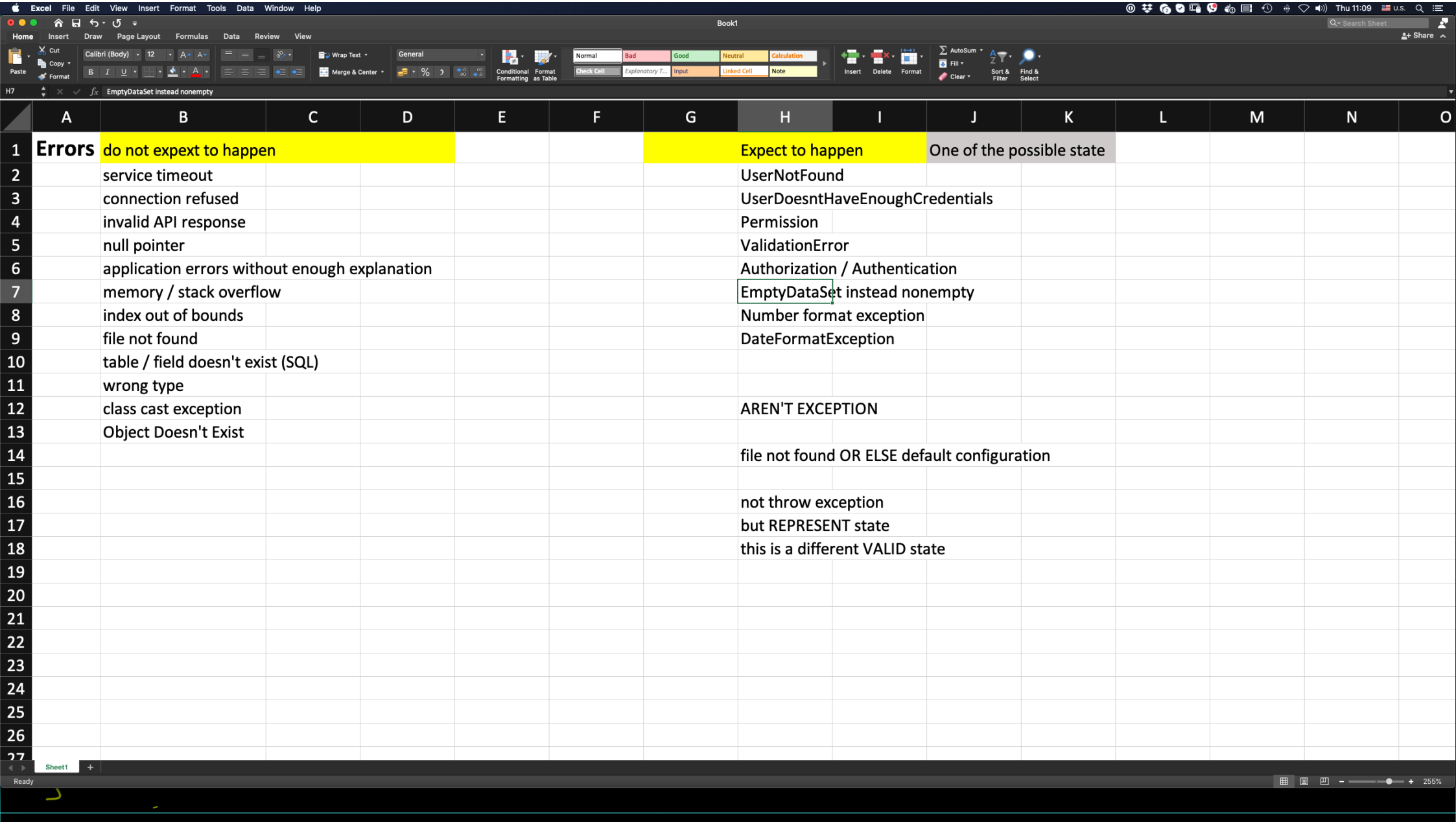

Excel File Edit View Insert Format Tools Data Window Help
Thu 11:09
Book1
Search Sheet
Home Insert Draw Page Layout Formulas Data Review View
Share
Paste Cut Copy Format
Calibri (Body) 12
Wrap Text
Merge & Center
General
Conditional Formatting
Format as Table
Normal Bad Good Neutral Calculation
Check Cell Explanatory T... Input Linked Cell Note
Insert Delete Format
AutoSum Fill Clear
Sort & Filter
Find & Select
H7
EmptyDataSet instead nonempty
Errors
do not expext to happen
service timeout
connection refused
invalid API response
null pointer
application errors without enough explanation
memory / stack overflow
index out of bounds
file not found
table / field doesn't exist (SQL)
wrong type
class cast exception
Object Doesn't Exist
Expect to happen
One of the possible state
UserNotFound
UserDoesntHaveEnoughCredentials
Permission
ValidationError
