→ L4 (Sde 2) → Coinbase] → LLD

↳ DSA Tist] → 4 ques

↳ Online MCQ

↳ interrelated

Easy

X Q1 → ✓ 250 1-3 f

Q2 x

Q3 x x

Q4 x

1000

700 f

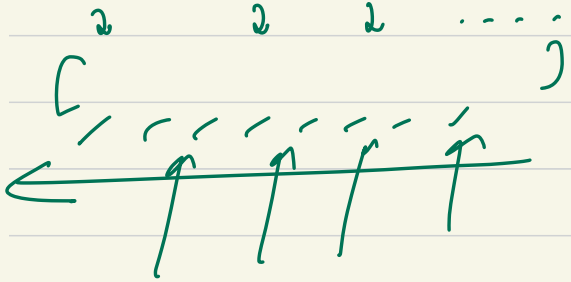
500 x

DSA

DSA + LLD

HM X

Q: You have a consumer, which is consuming the data from a data stream (assume it as a list) We have to read the data one by one. How to do it??

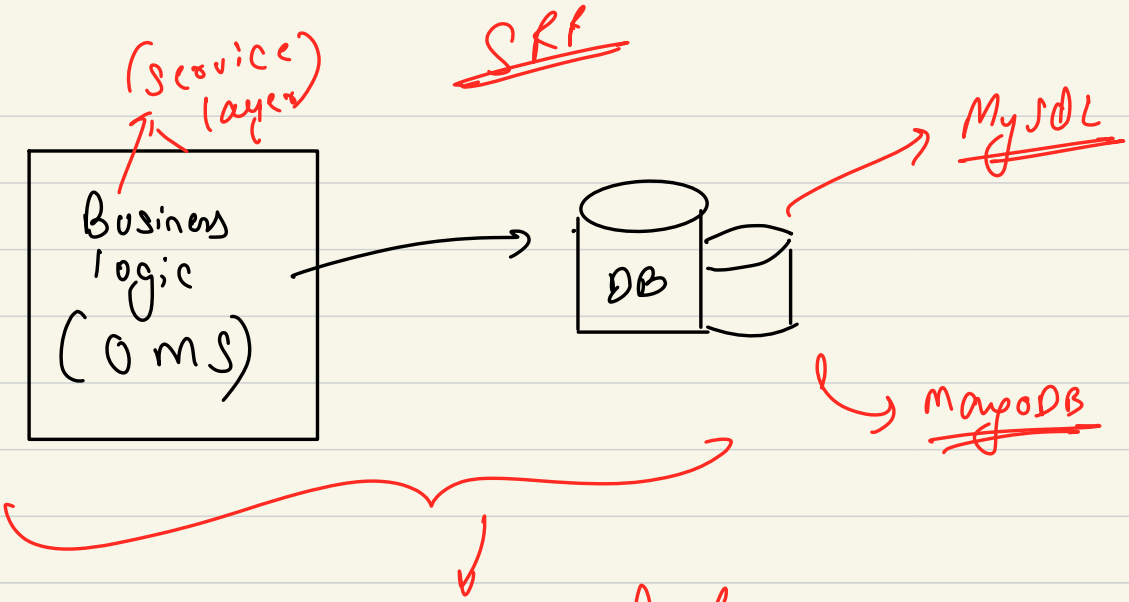


Paginated APIs

