

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 Алеевич

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

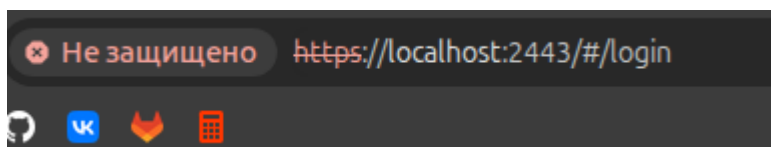
Установили Python и Selenium.

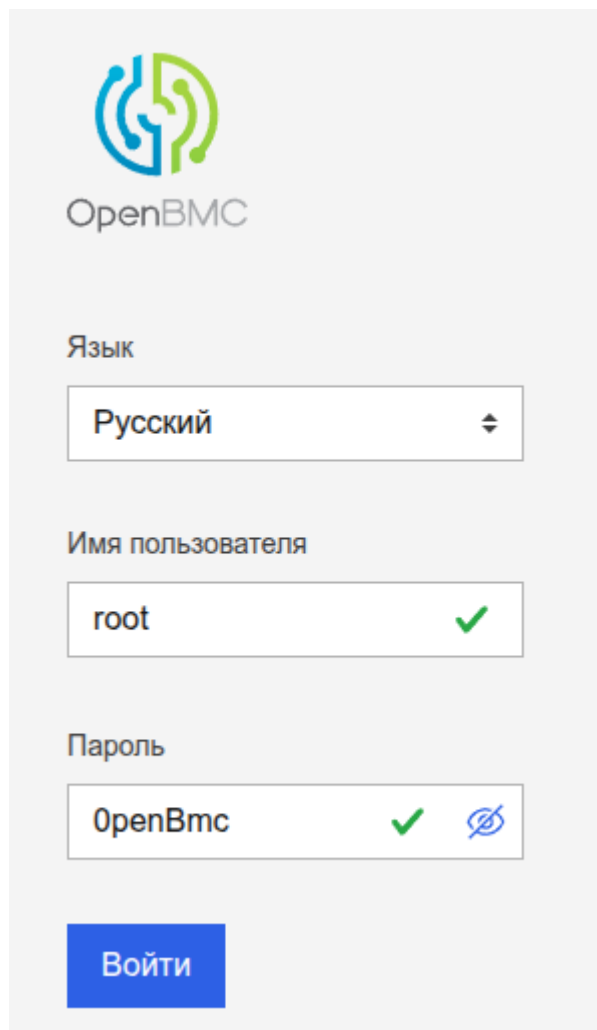
```
anr1@u24:~$ python3 --version
Python 3.12.3
anr1@u24:~$ pip show selenium
Name: selenium
Version: 4.18.1
Summary: Python bindings for Selenium
Home-page: https://github.com/SeleniumHQ/selenium
Author:
Author-email:
License: Apache 2.0
Location: /usr/lib/python3/dist-packages
Requires:
Required-by: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





The image shows the OpenBMC login interface. At the top is the OpenBMC logo, which consists of a stylized 'G' and 'B' in blue and green. Below the logo is the text 'OpenBMC'. The form has three input fields: 'Язык' (Language) with a dropdown menu showing 'Русский'; 'Имя пользователя' (Username) with the text 'root' and a green checkmark; and 'Пароль' (Password) with the text 'OpenBmc', a green checkmark, and an eye icon. At the bottom is a blue button labeled 'Войти' (Login).

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
```

