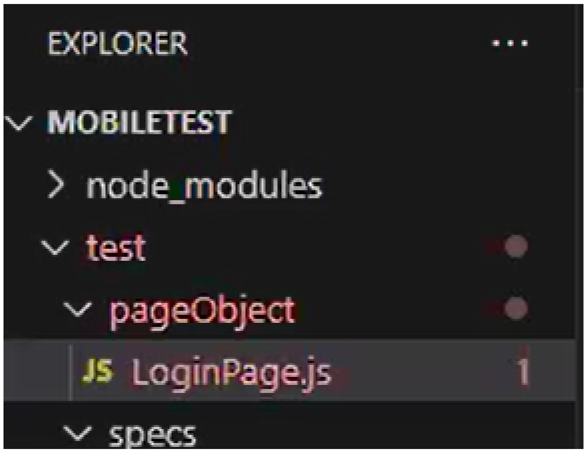
Automating Native Apps and Web Apps Using JavaScript with Appium

Create a new folder named pageObject in a test folder. Create a loginPage.js in a pageObject folder.



Script in the loginPage.js:

```
class loginPage{
     //input [@name = ‘username’] -> username ele
     get usernameField(){
          return
$(‘//android.view.View[@resource-id=”app”]/android.view.
View/android.view.View/android.view.View[3]/android.widg
et.EditText’)
     }
     get passwordField(){
          return
$(‘//android.view.View[@resource-id=”app”]/android.view.
View/android.view.View/android.view.View[6]/android.widg
et.EditText’)
     }
     get loginButton(){
          return
$(‘//android.widget.Button[@text=”Login”]’)
```

```
    }
    get dashboardText(){
        return
$(‘//android.widget.TextView[@text=”Dashboard”]’)
    }
    async enterUsername(username){
        await this.usernameField.setValue(username)
    }
    async enterPassword(password){
        await this.passwordField.setValue(password)
    }
    async clickLoginButton(){
        await this.loginButton.click();
    }
    async verifyLogin(){
        return this.dashboardText;
    }
    async login(username, password){
        await this.enterUsername(username)
        await this.enterPassword(password)
        await this.clickLoginButton();
    }
}
module.export = new loginPage();
```

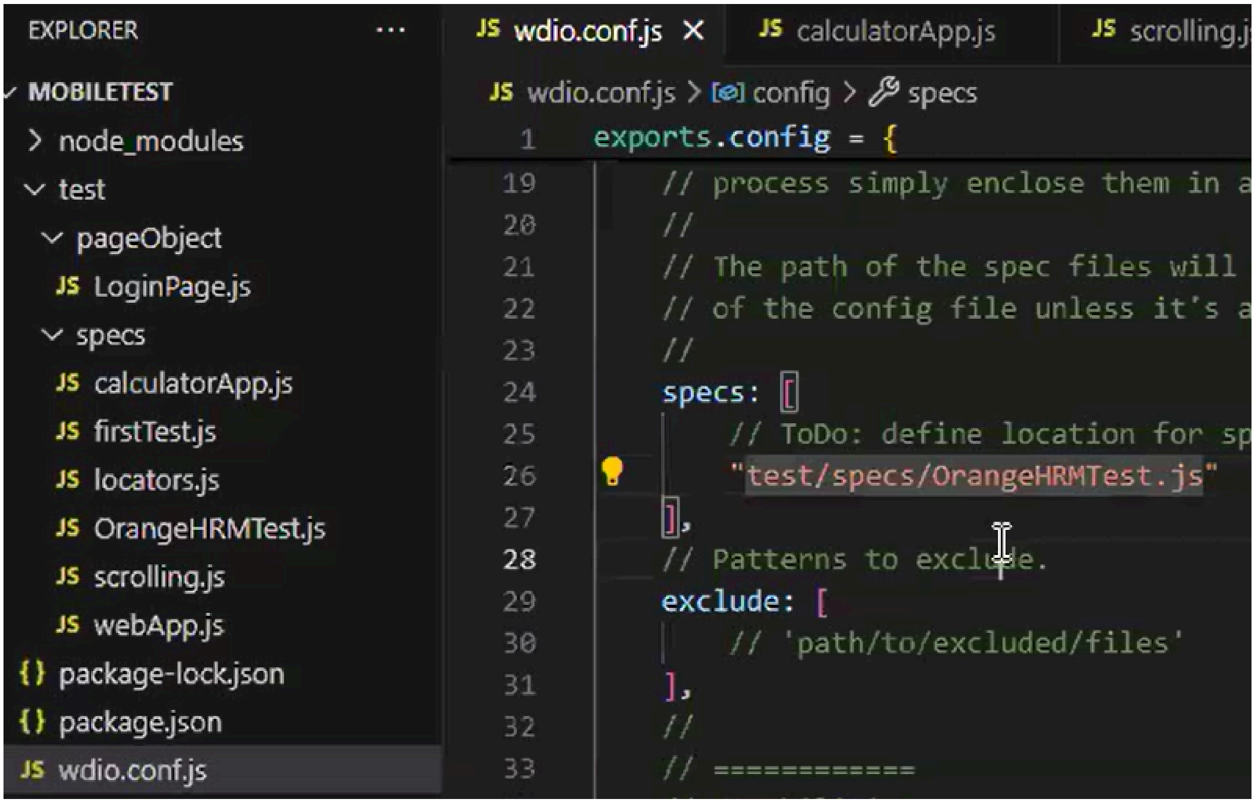
Create OrangeHRMTest.js in the specs folder:

Script in the OrangeHRMTest.js:

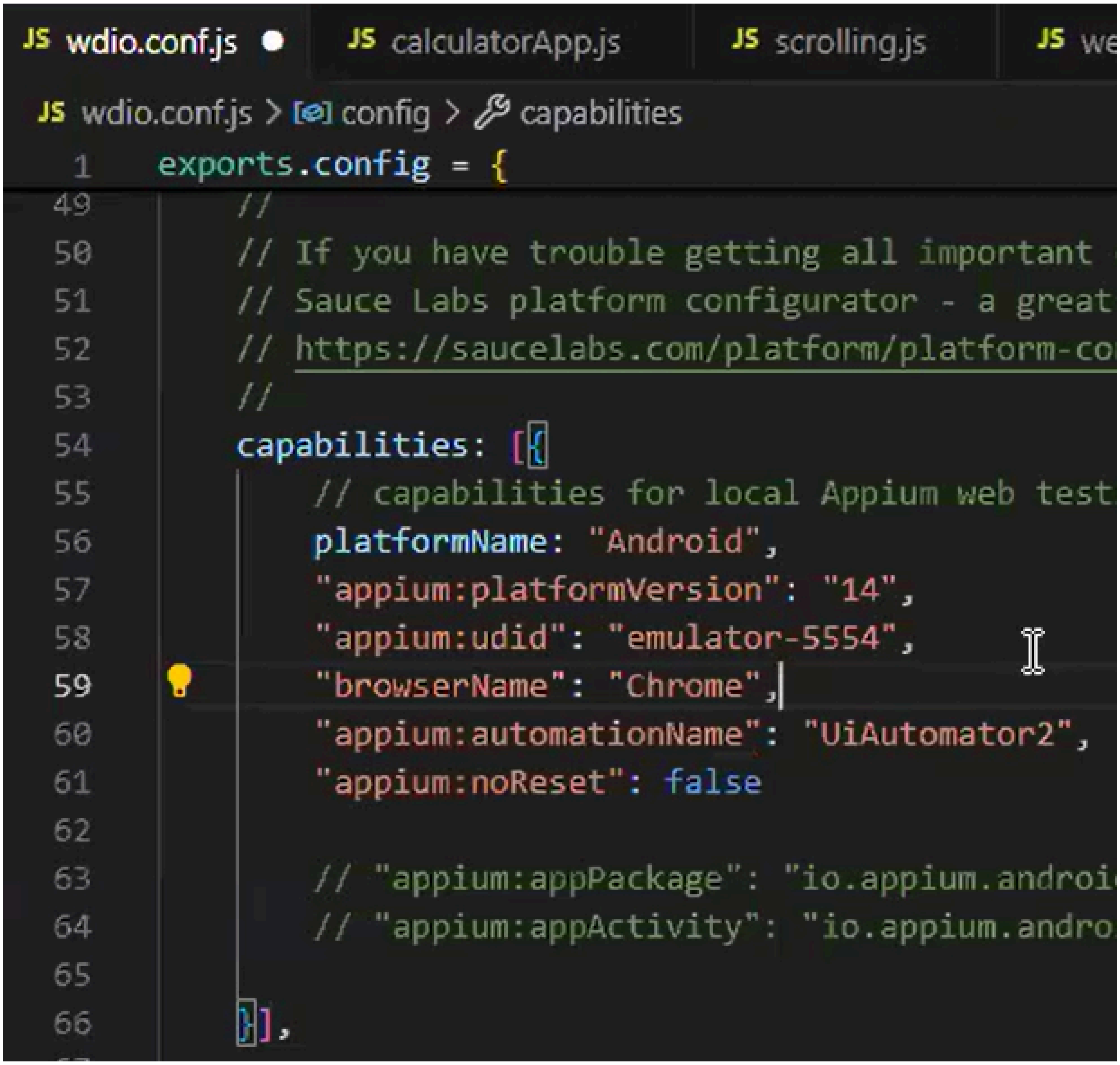
```
const LoginPage = require(‘../pageObject/LoginPage’)
describe(‘Testing Orange HRM’, () => {
    it(‘Login page test’, async() => {
        await browser.pause(2000);
        await
$(‘//android.widget.TextView[@context-desc=”Chrome”]’).c
lick()
        await browser.pause(2000)

        await
    browser.url(“https://opensource-demo.orangehrmlive.c
    om/web/index.php/auth/login”)
        await browser.pause(2000)
        await LoginPage.login(‘Admin’, ‘admin123’)
        await browser.pause(3000)
        expect(LoginPage.dashboardText).toExist()
    })
})
```

