Міністерство освіти і науки україни

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Математичний факультет

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Дисципліна «Автоматизоване тестування»

Лабораторна робота № 5

Виконав студент гр. 8.1214

Гавриш Іван Васильович

Перевірив: Викладач

Лебідь Максим Володимирович

Запоріжжя

2024

**Посилання на репозиторій:** [givtszs/automated-testing-lab5](https://github.com/givtszs/automated-testing-lab5)

**Завдання:**

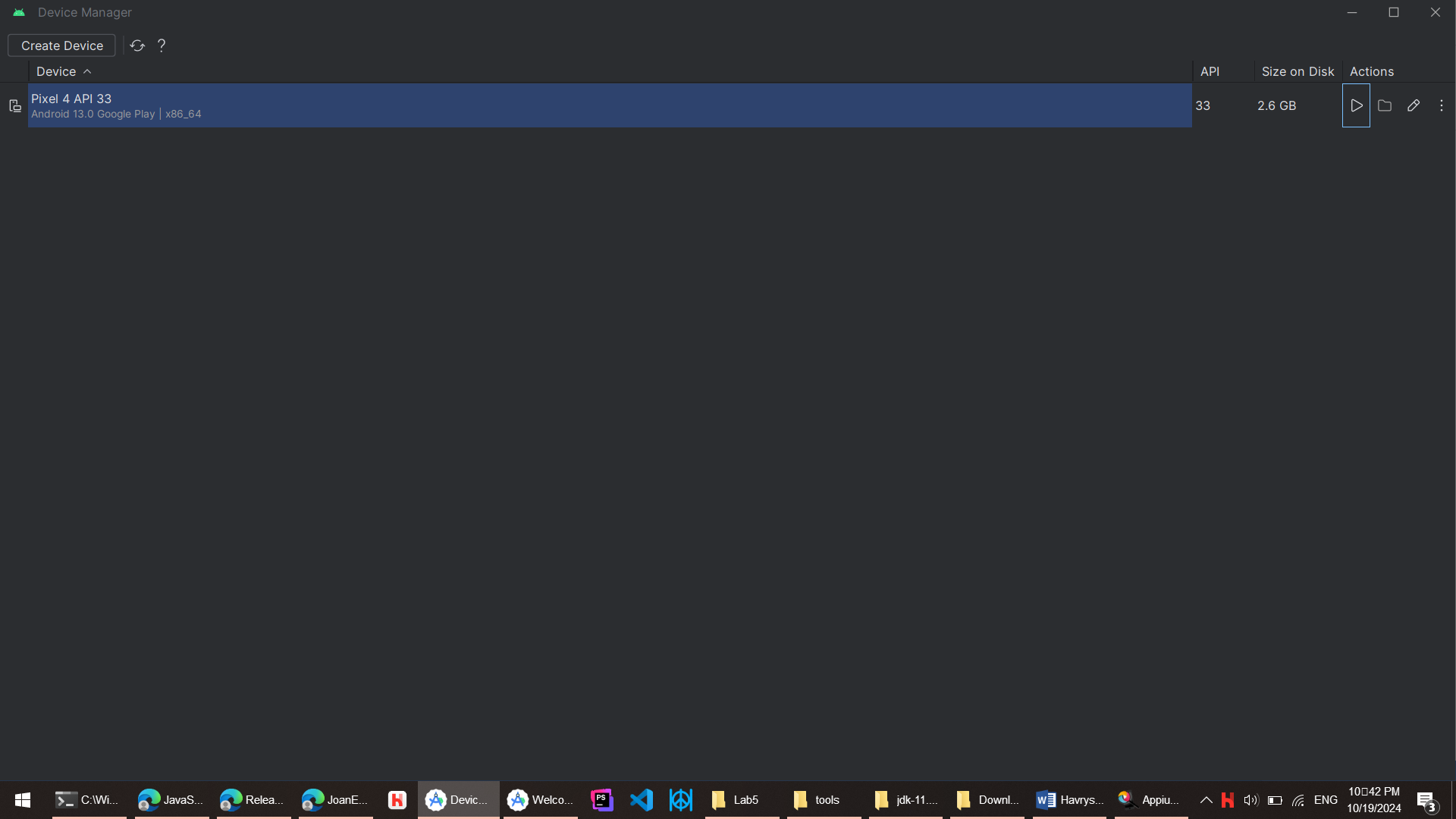
1. Реалізувати скрипт для автоматизації наступних дій з застосунком API-Demos:

* Отримати доступ до текстового поля із використанням декількох стретегій пошуку елементів: Views->Auto Complete -> Screen Top.
* Ввести назву країни.
* Перевірити введену назву за допомогою assert метода.
* На вибір протестувати ще 2 пункти головного меню застосунку.

**Хід роботи:**

1. Виконаємо створення проєкту, інсталяцію та налаштування Appium, WebDriverIO.

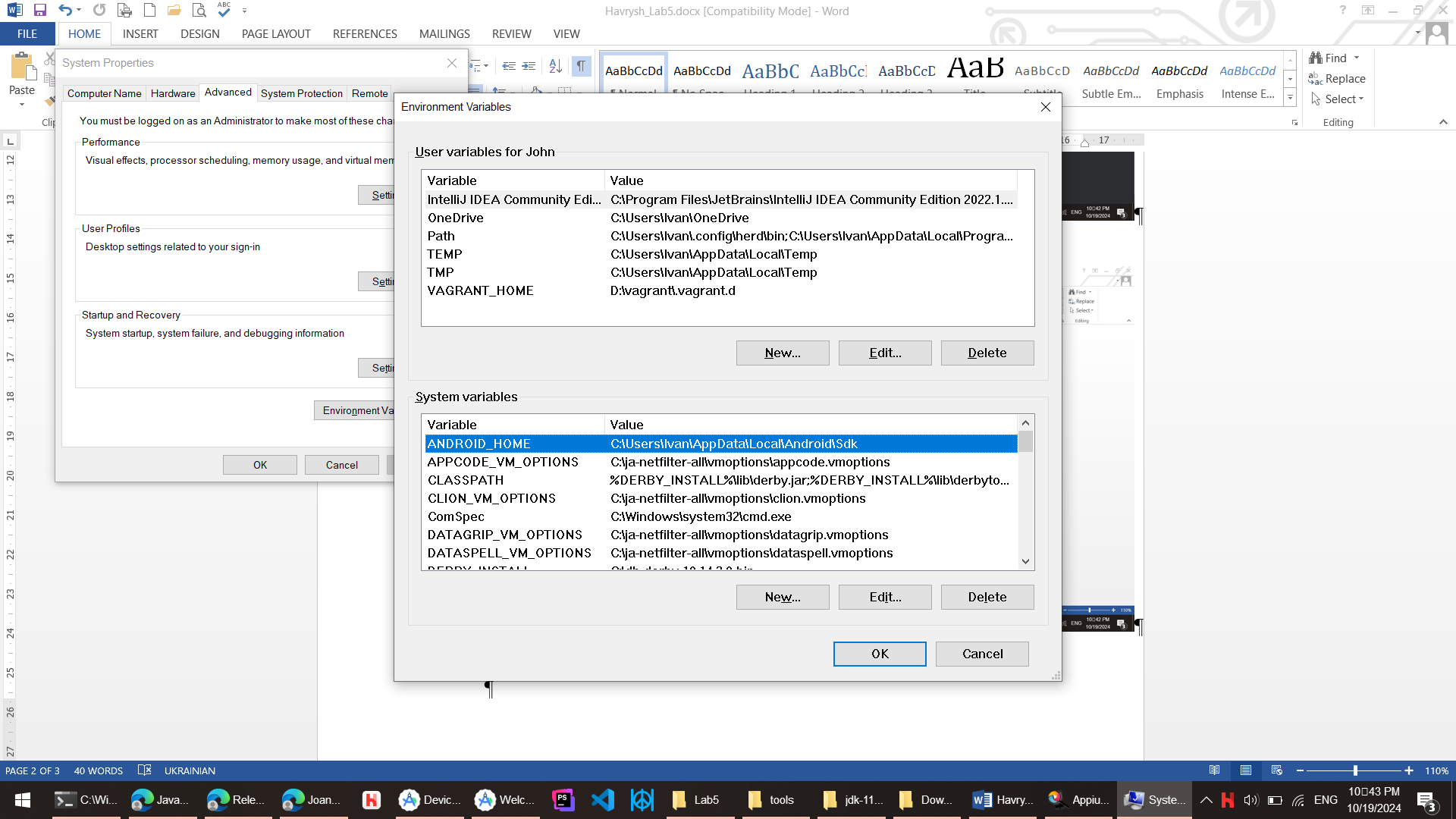
1.1 Створюємо емулятор Android пристрою:



