Nhóm 23

Đề tài: Phát triển service xử lý và lưu trữ video

Họ và tên: Hà Vĩ Khang

MSSV: 20110657

Milestone 4

Task 7: Xây dựng test case và tiến hành test cho chức năng tải lên hình ảnh có sẳn để tạo preview thumbnail (p2)

**4**. Kiểm tra tải ảnh lên tạo preview thumbnail lỗi xung đột

@pytest.mark.parametrize('projects', [({'file': 'sample\_0.mp4', 'duplicate': False},)], indirect=True)

@pytest.mark.parametrize('filestreams', [('sample\_0.jpg',)], indirect=True)

def test\_upload\_custom\_preview\_thumbnail\_409\_resp(test\_app, client, projects, filestreams):

project = projects[0]

jpg\_stream = filestreams[0]

# since we use CELERY\_TASK\_ALWAYS\_EAGER, task will be executed immediately,

# it means next request will return a finshed result,

# since we want to test 409 response, we must set processing flag in db directly

test\_app.mongo.db.projects.find\_one\_and\_update(

{'\_id': ObjectId(project['\_id'])},

{'$set': {'processing.thumbnail\_preview': True}}

)

with test\_app.test\_request\_context():

url = url\_for('projects.retrieve\_or\_create\_thumbnails', project\_id=project['\_id'])

resp = client.post(

url,

data={

'file': (BytesIO(jpg\_stream), 'sample\_0.jpg')

},

content\_type='multipart/form-data'

)

assert resp.status == '409 CONFLICT'

**5**. Kiểm tra tải ảnh lên tạo preview thumbnail thành công và xóa preview thumbnail cũ

@pytest.mark.parametrize('projects', [({'file': 'sample\_0.mp4', 'duplicate': False},)], indirect=True)

@pytest.mark.parametrize('filestreams', [('sample\_0.jpg',)], indirect=True)

def test\_upload\_custom\_preview\_remove\_old\_thumbnail(test\_app, client, projects, filestreams):

project = projects[0]

jpg\_stream = filestreams[0]

position = 3

with test\_app.test\_request\_context():

# capture preview

url = url\_for(

'projects.retrieve\_or\_create\_thumbnails', project\_id=project['\_id']

) + f'?type=preview&position={position}'

resp = client.get(url)

assert resp.status == '202 ACCEPTED'

# get storage\_id of captured preview thumbnail

resp = client.get(

url\_for('projects.retrieve\_edit\_destroy\_project', project\_id=project['\_id'])

)

resp\_data = json.loads(resp.data)

captured\_thumb\_strage\_id = resp\_data['thumbnails']['preview']['storage\_id']

# upload custom preview thumbnail

url = url\_for('projects.retrieve\_or\_create\_thumbnails', project\_id=project['\_id'])

resp = client.post(

url,

data={

'file': (BytesIO(jpg\_stream), 'sample\_0.jpg')

},

content\_type='multipart/form-data'

)

resp\_data = json.loads(resp.data)

assert resp.status == '200 OK'

assert test\_app.fs.get(resp\_data['storage\_id']).\_\_class\_\_ is bytes

# check that old thumbnail is deleted

with pytest.raises(FileNotFoundError):

test\_app.fs.get(captured\_thumb\_strage\_id)