**Config Analyst (Assignment)**

**--------------------------------------------------------------------------------------------------------------------------**

***Name: Patel Arun Ramjanak***

**----------------------------------------------------------------------------------------------------------------------------**

**Write down the syntax for each of the question below with answer/output**

**----------------------------------------------------------------------------------------------------------------------------**

(1) create database “policyDB”

**ANS:- use policyDB**

----------------------------------------------------------------------------------------------------------------------------

(2) create collection named “policyHolders”

**FieldsName:**

"insuredFirstName" :

"insuredLastName" :

“Age”:

“DateOfBirth”:

"State" :

“email" :

"PhoneNumber" :

"Status" :

"Beneficiaries" :

“Relationship”:

**ANS:- db.createCollection("policyHolders")**

----------------------------------------------------------------------------------------------------------------------------

(3) create collection named “claims”

**FieldsName:**

“ID”:

"claimNumber" :

" lineOfBusiness" :

"insuredFullName" :

"insuredFName" :

"insuredLName" :

"insuredDOB" :

"dateOfDeath" :

" Status" :

"effectiveDate" :

"insuredState" :

**ANS:- db.createCollection("claims")**

----------------------------------------------------------------------------------------------------------------------------

***Queries***

----------------------------------------------------------------------------------------------------------------------------

(1) Insert at least 10 documents in the collection policyHolders

**ANS:- db.policyHolders.insertMany([**

**{ "insuredFirstName": "insuredFirstName1", "insuredLastName": "insuredLastName1", "Age": 20, "DateOfBirth": ISODate("1989-05-15"), "State": "NY", "email": "john.doe@example.com", "PhoneNumber": "1234567890", "Status": "Active", "Beneficiaries": ["Alice", "Bob"], "Relationship": "Unknown" },**

**{ "insuredFirstName": "insuredFirstName2", "insuredLastName": "insuredLastName2", "Age": 20, "DateOfBirth": ISODate("1989-05-15"), "State": "NY", "email": "john.doe@example.com", "PhoneNumber": "1234567890", "Status": "Active", "Beneficiaries": ["Alice", "Bob"], "Relationship": "Unknown" },**

**{ "insuredFirstName": "insuredFirstName3", "insuredLastName": "insuredLastName3", "Age": 85, "DateOfBirth": ISODate("1989-05-15"), "State": "NY", "email": "john.doe@example.com", "PhoneNumber": "1234567890", "Status": "Active", "Beneficiaries": ["Alice", "Bob"], "Relationship": "Unknown" },**

**{ "insuredFirstName": "AAinsuredFirstName4", "insuredLastName": "insuredLastName4", "Age": 20, "DateOfBirth": ISODate("1989-05-15"), "State": "NY", "email": "john.doe@example.com", "PhoneNumber": "1234567890", "Status": "Active", "Beneficiaries": ["Alice", "Bob"], "Relationship": "Unknown" },**

**{ "insuredFirstName": "insuredFirstName5", "insuredLastName": "insuredLastName5", "Age": 30, "DateOfBirth": ISODate("1989-05-15"), "State": "NY", "email": "john.doe@example.com", "PhoneNumber": "1234567890", "Status": "Active", "Beneficiaries": ["Alice", "Bob"], "Relationship": "Unknown" },**

**{ "insuredFirstName": "insuredFirstName6", "insuredLastName": "insuredLastName6", "Age": 35, "DateOfBirth": ISODate("1989-05-15"), "State": "NY", "email": "john.doe@example.com", "PhoneNumber": "1234567890", "Status": "Active", "Beneficiaries": ["Alice", "Bob"], "Relationship": "Unknown" },**

**{ "insuredFirstName": "insuredFirstName7", "insuredLastName": "insuredLastName7", "Age": 20, "DateOfBirth": ISODate("1989-05-15"), "State": "NY", "email": "john.doe@example.com", "PhoneNumber": "1234567890", "Status": "Active", "Beneficiaries": ["Alice", "Bob"], "Relationship": "Unknown" },**

**{ "insuredFirstName": "insuredFirstName8", "insuredLastName": "insuredLastName8", "Age": 20, "DateOfBirth": ISODate("1989-05-15"), "State": "NY", "email": "john.doe@example.com", "PhoneNumber": "1234567890", "Status": "Active", "Beneficiaries": ["Alice", "Bob"], "Relationship": "Unknown" },**

**{ "insuredFirstName": "insuredFirstName9", "insuredLastName": "insuredLastName9", "Age": 20, "DateOfBirth": ISODate("1989-05-15"), "State": "NY", "email": "john.doe@example.com", "PhoneNumber": "1234567890", "Status": "Active", "Beneficiaries": ["Alice", "Bob"], "Relationship": "Unknown" },**

**{ "insuredFirstName": "insuredFirstName10", "insuredLastName": "insuredLastName10", "Age": 20, "DateOfBirth": ISODate("1989-05-15"), "State": "NY", "email": "john.doe@example.com", "PhoneNumber": "1234567890", "Status": "Active", "Beneficiaries": ["Alice", "Bob"], "Relationship": "Unknown" },**

**])**

----------------------------------------------------------------------------------------------------------------------------

(2) Insert at least 10 documents in the collection claims.

