

Data management

Validating user_create

Why validate?

- create_user method takes a dict
- what could go wrong?

```
class UserManager:  
    # [...]  
    def create_user(self, user_dict:dict) -> str | None:  
        pass
```

Validating input

We want to make sure that only certain data makes it into the DB, e.g.:

- right names for keys
- right data types for values
- all required fields are included

UserIn model

- Pydantic can be used for validation
 - by default, extra fields will be dropped on conversion
 - types are checked

```
# user_models.py
class UserIn(BaseModel):
    username:str
    password:str
```

Validating DB input

- convert to a UserIn and back to a dict to validate
- create a string `_id` before insertion

```
# validate and insert
data1 = {'username': 'joe', 'password': 'schmoe'}
u1 = UserIn(**data1)
#-----
u1d = u1.model_dump()
u1d['_id'] = str(ObjectId())
x = mycol.insert_one(u1d)
```

Validating results

- We also want to validate results.
- This includes
 - status results from create_user
 - data from read_user

Operation results

- CreateUserResult inherits status and message from OperationResult
- message and inserted_id are both optional
 - message should have a value if status is 0 (fail)
 - inserted_id should have a value if status is 1 (success)

```
class OperationResult(BaseModel):  
    status: int  
    message: Optional[str]=None  
  
class CreateUserResult(OperationResult):  
    inserted_id: Optional[str] = None
```

Implementing create_user

```
def create_user(self, user:UserIn) -> CreateUserResult:  
    '''create user and return result'''  
  
    # dump user to dict  
    # use a try-except block to return the appropriate result
```


Testing

- The tests will be almost the same, except for:
 - creating UserIn to pass to your create_user method
 - dealing with CreateUserResult rather than a dict

```
class TestUserManager(unittest.TestCase):  
    # [...]  
    def test_basic(self):  
        ''' Initialize, create user, read user'''
```

Activity

- Complete implementation and testing for create and read.
- Do the same for update and delete.