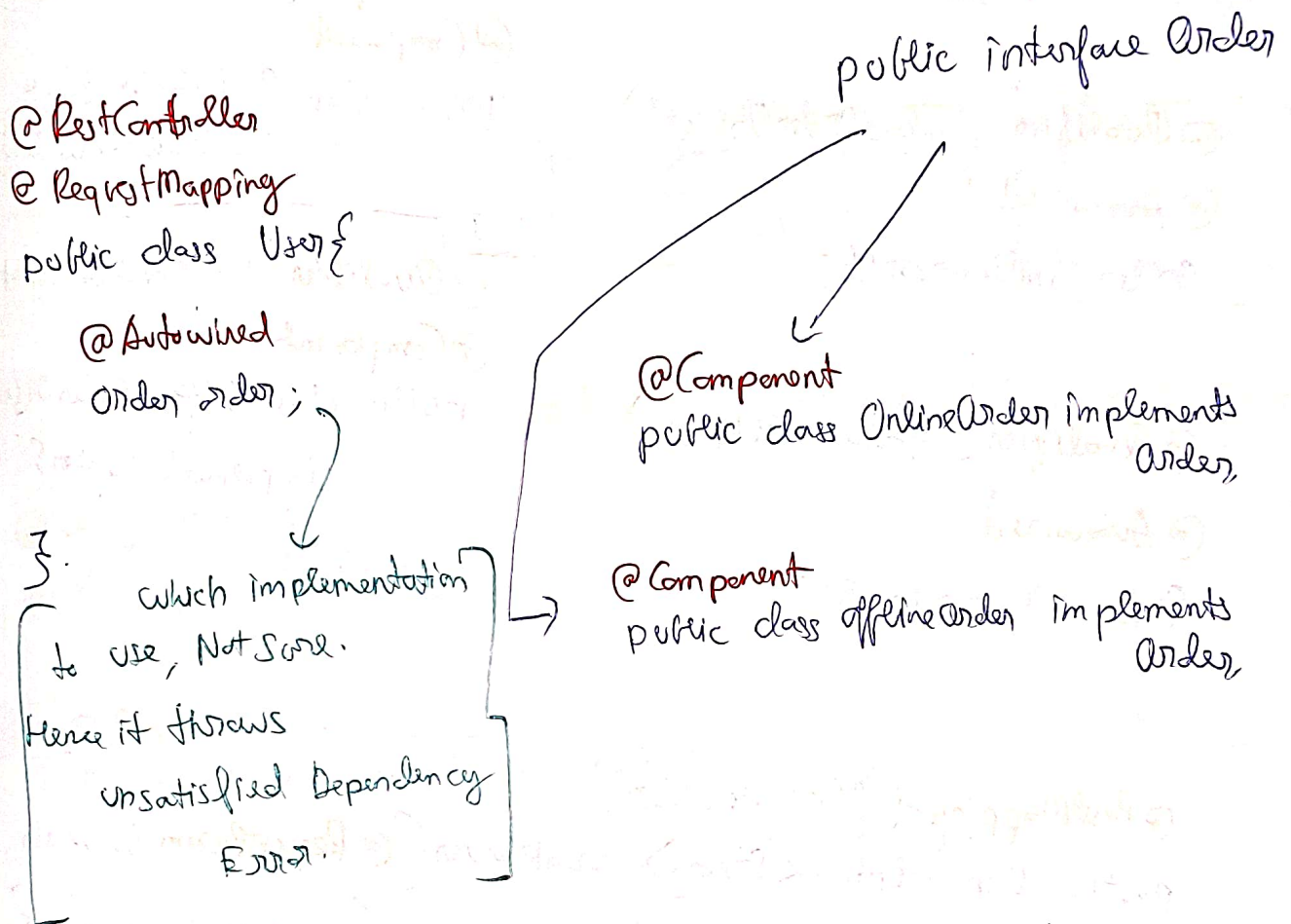


[Video 8. Spring Boot. Dynamically Initialised Beans | Value Annotation.]

Let's first remember Unsatisfied Dependency problem.



Now if you remember we resolved this using the @Qualifier annotation.

```
@Qualifier("onlineOrder")  
@Autowired
```

```
Order order;
```

But problem is that it is hardcoded. So it breaks Dependency Inversion.

```
@Qualifier("onlineOrder")
```

```
@Component  
public class OnlineOrder implements  
Order,
```

```
@Qualifier("offlineOrder")
```

```
@Component  
public class OfflineOrder implements  
Order,
```

1st Solution → To follow Dependency Inversion.

@ Component

public class User {

@Qualifier("onlineOrderObject")

@Autowired

Order onlineOrderObj;

@Qualifier("offlineOrderObject")

@Autowired,

Order offlineOrderObj;

@Qualifier("onlineOrderObject")

@Component

public class OnlineOrderObject
implements Order {

@Qualifier("offlineOrderObject")

@Component

public class OfflineOrderObject
implements Order {

@PostMapping("/createOrder")

public ResponseEntity<String> createOrder(@RequestParam boolean
isOnlineOrder) {

if (isOnlineOrder) {

onlineOrderObj.createOrder();

}

else {

offlineOrderObj.createOrder();

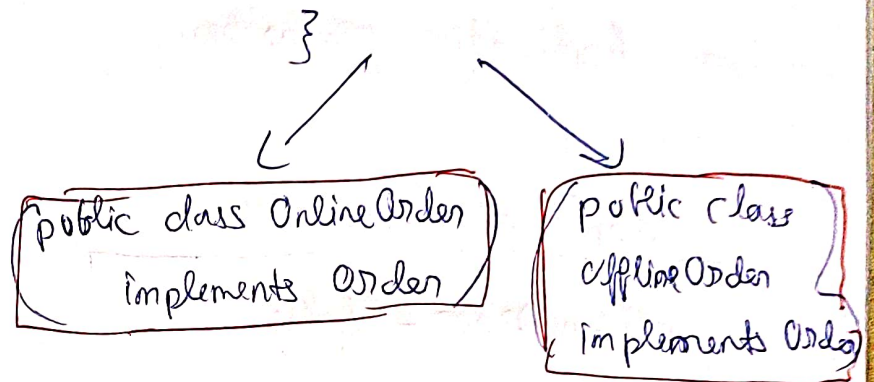
}

}

Using the
request Params
we are deciding
with Implementation
to use

2nd Solution : @Value annotation.

public interface Order {



Notice no @Component bean is there

@Configuration
public class AppConfig {

@Bean
public Order createOrderBean(@Value("\${isOnlineOrder}") boolean
isOnlineOrder) {

if (isOnlineOrder) {
 return new OnlineOrder();
}
else {
 return new OfflineOrder();
}

}

@Value: It is used to inject values from various sources like property variables or inline literals.

Like Application-properties.

application-properties

isOnlineOrder = false

Inline Literal Example

@Bean

```
public Order createOrderBean (@Value("false") boolean isOnlineOrder) {
```

```
    if (isOnlineOrder) {
```

```
        return new OnlineOrder();
```

```
    } else {
```

```
        return new OfflineOrder();
```

```
    }
```

```
}
```