2.1 - Tecnologías de Servicios Web

Tema 4 – Bases de datos

Universidad Rey Juan Carlos

Micael Gallego
micael.gallego@urjc.es
@micael_gallego

code URJC

Tema 4 – Bases de datos

Tema 4.3 – MongoDB en Spring

Universidad Rey Juan Carlos

Micael Gallego
micael.gallego@urjc.es
@micael_gallego

code URJC

MongoDB

code URJC

https://www.mongodb.com/

MongoDB

code URJC

- Instalación

  https://docs.mongodb.com/manual/installation/#mongodb-community-edition-installation-tutorials

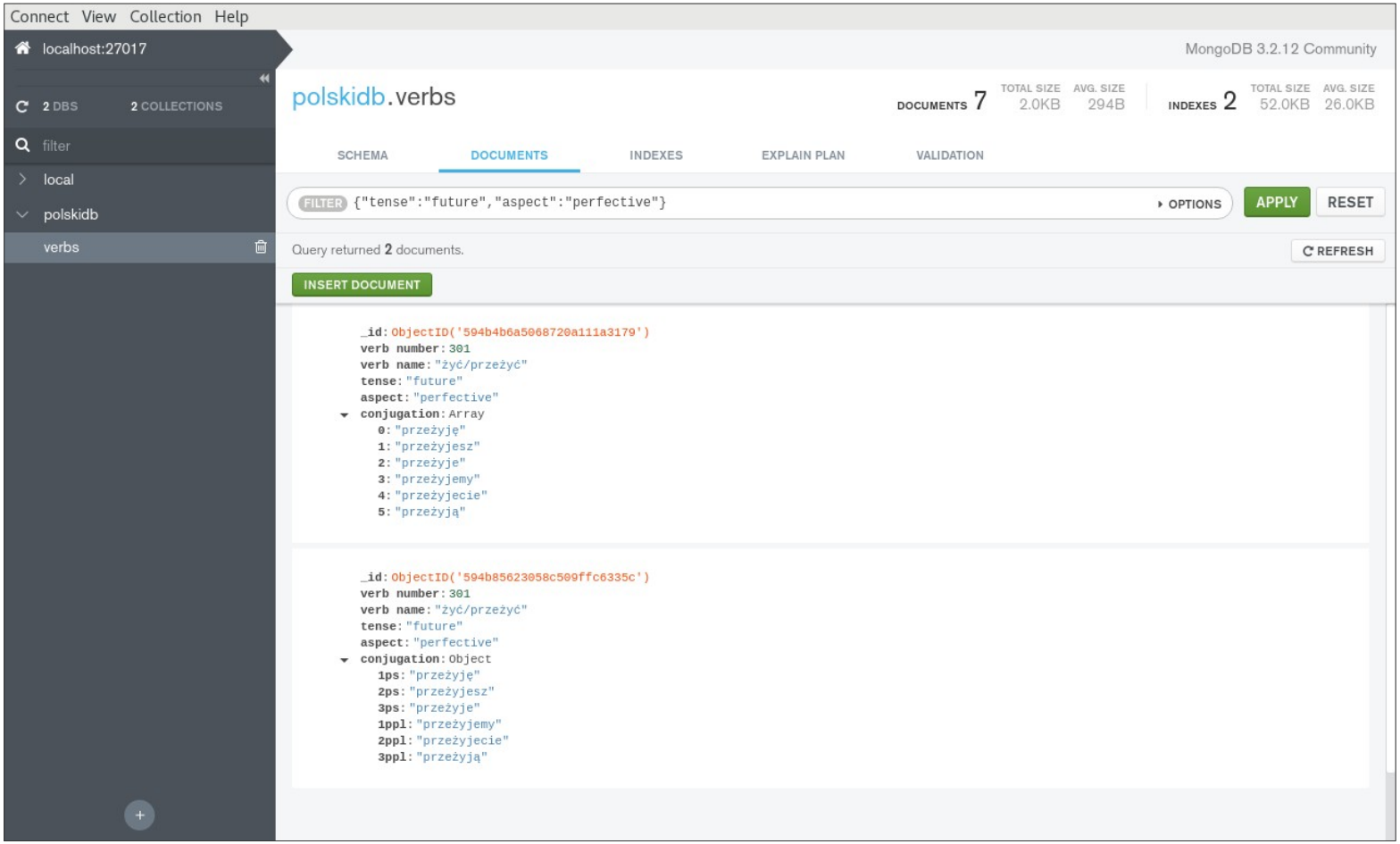
- Docker

```
$ docker run --rm -p 27017:27017 -d mongo:4.4-bionic
```

