

Data management

Validating user_update

update_user: validating updates

- We want to be sure that only certain fields can be updated
- And that they are updated with valid values
- In your user_models:

```
class UserUpdate(BaseModel):  
    password: Optional[str] = None  
    address: Optional[Address] = None
```

Creating a UserUpdate object

- missing fields will be set to None
- But we don't want to set them to None in the DB

```
updates = UserUpdate(password='newpw')  
print(updates)  
# password='newpw' address=None
```

Excluding none

- We need to exclude fields with None before applying the update
- In UserManager class:

```
def update_user(self, user_id, updates: UserUpdate):  
  
    updates = updates.model_dump(exclude_none=True)  
    result = self.col.update_one({'_id': user_id}, {'$set': updates})  
  
    # TODO: what to return here?
```

Validating results

- create_user and update_user return results with 'status' and other fields.
- We should validate results to be sure we are providing the right fields and values.

```
# user_models.py
class UpdateUserResult(OperationResult):
    modified_count: Optional[int] = None
```