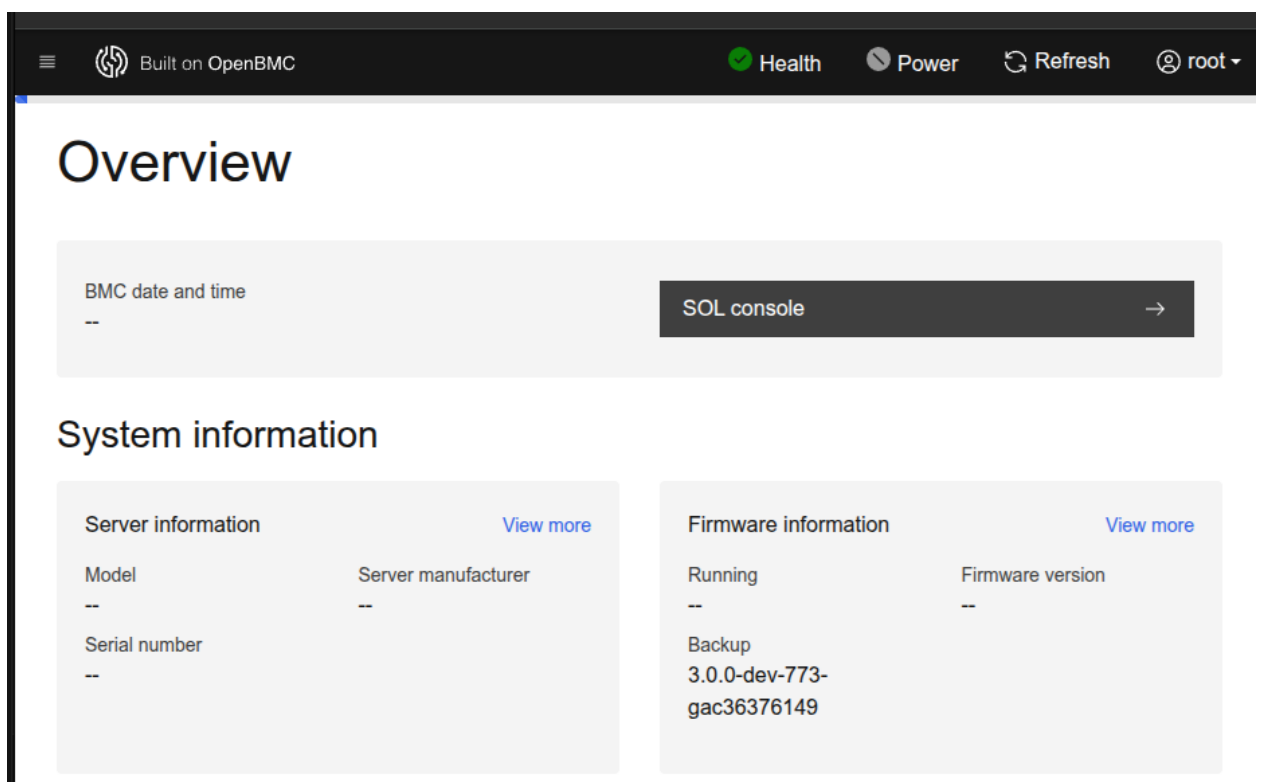
Написали фикстуру для подготовки драйвера и передачи его в остальные тесты. Фикстура — функция, которая подготавливает данные или настройки для тестов.

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("0penBmc")
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.



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 тесте. Если остались на странице входа — тест пройден.

```

===== 3 passed in 15.97s =====
anr1@u24:~/Desktop/test/4lab$ pytest tests_openbmc.py -v
===== test session starts =====
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

```

5 anr1@u24:~$ ipmitool -C 17 -H localhost -p 2623 -I lanplus -U root -P 0penBmc us
er set name 10 testuser
anr1@u24:~$ ipmitool -C 17 -H localhost -p 2623 -I lanplus -U root -P 0penBmc us
er 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 0penBmc us
er enable 10

```

```

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

```

Настройки политики учётной записи

Максимальное количество попыток входа

Значение должно быть между 0 – 65535

3 ✓

Метод разблокировки пользователя

☐ Ручное
☒ Автоматически после истечения таймаута

Продолжительность таймаута (в секундах)

100 ✓

Отменить

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("0penBmc")
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

```

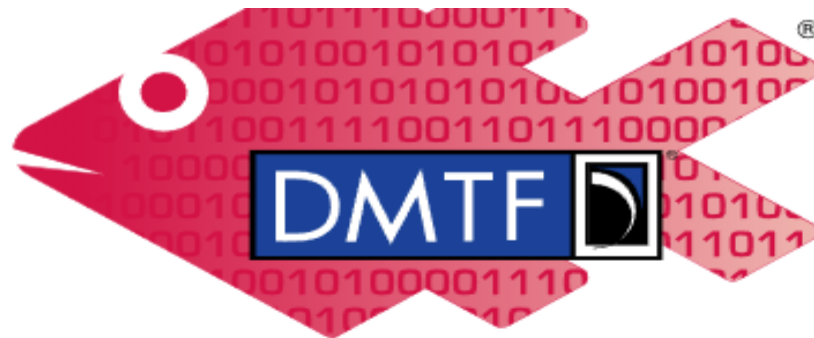
def test_temperature_redfish(driver):
    driver.get("https://localhost:2443/redfish/v1/Chassis/chassis/Thermal")
    time.sleep(2)

    if "login" in driver.current_url:
        driver.find_element(By.ID, "username").send_keys("root")
        driver.find_element(By.ID, "password").send_keys("0penBmc")
        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.