- Cliente gráfico (opcional)

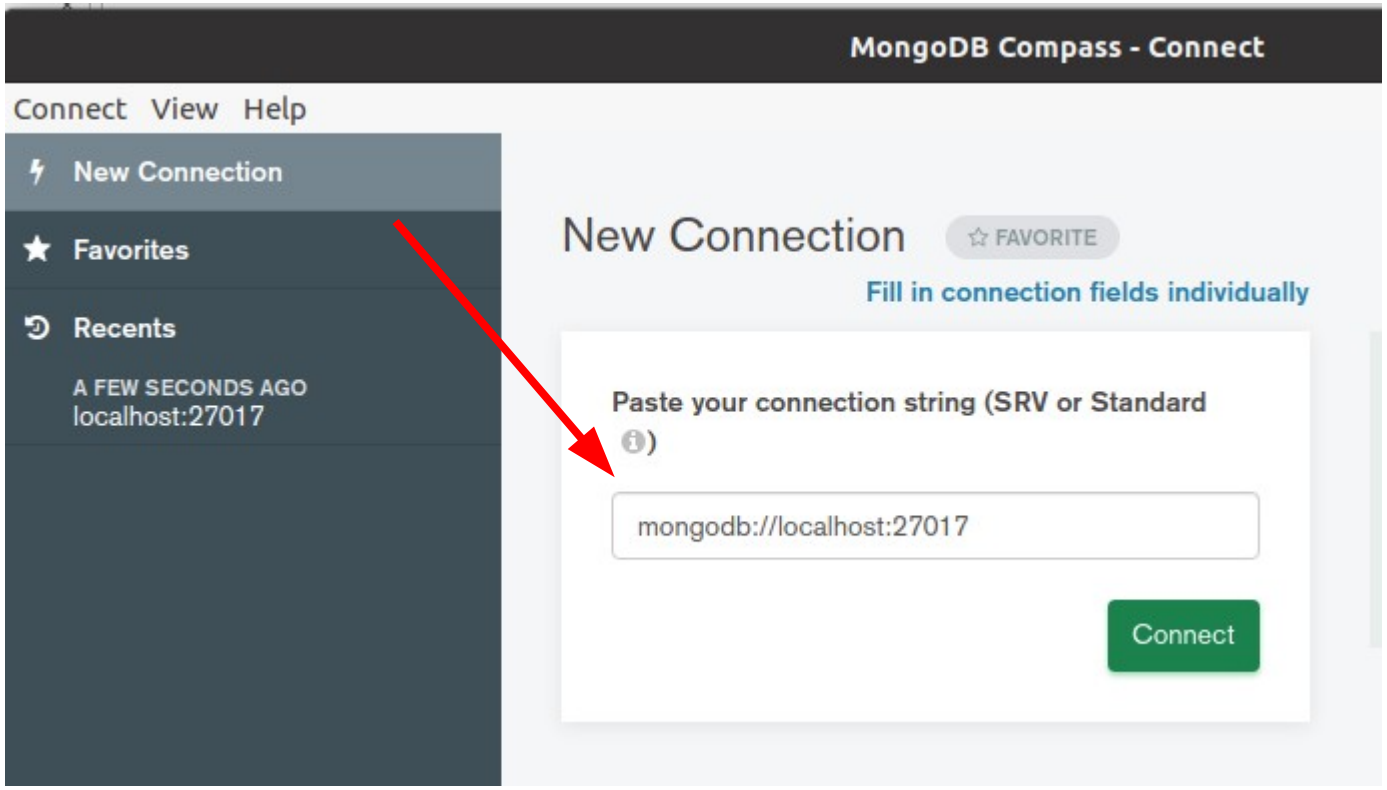
https://www.mongodb.com/products/compass

