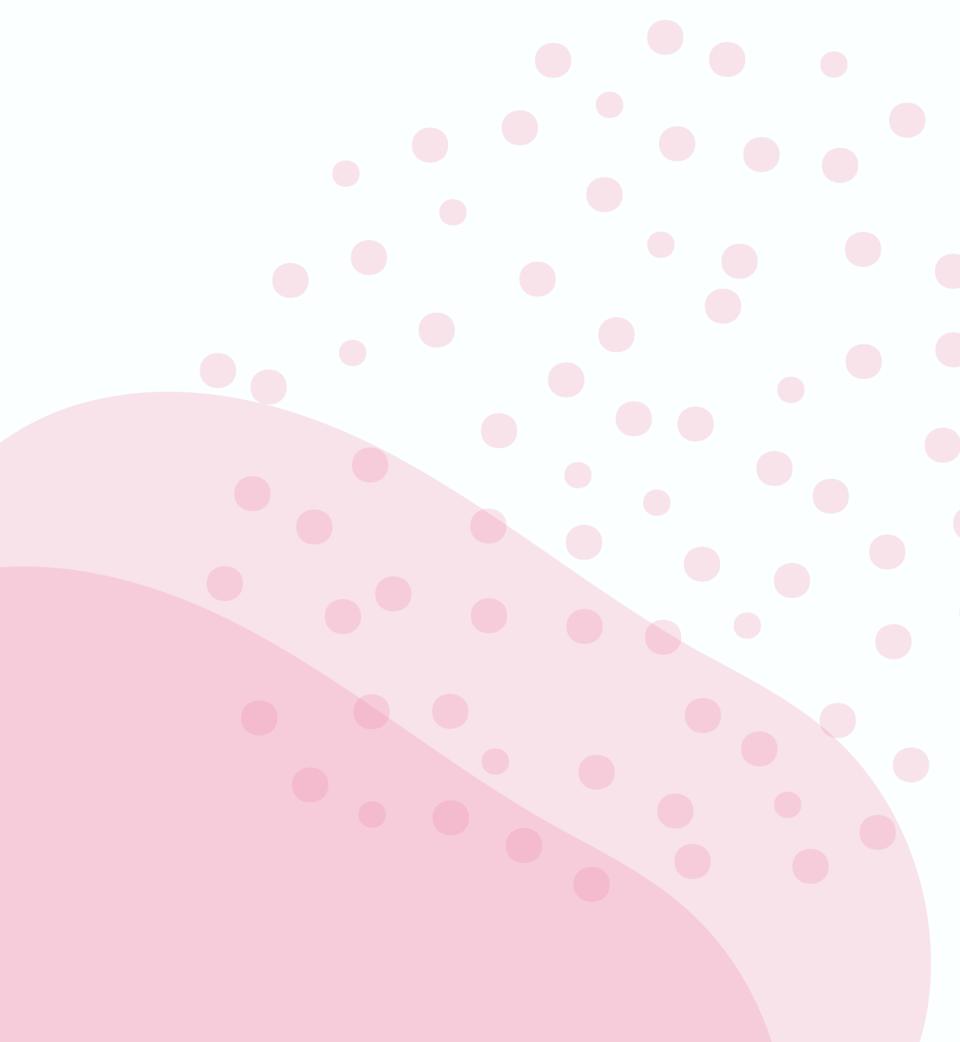


Mock DB for testing HTTP API in Go

BACKEND MASTER CLASS