Make change in spec:



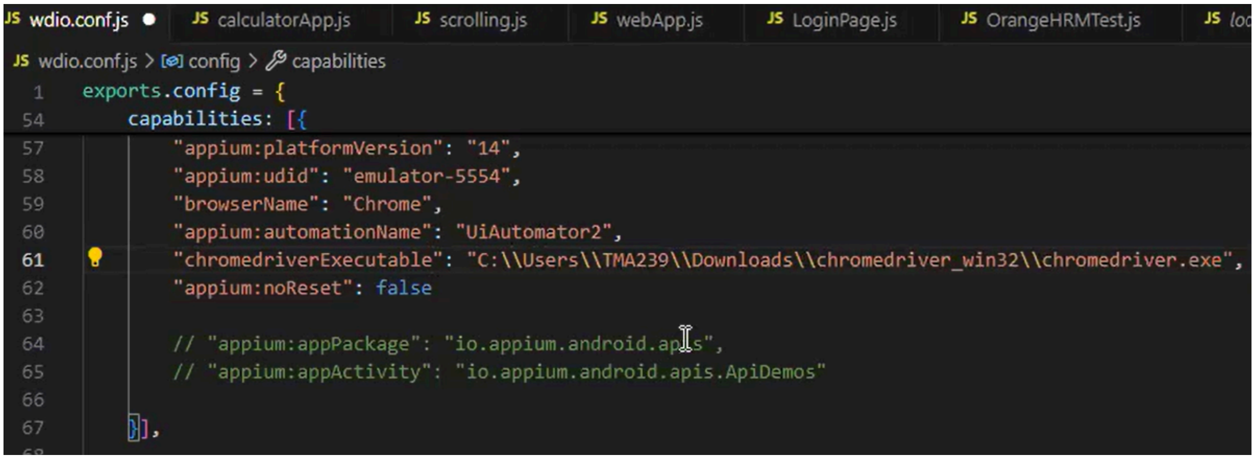
```
exports.config = {
    // process simply enclose them in a
    //
    // The path of the spec files will
    // of the config file unless it's a
    //
    specs: [
        // ToDo: define location for sp
        "test/specs/OrangeHRMTest.js"
    ],
    // Patterns to exclude.
    exclude: [
        // 'path/to/excluded/files'
    ],
    //
    // ===========
```



```
exports.config = {
    //
    // If you have trouble getting all important
    // Sauce Labs platform configurator - a great
    // https://saucelabs.com/platform/platform-co
    //
    capabilities: [{
        // capabilities for local Appium web test
        platformName: "Android",
        "appium:platformVersion": "14",
        "appium:udid": "emulator-5554",
        "browserName": "Chrome",
        "appium:automationName": "UiAutomator2",
        "appium:noReset": false

        // "appium:appPackage": "io.appium.androi
        // "appium:appActivity": "io.appium.andro

    }],
```



```
exports.config = {
    capabilities: [{
        "appium:platformVersion": "14",
        "appium:udid": "emulator-5554",
        "browserName": "Chrome",
        "appium:automationName": "UiAutomator2",
        "chromedriverExecutable": "C:\\Users\\TMA239\\Downloads\\chromedriver_win32\\chromedriver.exe",
        "appium:noReset": false

        // "appium:appPackage": "io.appium.android.apis",
        // "appium:appActivity": "io.appium.android.apis.ApiDemos"

    }],
```

In the git bash under the TERMINAL:

```
> npx install chromedriver - -save-dev
> npx wdio
```