**ANS:- db.claims.insertMany([**

**{ "claimNumber": "C001", "lineOfBusiness": "Life Insurance", "insuredFullName": "John Doe", "insuredFName": "John", "insuredLName": "Doe", "insuredDOB": ISODate("1989-05-15"), "dateOfDeath": ISODate("2024-02-10"), "Status": "Pending", "effectiveDate": ISODate("2022-01-01"), "insuredState": "NY" },**

**{ "claimNumber": "C002", "lineOfBusiness": "Life Insurance", "insuredFullName": "John Doe", "insuredFName": "John", "insuredLName": "Doe", "insuredDOB": ISODate("1989-05-15"), "dateOfDeath": ISODate("2022-02-10"), "Status": "Pending", "effectiveDate": ISODate("2022-01-01"), "insuredState": "NY" },**

**{ "claimNumber": "C003", "lineOfBusiness": "Life Insurance", "insuredFullName": "John Doe", "insuredFName": "John", "insuredLName": "Doe", "insuredDOB": ISODate("1989-05-15"), "dateOfDeath": ISODate("2021-02-10"), "Status": "Pending", "effectiveDate": ISODate("2022-01-01"), "insuredState": "NY" },**

**{ "claimNumber": "C004", "lineOfBusiness": "Life Insurance", "insuredFullName": "John Doe", "insuredFName": "John", "insuredLName": "Doe", "insuredDOB": ISODate("1989-05-15"), "dateOfDeath": ISODate("2025-02-10"), "Status": "Pending", "effectiveDate": ISODate("2022-01-01"), "insuredState": "NY" },**

**{ "claimNumber": "C005", "lineOfBusiness": "Life Insurance", "insuredFullName": "John Doe", "insuredFName": "John", "insuredLName": "Doe", "insuredDOB": ISODate("1989-05-15"), "dateOfDeath": ISODate("2019-02-10"), "Status": "Pending", "effectiveDate": ISODate("2022-01-01"), "insuredState": "NY" },**

**{ "claimNumber": "C006", "lineOfBusiness": "Life Insurance", "insuredFullName": "John Doe", "insuredFName": "John", "insuredLName": "Doe", "insuredDOB": ISODate("1989-05-15"), "dateOfDeath": ISODate("2023-02-10"), "Status": "Pending", "effectiveDate": ISODate("2022-01-01"), "insuredState": "NY" },**

**{ "claimNumber": "C007", "lineOfBusiness": "Life Insurance", "insuredFullName": "John Doe", "insuredFName": "John", "insuredLName": "Doe", "insuredDOB": ISODate("1989-05-15"), "dateOfDeath": ISODate("2015-02-10"), "Status": "Pending", "effectiveDate": ISODate("2022-01-01"), "insuredState": "NY" },**

**{ "claimNumber": "C008", "lineOfBusiness": "Life Insurance", "insuredFullName": "John Doe", "insuredFName": "John", "insuredLName": "Doe", "insuredDOB": ISODate("1989-05-15"), "dateOfDeath": ISODate("2024-02-10"), "Status": "Pending", "effectiveDate": ISODate("2022-01-01"), "insuredState": "NY" },**

**{ "claimNumber": "C009", "lineOfBusiness": "Life Insurance", "insuredFullName": "John Doe", "insuredFName": "John", "insuredLName": "Doe", "insuredDOB": ISODate("1989-05-15"), "dateOfDeath": ISODate("2020-02-10"), "Status": "Pending", "effectiveDate": ISODate("2022-01-01"), "insuredState": "NY" },**

**{ "claimNumber": "C0010", "lineOfBusiness": "Life Insurance", "insuredFullName": "John Doe", "insuredFName": "John", "insuredLName": "Doe", "insuredDOB": ISODate("1989-05-15"), "dateOfDeath": ISODate("2022-02-10"), "Status": "Pending", "effectiveDate": ISODate("2022-01-01"), "insuredState": "NY" },**

**])**

----------------------------------------------------------------------------------------------------------------------------

(3) Find record from policyHolders

1. All policyholders info

**ANS : - db.policyHolders.find({}).pretty()**

1. Find policy holder by insuredFirstName

**ANS: - db.policyHolders.find({"insuredFirstName": "insuredFirstName3"}).pretty()**

1. Find policy holder by Status

**ANS:- db.policyHolders.find({"Status": "Active"})**

1. Find policy holder with date range (From date to Date)

**ANS: - db.policyHolders.find({"DateOfBirth": {"$gte": ISODate("1980-01-01"), "$lte": ISODate("2000-01-01")}})**

----------------------------------------------------------------------------------------------------------------------------

