

Министерство цифрового развития, связи и массовых коммуникаций Федеральное  
государственное образовательное бюджетное учреждение  
высшего профессионального образования  
«Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

Отчёт

Лабораторная работа №4

«Разработка автотестов для Web UI с использованием Selenium»

Выполнил студент

группы ИП-311:  
Терентьев Андрей

Проверил:

Сентюров Святослав Алесеевич

Новосибирск 2025 г.

## Цель работы

Освоить навыки разработки автотестов для веб-интерфейса с использованием Selenium на примере функционала авторизации в OpenBMC.

## Задачи

- Изучить основы работы с Selenium.
- Разработать автотесты для проверки функционала авторизации в Web UI OpenBMC.
- Покрыть основные сценарии авторизации (успешная авторизация, неверные данные, блокировка учетной записи).
- Запустить автотесты и проанализировать результаты.

## Ход работы

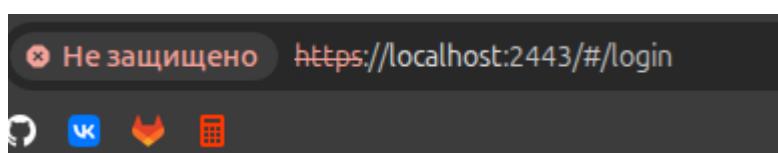
Установили Python и Selenium.

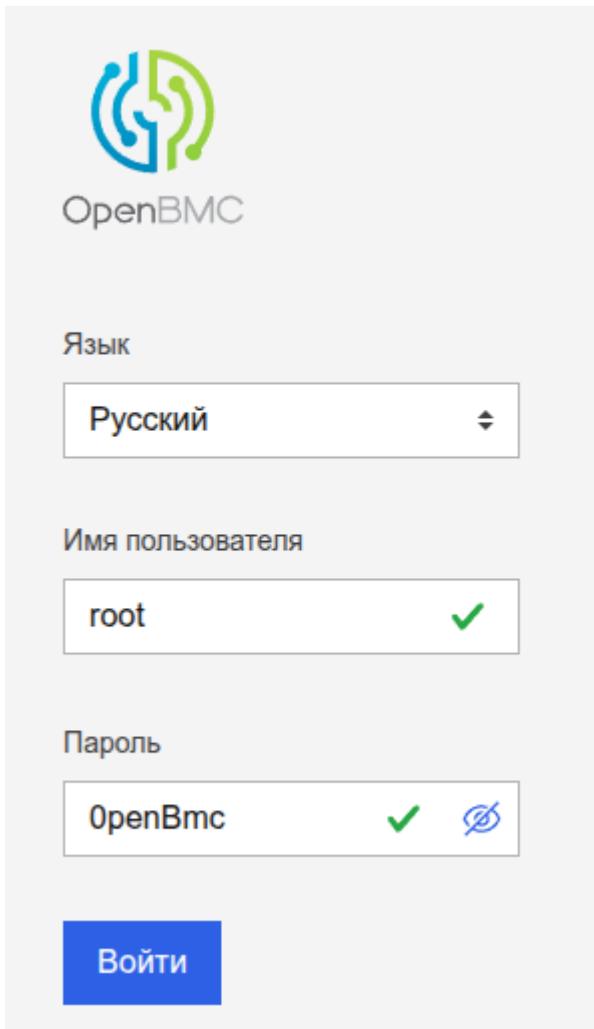
```
anr1@u24:~$ python3 --version
Python 3.12.3
anr1@u24:~$ pip show selenium
Name: selenium
Version: 4.18.1
Summary: Python bindings for Selenium
Home-page: https://github.com/SeleniumHQ/selenium
Author:
Author-email:
License: Apache 2.0
Location: /usr/lib/python3/dist-packages
Requires:
Required-by:
anr1@u24:~$
```

```
anr1@u24:~$ chromedriver --version
ChromeDriver 140.0.7339.127 (9412745860d8c3dfed9cf38f5daa943b163f8c69-refs/branch-heads/7339@{#1849})
anr1@u24:~$
```

Проверили webdriver.

Перешёл по адресу localhost:2443 и открыл OpenBMC





Создали файл tests\_openbmc.py. Далее написали следующие тесты:

### Настройка WebDriver

```
8  @pytest.fixture
9  def driver():
10     options = Options()
11     options.add_argument('--no-sandbox')
12     options.add_argument('--ignore-certificate-errors')
13
14     driver = webdriver.Chrome(service=Service('/usr/bin/chromedriver'), options=options)
15     yield driver
16     driver.quit()
17
18 def test_find_openbmc_web(driver):
19     driver.get("https://localhost:2443")
20     time.sleep(2)
21     assert any(word in driver.page_source.lower()
22               for word in ['openbmc', 'username', 'password']), "OpenBMC не найден"
23
```

Написали фикстуру для подготовки драйвера и передачи его в остальные тесты. Фикстура — функция, которая готовит данные или настройки для тестов.