1.2 Створюємо нову системну змінну JAVA\_HOME:



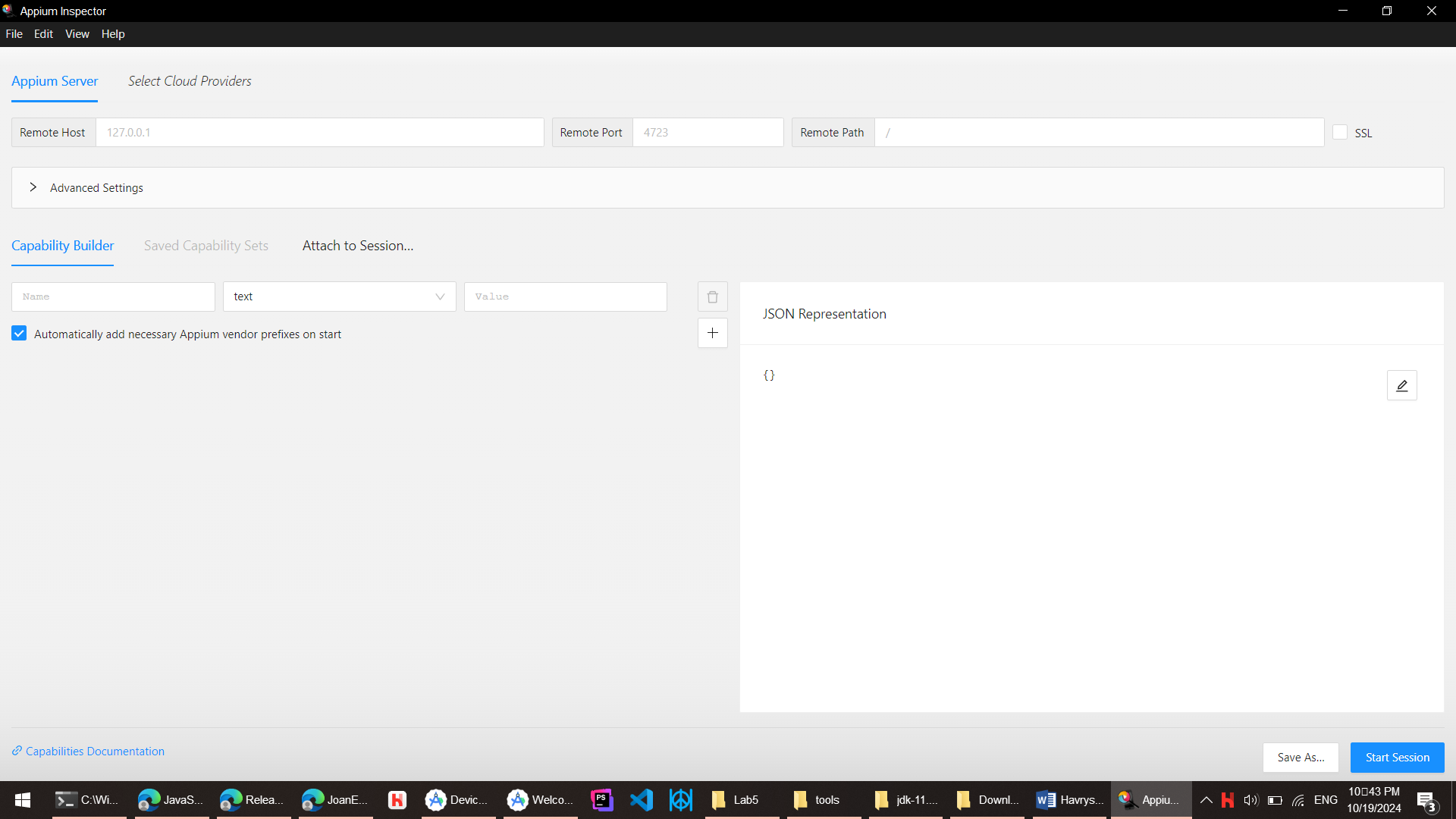
1.3 Створюємо нову системну змінну ANDROID\_HOME:



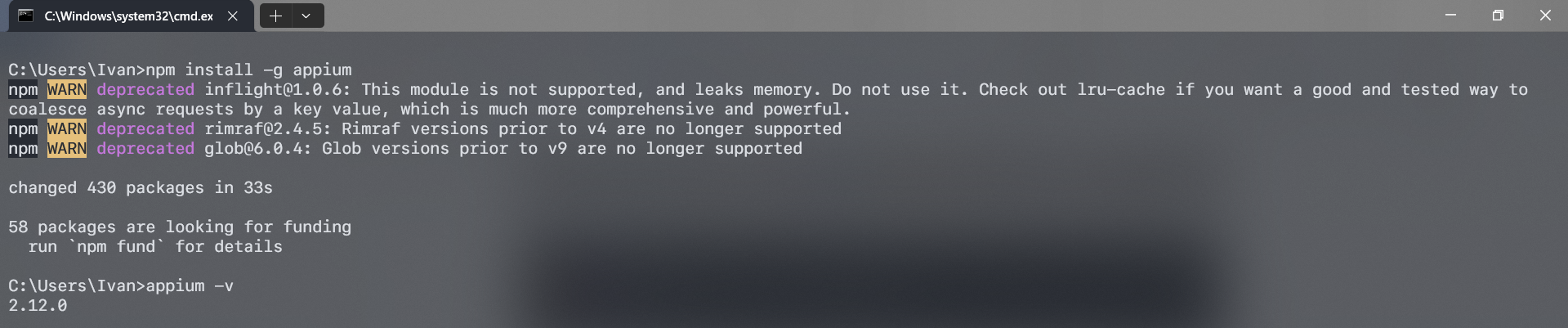
1.4 Додаємо записи до змінної Path:



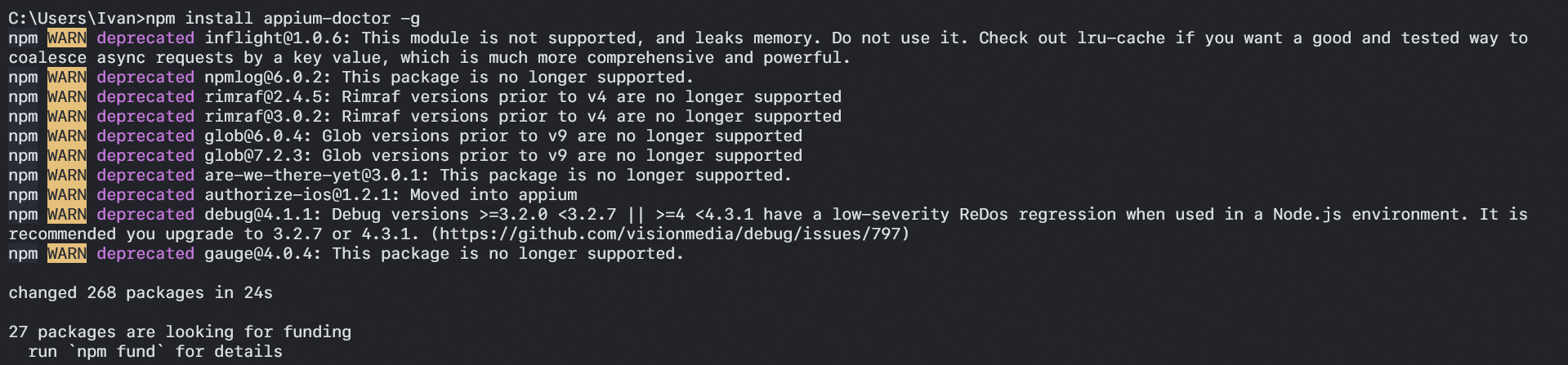
1.5 Встановлюємо AppiumInspector:



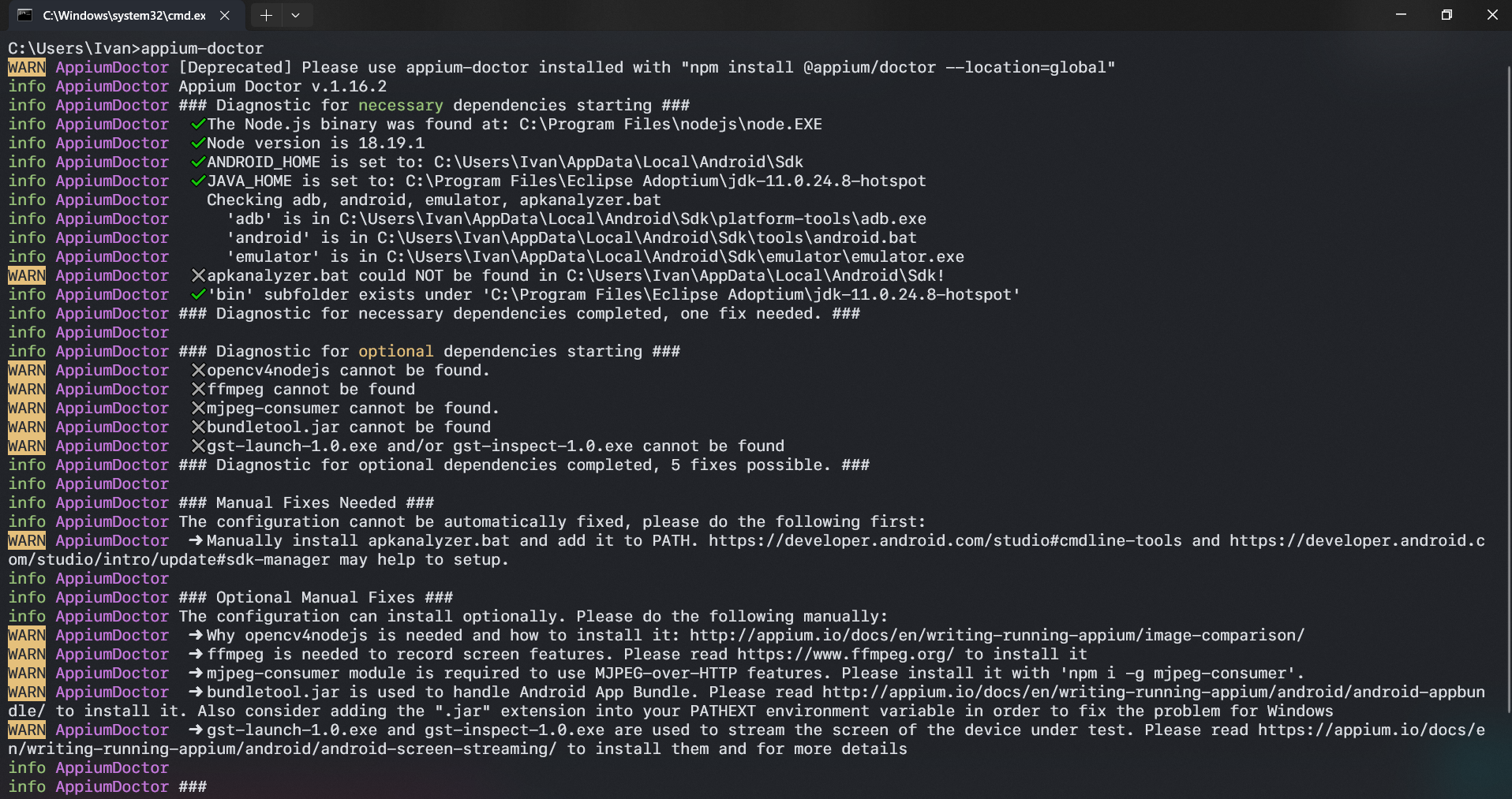
1.6 Встановлюємо Appium:



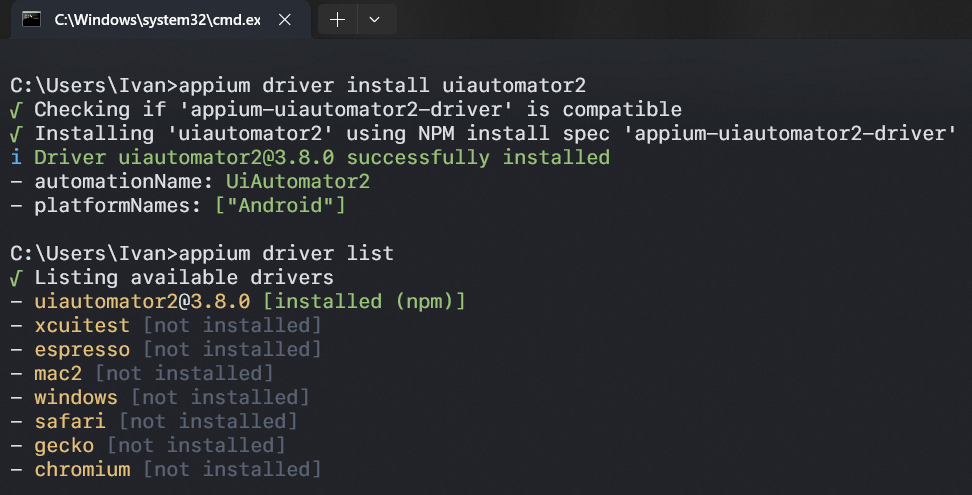
1.7 Встановлюємо Appium Doctor:



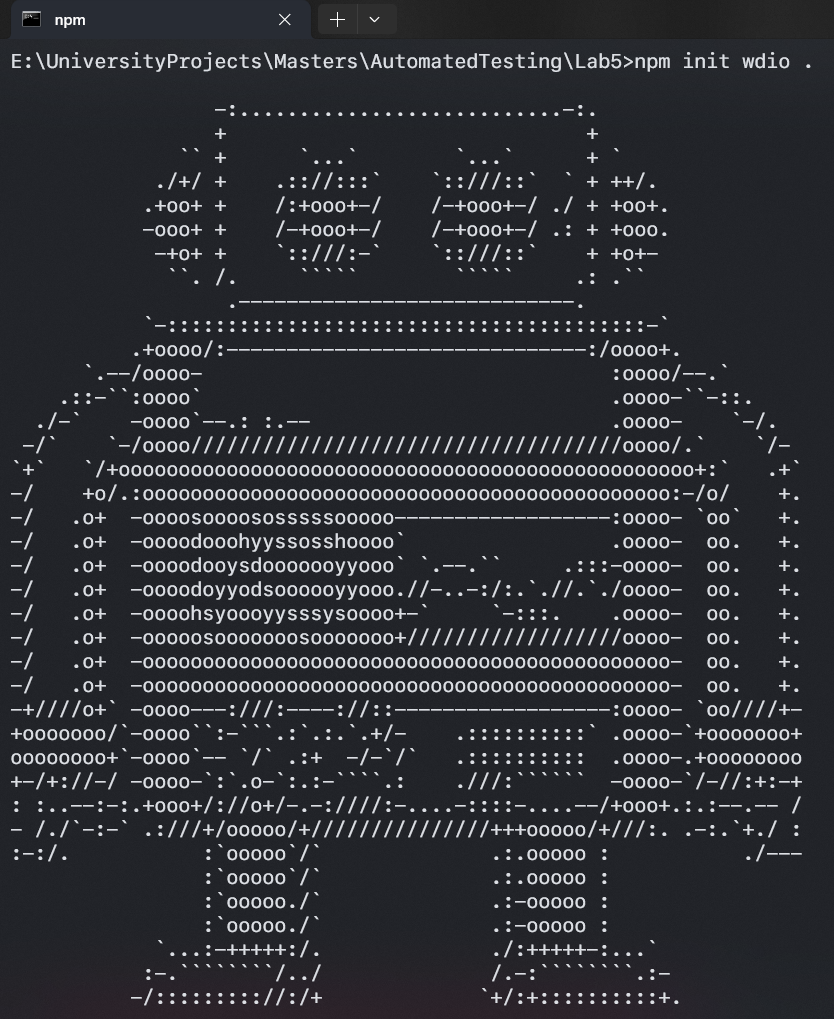
1.8 Запускаємо перевірку appium doctor:

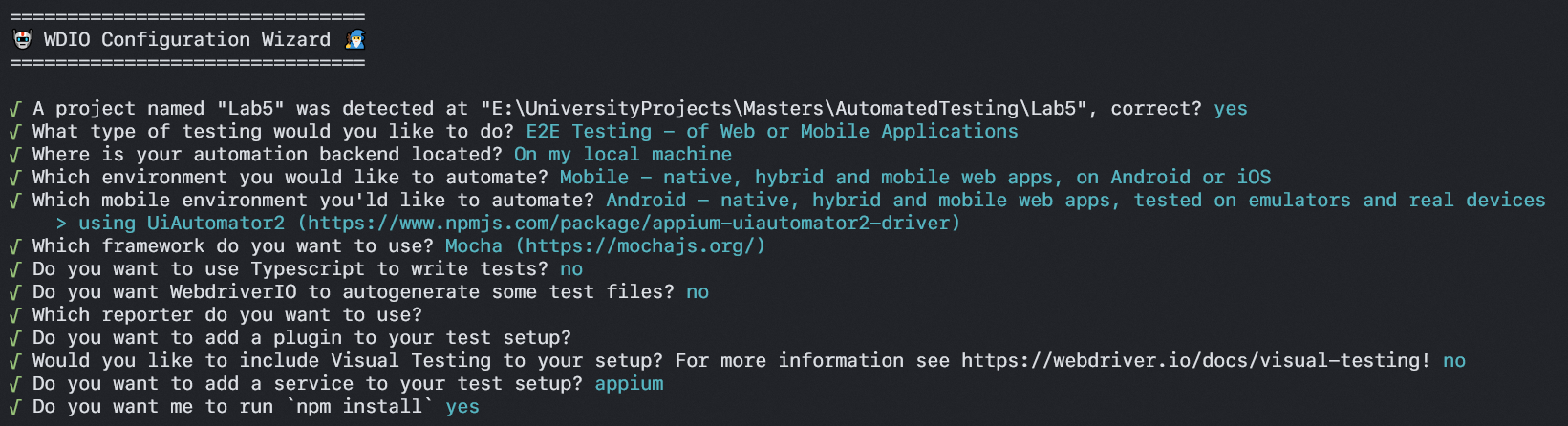


1.9 Встановлюємо драйвер uiautomator2 для appium:

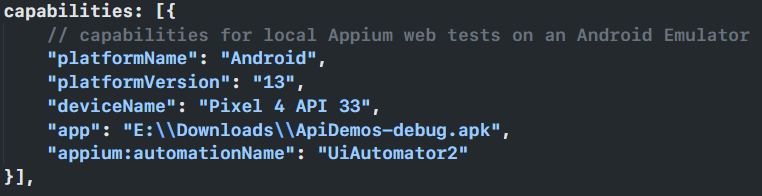


1.10 Створюємо новий проєкт і встановлюємо WebDriverIO:

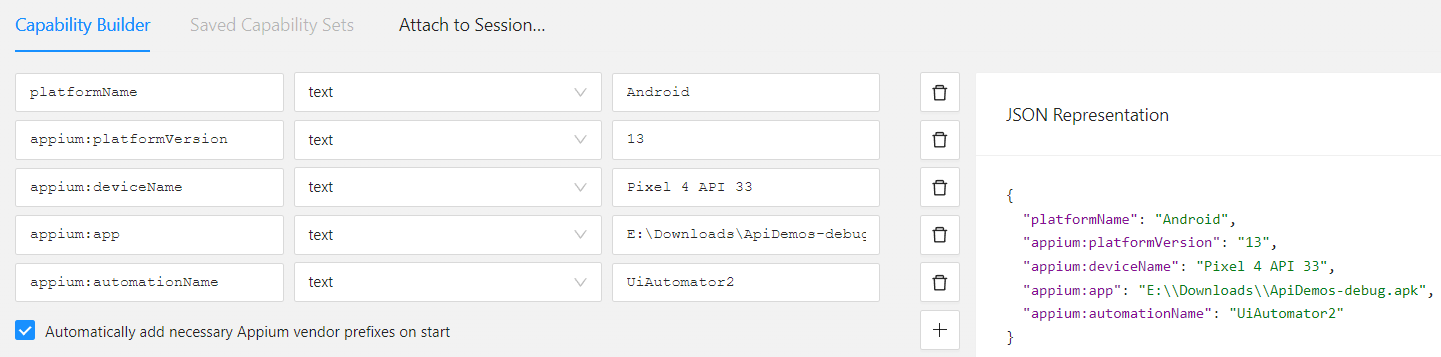




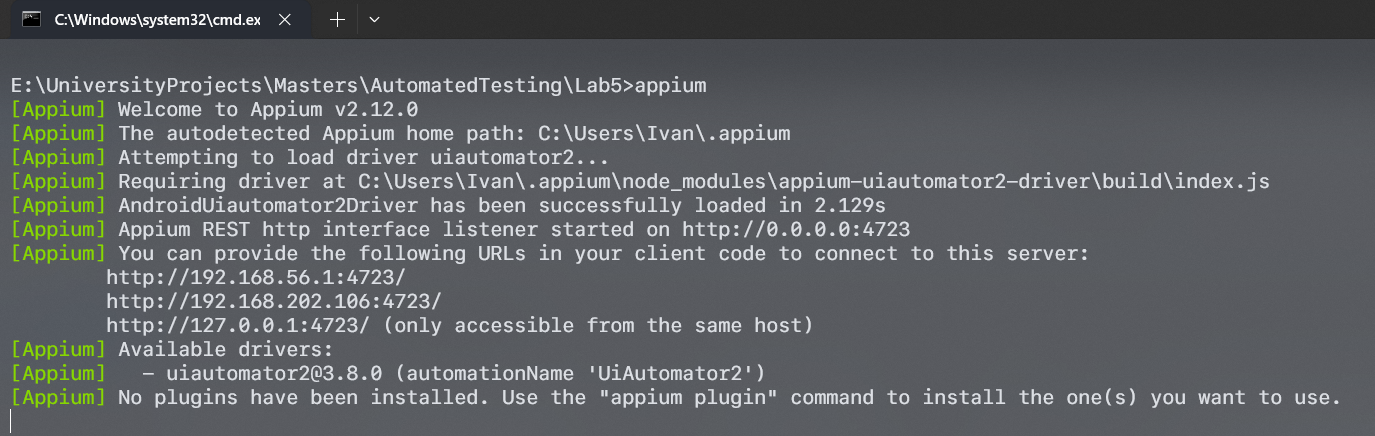
1.11 Змінюємо capabilities у wdio.conf.js згідно створеного емулятору та застосунку, що тестуватиметься:

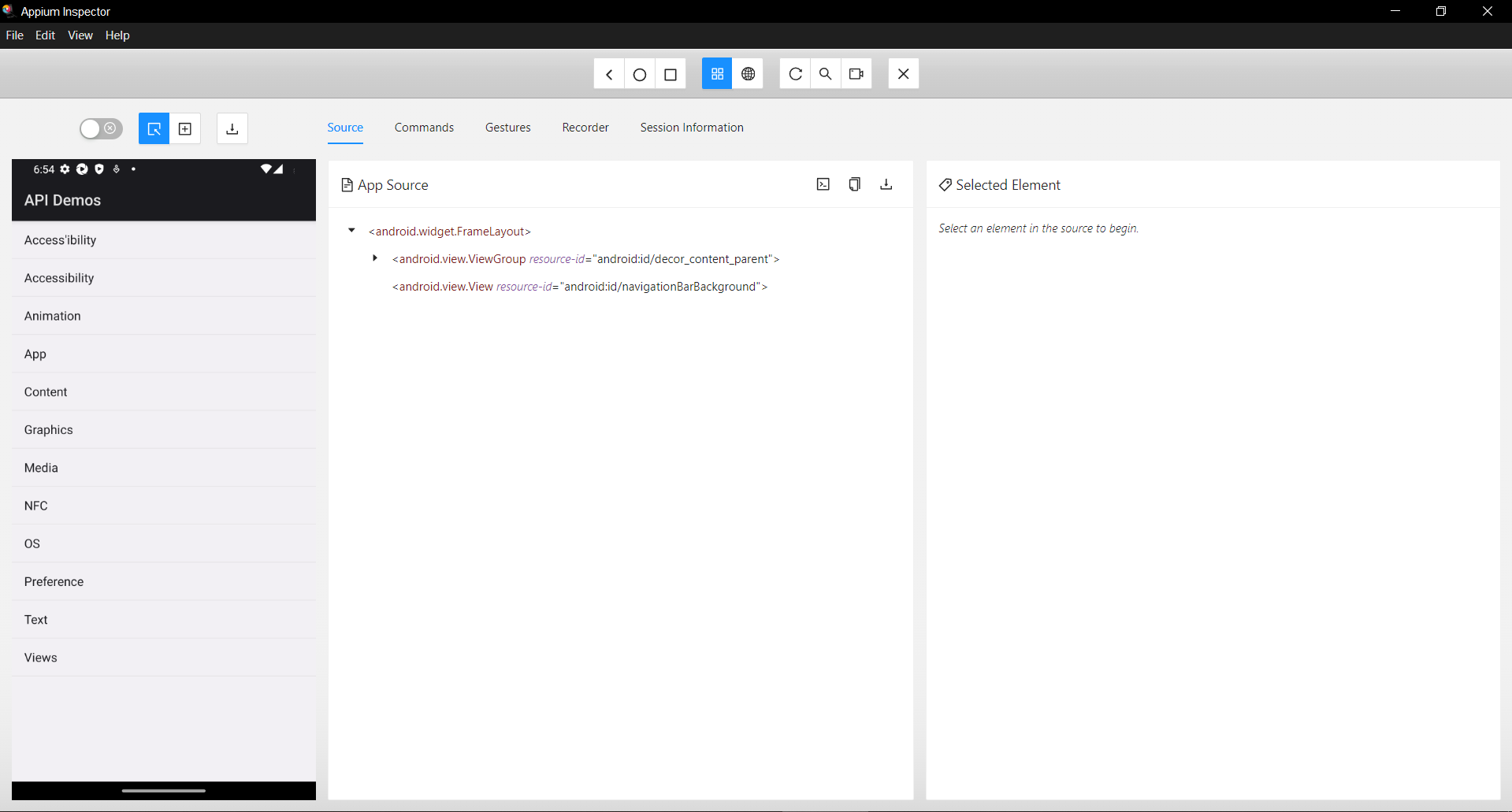


1.12 Змінюємо capabilities у Appium Inspector згідно створеного емулятору та застосунку, що тестуватиметься:

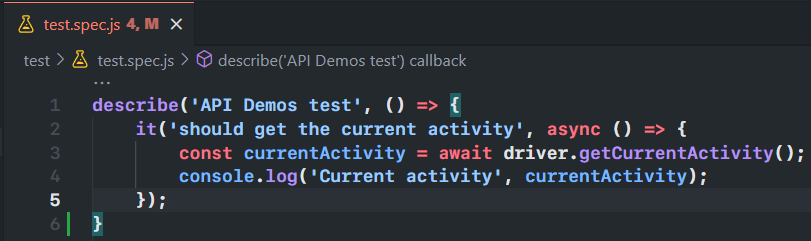


1.13 Запускаємо сервер appium та перевіряємо роботу Appium Inspector:

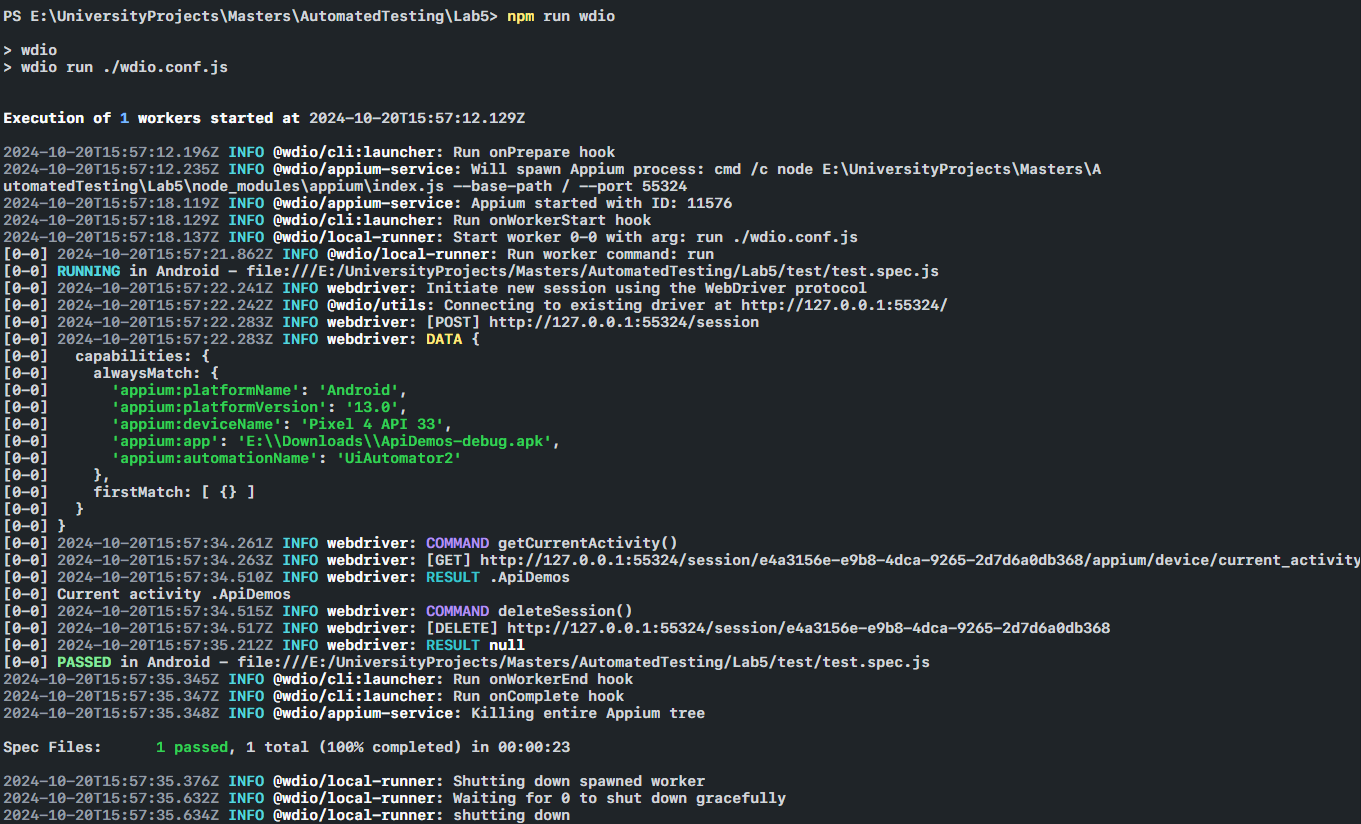




1.14 Створюємо тестовий тест та перевіряємо роботу wdio:



Виконуємо тест:



1. Створюємо скрипт згідно завдання:

**it('should find the "Country" AutoCompleteTextView, input a country name and verify its value', async () => {**

**// access the "Views" TextView by the xpath**

**const viewsTextView = await $('//android.widget.TextView[@content-desc="Views"]');**

**const viewsText = await viewsTextView.getText();**

**console.log('Views TextView text:', viewsText);**

**expect(viewsText).toEqual('Views');**

**// tap on the Views**

**viewsTextView.click();**

**// access the "Auto Complete" TextView by its accessibility id**

**const autoCompleteTextView = await $('~Auto Complete');**

**const autoCompleteText = await autoCompleteTextView.getText();**

**console.log('Auto Complete TextView text:', autoCompleteText);**

**expect(autoCompleteText).toEqual('Auto Complete');**

**// tap on the Auto Complete**

**autoCompleteTextView.click();**

**// access the "Screen Top" TextView by its text**

**const screenTopTextView = await $('//\*[@text="1. Screen Top"]');**

**const screenTopText = await screenTopTextView.getText();**

**console.log('Screen Top TextView text:', screenTopText);**

**expect(screenTopText).toEqual('1. Screen Top');**

**// tap on the Screen Top**

**screenTopTextView.click();**

**// access the AutoCompleteTextView element using UiSelector**

**const selector = 'new UiSelector().resourceId("io.appium.android.apis:id/edit")';**

**const autoCompleteTextViewSelector = await $(`android=${selector}`);**

**console.log('AutoCompleteTextView text:', await autoCompleteTextViewSelector.getText());**

**expect(await autoCompleteTextViewSelector.getText()).toEqual('');**

**// input the country name**

**const inputValue = 'Ukraine';**

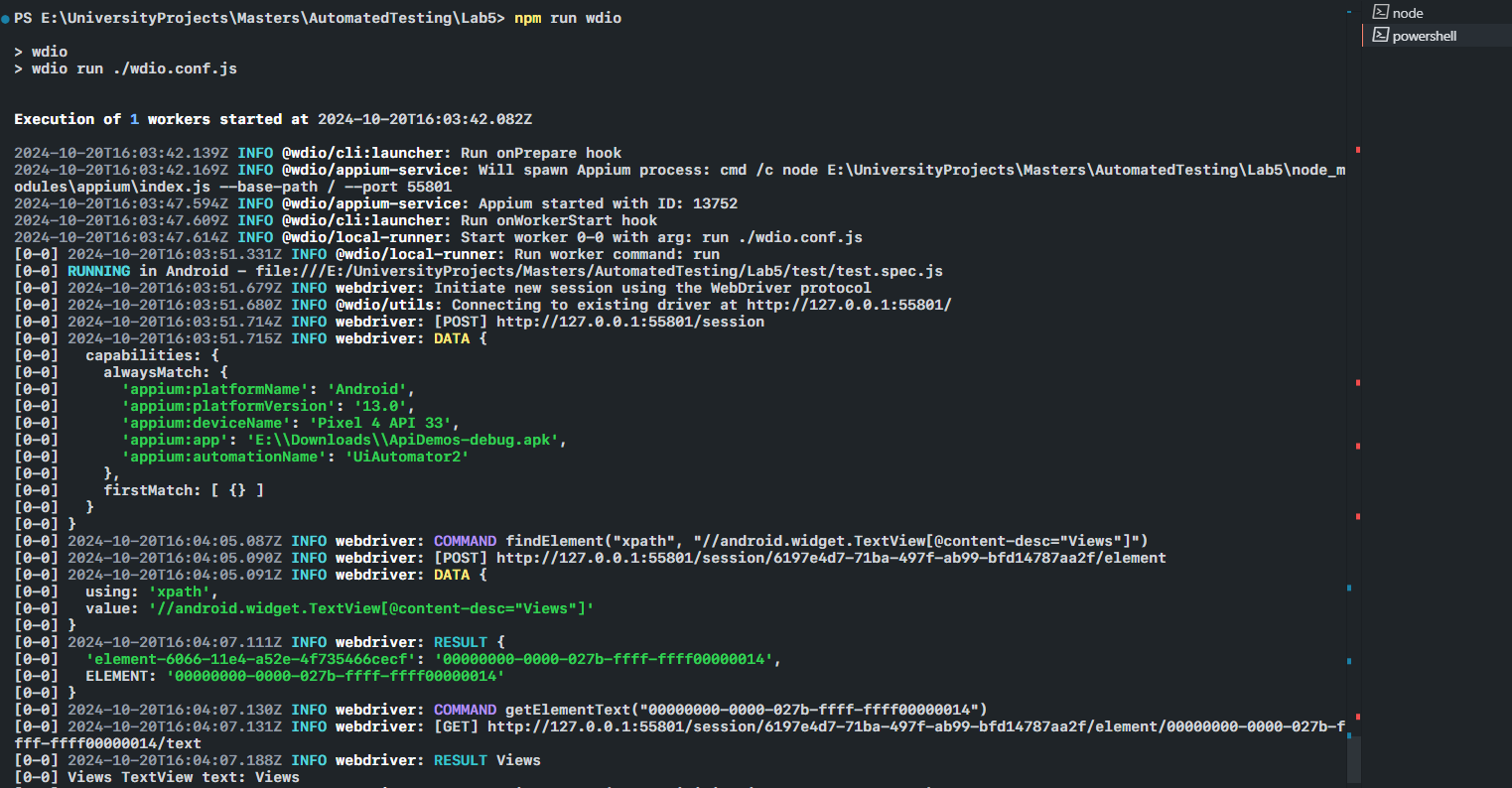
**await autoCompleteTextViewSelector.setValue(inputValue);**

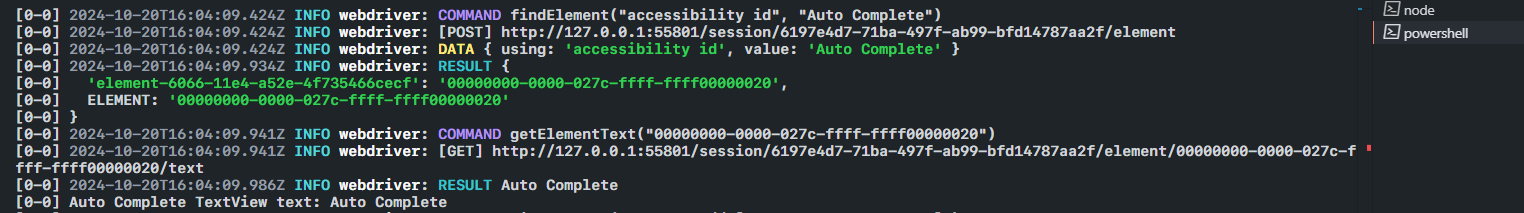
**console.log('Verify AutoCompleteTextView value:', await autoCompleteTextViewSelector.getText());**