LECTURE 13

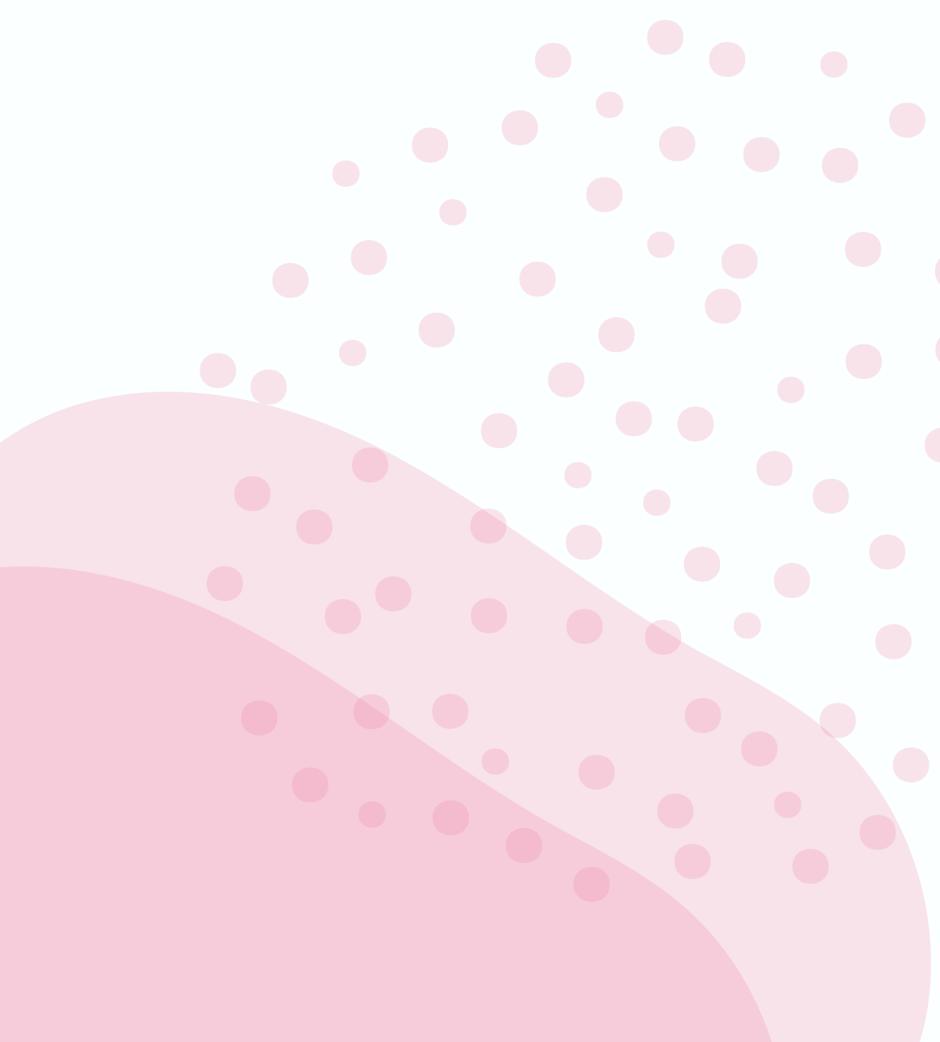
Why mock database?



Why mock database?

