## JSON

1. Create a JSON objects “user”: username, display name, Email, created date
2. Create a JSON objects “post”: title, content, published Date, Edited Date, Owner (which is username)

{

"title": "shop annguyen",

"content": "shoes",

"published Date": "01/03/2019",

"Edited Date": "01/03/2019",

"Owner": {

"username": "annguyen"

}

}

1. Create a JSON objects “post”: title, content, published Date, Edited Date, Owner (which is username), comments (text, date, owner)

{

"title": "shop annguyen",

"content": "shoes",

"published Date": "01/03/2019",

"Edited Date": "01/03/2019",

"Owner": {

"username": "annguyen"

},

"comments ": {

"text": "annguyen",

"date": "02/03/2019",

"owner": "Nguyen"

}

}

]

1. Create a JSON objects “post”: title, content, published Date, Edited Date, Owner (which is username), comments (text, date, owner contains both username & display name), tags (username, display name)

{

"title": "shop annguyen",

"content": "shoes",

"published Date": "01/03/2019",

"Edited Date": "01/03/2019",

"Owner": {

"username": "annguyen"

},

"comments ": {

"text": "annguyen",

"date": "02/03/2019",

"owner": {

"username ": "annguyen",

"displaynname": "An Nguyen"

}

},

"tags": {

"username": "annguyen",

"display name": "An Nguyen"

}

}

1. Create a JSON objects “post”:
   * title,
   * content,
   * published Date,
   * Edited Date,
   * Owner: which is username,
   * Comments:
     + text,
     + date,
     + owner: contains both username & display name,
     + replies:
       - reply1: { text: “Reply 1”, “owner”: “annguyen”}
       - reply2: {text: “reply 2”, “owner”:”hoaphan”}
     + tags:
       - tag1: username: hoaphan, display name: Hoa Phan
       - tag2: username:khanhnguyen, displayname: Khanh Nguyen

## MongoDB

Design data model for a Blogging with objects:

* User who is the post authors: username, display name, email, create date
* Post: Title, content, published Date, Edited Date, Owner
* Comments: comment, date, owner, post
* Tags: name, URL

Complete steps below:

1. mongod.exe --dbpath " c:\Khanh\training\SQL\_NoSQL\MongoDB\Data"
2. Start Mongo server: mongo.exe
3. connect to mongo DB: db.stats()
4. connect to mongo DB
5. Write a MongoDB create to create new database “Blog” : use Blog
6. Show the current Database : db
7. Drop current Database : db.dropDatabase()
8. Create “**user**” collection and insert 5 users document into that collection with **\_id = username**

db.createCollection("user")

db.user.insert({ username : "annguyen", "display name" : "An Nguyen", "Email" : "Anguyen@csc.com", "created date" : 10/03/2019 })

db.user.insert({ username : "anhnguyen", "display name" : "Anh Nguyen", "Email" : "Anguyen1@csc.com", "created date" : 10/03/2019 })

db.user.insert({ username : " hoaphan", "display name" : "Hoa Phan", "Email" : "[hphan@csc.com](mailto:hphan@csc.com)", "created date" : 10/03/2019 })

|  |  |  |  |
| --- | --- | --- | --- |
| User name | Display Name | Email | Create date |
| annguyen | An Nguyen | [Anguyen@csc.com](mailto:Anguyen@csc.com) | today |
| anhnguyen | Anh Nguyen | [Anguyen1@csc.com](mailto:Anguyen1@csc.com) | today |
| hoaphan | Hoa Phan | [hphan@csc.com](mailto:hphan@csc.com) | today |
| Hungnguyen | Hung Nguyen | [hnguyen@csc.com](mailto:hnguyen@csc.com) | Today |
| khanhnguyen | Khanh Nguyen | [knguyen@csc.com](mailto:knguyen@csc.com) | today |
|  |  |  |  |

1. Create a “**post**” collection and insert 5 posts document into that collection with detail below:
   1. Post 1: owner store username of user =annguyen that created in previous steps
   2. Post 2: owner stores user info {username, display name} of username = khanhnguyen
   3. Post 3: owner stores user info {username, display name} of username = hoaphan
   4. Post 4:
      1. owner store username of user =anhnguyen that created in previous steps
      2. this post has 2 comments: [

{ “comment”: “ this is the first comment”,

“date”: <today>,

“owner”: “annguyen”

},

{ “comment”: “ this is the second comment”,

“date”: <today>,

“owner”: { “username”: “hoaphan”

“displayname”: Hoa Phan

},

“replies”: { “reply”: “Reply 1”,

“date”: <today>

“owner”: “annguyen”

}

}

]

1. Write a MongoDB query to display all the **user** documents in the collection user.

db.user.find().pretty()

1. Write a MongoDB query to display username, display name, email of the first 2 employees

db.user.find().limit(2).sort({"username":-1})

1. Write a MongoDB query to display posts created by “anhnguyen”

db.user.find({"username":"anhnguyen"})

1. Write a MongoDB query to display posts created by “anhnguyen” and Email is Anguyen@csc.com

db.user.find({$and: [{"username":"annguyen"}, {"Email":"Anguyen@csc.com"}]})

1. Write a MongoDB query to delete post created by hoanguyen

db.user.remove({"username":"hoaphan"})

db.user.remove({"username":"hoaphan"},1)

## Access Data with MongoDB using Spring Boot and Spring Data MongoDB

Requirements:

* Maven 3.x
* Spring Boot 4.x
* Valid MongoDB server

Steps to implement:

1. Design a domain object to present the Product: id, name, title, description, imageUrl, price.
2. Create Product Repository:

* ProductRepository extends the MongoRepository interface and plugs in the type of values and id it works with: Product and String
* Implement the Product findById(String id) method. This method finds the Product object from the mongoDB, converts it into a Product object, and returns the Product object. If no Product s found, this method throws the NotFoundException.
* Implement the List<Product> searchByPrice(fromPrice, toPrice) to display products math the condition.

1. Start MongoDB server
2. Make the application executable.

* Add @SpringBootApplication