MongoDB y Java: Parte IV, consultas básicas a la base de datos

Una vez configurado el entorno y probado que podemos conectarnos y acceder a la base de datos, vamos a ver como realizar consultas básicas desde Java a MongoDB. En la parte III vimos como conectarnos, seleccionar una colección y contar el número de documentos en ella.

Obtener todos los elementos de la colección

El método find() nos permite obtener todos los documentos de una colección, y first() devuelve el primer elemento de cualquier consulta. Llamar a first() sobre una consulta es útil cuando esta debería devolver un solo elemento o queremos el primer elemento de dicha consulta. En el caso de que la consulta no devuelva resultados, first() devuelve null.
[crayon-55a8f274cbbc3990997843/]

Crear e insertar un nuevo documento usando Document BSON

Para insertar un documento en la base de datos, podemos hacerlo creando un Document BSON y añadiendo a este los datos que queramos. Vamos a añadir un restaurante con el formato que se usa en el primer documento de la colección pero omitiendo algunos campos:
[crayon-55a8f274cbbd1299617146/]
Creamos un nuevo documento y añadimos dos campos, "borough" y "cuisine" con sus respectivos valores. Como las puntuaciones son una lista, creamos un ArrayList() para rellenarla con ellas. Para crear una puntuación, haremos uso de documentos embebidos de MongoDB, creando un nuevo Document y poblándolo con los datos de la puntuación para luego añadirlo al documento principal.

Como vemos en la línea "grade.append("date", new Date());", para representar la fecha utilizamos directamente la clase Date de Java, al igual que para un Array BSON utilizamos List. Para ver la correspondencia entre los tipos de datos BSON y Java podemos consultar el enlace al final de esta entrada. Finalmente añadimos el documento de la puntuación a la lista, la lista al documento principal e insertamos este en la colección.

Recuperar y borrar un documento usando un filtro

Ahora vamos a recuperar el documento que hemos insertado y mostrarlo. La forma más fácil es hacer una consulta y usar un filtro, en nuestro caso filtraremos por el nombre, y debe ser igual. El filtro está definido por "eq("name", "Cafetería FIC")" y se lo debemos pasar al metodo find() como parámetro. Como vemos usamos first() para obtener el primer elemento y evitarnos iterar sobre el resultado.

Por último borramos el documento usando deleteOne() y el mismo filtro. En caso de querer eliminar todos los documentos que coincidan con el filtro usaríamos deleteMany().
[crayon-55a8f274cbbda537584160/]

Crear e insertar un nuevo documento usando un objeto JSON representado por un string y JSONObject

Para añadir un documento a partir de una representación en una string de un objeto JSON, primero deberemos convertirla en un JSONObject. A continuación utilizamos la función Document.parse() para convertirlo a un Document BSON e insertarlo.
[crayon-55a8f274cbbe0677475104/]

El ejemplo completo lo podéis descargar al final de esta entrada. El resultado de la ejecución debería ser el siguiente:

Descarga del ejemplo TestMongo.java: TestMongo

Fuentes:

MongoDB Java driver CRUD

MongoDB Java driver quick tour

Correspondencia de tipos BSON Java

Model One-to-Many Relationships with Embedded Documents

MongoDB Java Filters