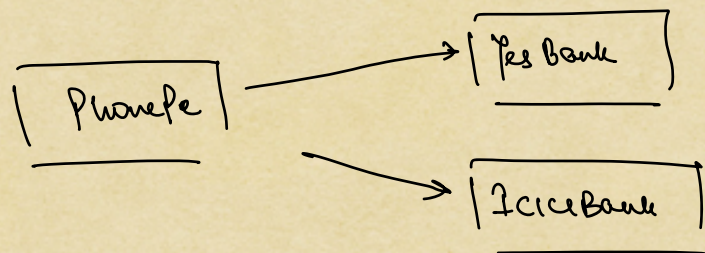# AGENDA

1. Adapter pattern with Client

2. Complete all REST APIs

3. Implement Exception Handler with @ControllerAdvice

4. Config Properties

5. Protocol Buffers or, ProtoBufs

=) <u>Adapter Design Pattern</u>

```
┌──────────┐         ┌──────────┐
│ PhonePe  │ ──────→ │ Yes Bank │
└──────────┘    ↘    └──────────┘
                 ↘   ┌──────────┐
                  ──→│ IcicBank │
                     └──────────┘
```

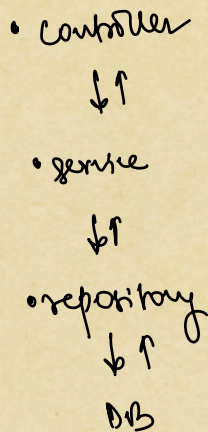* Adapter is used whenever we want to call a 3rd party.

* Service layers should hold only business logic
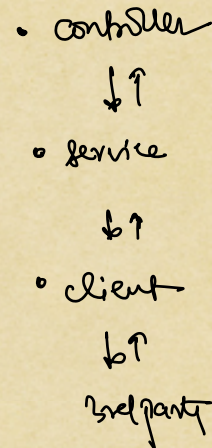
→ call to 3rd party in our service layer (X)

→ (products ⇒ (items ⇒ these kind of changes should not impact service layer

- Whenever we are calling a 3rd party for any operations we use a "client" class for that 3rd party.

- This client class has methods to do operations wrt 3rd party

- This client class acts as your adapter.

DB

- controller
  ↓↑
- service
  ↓↑
- repository
  ↓↑
  DB

3rd party

- controller
  ↓↑
- service
  ↓↑
- client
  ↓↑
  3rd party

⇒ Code flow :-
_____

ProductRequestDTO   FakestoreProductRepDTO

(API) → Controller ⟶ Service ⟶ APIClient ⟶ 3rd party
  ⟵   ⟵   ⟵   ↓ ↑   ⟵
                 ProductRespDTO   mapper   FakestoreProductRespDTO