Nhóm 23

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

Họ và tên: Hà Vĩ Khang

MSSV: 20110657

Milestone 4

Task 6: Xây dựng test case và tiến hành test cho chức năng tải lên hình ảnh có sẳn để tạo preview thumbnail (p1)

1. Kiểm tra tải ảnh lên tạo preview thumbnail thành công

@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\_success(test\_app, client, projects, filestreams):

project = projects[0]

jpg\_stream = filestreams[0]

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'

)

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

assert resp.status == '200 OK'

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

1. Kiểm tra tải ảnh lên tạo preview thumbnail lỗi không tìm thấy file

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

def test\_upload\_custom\_preview\_thumbnail\_no\_file(test\_app, client, projects):

project = projects[0]

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

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

resp = client.post(

url,

data={},

content\_type='multipart/form-data'

)

assert resp.status == '400 BAD REQUEST'

1. Kiểm tra tải ảnh lên tạo preview thumbnail lỗi sai codec

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

@pytest.mark.parametrize('filestreams', [('sample\_0.mp4',)], indirect=True)

def test\_upload\_custom\_preview\_thumbnail\_wrong\_codec(test\_app, client, projects, filestreams):

project = projects[0]

jpg\_stream = filestreams[0]

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'

)

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

assert resp.status == '400 BAD REQUEST'

assert resp\_data == {'file': ["Codec: 'h264' is not supported."]}