Nhóm 23

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

Họ và tên: Hà Vĩ Khang

MSSV: 20110657

Milestone 4

Task 5: Xây dựng test case và tiến hành test cho chức năng cắt khung ảnh video

1. Kiểm tra cắt khung ảnh video thành công

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

def test\_edit\_project\_crop\_success(test\_app, client, projects):

project = projects[0]

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

# edit request

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

resp = client.put(

url,

data=json.dumps({

"crop": "0,0,640,480"

}),

content\_type='application/json'

)

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

assert resp.status == '202 ACCEPTED'

assert resp\_data == {'processing': True}

# get details

resp = client.get(url)

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

assert resp\_data['metadata']['width'] == 640

assert resp\_data['metadata']['height'] == 480

1. Kiểm tra cắt khung ảnh video thất bại do các lỗi sau: (nội dung cắt phải là chuỗi, các ký tự hợp lệ, x,y phải lơn hơn 0, chiều cao và rộng của khung ảnh phải nằm trong giới hạn của video)

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

def test\_edit\_project\_crop\_fail(test\_app, client, projects):

project = projects[0]

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

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

# incorrect type

resp = client.put(

url,

data=json.dumps({

"crop": 1

}),

content\_type='application/json'

)

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

assert resp.status == '400 BAD REQUEST'

assert resp\_data == {'crop': ["must be of string type"]}

# malformed format

resp = client.put(

url,

data=json.dumps({

"crop": "x,y,w,h"

}),

content\_type='application/json'

)

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

assert resp.status == '400 BAD REQUEST'

assert resp\_data == {'crop': ["value does not match regex '^\\d+,\\d+,\\d+,\\d+$'"]}

# edit request

resp = client.put(

url,

data=json.dumps({

"crop": "2000,0,640,480"

}),

content\_type='application/json'

)

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

assert resp.status == '400 BAD REQUEST'

assert resp\_data == {'crop': ['x is less than minimum allowed crop width']}

# edit request

resp = client.put(

url,

data=json.dumps({

"crop": "0,1000,640,480"

}),

content\_type='application/json'

)

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

assert resp.status == '400 BAD REQUEST'

assert resp\_data == {'crop': ['y is less than minimum allowed crop height']}

# edit request

resp = client.put(

url,

data=json.dumps({

"crop": "300,0,1000,480"

}),

content\_type='application/json'

)

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

assert resp.status == '400 BAD REQUEST'

assert resp\_data == {'crop': ["width of crop's frame is outside a video's frame"]}

# edit request

resp = client.put(

url,

data=json.dumps({

"crop": "0,200,640,600"

}),

content\_type='application/json'

)

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

assert resp.status == '400 BAD REQUEST'

assert resp\_data == {'crop': ["height of crop's frame is outside a video's frame"]}

# edit request

resp = client.put(

url,

data=json.dumps({

"crop": "0,200,10000,600"

}),

content\_type='application/json'

)

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

assert resp.status == '400 BAD REQUEST'

assert resp\_data == {'crop': ['width 10000 is greater than maximum allowed crop width (3840)']}

# edit request

resp = client.put(

url,

data=json.dumps({

"crop": "0,0,300,600"

}),

content\_type='application/json'

)

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

assert resp.status == '400 BAD REQUEST'

assert resp\_data == {'crop': ['width 300 is less than minimum allowed crop width (320)']}

# edit request

resp = client.put(

url,

data=json.dumps({

"crop": "0,0,640,100"

}),

content\_type='application/json'

)

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

assert resp.status == '400 BAD REQUEST'

assert resp\_data == {'crop': ['height 100 is less than minimum allowed crop height (180)']}

1. Kiểm tra cắt khung ảnh video

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

def test\_ffmpeg\_video\_editor\_crop(test\_app, filestreams):

editor = FFMPEGVideoEditor()

mp4\_stream = filestreams[0]

with test\_app.app\_context():

content, metadata = editor.edit\_video(

stream\_file=mp4\_stream,

filename='test\_ffmpeg\_video\_editor\_sample.mp4',

crop={

'x': 0,

'y': 0,

'width': 640,

'height': 480

}

)

assert metadata['width'] == 640

assert metadata['height'] == 480