→ extendibility

→ limit, offset

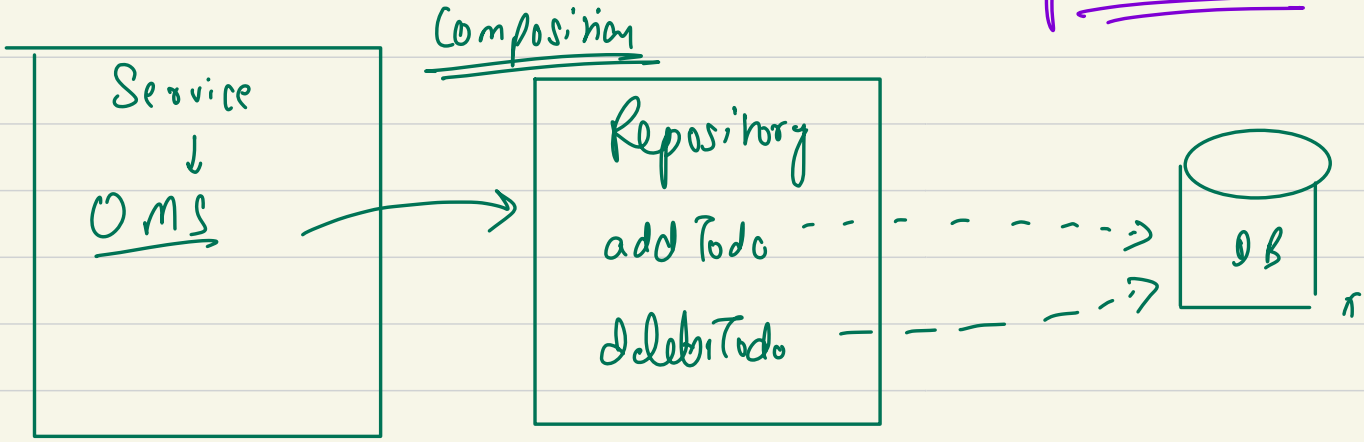
limit	offset	
10	0	→ first 10 records
10	10	→ next 10 records



Not a good practice

Repository Pattern → DAO (data access object)

Repository classes



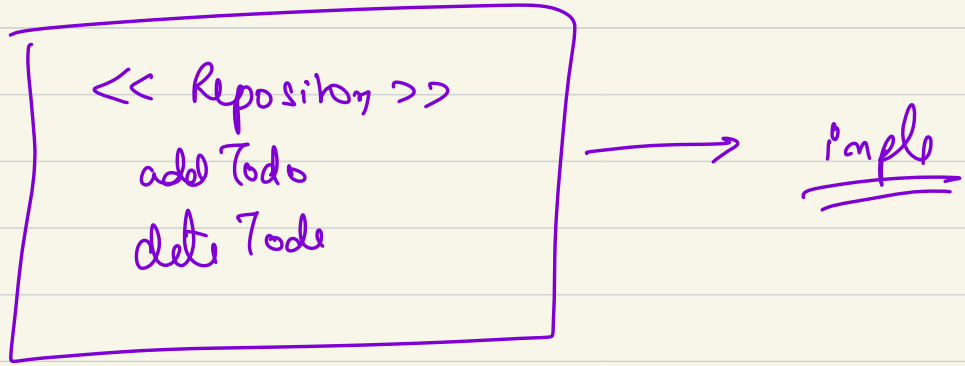
for any db operation, service layer will not call db directly instead call the repository class.

class OMS {

private final MySQLRepository; → ??

}

DIP X



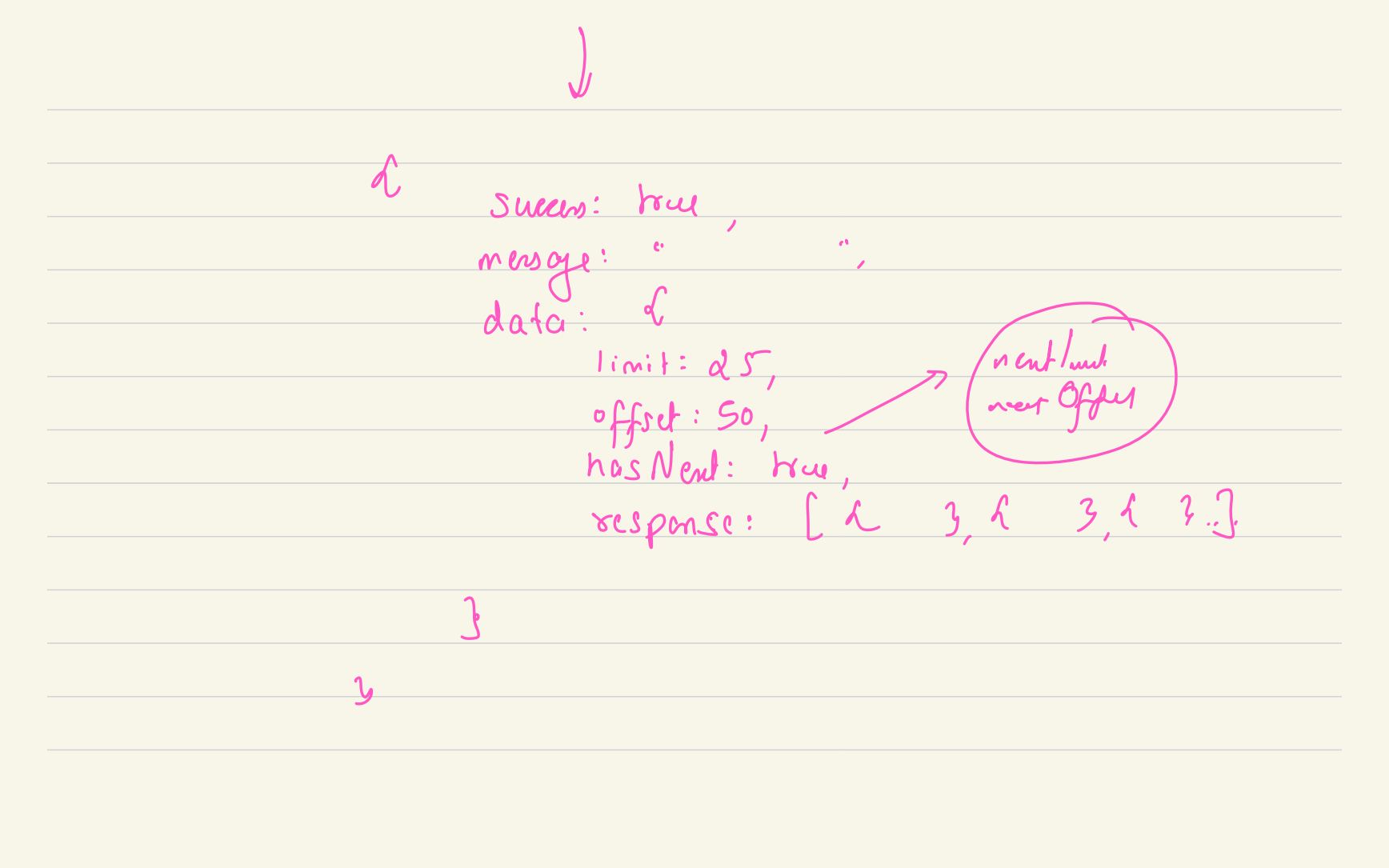
```
class OMS {

    private final Reposiroy r;

}
```

→ Cut → How can we improve?? → lacks

/api/v1/products ? limit = 25 & offset = 50

Response →

```
{
    success: true
    message: "_____"
    data: [ { ___ },
            { ___ },
              ⋮
            { ___ },
    ]
}
```


{
success: true,
message: "",
data: {
limit: 25,
offset: 50,
hasNext: true,
response: [{}, {}, {}...]
next/next page offset
}
}

Iterator design pattern

```
interface Iterator {

    boolean hasNext();
    List<String> next();

}

class SequentialIterator implements Iterator {

}
```

class RandomizedIterator implements Iterator {

}

class JumperIterator implements Iterator {

}

How do we know if a structure supports iterators?

How can we force a structure to support Iterators?

```
class LinkedList implement Iterable {

}

interface Iterable {
    Iterator getIterator();
}
```