(3) Find record from claims

1. All claims info

**ANS:- db.claims.find({})**

1. Find claims by claimNumber

**ANS:- db.claims.find({"claimNumber": "C001"})**

1. Find claims by insuredDOB

**ANS:- db.claims.find({"insuredDOB": ISODate("1989-05-15")})**

1. Find claims with dateOfDeath (From date to Date)

**Ans: db.claims.find({"dateOfDeath": {"$gte": ISODate("2022-01-01"), "$lte": ISODate("2022-12-31")}})**

----------------------------------------------------------------------------------------------------------------------------

(4) Update policyholders with at least 3 parameters separately

**ANS:-**

**db.policyHolders.updateOne({"insuredFirstName": "insuredFirstName1"}, {"$set": {"Status": "Inactive"}})**

**db.policyHolders.updateOne({"insuredFirstName": "insuredFirstName1"}, {"$set": {"PhoneNumber": "987-654-3210"}})**

**db.policyHolders.updateOne({"insuredFirstName": "insuredFirstName1"}, {"$push": {"Beneficiaries": "Charlie"}})**

----------------------------------------------------------------------------------------------------------------------------

(4) Update claims with at least 3 parameters separately

**ANS:-**

**db.claims.updateOne({"claimNumber": "C001"}, {"$set": {"Status": "Approved"}})**

**db.claims.updateOne({"claimNumber": "C001"}, {"$set": {"effectiveDate": ISODate("2022-02-01")}})**

**db.claims.updateOne({"claimNumber": "C001"}, {"$set": {"insuredState": "CA"}})**

----------------------------------------------------------------------------------------------------------------------------

(6) Delete the PolicyHolder by name starting with letter A.

**ANS- db.policyHolders.deleteMany({"insuredFirstName": /^A/})**

----------------------------------------------------------------------------------------------------------------------------

(7) Delete a claim which was created on a particular date.

**ANS:- db.claims.deleteOne({"effectiveDate": ISODate("2022-01-01")})**

----------------------------------------------------------------------------------------------------------------------------

(8) Delete all from PolicyHolders collection who have age greater than “ 80”

**ANS:- db.policyHolders.deleteMany({"Age": {"$gt": 80}})**

----------------------------------------------------------------------------------------------------------------------------

(9) Update the PolicyHolders Relationship to Married where Age is > 25.

**ANS:- db.policyHolders.updateMany({"Age": {"$gt": 25}}, {"$set": {"Relationship": "Married"}})**

----------------------------------------------------------------------------------------------------------------------------

(10) Update the claims where dateOfDeath is 1 year before the current date

**ANS:-**

**db.claims.updateMany({**

**"dateOfDeath": {**

**"$lte": new Date(new Date().setFullYear(new Date().getFullYear() - 1))**

**}},**

**{"$set": {"Status": "Expired"}}**

**)**

----------------------------------------------------------------------------------------------------------------------------

JSON:

----------------------------------------------------------------------------------------------------------------------------

Create a JSON format report for

1. policyHolders and
2. claims

----------------------------------------------------------------------------------------------------------------------------

XML:

----------------------------------------------------------------------------------------------------------------------------

Create a XML format report for

1. policyHolders and
2. claims

----------------------------------------------------------------------------------------------------------------------------

API:

----------------------------------------------------------------------------------------------------------------------------

Create API for policyHolders and claims includes the following methods:

GET,

POST,

PUT And

Delete

Consider the Domain: <https://abctestapi.com/>

For POST show the payload

For GET show the params used

Share your result/response in JSON format

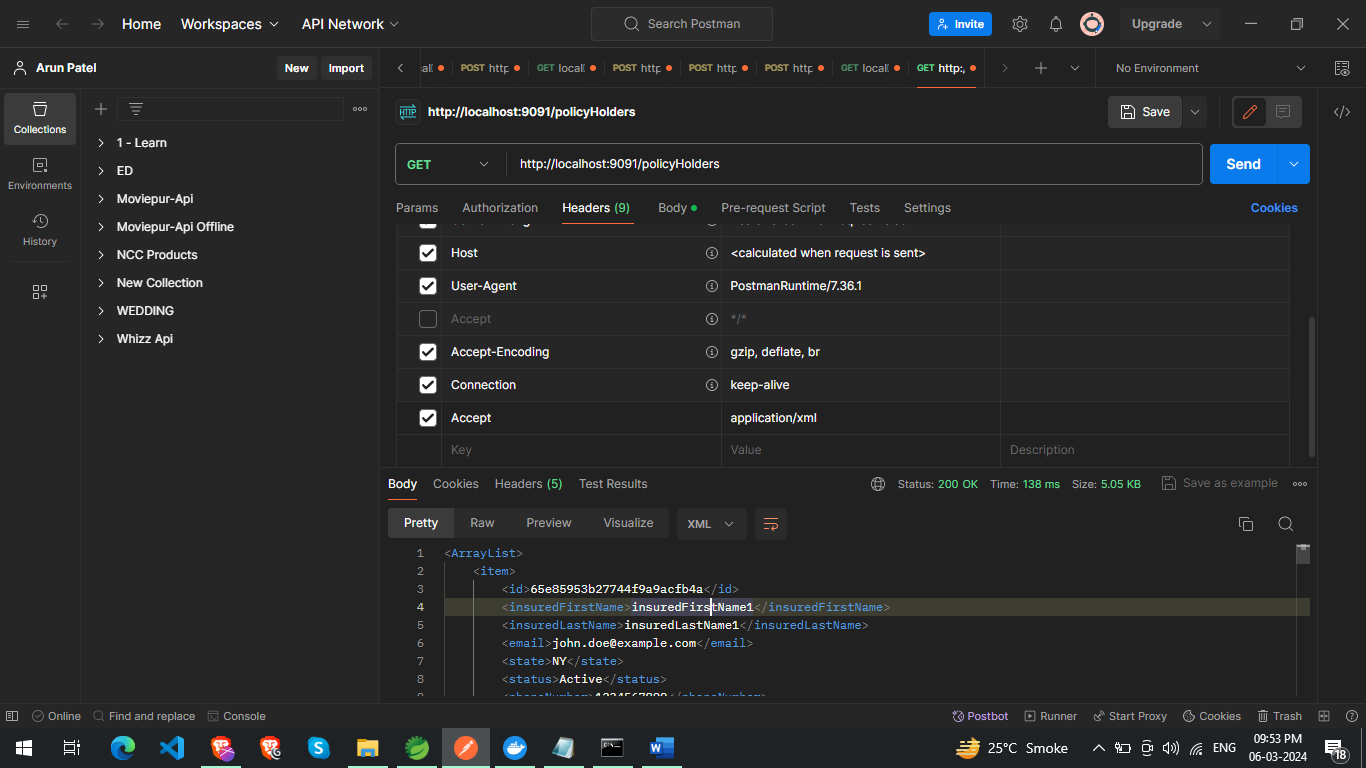
**===================================**

By defult api return type is json

if you want XML response then pass

**Accept = application/xml**

in header



=============claims Api============

GET <== Find All ==> http://localhost:9091/claims

GET <== Find By Id ==> http://localhost:9091/claims/{id}

GET <== Find By Claim Number ==> http://localhost:9091/claims/findByClaimNumber/{claimNumber}

POST <== Create New => http://localhost:9091/claims

{

"claimNumber": "C002",

"lineOfBusiness": "Life Insurance Add",

"insuredFullName": "John Doe add",

"insuredFName": "John add",

"insuredLName": "Doe aaa",

"insuredDOB": "1989-05-15T00:00:00.000+00:00",

"dateOfDeath": "2022-02-10T00:00:00.000+00:00",

"effectiveDate": "2022-01-01T00:00:00.000+00:00",

"insuredState": "NY",

"status": "Expired"

}

PUT <====== Update claimNumber and dateOfDeath === > http://localhost:9091/claims/{id}

{

"claimNumber": "C0010",

"dateOfDeath": "2020-02-10T00:00:00.000+00:00"

}

DELETE <== Delete By Id ==> http://localhost:9091/claims/{id}

DELETE <== Delete All ==> http://localhost:9091/claims/deleteAll

=============policyHolders api============

GET <== Find All ==> http://localhost:9091/policyHolders

GET <== Find By Id ==> http://localhost:9091/policyHolders/{id}

POST <== Create New => http://localhost:9091/policyHolders

{

"insuredFirstName": "insuredFirstName new new",

"insuredLastName": "insuredLastName new new",

"email": "john.doe@example.com new new",

"state": "NY",

"status": "Active",

"phoneNumber": "1234567890",

"dateOfBirth": "1989-05-15",

"age": 20,

"beneficiaries": [

"Alice",

"Bob"

],

"relationship": "Unknown"

}

PUT <====== Update email, state, status, phoneNumber === > http://localhost:9091/policyHolders/{id}

{

"email": "john.doe@example.com new new",

"state": "NY",

"status": "Active",

"phoneNumber": "1234567890"

}

DELETE <== Delete By Id ==> http://localhost:9091/policyHolders/{id}

DELETE <== Delete All ==> http://localhost:9091/policyHolders/deleteAll