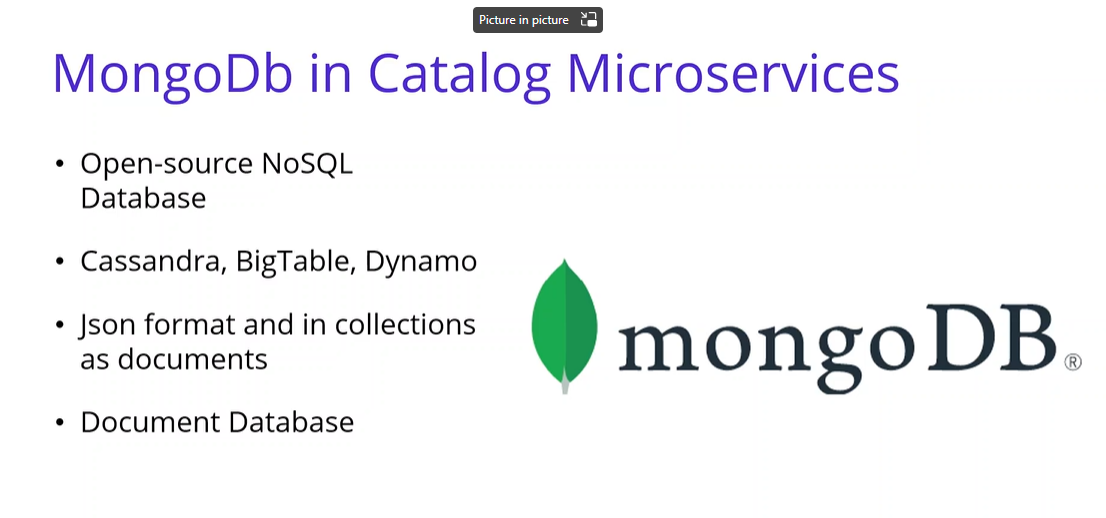
**Develop Micro services With Mongo DB**

**Notes:-**



**We see that mongoDb is an open data source that is designed for easy development and it’s the data on JSON format and its contains collection and inside documents**

**Collection 🡪 tables**

**Document 🡪 rows**

**Steps**

**1-we go to docker hub and install docker**

**2-we install docker and search for mongo and then install mongo**

**docker pull mongo //it will create image for mongo**

**docker ps //to get list of containers**

**docker images //to get list of images**

**//to create new instance of image mongo on port 27017 based on mongoDb image**

**docker run -d -p 27017:27017 --name shopping-mongo mongo**

**//to get basic info of the container shopping-mongo**

**docker logs -f shopping-mongo**

**//to get access to internal mongoDb bash portal to execute commands**

**docker exec -it shopping-mongo /bin/bash**

**(We can create collection, edit collection, list collection, etc…)**

**show dbs //it will get list of databases**

**use CatalogDb //it will switch and create new database CatalogDb if not exist**

**db.createCollection('Products') //it will create collection called Products**

**//it will create documents inside this collection called Products**

**db.Products.insertMany([{ 'Name':'Asus Laptop','Category':'Computers', 'Summary':'Summary', 'Description':'Description', 'ImageFile':'ImageFile', 'Price':54.93 }, { 'Name':'HP Laptop','Category':'Computers', 'Summary':'Summary', 'Description':'Description', 'ImageFile':'ImageFile', 'Price':88.93 } ])**