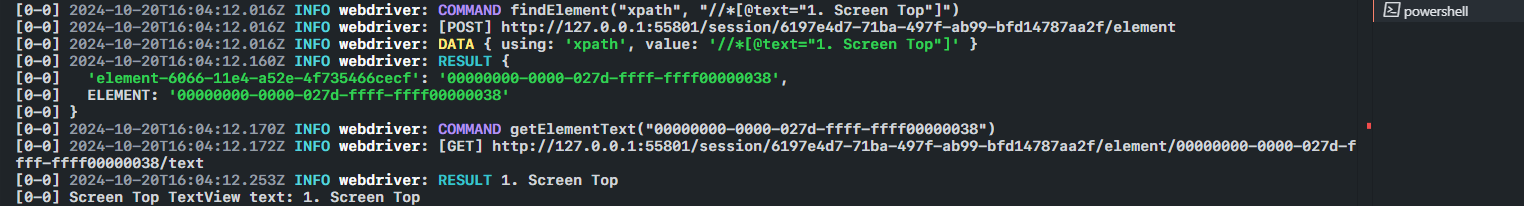
**expect(await autoCompleteTextViewSelector.getText()).toEqual(inputValue);**

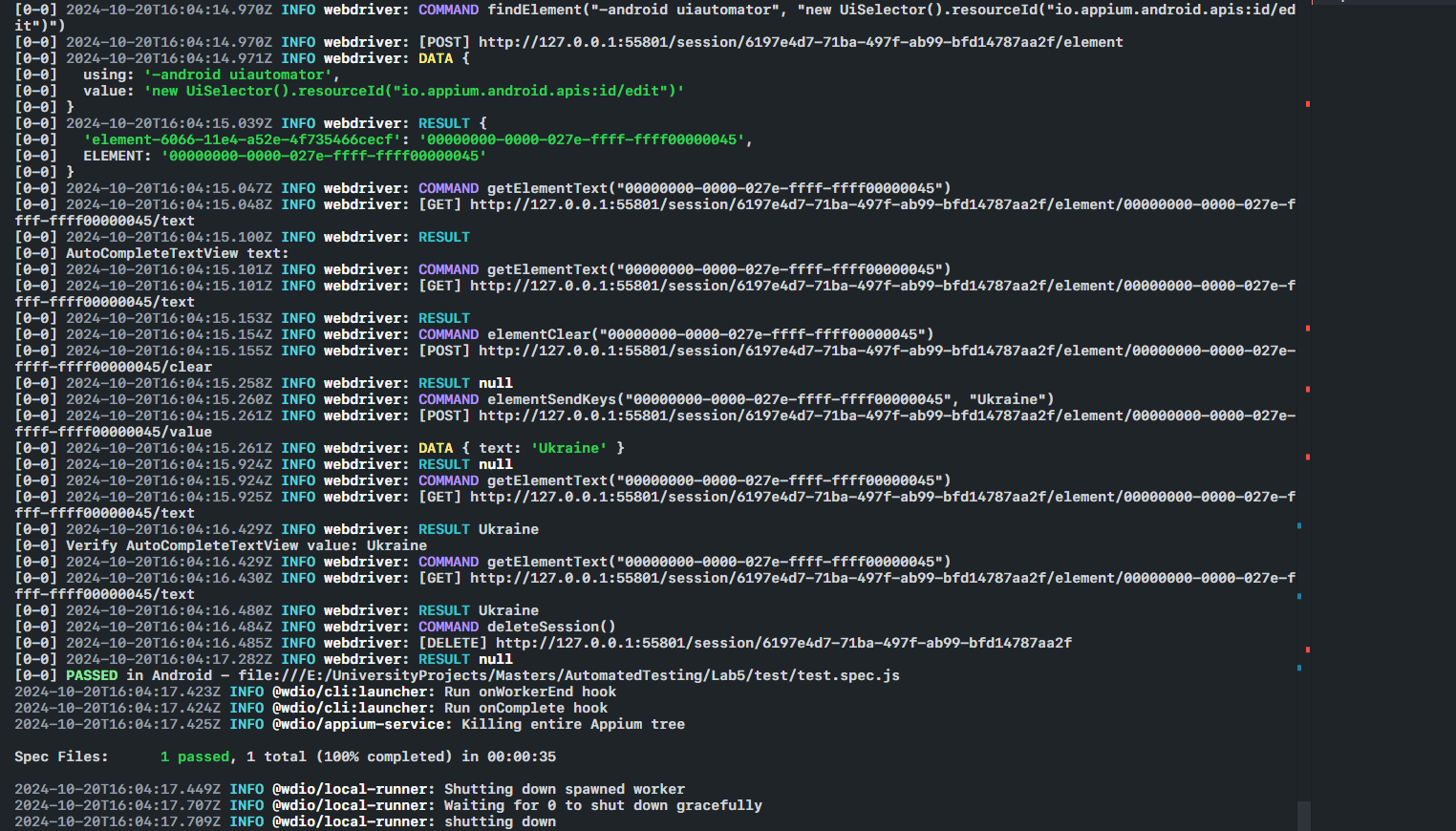
**});**

Виконуємо тест:









1. Створюємо тест для перевірки, що діалог за шляхом App -> Activity -> Dialog відображається і додає елементи при натисканні кнопки Add content:

**it('should find and open the "App/Activity/Dialog" Dialog and get its title value', async () => {**

**// access the "App" TextView**

**const appTextView = await $('//android.widget.TextView[@content-desc="App"]');**

**expect(await appTextView.getText()).toEqual('App');**

**await appTextView.click();**

**// access the "Activity" TextView**

**const activityTextView = await $('~Activity');**

**expect(await activityTextView.getText()).toEqual('Activity');**

**await activityTextView.click();**

**// access the "Dialog" TextView**

**const dialogTextView = await $('~Dialog');**

**expect(await dialogTextView.getText()).toEqual('Dialog');**

**await dialogTextView.click();**

**// access the dialog title TextView**

**const titleSelector = 'new UiSelector().resourceId("android:id/title")';**

**const titleTextView = await $(`android=${titleSelector}`);**

**expect(await titleTextView.getText()).toEqual('App/Activity/Dialog');**

**// access the dialog 'Add content' button**

**const buttonSelector = 'new UiSelector().resourceId("io.appium.android.apis:id/add")';**

**const button = await $(`android=${buttonSelector}`);**

**expect(await button.getText()).toEqual('Add content');**

**await button.click();**

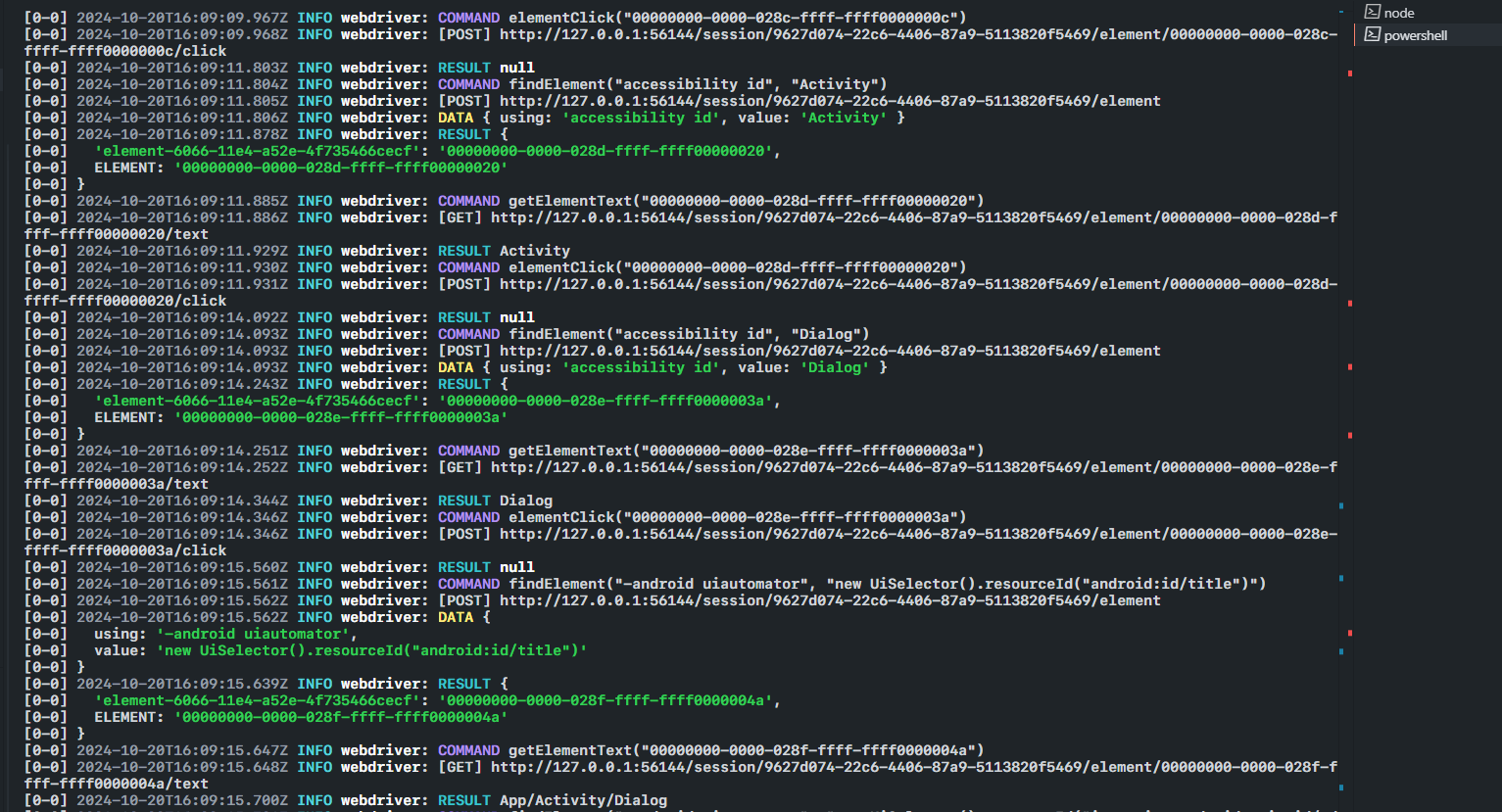
**const imageView = await $('//android.widget.LinearLayout[@resource-id="io.appium.android.apis:id/inner\_content"]/android.widget.ImageView');**

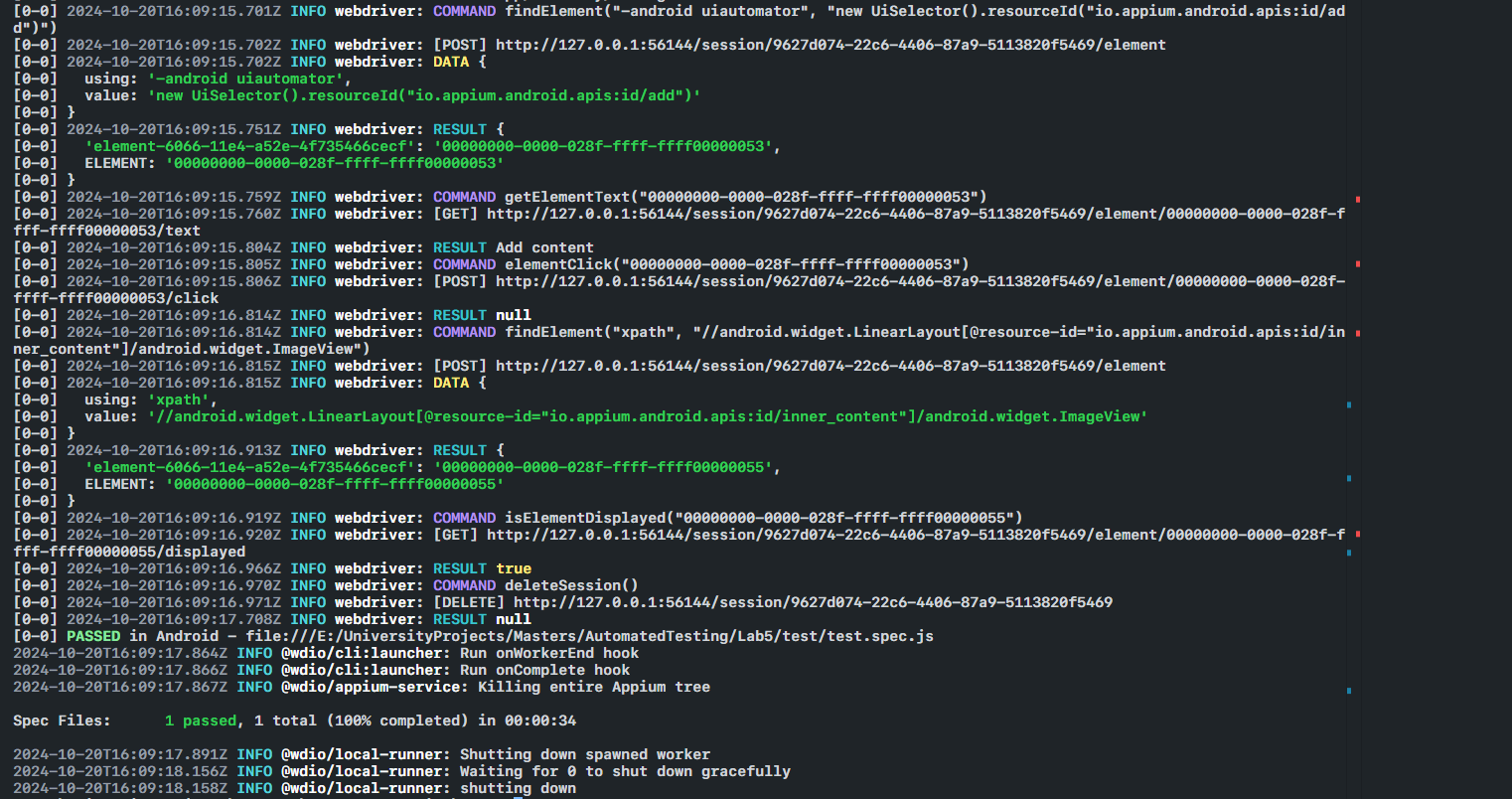
**expect(await imageView.isDisplayed()).toBe(true);**

**});**

Виконуємо тест:







1. Створюємо тест для перевірки, що чекбокс за шляхом Preference -> Default values -> Checkbox відображається і змінює стан при натисканні на нього:

**it.only('should find the "Checkbox preference" checkbox and uncheck it', async () => {**

**// access the "Preference" TextView**

**const preferenceTextView = await $('~Preference');**

**expect(await preferenceTextView.getText()).toEqual('Preference');**

**await preferenceTextView.click();**

**// access the "Default values" TextView**

**const defaultValuesTextView = await $('~4. Default values');**

**expect(await defaultValuesTextView.getText()).toEqual('4. Default values');**

**await defaultValuesTextView.click();**

**// access the checkbox**

**const checkboxSelector = 'new UiSelector().resourceId("android:id/checkbox")';**

**const checkbox = await $(`android=${checkboxSelector}`);**

**expect(await checkbox.getAttribute('checked')).toEqual('true');**

**// change its value**

**await checkbox.click();**

**expect(await checkbox.getAttribute('checked')).toBe('false');**

**});**

Виконуємо тест:

