Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №3**

По дисциплине: “РСиОТ”

По теме: «Разработка клиентского интерфейса для веб-приложения»

**Выполнил**:

студент 4 курса

группы ПО-7

Комиссаров А.Е.

**Проверил:**

Козик И.Д.

Брест 2024

**Цель.** Разработать клиентскую часть веб-приложения для взаимодействия с API, созданным в рамках первой лабораторной работы. Клиентское приложение должно обеспечивать удобный интерфейс для взаимодействия с функционалом сервера.

**Ход работы:**

Веб-интерфейс разработан на фреймворке flask для python. Обращение от клиента к серверу происходит через модуль requests встроенный в python.

Код программы:

from flask import Flask, request, redirect, render\_template

import requests

from markupsafe import escape

from flask\_wtf import FlaskForm

from wtforms import StringField

from wtforms.validators import DataRequired

import json

SERVER\_URL = "http://192.168.0.150:8000/"

global userid, useraccess

userid = None

useraccess = None

app = Flask(\_\_name\_\_)

def header(title):

global userid, useraccess

credentials = ""

if userid!=None:

credentials = f"""Logged in as: {useraccess}. User id: {userid}. <a href="/logout">logout</a>"""

else:

credentials = """Not logged in."""

header = f"""<h2><a href="/">Lab 3 Komissarov.</a> {title}</h2>

{credentials}<hr>

"""

return header

@app.get('/')

def index\_page():

if userid==None:

returned = header("Index page.")+"""

Welcome to lab 3. You can register or login:<br>

<a href="/login" class="button">LOGIN</a><br>

<a href="/register" class="button">REGISTER</a><br>

"""

return render\_template("template.html", content=returned)

else:

if useraccess=="admin":

returned = header("Index page.")+"""

Select action:

<a href="/allbugs" class="button">List all bugs</a><br>

<a href="/appbugs" class="button">List bugs from app</a><br>

<a href="/openbugs" class="button">List open bugs</a><br>

<a href="/fixedbugs" class="button">List fixed bugs</a><br>

<a href="/addbug" class="button">Add bug</a><br>

<a href="/delbug" class="button">Delete bug</a><br>

<a href="/editbug" class="button">Edit bug</a><br>

<a href="/regadmin" class="button">Register new admin</a><br>

<a href="/logout" class="button">Log out</a><br>

"""

return render\_template("template.html", content=returned)

else:

returned = header("Index page.")+"""

Select action:

<a href="/appbugs" class="button">List bugs from app</a><br>

<a href="/openbugs" class="button">List open bugs</a><br>

<a href="/fixedbugs" class="button">List fixed bugs</a><br>

<a href="/addbug" class="button">Add bug</a><br>

<a href="/logout" class="button">Log out</a><br>

"""

return render\_template("template.html", content=returned)

@app.get('/login')

def login\_get():

returned = header("Login.")+"""

<form method="post" action="/login">

Login: <input type="text" name="username"></input><br>

Password: <input type="password" name="password"></input><br>

<button type="submit">Submit</button>

</form>

"""

return render\_template("template.html", content=returned)

@app.post('/login')

def login\_post():

global userid,useraccess

username = request.form.get("username")

password = request.form.get("password")

print(username + "!!" + password)

resp = requests.post(SERVER\_URL+"users/login", data={"name":username, "password":password})

if resp.json().get("status") != "failed":

creds = list(resp.json().get("status"))

userid = creds[0]

useraccess = creds[1]

return redirect("/")

else:

returned = header("Log in (failed).")+"""

Failed to login, you can try again: <a href="/login" method="post">TRY AGAIN</a>

"""

return render\_template("template.html", content=returned)

@app.get('/register')

def register\_get():

returned = header("Register.")+"""

<form method="post" action="/register">

Login: <input type="text" name="username"></input><br>

Password: <input type="password" name="password"></input><br>

<button type="submit">Submit</button>

</form>

"""

return render\_template("template.html", content=returned)

@app.post('/register')

def register\_post():

username = request.form.get("username")

password = request.form.get("password")

resp = requests.post(SERVER\_URL+"users/register", data={"name":username, "password":password, "access":"user"})

if resp.json().get("status") == "success":

returned = header("Register.")+"""Registered successfully. You can <a href="/">return to index page</a> now."""