MongoDB



https://www.mongodb.com/products/compass

MongoDB



mongodb://localhost:27017

MongoDB

code URJC

- En MongoDB se manejan **colecciones**, que guardan **documentos**.
  - Una **base de datos** Mongo tiene un conjunto de **colecciones**
  - Una **colección** tiene un conjunto de **documentos**
  - Un **documento** es un binario serializado en formato **BSON** (*Binary jSON*)

https://mongodb.github.io/node-mongodb-native/3.3/api/Collection.html

MongoDB

code URJC

- Documentos

```
{
    name: "patxi",
    age: 37,
    preferences: [
        "functional programming",
        "nosql",
        "java"
    ]
}
```

MongoDB

https://docs.mongodb.com/manual/reference/bson-types

- BSON
  - Double
  - String
  - Object
  - Array
  - Binary data
  - Undefined
  - ObjectId
  - Boolean
  - Date
  - Null
  - Regular Expression
  - DBPointer
  - JavaScript
  - Symbol
  - JavaScript (with scope)
  - 32-bit integer
  - Timestamp
  - 64-bit integer
  - Decimal128
  - Min key
  - Max key

MongoDB

code URJC

https://docs.mongodb.com/manual/reference/bson-types

- BSON
  - Double
  - **String**
  - **Object**
  - Array
  - Binary data
  - **Undefined**
  - ObjectId
  - **Boolean**
  - Date
  - **Null**
  - Regular Expression
  - DBPointer
  - JavaScript
  - **Symbol**
  - JavaScript (with scope)
  - 32-bit integer
  - Timestamp
  - 64-bit integer
  - Decimal128
  - Min key
  - Max key

- JSON
  - Boolean
  - Null
  - Undefined
  - Number
  - String
  - Symbol (ES6)
  - Object: Object, Array, Function…

MongoDB

- Operaciones sobre una BBDD MongoDB:
  - Añadir una colección
  - Eliminar una colección
  - Añadir un documento a una colección
  - Buscar documentos en una colección
  - Actualizar un documento en una colección
  - Eliminar un documento de una colección
  - Manejo de arrays

MongoDB

- **Shell para interactuar con Mongo**

```
$ mongo
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> db
test
>_
```

Si usamos Mongo en Docker podemos instalar únicamente el clientes mongo

```
$ sudo apt install mongodb-clients
```

MongoDB

code URJC

- **Creación de bases de datos y colecciones**
  - Las bases de datos y las colecciones no se crean de forma explícita
  - Se define la base de datos a usar
  - Se inserta un documento en alguna colección

```
> use mca
> db.users.insert({"name":"patxi", "age":36})
> show dbs
> show collections
```

MongoDB

code URJC

- En una colección puede haber **documentos con estructura diferente**

```
>db.users.insert({"name":"carlos","age":25,"preferences":["skyiing","swimming","travelling"]})
>db.users.find()
{ "_id" : ObjectId("533a8b942ef10f2fc063eb24"), "name" : "patxi", "age" : 37 }
{ "_id" : ObjectId("533a922d2ef10f2fc063eb25"), "name" : "carlos", "age" : 25, "preferences" : [
"skyiing",  "swimming",  "travelling" ] }
```

MongoDB

code URJC

- Puede haber documentos embebidos

```
> db.users.insert({"name":"alejandro","age":2,"parent":{"name":"patxi","age":37}})
> db.users.find().pretty()
{
     "_id" : ObjectId("533a8b942ef10f2fc063eb24"),
     "name" : "patxi",
     "age" : 37
}
...
{
     "_id" : ObjectId("533a92f82ef10f2fc063eb26"),
     "name" : "alejandro",
     "age" : 2,
     "parent" : {
          "name" : "patxi",
          "age" : 37
     }
}
```

