Unit testing via Python requests

P0 Login in Python requests

P1 Request in Python requests

P2 UserRoles in Python requests

P3 Roles in Python requests

P4 Upload [Access] in Python requests

P5 Upload [file upload] in Python requests

P6 Download [Access] in Python requests

P7 Download [file download] in Python requests

Python Requests https://pypi.org/project/requests/

Test case	Condition	Result
Login	Success	True
Request	Accessible (Get entries from database)	True
UserRoles	Accessible (Get users and roles from database)	True
Roles	Accessible (Get roles from database)	True
Upload [Access]	Accessible	True
Upload [File upload]	File uploaded via post request	False
Download [Access]	Accessible (Get book from database)	True
Download [File download]	File downloaded via post request	True

Python request - login test code

```
response = session.get(login.url, verify=False)

token = BeautifulSoup(response.text, features='html.parser').find('input', {'name':
'__RequestVerificationToken'})['value']

states = ["__RequestVerificationToken", "Email", "RememberMe"]

login_data = {"Input.Email": usr, "Input.Password": password, "__RequestVerificationToken": token}

post_request = session.post(login.url, data=login_data, verify=False)

soup = BeautifulSoup(post_request.content, features="html.parser")

result = (post_request.status_code == 200)

print("Login test : %s" % result)
```

```
Login test : True
```

Python request - Request test code

```python
r = session.get(request.url, verify=False)

soup = BeautifulSoup(r.content, features="html.parser")

data = []

table = soup.find('table', attrs={'class': 'table'})

table_data = table.tbody.find_all("tr")

head = []

td = table_data[0].find("td", attrs={'class': 'SubmittorEmail'})

td_list = td.text.replace(' ', '').replace('\r', '').split('\n')

test = [i for i in td_list if i]

print("Request test : %s" % test[0])
```

```
Request test : ictadmin@student.mahidol.ac.th
```

Python request - UserRoles test code

```
r = session.get(userroles.url, verify=False)

soup = BeautifulSoup(r.content, features="html.parser")

print("UserRoles test : %s" % (soup.title.text == "UserRoles - SeniorLibrary"))

UserRoles test : True
```

Python request - Roles test code

```
r = session.get(roles.url, verify=False)

soup = BeautifulSoup(r.content, features="html.parser")

print("Roles test : %s" % (soup.title.text == "Roles - SeniorLibrary"))

Roles test : True
```

Python request - Upload access test code

```python
r = session.get(upload.url, verify=False)

soup = BeautifulSoup(r.content, features="html.parser")

print("Upload test [Access] : %s" % (soup.title.text == "Index - SeniorLibrary"))
```

```
Upload test [Access] : True
```

Python request - Upload file test code

```python
r = session.post(upload.url, files={"files": open("SP-Webtest.pdf", 'rb')})

if r.ok:

    print("Upload test [file upload] : True")

else:

    print("Upload test [file upload] : False")
```

```
Upload test [file upload] : Failed
```

Python request - Download access test code

```python
r = session.get(download.url, verify=False)

soup = BeautifulSoup(r.content, features="html.parser")

print("Download test [Access] : %s" % (soup.title.text == "Download - SeniorLibrary"))
```

```
Download test [Access] : True
```

Python request - Download file test code

```python
r = session.post(download.url, verify=False)

print("Download test [file download] : %s" % (type(r.content) == bytes))
```

```
Download test [file download] : True
```

Unit testing via Postman

P0 Login in Postman

P1 Request in Postman

P2 Roles in Postman

P3 UserRoles in Postman

P4 Upload [Access] in Postman

P5 Upload [file upload] in Postman

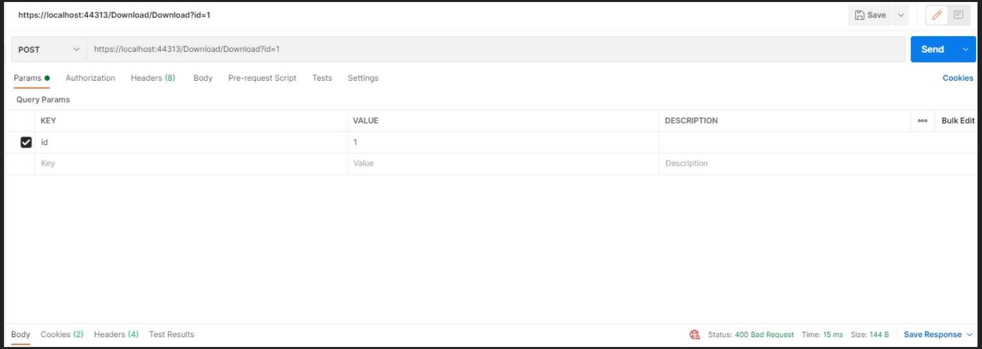
P6 Download [Access] in Postman

P7 Download [file download] in Postman

Postman https://www.postman.com/

Test case	Condition	Result
Login	Success	True
Request	Accessible (Get entries from database)	True
UserRoles	Accessible (Get users and roles from database)	True
Roles	Accessible (Get roles from database)	True
Upload [Access]	Accessible	True
Upload [File upload]	File uploaded via post request	False
Download [Access]	Accessible (Get book from database)	True
Download [File download]	File downloaded via post request	False

Postman - test failures



https://localhost:44313/Download/Download?id=1

POST https://localhost:44313/Download/Download?id=1

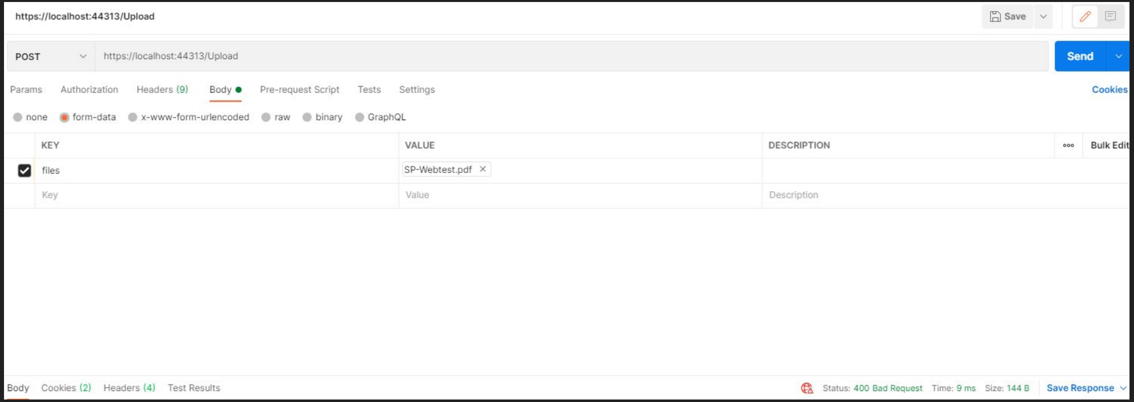
Query Params

	KEY	VALUE	DESCRIPTION
☑	id	1	
	Key	Value	Description

Status: 400 Bad Request Time: 15 ms Size: 144 B

File download failed

File upload failed



https://localhost:44313/Upload

POST https://localhost:44313/Upload

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	files	SP-Webtest.pdf ×	
	Key	Value	Description

Status: 400 Bad Request Time: 9 ms Size: 144 B

System Integration test

Login via web browser
Request via web browser
UserRoles via web browser
Roles via web browser
Upload [Access] via web browser
Upload [File upload] via web browser
Download [Access] via web browser
Download [File download]via web browser

System Integration test

Test case	Condition	Result
Login via web browser	Success	True
Request via web browser	Accessible (Get entries from database)	True
UserRoles via web browser	Accessible (Get users and roles from database)	True
Roles via web browser	Accessible (Get roles from database)	True
Upload [Access] via web browser	Accessible	True
Upload [File upload] via web browser	File uploaded via post request	True
Download [Access] via web browser	Accessible (Get book from database)	True
Download [File download]via web browser	File downloaded via post request	True