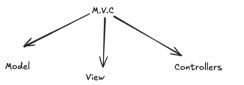
Agenda:

- MVC architecture
- How MVC is implemented in prod grade soft
- Controllers, Services, Routers, Repositories, DTO's, etc

MVC Architecture

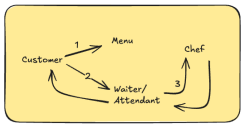


Nick names

waiter <-----> controller

chef <----> Model

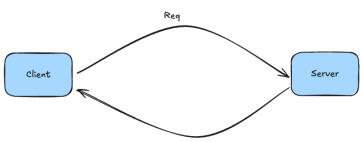
Menu <-----> View ——> No need to bother about it



Separation of concerns

Restaurant

1. We take a look at the menu
2. The waiter/attendant collects the order details from us
3. The waiter is responsible to take the order to the chef and then chef preps the order



Mongo

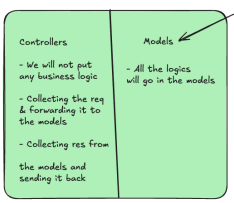
Redis

Doesn't talk about how the server process the req

We need a mechanism to explain the internal processing

MVC is one way in which we can define the internal working of a server

We are going to write the code in a way



Business Logics

HSBC Bank

Sanket wants to send money to Saurav (not a cross bank txn)

1. Check if Sanket has a valid working account or not
2. Check if Saurav has a valid working acc or not
3. Check if Sanket has enough money in the bank acc or not
4. Check if the details provided by sanket (CVV, OTP) are they correct or not

.....

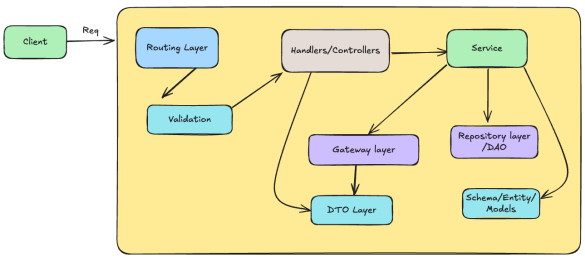
- MVC is over simplistic

Spring boot, Dot net, Ruby on rails, Django, Laravel etc

Q -> How work is done in the industry ?

-> In actual projects, we divide the controllers and models into further smaller layers with smaller responsibilities

Server



DAO -> data access object

emailId
subject
content

/persistence

/repositories
/entities

Schema is the representation of data in the datastore

1. Routing layer identifies, which controller should be triggered for the incoming req: Spring-> Dispatcher Servelet

We will talk about it later

So routing layer collects the req and forwards it to the further processing to be done

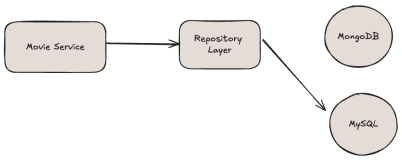
2. Handlers/Controllers -> Their responsibility is to take the incoming req, forward to the business logic i.e. Service layer, and then wait for the processing in that.Once service layer has done the processing then it takes the response from service layer and send it back.

Apart from this we can give controller the responsibility to prepare the response structure.

```
{
   data: {
      name: "Sanket"
   },
   success: true,
   message: "Name fetched successfully"
}
```

3. Service layer-> this is responsible for all the business logic calculations and processing. Service layer is not responsible for communicating to the data storage or 3rd party apis.

select * from movies;



Repository pattern : Direct application of DIP

```
Class MovieService {

    private MovieRepository sqlrepo;
    private MovieRepository nosqlrepo;

    MovieService(MovieRepository _sqlrepo, ...) {
          this.sqlrepo = _sqlrepo;
          ....
    }

}
```

```
interface MovieRepository {

    Movie createMovie(....);

    List<Movie> getAllMovies();

    void deleteMovie(Integer id);

}
```

```
class MySQLMovieRepo implements  MovieRepository { ... }
```

```
class MongoMovieRepo implements  MovieRepository { ... }
```

```
MovieService s = new MovieService(new MongoMovieRepo());
```

We need to define structures of data which might be sent from our system to third part and from third party to our's

DTO -> Data transfer object

```
Class MyMovieService {

    private MovieRepo repo;
    private MovieGateway gateway;

    void fetchMovies(string name) {

        List<Movie> movies = repo.findMovies(name);

        if(movies.size() == 0) {

            List<OMDBFetchMovieResponseDTO> newmovies = gateway.findMovies(name);// 3rd party api call

            repo.saveAllMovies(newmovies);


        }

        return ..

    }


}
```

```
class OMDBFetchMovieRequestDTO {

    private string title;
    ....
}
```

```
class OMDBFetchMovieResponseDTO {

    private string title;
private string year;
private string rated;
    ....
}
```

```
class Movie {
    // entity

    private string title;
    private string year;
    private string rated;
    private Date createdAt;


}
```

```
interface  MovieGateway {

    List<OMDBFetchMovieResponseDTO> fetchMovies;

}

class OMDBMovieGateway implements MovieGateway {


}
```