

Gestión de la Información en la Web MongoDB desde Python (consultas)

Fecha de entrega: martes 12 de enero de 2016, 13:55h

Entrega de la práctica

La entrega de la práctica se realizará a través del Campus Virtual de la asignatura mediante un fichero **grupoXX.zip** donde **XX** es el numero de grupo. Este fichero constará de un fichero **consultas.py** con el código del servidor web, cuyo esqueleto se puede descargar del Campus Virtual. Este fichero **consultas.py** comenzará con un comentario indicando los autores, el grupo y una **declaración de integridad** expresando que el código es fruto del trabajo de sus miembros. Además del servidor web, el fichero **ZIP** contendrá las vistas necesarias para mostrar los datos adecuadamente (si las habéis utilizado).

Objetivos mínimos

Realizar consultas en MongoDB desde Python a través de la librería **pymongo**.

Calificación

Los apartados del 1 al 5 aportan el 13% cada uno, el apartado 6 un 15% y el 7 un 20%.

Otras consideraciones

El código entregado será resultado exclusivamente del trabajo de sus miembros. No se permite ningún tipo de colaboración entre grupos. Cualquier fragmento de código que aparezca repetido en distintas prácticas o copiado de alguna fuente externa implicará **automáticamente** una calificación de cero y se trasladará dicha situación a las autoridades académicas competentes.

En esta práctica consultaremos un servidor MongoDB usando Python mediante la librería pymongo. Para ello diseñaremos un servidor que conteste a distintas peticiones GET y se conecte con el servidor MongoDB para realizar consultas sobre sus colecciones.

El primer paso será descargar el fichero con la colección **users.json**. Debéis importar estos datos en la colección **users** dentro de la base de datos **giw** utilizando **mongoimport**. Tendréis que inspeccionar dicha colección para conocer el esquema implícito de los documentos (todos siguen el mismo patrón), teniendo especial cuidado en detectar qué atributos son cadenas de texto y cuáles son números. **Es muy importante respetar el nombre de la base de datos y de la colección.**

Para realizar la práctica debéis descargar el esqueleto básico del servidor web del Campus Virtual y usarlo como base. Este esqueleto incluye las 7 rutas en las que el servidor web debe responder a peticiones **GET** (no podéis cambiar las rutas, el método HTTP ni añadir nuevas rutas):

1.- /find_user_id

Busca un usuario a través de su identificador. Esta consulta acepta como parámetro **_id** el identificador buscado. El resultado es una página web mostrando los valores de ese usuario en una **lista no numerada** o un mensaje de que no existe en la base de datos.

*Datos del usuario de identificador **user_1***

http://localhost:8080/find_user?_id=user_1

2.- /find_users

Busca usuarios que tengan determinados valores en sus campos. Esta consulta acepta como parámetros cualquier combinación de **_id**, **email**, **gender** y **year**; por ejemplo:

*Todos los usuarios de sexo **Male***

http://localhost:8080/find_users?gender=Male

*Todos los usuarios de sexo **Male** Y año **2009***

http://localhost:8080/find_users?gender=Male&year=2009

Si se pasa algún argumento no definido se mostrará un mensaje de error indicando el argumento o argumentos inválido, como por ejemplo en la consulta:

http://localhost:8080/find_users?gender=Male&drink=water&year=2009

Si no existe ningún error el resultado será una página web que contiene una tabla. Cada fila será un usuario y sus columnas serán:

- ID
- e-mail
- Password Hash
- Gender
- Country
- ZIP
- Year
- Likes

Antes de la tabla aparecerá un mensaje indicando el número de resultados encontrados.

3.- /find_users_or

Igual que la ruta anterior salvo que en este caso la condición es disyuntiva en lugar de conjuntiva; es decir, encontrar los documentos que tienen unos determinados valores **en alguno de sus campos**:

*Todos los usuarios de sexo **Male** O cuyo año es **2000***

http://localhost:8080/find_users_or?gender=Male&year=2000

4.- /find_like

Encuentra todos aquellos usuarios aficionados a una determinada actividad. Esta consulta acepta un parámetro **like** que contiene el nombre de una actividad. El resultado debe ser una página web con el número de resultados seguido de una tabla con los datos de los usuarios (como en el apartado 2).

*Todos los usuarios a los que les gusta el **fútbol***

http://localhost:8080/find_like?like=football

5.- find_country

Devuelve todos los usuarios de un determinado país que se pasa como parámetro **country**. El resultado debe ser una página web con el número de resultados seguido de una tabla con los datos de los usuarios (como en el apartado 2).

*Todos los usuarios de **China***

http://localhost:8080/find_country?country=China

6.- /find_email_year

Busca únicamente el **identificador** y el **e-mail** de todos los usuarios de un determinado año que se pasa como parámetro **year**. El resultado es una página web con el número de resultados seguido de una tabla con los datos de los usuarios. Cada fila será un usuario y tendrá dos columnas llamadas **ID** y **e-mail**.

Identificador y correo electrónico de los usuarios del año 1992

http://localhost:8080/find_email_year?year=1992

7.- /find_country_limit_sorted

Encuentra los primeros usuarios de un determinado país según ordenados por año según el orden especificado. Esta consulta recibe 3 parámetros:

- **country**: país seleccionado
- **limit**: número de usuarios a mostrar como máximo
- **ord**: orden a aplicar sobre el año, que puede ser ascendente **asc** o descendente **desc**.

El resultado debe ser una página web con el número de resultados seguido de una tabla con todos los datos de los usuarios (como en el apartado 2).

Primeros 20 usuarios de España por orden ascendente de año

http://localhost:8080/find_country_limit_sorted?country=Spain&limit=20&ord=asc