Запускаем в Chrome через ChromeDriver, переходим на адрес OpenBMC, ищем ключевые слова в коде страницы. Если нашли — тест пройден.

## Тест успешной авторизации

```
24  def test_successful_login(driver):
25      driver.get("https://localhost:2443")
26      time.sleep(1)
27      driver.find_element(By.ID, "username").send_keys("root")
28      driver.find_element(By.ID, "password").send_keys("OpenBmc")
29      driver.find_element(By.XPATH, "//button[@type='submit']").click()
30      time.sleep(2)
31
32      assert "login" not in driver.current_url, "Авторизация не удалась"
33
```

Запускаем страницу OpenBMC, вводим логин и пароль, нажимаем кнопку Log In. Если ушли со страницы входа — тест пройден.

The screenshot shows the OpenBMC web interface. At the top, there is a navigation bar with icons for Health (green checkmark), Power (power button), Refresh, and a user dropdown for 'root'. Below the header, the word 'Overview' is prominently displayed. On the left, there is a section titled 'System information' containing two tables: 'Server information' and 'Firmware information'. The 'Server information' table shows 'Model' and 'Serial number' both as '--'. The 'Firmware information' table shows 'Running' and 'Backup' status as '--', and the 'Firmware version' as '3.0.0-dev-773-gac36376149'. At the bottom of the main content area, there is a dark button labeled 'SOL console' with a right-pointing arrow.

## Тест авторизации с неверным именем и паролем

```

34     def test_wrong_data(driver):
35         driver.get("https://localhost:2443")
36         time.sleep(1)
37         driver.find_element(By.ID, "username").send_keys("tank")
38         driver.find_element(By.ID, "password").send_keys("T-34-85")
39         driver.find_element(By.XPATH, "//button[@type='submit']").click()
40         time.sleep(5)
41
42     assert "login" in driver.current_url, "Как ты зашёл с названием танка вместо пароля?"
43

```

Тот же алгоритм, как в прошлом teste. Если остались на странице входа — тест пройден.

```

===== 3 passed in 15.97s =====
anr1@u24:~/Desktop/test/4lab$ pytest tests_openbmc.py -v
=====
platform linux -- Python 3.12.3, pytest-7.4.4, pluggy-1.4.0 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /home/anr1/Desktop/test/4lab
collected 3 items

tests_openbmc.py::test_find_openbmc_web PASSED
tests_openbmc.py::test_successful_login PASSED
tests_openbmc.py::test_wrong_data PASSED

===== 3 passed in 15.94s =====

```

### Тест блокировки учетной записи

```

anr1@u24:~$ ipmitool -C 17 -H localhost -p 2623 -I lanplus -U root -P OpenBmc user set name 10 testuser
anr1@u24:~$ ipmitool -C 17 -H localhost -p 2623 -I lanplus -U root -P OpenBmc user set password 10 TestPass123
Set User Password command successful (user 10)
anr1@u24:~$ ipmitool -C 17 -H localhost -p 2623 -I lanplus -U root -P OpenBmc user enable 10

romulus login: root
Password:
root@romulus:~# busctl set-property xyz.openbmc_project.User.Manager /xyz/openbmc_project/user xyz.openbmc_project.User.AccountPolicy MaxLoginAttemptBeforeLockout q 3

```

### Настройки политики учётной записи

Максимальное количество попыток входа	Метод разблокировки пользователя
Значение должно быть между 0 – 65535	<input type="radio"/> Ручное
<input type="text" value="3"/>	<input checked="" type="radio"/> Автоматически после истечения таймаута
	Продолжительность таймаута (в секундах)
	<input type="text" value="100"/>
<input type="button" value="Отменить"/> <input type="button" value="Сохранить"/>	

```

44 def test_account_block(driver):
45     for i in range(3):
46         driver.get("https://localhost:2443")
47         driver.find_element(By.ID, "username").send_keys("testuser")
48         driver.find_element(By.ID, "password").send_keys("i")
49         driver.find_element(By.XPATH, "//button[@type='submit']").click()
50         time.sleep(2)
51         assert "login" in driver.current_url
52
53     driver.get("https://localhost:2443")
54     driver.find_element(By.ID, "username").send_keys("testuser")
55     driver.find_element(By.ID, "password").send_keys("TestPass123")
56     driver.find_element(By.XPATH, "//button[@type='submit']").click()
57     time.sleep(2)
58
59     assert "login" in driver.current_url, "Аккаунт не заблокирован!"
60     print("✅ Аккаунт заблокирован!")
61

```

3 попытки входа. Далее блокировка — тест пройден.