Authorization / Authentication
EmptyDataSet instead nonempty
Number format exception
DateFormatException
AREN'T EXCEPTION
file not found OR ELSE default configuration
not throw exception
but REPRESENT state
this is a different VALID state
Sheet1
Ready
255%

try {

getUser(id) → User

we keep box "closed" until we need it

Result Box

getUser(id)

User → happy path

None/Empty → exception / null / "magic value"

Optional<User>

if {
  if {

business idea

all the errors caught here

} catch (*) {
  Response 500
}

} }

HTTP Get /user/123 → user found
↘ user NotFound

200 + {id: 123, name: "Jim"}

404 → user not found
↘ url not found

it's not accurate

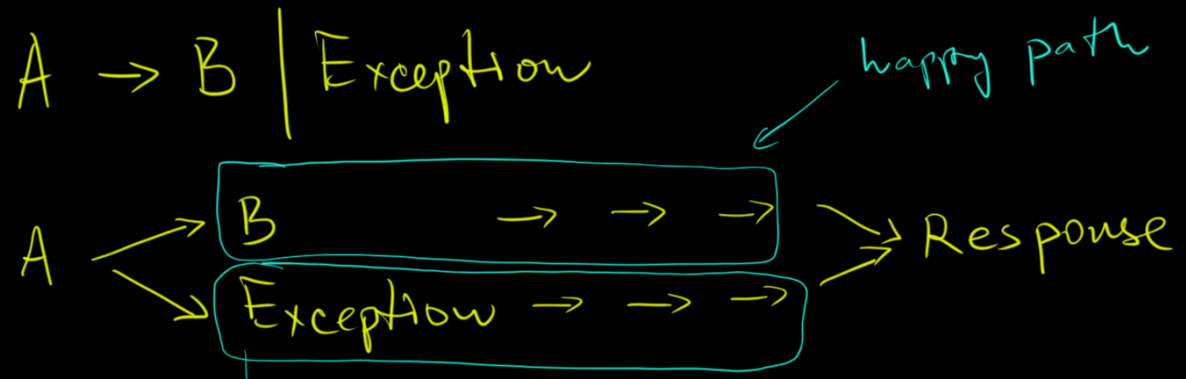
204 - user
404 - url
500 - critical error

service Layer
int => Option<User>

http level
Request => Response
no id => 404
int id => Optional<User> → Option.of(user) => 200 + {user}
↘ Option.empty => 204