INDEPENDENT TESTS

Isolate tests data to avoid conflicts



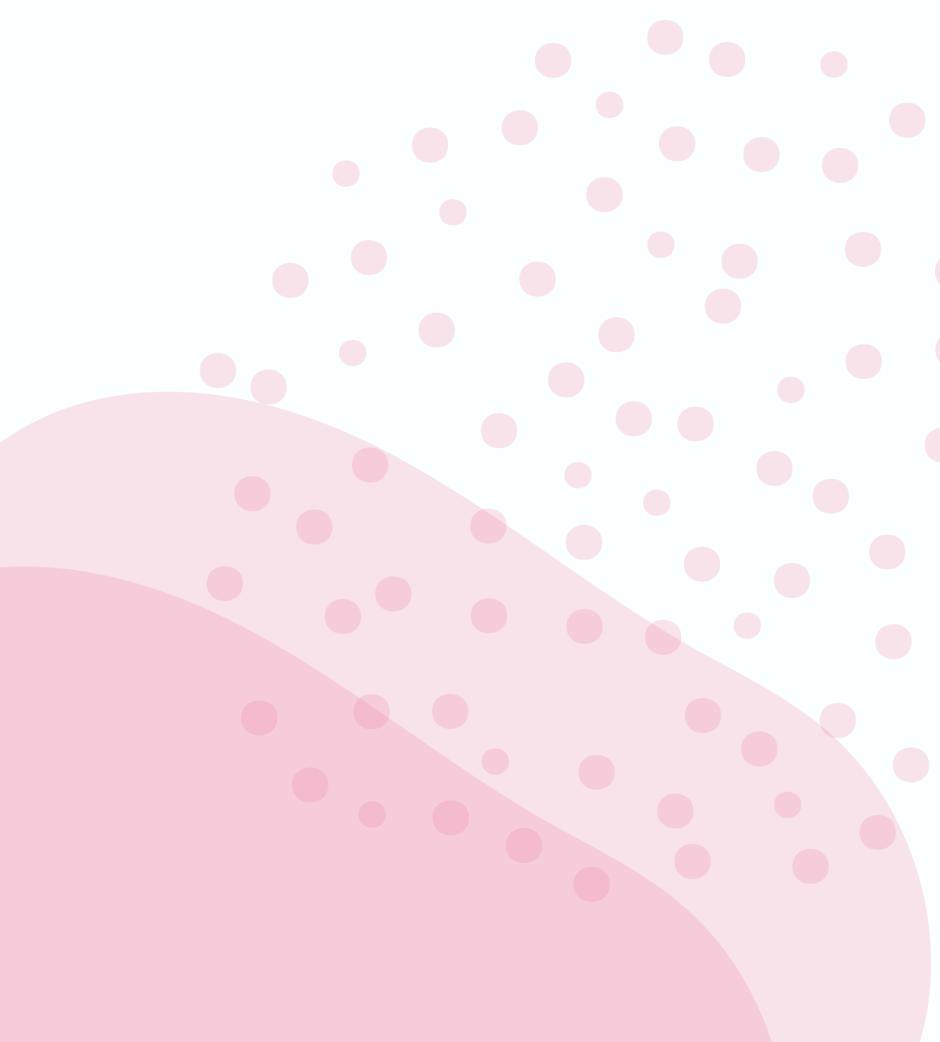
Why mock database?

INDEPENDENT TESTS

Isolate tests data to avoid conflicts

FASTER TESTS

Reduce a lot of time talking to the database



Why mock database?

