

Project - Learn to use MongoDB

Programming Language and Tools

Java, Spring(Spring Data MongoDB), MongoDB, Robo3T, Jackson

Pseudo Code and Structure

The code runs with following sequence

1. Java Jar connects to MongoDB database using connection string, database ip, port, username and password
2. The jar reads all the files placed in the 'files' folder and converts the csv files to equivalent Java objects and saves the data to MongoDB in normalized form. The respective CollectionName/Java Object Names are as follows:
 - EmployeeNormal
 - DepartmentNormal
 - ProjectNormal
 - WorksOnNormal
3. Once csv files are stored in MongoDB, a function call is made to create nested documents of Project, Employee and Department.
4. For Task - 1 Project and Nested Employee structure a join is created on ProjectNormal and WorksOnNormal by Pno(Project Number) and List<SSN> is retrieved of the employees working on that project. Using this SSN all Employees are fetched who are working under that specific project. A java object of Project is created which contains List<Employee> nested inside it. All the required data is populated and data is persisted into MongoDB new collection - **PROJECTS**.
5. Similar approach is done for rest 2 tasks.
6. XML file of Department collection generated at Jar level

Sample Queries

Fetching all projects between Project No 1 and 30

```
db.getCollection('PROJECTS').find({PNUMBER:{"$gt": 1, "$lt": 30}})
```

Fetching all projects with department name = Research or Software

```
db.getCollection('PROJECTS').find({DNAME:{$in:["Research", "Software"]}})
```

Fetching count of projects with department name = Software

```
db.getCollection('PROJECTS').find({DNAME:"Software"}).count()
```

Sorting EMPLOYEE collection by Employee Salary in descending

```
db.getCollection('EMPLOYEES').find().sort({EMP_SALARY:-1})
```

Counting employees by distinct SSN

```
db.getCollection('EMPLOYEES').distinct('EMP_SSN').length
```

Fetching count of employees whose sex = M

```
db.getCollection('EMPLOYEES').find({EMP_SEX:'M'}).count()
```

Fetching employees whose salary >= 90000 And DNO = 7

```
db.getCollection('EMPLOYEES').find({
  $and: [
    {EMP_SALARY: { $gte: 90000 }},
    {EMP_DNO:7}
  ]
})
```