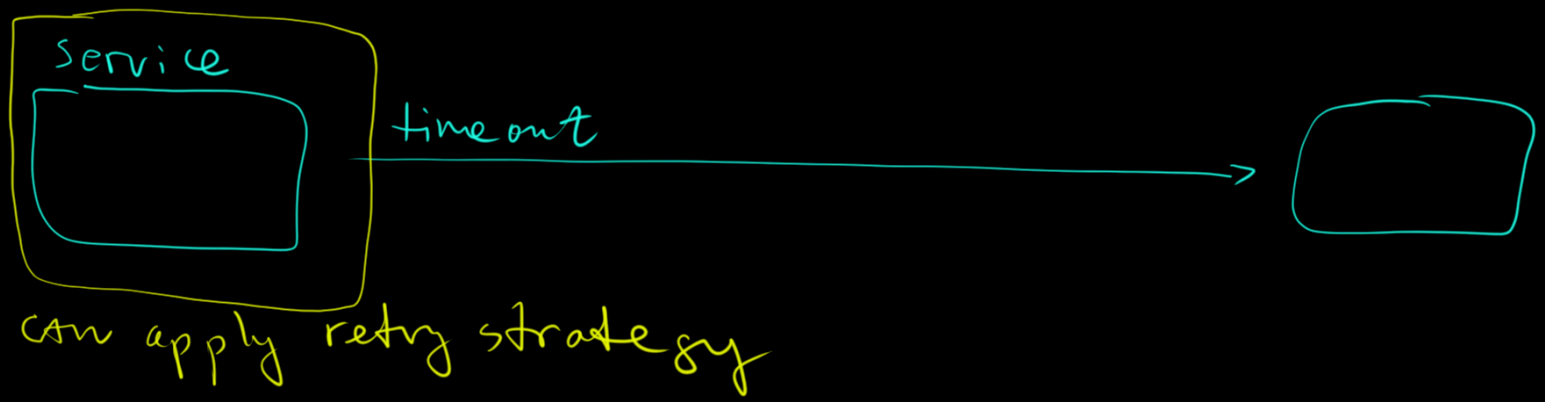
Not open the box "Optional" until you really need (http)

f: A → B | Exception

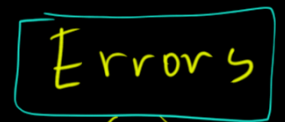


if we have scenario to handle
we treat as a VALID STATE
and handle IT (message, retry ...)

if we don't have scenario
we consider it I/O error
we don't handle it
we allow them to blow up.



can apply retry strategy



we don't expect to happen

we don't care

we EXPECT to happen (I know how to handle)

we HANDLE

Http layer

```
new *
@GetMapping
public ResponseEntity<?> handle(@RequestParam("id") int id) {
  Optional<User> maybeUser = ???; // gotten from the service layer
  return maybeUser
      .map(u -> ResponseEntity.ok(u))
      .orElse(ResponseEntity.status(204).build());
}
```

Optional<Int> ≡ List<Int>

Optional.empty ≡ empty list

Optional.of(33) ≡ list contains only ONE element

Errors we expect actually not errors

they are just another VALID state

At any cost don't use exceptions, REPRESENT THEM

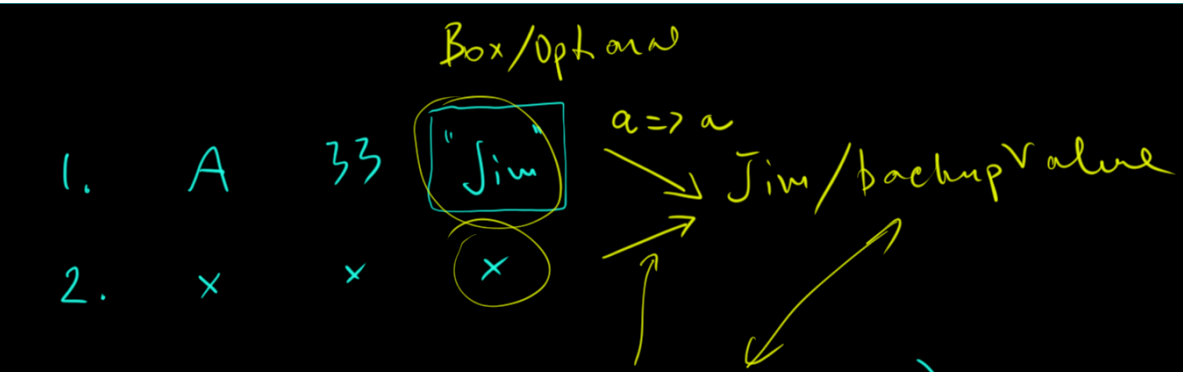
Optional<A> is one of the possible cases

---

the same exception

- on the one layer → you don't know how to handle
- on the higher layer → you MAY know how to handle.

Optional<A> is the way to DEFER solution place



Either.fold(Right → A,
Left → A)

Int => A
Exception => A