return render\_template("template.html", content=returned)

else:

returned = header("Register (failed).")+"""

Registering failed. <a href="/register">Try again?</a><br>

Error message:"""+str(resp.json().get("status"))

return render\_template("template.html", content=returned)

@app.get('/logout')

def logout():

global userid, useraccess

userid=None

useraccess=None

returned = header("Logout.")+"""You are logged out now. You can return to the <a href="/">index page</a>"""

return render\_template("template.html", content=returned)

@app.get('/allbugs')

def all\_bugs():

resp = requests.get(SERVER\_URL+"bugs/list/all")

if (resp.json().get("success")):

bugs = eval(resp.json().get("success"))

result = "<table>"

result += "<tr><th>Bug id</th><th>App id</th><th>Author id</th><th>Title</th><th>Status</th></tr>"

for bug in bugs:

bugid = bug[0]

appid = bug[1]

author = bug[2]

title = bug[3]

status = bug[4]

result+="<tr>"

result+=f"<td>{bugid}</td><td>{appid}</td><td>{author}</td><td>{title}</td><td>{status}</td>"

result+="</tr>"

result += "</table>"

returned = header("List all bugs.")+result

return render\_template("template.html", content=returned)

else:

returned = header("List all bugs (failed).")+str(resp.json())

return render\_template("template.html", content=returned)

@app.get('/appbugs')

def app\_bugs():

returned = header("List bugs from app.")+"""

<form method="post" action="/appbugs">

Enter app id: <input type="text" name="app"></input><br>

<button type="submit">submit</button>

</form>

"""

return render\_template("template.html", content=returned)

@app.post('/appbugs')

def app\_bugs\_post():

app = request.form.get("app")

resp = requests.get(SERVER\_URL+"bugs/list/app", params={"appid":app})

if (resp.json().get("success")):

bugs = eval(resp.json().get("success"))

result = "<table>"

result += "<tr><th>Bug id</th><th>App id</th><th>Author id</th><th>Title</th><th>Status</th></tr>"

for bug in bugs:

bugid = bug[0]

appid = bug[1]

author = bug[2]

title = bug[3]

status = bug[4]

result+="<tr>"

result+=f"<td>{bugid}</td><td>{appid}</td><td>{author}</td><td>{title}</td><td>{status}</td>"

result+="</tr>"

result += "</table>"

returned = header("List bugs from app.")+result

return render\_template("template.html", content=returned)

else:

returned = header("List bugs from app (failed).")+str(resp.json())

return render\_template("template.html", content=returned)

@app.get('/openbugs')

def open\_bugs():

resp = requests.get(SERVER\_URL+"bugs/list/open")

if resp.json().get("success"):

bugs = eval(resp.json().get("success"))

result = "<table>"

result += "<tr><th>Bug id</th><th>App id</th><th>Author id</th><th>Title</th><th>Status</th></tr>"

for bug in bugs:

bugid = bug[0]

appid = bug[1]

author = bug[2]

title = bug[3]

status = bug[4]

result+="<tr>"

result+=f"<td>{bugid}</td><td>{appid}</td><td>{author}</td><td>{title}</td><td>{status}</td>"

result+="</tr>"

result += "</table>"

returned = header("List open bugs.")+result

return render\_template("template.html", content=returned)

else:

returned = header("List open bugs (failed).")+"There was an error processing request:"+str(resp.json())

return render\_template("template.html", content=returned)

@app.get('/fixedbugs')

def fixed\_bugs():

resp = requests.get(SERVER\_URL+"bugs/list/fixed")

if resp.json().get("success"):

bugs = eval(resp.json().get("success"))

result = "<table>"

result += "<tr><th>Bug id</th><th>App id</th><th>Author id</th><th>Title</th><th>Status</th></tr>"

for bug in bugs:

bugid = bug[0]

appid = bug[1]

author = bug[2]

title = bug[3]

status = bug[4]

result+="<tr>"

result+=f"<td>{bugid}</td><td>{appid}</td><td>{author}</td><td>{title}</td><td>{status}</td>"

result+="</tr>"

result += "</table>"

returned = header("List fixed bugs.")+result

return render\_template("template.html", content=returned)

else:

returned = header("List fixed bugs (failed).")+"There was an error processing request:"+str(resp.json())

return render\_template("template.html", content=returned)