### Включение сервера через WebUI

```

70
77 def test_power_control(driver):
78     driver.get("https://localhost:2443")
79     driver.find_element(By.ID, "username").send_keys("root")
80     driver.find_element(By.ID, "password").send_keys("OpenBmc")
81     driver.find_element(By.XPATH, "//button[@type='submit']").click()
82     time.sleep(3)
83
84     driver.get("https://localhost:2443/#/operations/server-power-operations")
85     time.sleep(2)
86     driver.find_element(By.XPATH, "//button[contains(text(), 'Power on')]").click()
87     time.sleep(5)
88
89     assert "on" in driver.page_source.lower(), "Питание не включилось"

```

Заходим на страницу питания, нажимаем кнопку, смотрим на включение.

### Проверка температуры

```

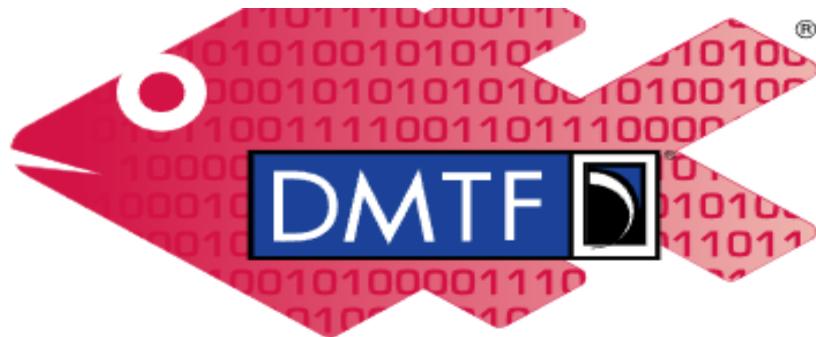
def test_temperature_redfish(driver):
    driver.get("https://localhost:2443/redfish/v1/Chassis/chassis/Thermal")
    time.sleep(2)

    if "login" in driver.current_url:
        driver.find_element(By.ID, "username").send_keys("root")
        driver.find_element(By.ID, "password").send_keys("OpenBmc")
        driver.find_element(By.XPATH, "//button[@type='submit']").click()
        time.sleep(3)
        driver.get("https://localhost:2443/redfish/v1/Chassis/chassis/Thermal")
        time.sleep(5)

        page_text = driver.page_source.lower()
        assert "thermal" in page_text or "temperature" in page_text, "Thermal endpoint не доступен"

```

Переходим на страницу температуры Redfish. Проверяем существование thermal, temperature.



# Redfish

```
{  
    "error": {  
        "@Message.ExtendedInfo": [  
            {  
                "@odata.type": "#Message.v1_1_1.Message",  
                "Message": "The requested resource of type named  
'Thermal' was not found.",  
                "MessageArgs": [  
                    "",  
                    "Thermal"  
                ],  
                "MessageId": "Base.1.19.ResourceNotFound",  
                "MessageSeverity": "Critical",  
                "Resolution": "Provide a valid resource  
identifier and resubmit the request."  
            }  
        ],  
        "code": "Base.1.19.ResourceNotFound",  
        "message": "The requested resource of type named  
'Thermal' was not found."  
    }  
}
```

```
anr1@u24:~/Desktop/test/4lab$ pytest tests_openbmc.py -v -s  
=====  
platform linux -- Python 3.12.3, pytest-7.4.4, pluggy-1.4.0 -- /usr/bin/python3  
cachedir: .pytest_cache  
rootdir: /home/anr1/Desktop/test/4lab  
collected 6 items  
  
tests_openbmc.py::test_find_openbmc_web PASSED  
tests_openbmc.py::test_successful_login PASSED  
tests_openbmc.py::test_wrong_data PASSED  
tests_openbmc.py::test_account_block ✅ Аккаунт заблокирован!  
PASSED  
tests_openbmc.py::test_temperature_redfish PASSED  
tests_openbmc.py::test_power_control PASSED
```

Все тесты успешно проходят.

## **Вывод**

В ходе лабораторной работы вы освоили навыки разработки автотестов для веб-интерфейса с использованием Selenium. Вы создали тесты для проверки функционала авторизации в OpenBMC, покрыли основные сценарии и проанализировали результаты. Эти навыки могут быть применены для тестирования других веб-приложений.