**//it will get list of products**

**db.Products.find({}).pretty()**

**//it will get products based on id**

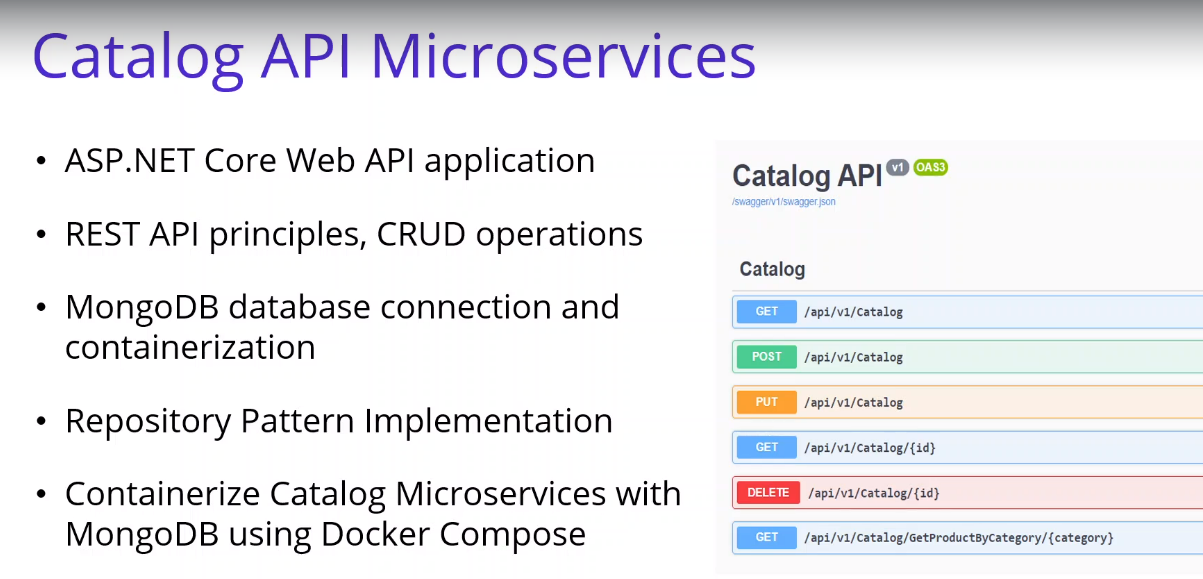
**db.Products.find({\_id : ObjectId("62c0f4e3b5de41af2ec82d2f")}).pretty()**

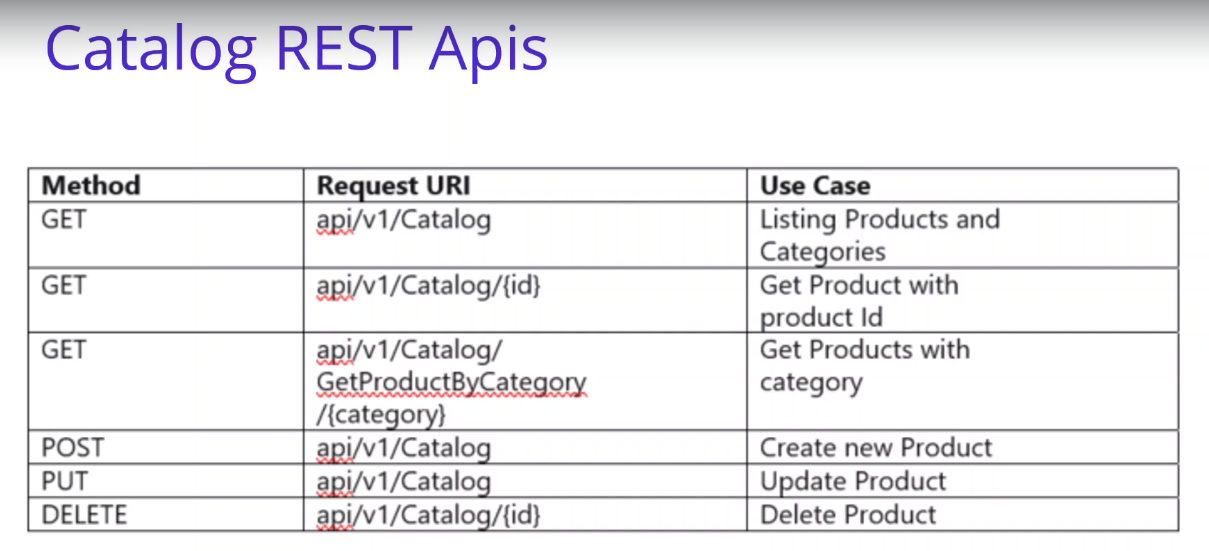
**//it will remove all documents inside collection products**