@app.get('/addbug')

def add\_bugs():

returned = header("Add bug.")+"""

<form method="post" action="/addbug">

App id: <input type="text" name="appid"></input><br>

Bug title: <input type="text" name="title"></input><br>

<button type="submit">submit</button>

</form>

"""

return render\_template("template.html", content=returned)

@app.post('/addbug')

def add\_bugs\_post():

global userid, useraccess

appid = request.form.get("appid")

title = request.form.get("title")

resp = requests.post(SERVER\_URL+"bugs/add", data={"appid":appid, "authorid":userid, "title":title, "status":"open"})

returned = header("Add bug.")+str(resp.json().get("status"))

return render\_template("template.html", content=returned)

@app.get('/delbug')

def del\_bug():

returned = header("Delete bug.")+"""

<form method="post" action="/delbug">

Enter bug id: <input type="text" name="bug"></input><br>

<button type="submit">submit</button>

</form>

"""

return render\_template("template.html", content=returned)

@app.post('/delbug')

def del\_bug\_post():

bugid = request.form.get("bug")

resp = requests.post(SERVER\_URL+"bugs/delete", data={"id":bugid})

returned = header("Delete bug.")+str(resp.json().get("status"))

return render\_template("template.html", content=returned)

@app.get('/editbug')

def edit\_bug():

returned = header("Edit bug.")+"""

<form method="post" action="/editbug">

Bug id: <input type="text" name="bugid"></input><br>

App id: <input type="text" name="appid"></input><br>

Author id: <input type="text" name="author"></input><br>

Title: <input type="text" name="title"></input><br>

Status: <input type="text" name="status"></input><br>

<button type="submit">Submit</button>

</form>

"""

return render\_template("template.html", content=returned)

@app.post('/editbug')

def edit\_bug\_post():

bugid = request.form.get("bugid")

appid = request.form.get("appid")

authorid = request.form.get("author")

title = request.form.get("title")

status = request.form.get("status")

resp = requests.post(SERVER\_URL+"bugs/edit", data={"id":bugid, "appid":appid, "authorid":authorid, "title":title, "status":status})

if resp.json().get("status"):

returned = header("Edit bug.")+str(resp.json().get("status"))

else:

returned = header("Edit bug (failed).")+"""There was an error processing your request. Error: """+str(resp.json())

return render\_template("template.html", content=returned)

@app.get('/regadmin')

def register\_admin\_get():

returned = header("Register new admin.")+"""

<form method="post" action="/regadmin">

Login: <input type="text" name="username"></input><br>

Password: <input type="password" name="password"></input><br>

<button type="submit">submit</button>

</form>

"""

return render\_template("template.html", content=returned)

@app.post('/regadmin')

def register\_admin\_post():

username = request.form.get("username")

password = request.form.get("password")

resp = requests.post(SERVER\_URL+"users/register", data={"name":username, "password":password, "access":"admin"})

if resp.json().get("status") == "success":

returned = header("Register new admin.")+"""Registered successfully. You can <a href="/">return to index page</a> now."""

return render\_template("template.html", content=returned)

else:

returned = header("Register new admin (failed).")+"""

Registering failed. <a href="/regadmin">Try again?</a><br>

Error message:"""+str(resp.json().get("status"))

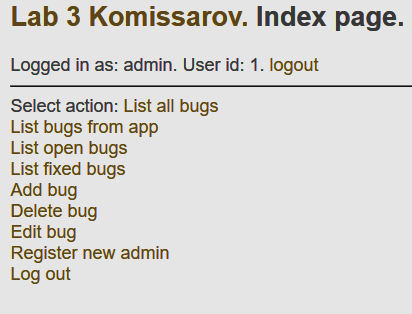
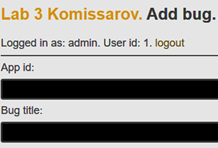
return render\_template("template.html", content=returned)

if \_\_name\_\_=='\_\_main\_\_':

app.run()

Результат работы программы:



**Вывод.** Разработал клиентскую часть веб-приложения для взаимодействия с API сервера.