. En el caso de ser un error en la query, se mostrará un “code = 400” y un “msg” explicando el por qué del error.

Peticiones posibles

A continuación, describiremos todas las peticiones GET posibles de nuestra API, considerando el dominio que señalamos al comienzo de este documento:

Petición: /select?q=id:Valor&start=0&rows=1000&indent=on&wt=json

Tipo: GET.

Notas de implementación: Obtiene una publicación o artículo con el id especificado.

Tipo de respuesta: JSON.

Parámetro: id : Valor. Debe ser un entero. En caso de no existir, no devolverá nada.

Error: HTTP 404, en caso de ingresar de manera incorrecta la URL. Status : 1, en el caso de que la petición falle. Code = 400, en caso de que la query sea incorrecta.

Petición: /select?q=Autor:"Valor"&start=0&rows=1000&indent=on&wt=json

Tipo: GET.

Notas de implementación: Obtiene la/s publicaciones o artículos con el nombre o parte del nombre especificado.

Tipo de respuesta: JSON.

Parámetro: Autor : “Valor”. El valor especificado tiene que estar entre comillas, por ser un string. En caso de no existir, no devolverá nada. Si el valor ingresado posee espacios, se agregará automáticamente a la URL un %20 por espacio.

Error: HTTP 404, en caso de ingresar de manera incorrecta la URL. Status : 1, en el caso de que la petición falle. Code = 400, en caso de que la query sea incorrecta.

Aclaración específica de Autor: Debido a que AUTOR es un campo multivaluado, en el caso de querer buscar una publicación que incluya dos o mas autores (en la petición a continuación, se muestra como sería para dos), la petición se realiza de la siguiente manera:

Petición para publicación que contenga todos los autores especificados:

/select?q=Autor:"Valor" AND Autor:"Valor"&start=0&rows=1000&indent=on&wt=json

En caso de buscar una publicación que incluya uno u otro autor:

Petición para que la publicación contenga aunque sea alguno de los autores especificados:

/select?q=Autor:"Valor" OR Autor:"Valor"&start=0&rows=1000&indent=on&wt=json

Petición: /select?q=Titulo:"Valor"&start=0&rows=1000&indent=on&wt=json

Tipo: GET.

Notas de implementación: Obtiene la/s publicaciones o artículos con título o parte del título especificado.

Tipo de respuesta: JSON.

Parámetro: Título : “Valor”. El valor especificado tiene que estar entre comillas, por ser un string. En caso de no existir, no devolverá nada. Si el valor ingresado posee espacios, se agregará automáticamente a la URL un %20 por espacio.

Error: HTTP 404, en caso de ingresar de manera incorrecta la URL. Status : 1, en el caso de que la petición falle. Code = 400, en caso de que la query sea incorrecta.

Peticion: /select?q=**ISBN:Valor**&start=0&rows=1000&indent=on&wt=json

Tipo: GET.

Notas de implementación: Obtiene la/s publicaciones o artículos con el ISBN o el ISSN especificado, o bien parte de él.

Tipo de respuesta: JSON.

Parámetro: ISBN: Valor. En caso de no existir, no devolverá nada. Si el valor ingresado posee espacios, se agregará automáticamente a la URL un %20 por espacio.

Error: HTTP 404, en caso de ingresar de manera incorrecta la URL. Status : 1, en el caso de que la petición falle. Code = 400, en caso de que la query sea incorrecta.

Petición: /select?q=**URL:"Valor"**&start=0&rows=1000&indent=on&wt=json

Tipo: GET.

Notas de implementación: Obtiene la/s publicaciones o artículos con la URL especificada.

Tipo de respuesta: JSON.

Parámetro: URL:"Valor". En caso de no existir, no devolverá nada. Si el valor ingresado posee espacios, se agregará automáticamente a la URL un %20 por espacio.

Error: HTTP 404, en caso de ingresar de manera incorrecta la URL. Status : 1, en el caso de que la petición falle. Code = 400, en caso de que la query sea incorrecta.

Petición: /select?q=**Año:Valor**&start=0&rows=1000&indent=on&wt=json

Tipo: GET.

Notas de implementación: Obtiene la/s publicaciones o artículos con el año especificado.

Tipo de respuesta: JSON.

Parámetro: Año : Valor. En caso de no existir, no devolverá nada. Si el valor ingresado posee espacios, se agregará automáticamente a la URL un %20 por espacio.

Error: HTTP 404, en caso de ingresar de manera incorrecta la URL. Status : 1, en el caso de que la petición falle. Code = 400, en caso de que la query sea incorrecta.

Petición: /select?q=***.***&start=0&rows=1000&indent=on&wt=json

Tipo: GET.

Notas de implementación: Obtiene TODAS las publicaciones o artículos que existan en el core.

Tipo de respuesta: JSON.

Parámetro: *.*. En caso de que el core esté vacío, no devolverá nada.

Error: HTTP 404, en caso de ingresar de manera incorrecta la URL. Status : 1, en el caso de que la petición falle. Code = 400, en caso de que la query sea incorrecta.

Aclaración General: En el caso de querer buscar por dos campos distintos, las query se realizan de la siguiente manera (de nuevo, utilizando un ejemplo para dos campos con sus respectivos valores):

Petición para publicación que contenga todos los CAMPOS y VALORES especificados:

/select?q=**campo:Valor AND campo:Valor**&start=0&rows=1000&indent=on&wt=json

Petición para publicación que contenga al menos alguno de los CAMPOS y VALORES especificados:

/select?q=**campo:Valor OR campo:Valor**&start=0&rows=1000&indent=on&wt=json