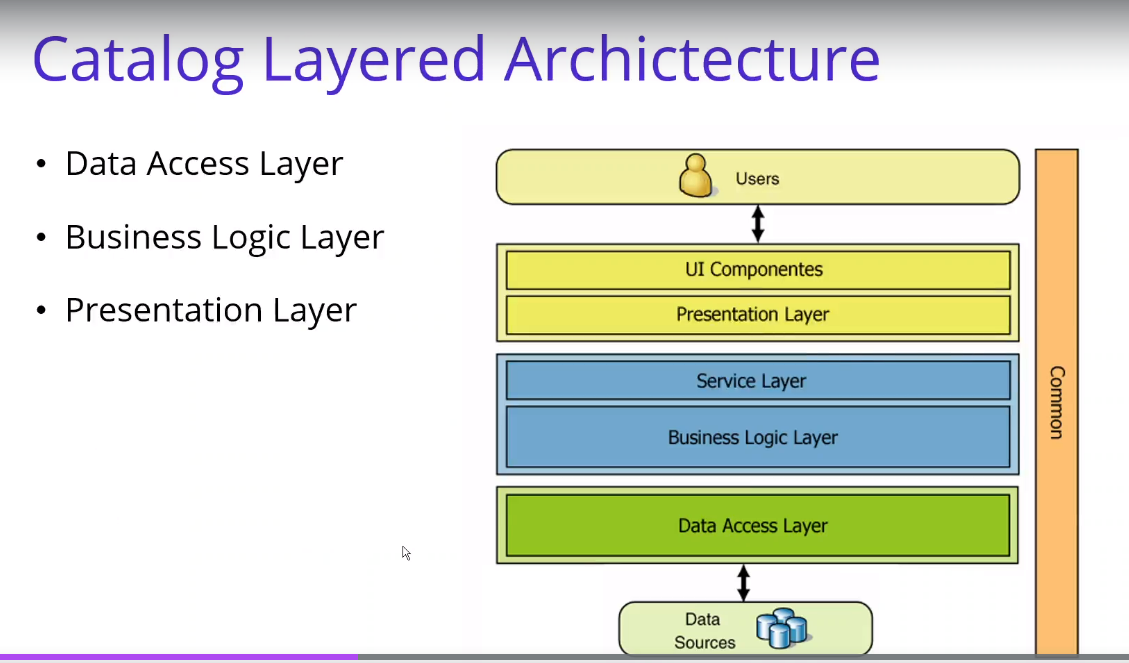
**db.Products.remove({})**

**//it will show product collections**

**show collections**

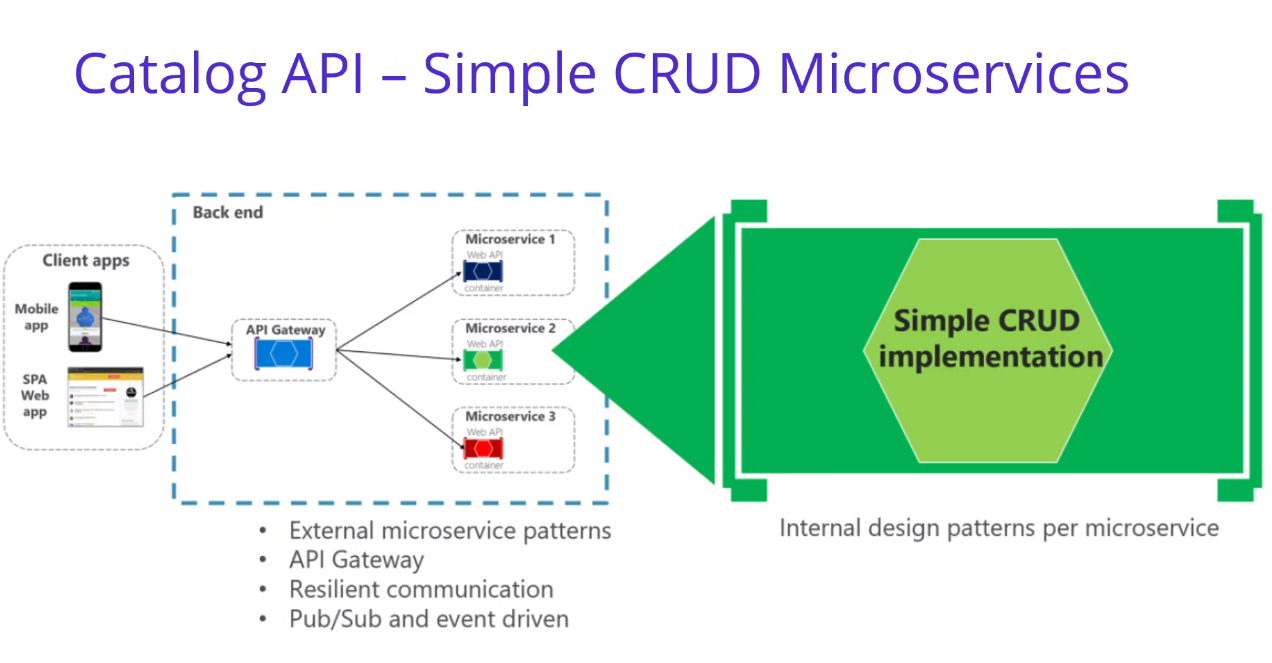




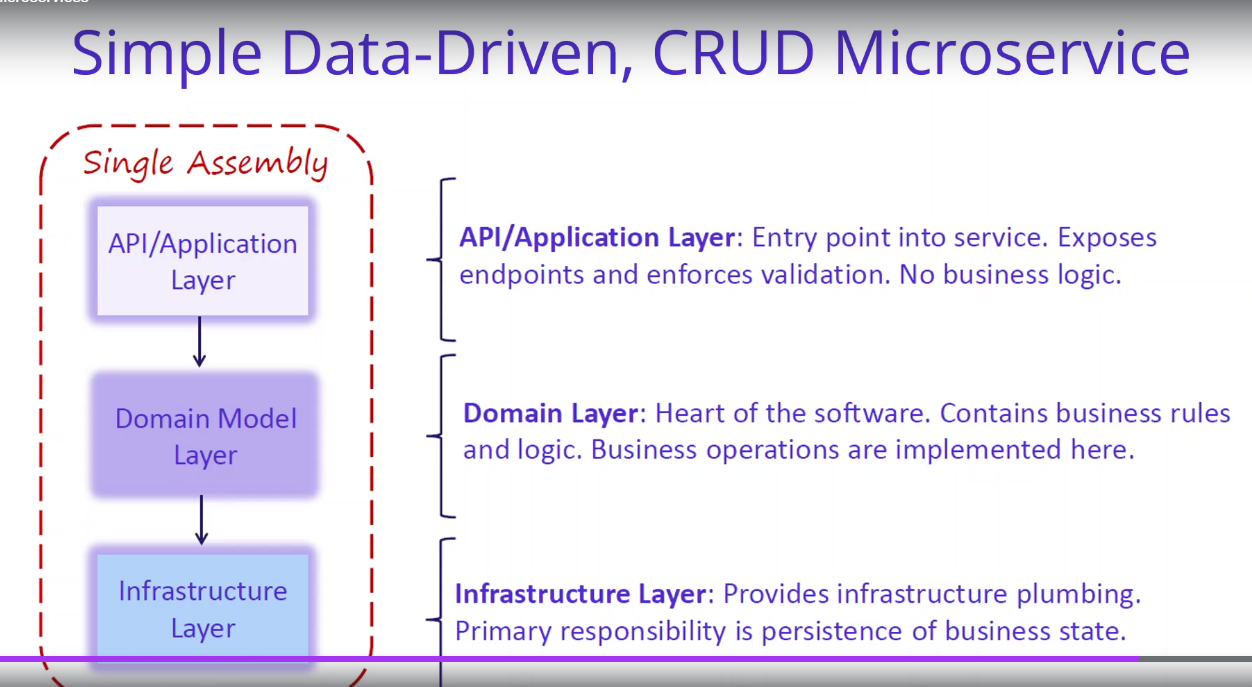


**We use N-layer architecture (motholonic architecture) (old structure traditional)**

**Micro services**



**We use clean micro service which contains API gateway which is responsible to handle request and detect the target micro service it target and publish message and on the target micro service it will subscribe the message and consume and execute it**

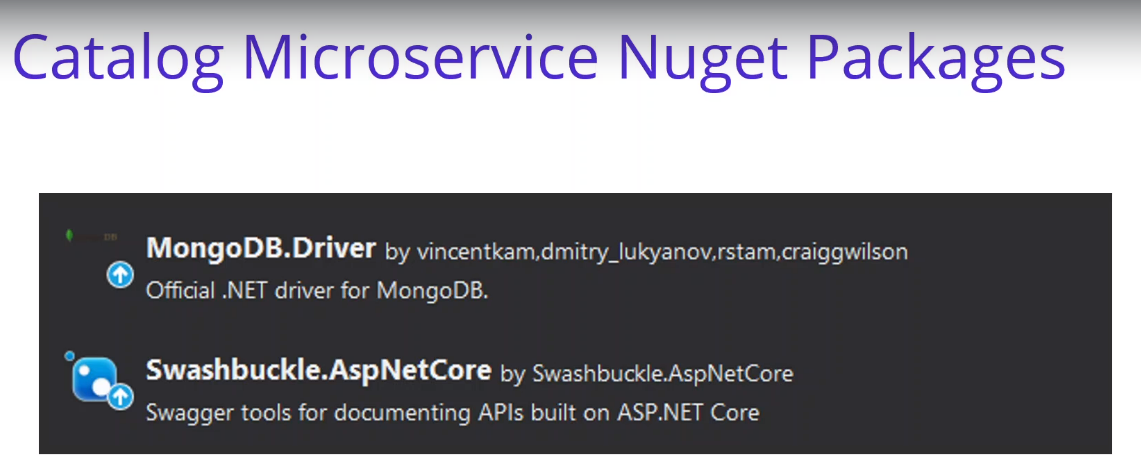


**We are using clean architecture**

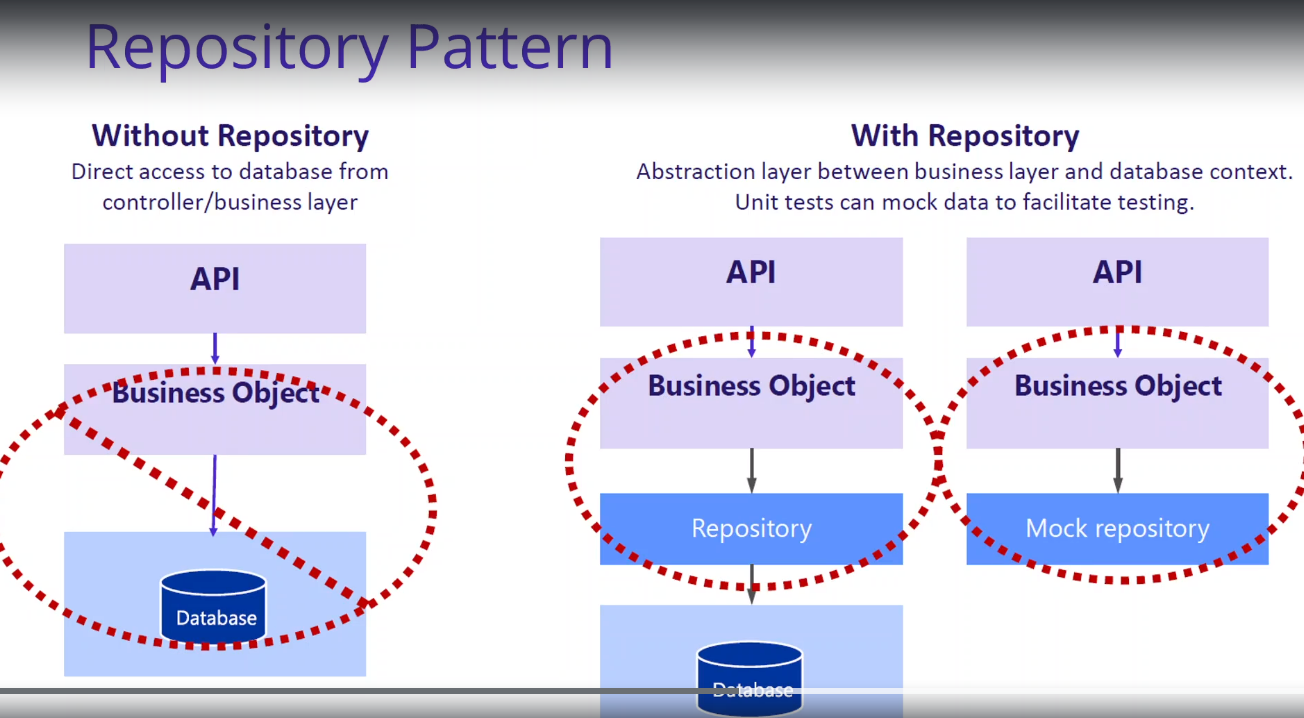
**Within it the API Layer only contains the endpoints and talking with Domain layer within CQRS**

**Domain layer contains the business rules and logic**

**Infrastructure layer responsible to communicate with database**



**Repository Pattern**



**1-Repository pattern is an abstraction between the business layer and database context layer (Middle layer between the above 2 layers).**

**2-Centerline the domain object make easy to maintain.**

**3-testing code is in easier way.**

**4-reducre application of code.**

**5-controller don’t care about what database you are using**

**Docker run –d –p 27017:27017 –name shopping-mongo mongo**

**Docker ps //list the Docker working containers**

**3fe91a216514 mongo "docker-entrypoint.s…" 4 hours ago Up 4 hours 0.0.0.0:27017->27017/tcp shopping-mongo**

**Docker stop 3fe9 //stop the Docker container**

**Docker start 3fe9 //start the Docker container**