INDEPENDENT TESTS

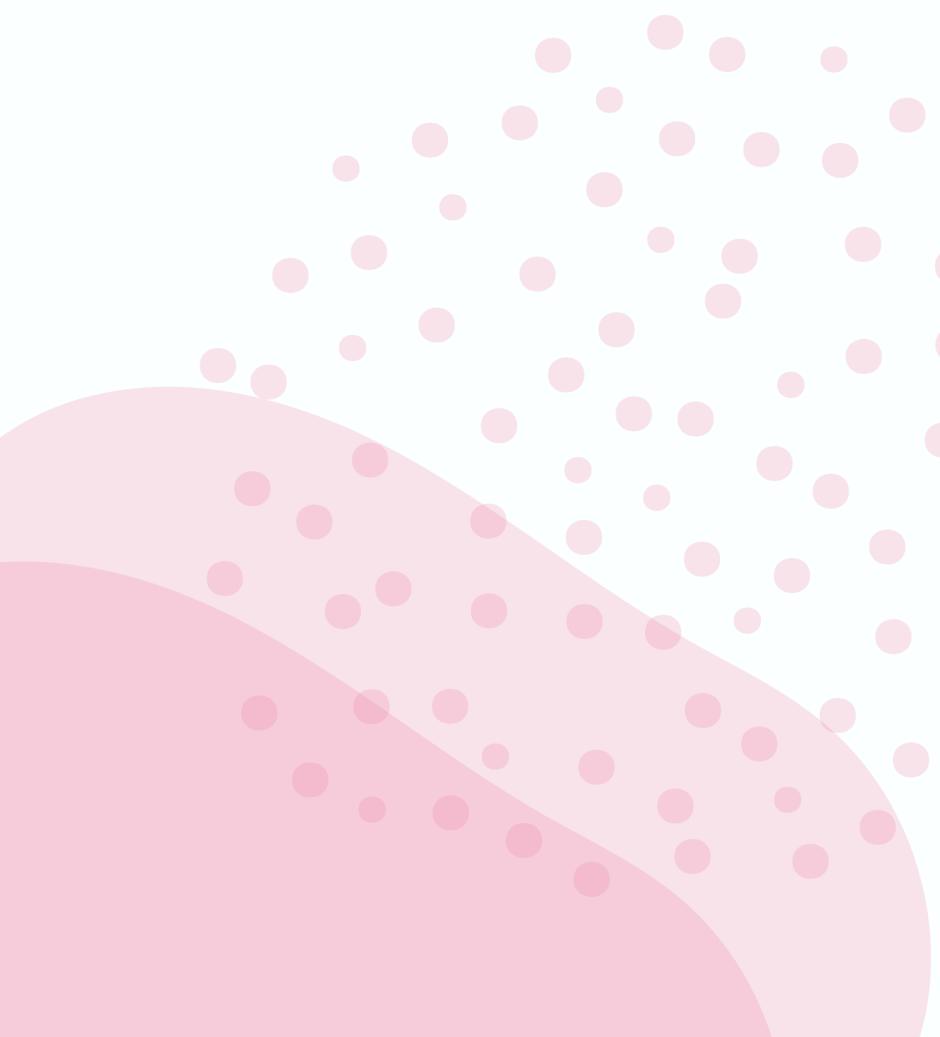
Isolate tests data to avoid conflicts

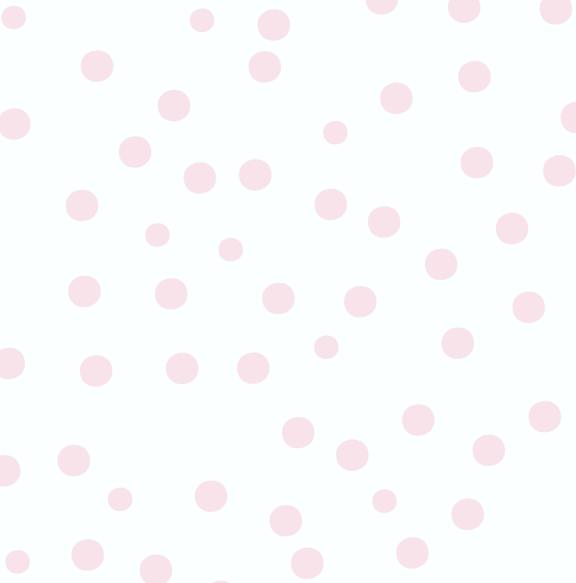
FASTER TESTS

Reduce a lot of time talking to the database

100% COVERAGE

Easily setup edge cases: unexpected errors





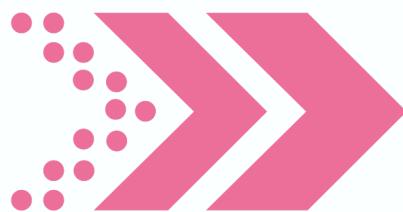
Is it good enough to test our API with a mock DB?

Is it good enough to test our API with a mock DB?

YES! OUR REAL DB STORE IS ALREADY TESTED

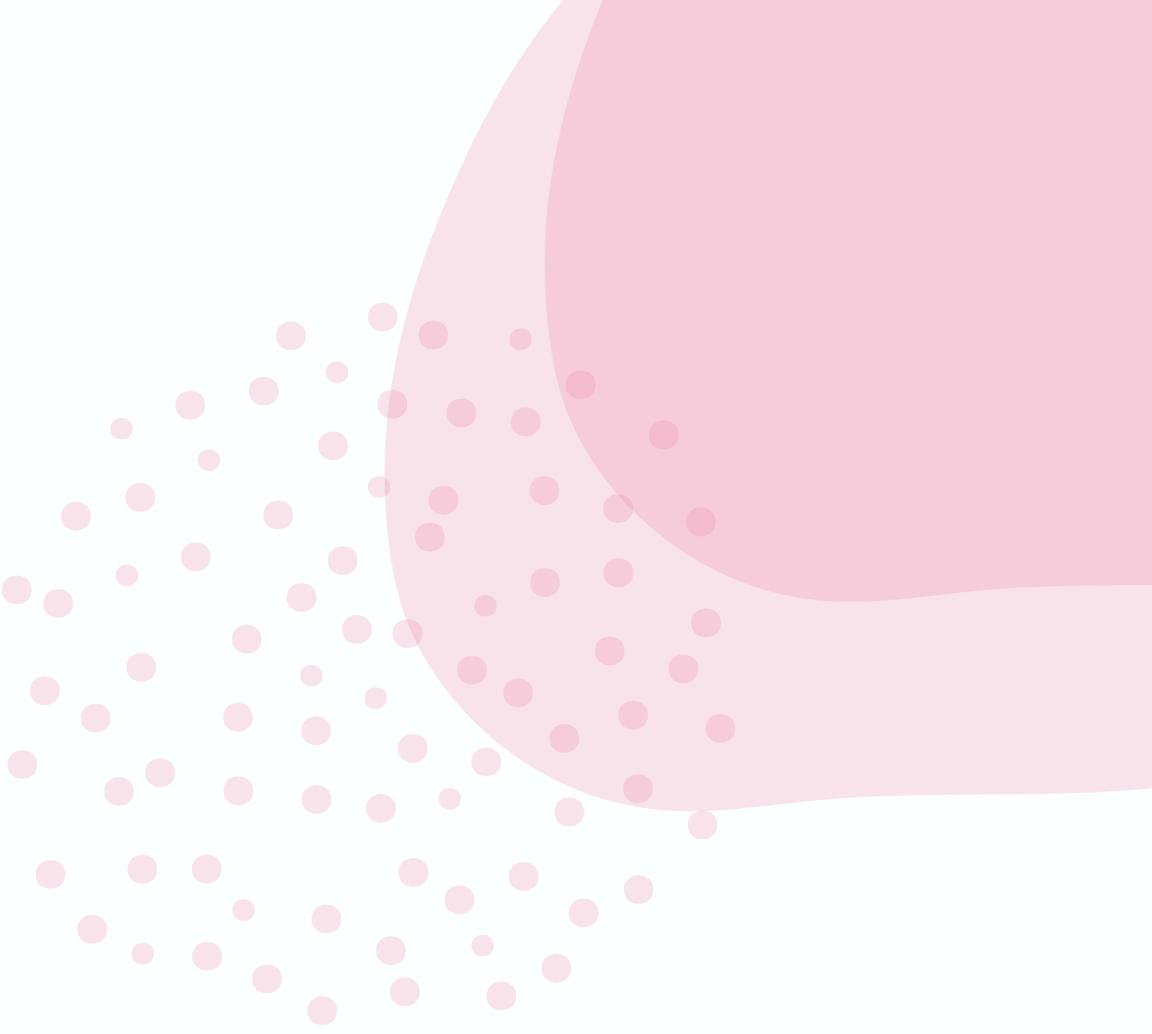
Is it good enough to test our API with a mock DB?

YES! OUR REAL DB STORE IS ALREADY TESTED



MOCK DB & REAL DB SHOULD
IMPLEMENT THE SAME INTERFACE

How to mock?



How to mock?