MongoDB

code URJC

- **Gestión de colecciones y bases de datos**
  - Listar colecciones de la database en uso

```
> db.getCollectionNames()
```

  - Borrar colección

```
> db.<COLLECTION_NAME>.drop()
```

  - Listar documentos de colección

```
> db.<COLLECTION_NAME>.find()
```

  - Borrar base de datos

```
> db.dropDatabase()
```

MongoDB

code URJC

- Consultas sobre documentos

```
> db.users.find({"name":"patxi"})
```

```
{ "_id" : ObjectId("533a8b942ef10f2fc063eb24"), "name" : "patxi", "age" : 36 }
```

MongoDB

code URJC

- **Operadores**

```
and: {age:37, gender:"M"}

or: {$or: [{zipcode:27001}, {zipcode: 27002}]}

in: {tags:{$in: ["databases", "nosql"]}}

lt, gt, lte, gte: {type:"flight",price: {$lt:50}}
```

MongoDB

code URJC

- Combinación de operadores

```
{
    type: "flight",
    from: "Madrid",
    $or: [
        price:{$lt: 100},
        destiny:{$in: ["Roma", "Praga"]}
    ]
}
```

MongoDB

- Búsquedas en subdocumentos

```
{"parent.name": "patxi"}
```

MongoDB

code URJC

- Seleccionar los campos a devolver
  - `db.users.find({"name": "patxi"}, {"name":1, "age":1})`
- Limitar el número de resultados
  - `db.users.find({"age":{$lt:40}}).limit(10)`
- Ordenarlos
  - `db.users.find({"type":"flight"}).sort({"price":1})`

MongoDB

code URJC

- Modificar un campo de un documento
  - `db.users.update({"name": "patxi"}, {"age":36})`

- Modificar múltiples documentos
  -
```
db.users.update({"age":{$lt:18}},
        {"drivingLicense":false},
        {multi: true})
```

MongoDB

code URJC

- Reemplazar un documento por otro
  - `db.users.save({_id:10,"name": "patxi", age:55})`
- Eliminar documentos
  - Eliminar todos los documentos cuyo atributo type sea “flight”
    - `db.users.remove({type:"flight"})`
  - Eliminar el primer documento que cumpla la condición
    - `db.users.remove({name:"patxi"}, true)`
  - Eliminar todos los documentos de la colección
    - `db.users.remove({})`

MongoDB en Spring

code URJC

- Spring Data tiene soporte para MongoDB
- Es bastante similar al uso de bases de datos relacionales
  - Entidades
  - Repositorios y consultas
  - Configuración de la ruta a la BD
  - Dependencias

MongoDB en Spring

code URJC

ejem23

- **Entidades**

La anotacion @Id está en el paquete **org.springframework.data.annotation.Id**

En JPA está en jakarta.persistence.Id

La entidad puede estar sin anotar.

Para configurar algunos aspectos se puede usar **@Document**

En JPA se usa @Entity

```
public class Post {

   @Id
   private String id;

   private String user;
   private String title;
   private String text;

}
```

El id es de tipo **String**

En JPA se usa int o long

MongoDB en Spring

code URJC

ejem23

- Repositorios y Consultas

Se extiende **MongoRepository**

En JPA se usa JpaRespository

```
public interface PostRepository extends MongoRepository<Post, String> {

}
```

MongoDB en Spring

code URJC

ejem23

- Configuración de la ruta a la BD

La configuración del log para que se muestren las consultas es específica de Mongo

```
logging.level.org.springframework.data.mongodb.core.MongoTemplate=DEBUG

spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
spring.data.mongodb.database=posts
#spring.data.mongodb.username=
#spring.data.mongodb.password=
```

La configuración de localización de la base de datos y credenciales es específica de Mongo

MongoDB en Spring

code URJC

ejem23

- Dependencias

Con esta dependencia es suficiente

En JPA es necesario una dependencia para JPA y otra para el driver de la BD

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-mongodb</artifactId>
</dependency>
```