

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

Fecha de entrega: martes 8 de diciembre de 2015, 13:55h

Entrega de la práctica

La entrega de la práctica se realizará a través del Campus Virtual de la asignatura. Constará únicamente de un fichero **.py** con el código del servidor web implementado utilizando el *framework* **bottle**. El fichero **.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.

Objetivos mínimos

Insertar y eliminar documentos en MongoDB desde Python a través de la librería **pymongo**.

Calificación

Cada función aporta un 20% de la nota total de la práctica.

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 aprenderemos a conectar con un servidor MongoDB usando Python mediante la librería **pymongo**. Para ello diseñaremos un servidor que conteste a distintas peticiones POST y se conecte con el servidor MongoDB para realizar modificaciones.

El primer paso será descargar el fichero con la colección de prueba **users.json**. Debéis importar estos datos en la colección **users** dentro de la base de datos **giw** (ver pistas al final del enunciado). Tendréis que inspeccionarla 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.

Para realizar la práctica debéis descargar el esqueleto básico de servidor web desde el Campus Virtual y usarlo como base. Este esqueleto incluye las 5 rutas en las que el servidor web debe responder a peticiones **POST** (no podéis cambiar las rutas ni el método HTTP):

1.- /add_user

Inserta un usuario en la colección **users**. Recibe los siguientes parámetros:

- **_id**: identificador del usuario
- **country**: país del usuario
- **zip**: código postal del usuario
- **email**: dirección de correo electrónico del usuario
- **gender**: sexo del usuario
- **likes**: cadena de texto separada por comas con las aficiones del usuario. Ejemplo: *"football,painting,baseball"*
- **password**: hash de la contraseña del usuario

- **year**: año de nacimiento del usuario

Con estos datos inserta el documento pertinente en la colección **users**. Si el usuario ya existe en la colección el servidor web mostrará un mensaje indicándolo. En otro caso mostrará un mensaje indicando que la inserción ha tenido éxito.

2.- /change_email

Modifica la dirección de un usuario que ya existe en la colección. Recibe dos parámetros:

- **_id**: identificador de usuario
- **email**: nueva dirección de correo electrónico

Como resultado de esta petición el servidor web debe mostrar el número de documentos modificados.

3.- /insert_or_update

Similar al punto 1 si el usuario no existe. Sin embargo si el usuario ya existe realiza una actualización de todos sus datos. Recibe los mismos parámetros detallados en el punto 1. El resultado de esta petición es un mensaje indicando si se ha producido una inserción o una modificación. Consideramos que un usuario existe si hay un documento con dicho identificador.

4.- /delete

Recibe un identificador (clave **_id**) y elimina dicho usuario de la colección. Como resultado el servidor web debe mostrar el número de documentos eliminados, que será 0 o 1.

5.- /delete_year

Recibe un año (clave **year**) y elimina todos los usuarios nacidos en ese año. El servidor web debe mostrar el número de documentos eliminados.

Pistas

a) Importar colecciones en MongoDB:

```
$ mongoimport --db giw --collection users users.json
```

b) Librerías y clases útiles:

- (pymongo) MongoClient
https://api.mongodb.org/python/current/api/pymongo/mongo_client.html
- (pymongo) Cursor
<https://api.mongodb.org/python/current/api/pymongo/cursor.html>
- (pymongo) Collection
<https://api.mongodb.org/python/current/api/pymongo/collection.html>
- (pymongo) Excepciones
<https://api.mongodb.org/python/current/api/pymongo/errors.html>
- (bottle) Request
<http://bottlepy.org/docs/dev/api.html#bottle.BaseRequest>

c) La forma más sencilla de realizar peticiones POST es crear una página web con un formulario que se comunique con la ruta adecuada. Por ejemplo:

```
...  
<form action="http://localhost:8080/add_user" method="post">  
...
```