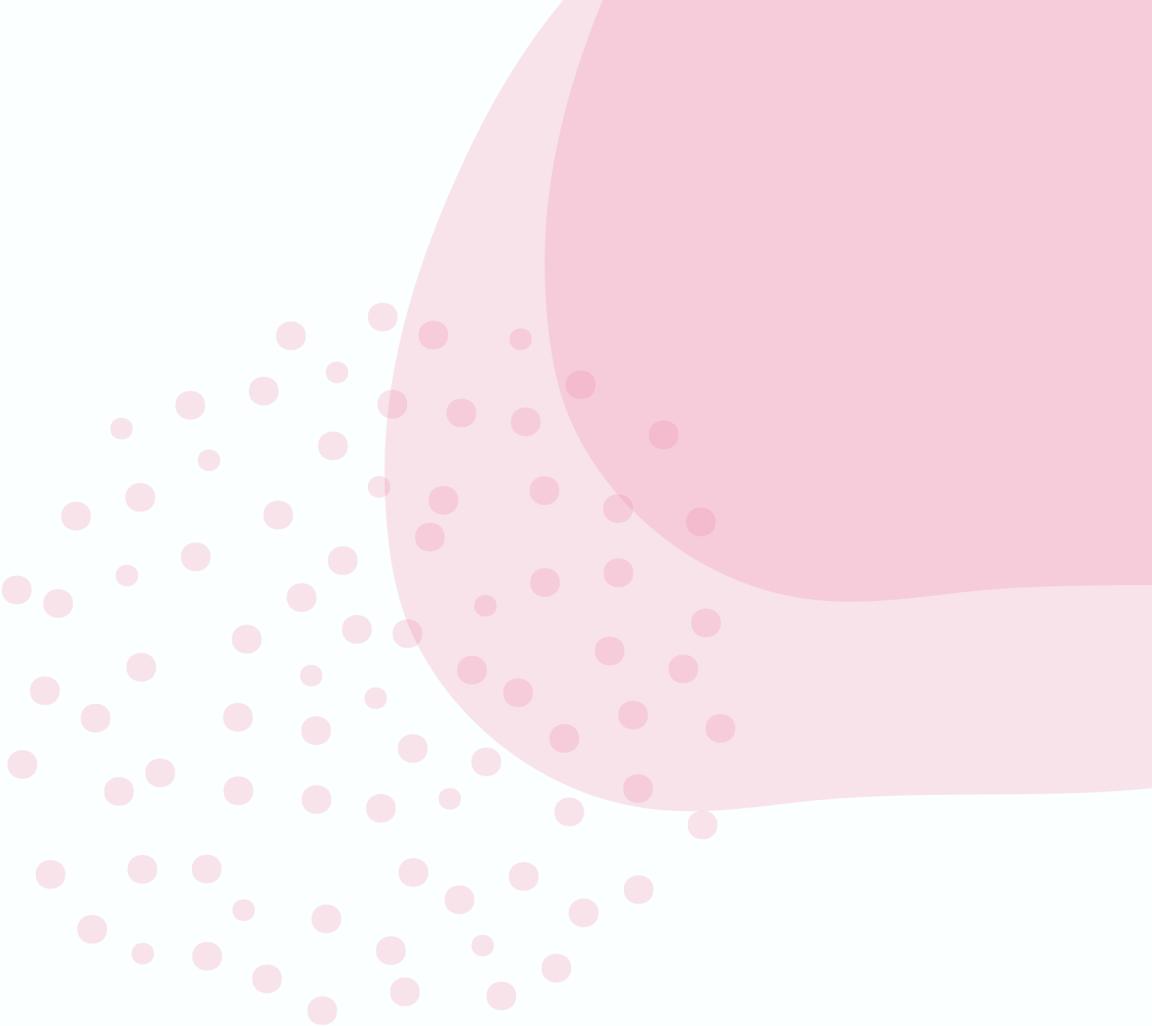
```
type Store interface {
    GetAccount(id int64) (Account, error)
}

type MemStore struct {
    data map[int64]Account
}

func (store *MemStore) GetAccount(id int64) (Account, error) {
    return store.data[id], nil
}
```

USE FAKE DB: MEMORY

Implement a fake version of DB:
store data in memory



How to mock?

```
type Store interface {
    GetAccount(id int64) (Account, error)
}

type MemStore struct {
    data map[int64]Account
}

func (store *MemStore) GetAccount(id int64) (Account, error) {
    return store.data[id], nil
}
```

```
func TestGetAccountAPI(t *testing.T) {
    ctrl := gomock.NewController(t)
    defer ctrl.Finish()

    store := mockdb.NewMockStore(ctrl)
    account := randomAccount()

    // Build stub: expect GetAccount() to be called
    // exactly 1 time with this input account.ID
    // and return this account object as output
    store.EXPECT().GetAccount(gomock.Eq(account.ID)).
        Times(1).Return(account)

    // TODO: use this mock store to test your API
}
```

<https://github.com/golang/mock>

USE FAKE DB: MEMORY

Implement a fake version of DB:
store data in memory

USE DB STUBS: GOMOCK

Generate and build stubs that
returns hard-coded values