selenium_testing

⏱ 45.707s 2 5

✓ 5 ✕ 0

MWM tests

\src\test\MWMselenium.ts

App run through

\src\test\MWMselenium.ts

⏱ 45.608s 5 ✓ 5

✓ Open MWM and check URL correct 5.587s ⏱

```
return __awaiter(void 0, void 0, void 0, function () {
                var driver, currentUrl;
                return __generator(this, function (_a) {
                    switch (_a.label) {
                        case 0: return [4 /*yield*/, new selenium_webdriver_1.Builder().forBrowser('chrome'
                        case 1:
                            driver = _a.sent();
                            return [4 /*yield*/, driver.get('http://2a246.yeg.rac.sh/')];
                        case 2:
                            _a.sent();
                            return [4 /*yield*/, driver.getCurrentUrl()];
                        case 3:
                            currentUrl = _a.sent();
                            // make sure correct URL opened
                            assert.strictEqual('http://2a246.yeg.rac.sh/', currentUrl, 'Correct URL not ope
                            return [2 /*return*/];
                    }
                });
            });
```

✓ MWM sign up 8.906s ⏱

```
return __awaiter(void 0, void 0, void 0, function () {
                var driver, nameElements, passwordElements, notMatchingPasswords;
                return __generator(this, function (_a) {
                    switch (_a.label) {
                        case 0: return [4 /*yield*/, new selenium_webdriver_1.Builder().forBrowser('chrome'
                        case 1:
                            driver = _a.sent();
                            return [4 /*yield*/, driver.get('http://2a246.yeg.rac.sh/')];
                        case 2:
                            _a.sent();
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.xpath("//a[@hre
                        case 3:
                        // opening signup page
```

```
    sleep(0.2);
    return [4 /*yield*/, nameElements[0].sendKeys('firstName')];
case 6:
    _a.sent();
    return [4 /*yield*/, nameElements[1].sendKeys('lastName')];
case 7:
    _a.sent();
    return [4 /*yield*/, nameElements[2].sendKeys(username)];
case 8:
    _a.sent();
    return [4 /*yield*/, driver.findElements(selenium_webdriver_1.By.id('outlined-a
case 9:
    passwordElements = _a.sent();
    // sending keys to passwords that do not match
    return [4 /*yield*/, passwordElements[0].sendKeys('password')];
case 10:
    // sending keys to passwords that do not match
    _a.sent();
    return [4 /*yield*/, passwordElements[1].sendKeys('password2')];
case 11:
    _a.sent();
    return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.className('MuiB
case 12:
// click signup button
return [4 /*yield*/, (_a.sent()).click()];
case 13:
    // click signup button
    _a.sent();
    sleep(2);
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
        // testing that passwords not matching brings up a message to the user sayi
    ];
case 14:
    notMatchingPasswords = _a.sent();
    // testing that passwords not matching brings up a message to the user saying p
    assert.strictEqual(notMatchingPasswords, 'Passwords must match!', 'Password not
    // clearing password elements to put a correct password in
    return [4 /*yield*/, passwordElements[0].sendKeys(selenium_webdriver_1.Key.CONT
case 15:
    // clearing password elements to put a correct password in
    _a.sent();
    return [4 /*yield*/, passwordElements[0].sendKeys(selenium_webdriver_1.Key.DELE
case 16:
    _a.sent();
    return [4 /*yield*/, passwordElements[1].sendKeys(selenium_webdriver_1.Key.CONT
case 17:
    _a.sent();
    return [4 /*yield*/, passwordElements[1].sendKeys(selenium_webdriver_1.Key.DELE
case 18:
    _a.sent();
    // putting in correct passwords
    return [4 /*yield*/, passwordElements[0].sendKeys(password)];
case 19:
    // putting in correct passwords
    _a.sent();
    return [4 /*yield*/, passwordElements[1].sendKeys(password)];
case 20:
```

```
                            return [2 /*return*/];
                    }
                });
            });
```

✓ Logs into app, showing error handling when incorrect password or empty field present 7.496s ⏱

```
return __awaiter(void 0, void 0, void 0, function () {
                var driver, userNameElement, passwordElement, loginButton, errorElement, errorMessage, curr
                return __generator(this, function (_a) {
                    switch (_a.label) {
                        case 0: return [4 /*yield*/, new selenium_webdriver_1.Builder().forBrowser('chrome'
                        case 1:
                            driver = _a.sent();
                            return [4 /*yield*/, driver.get('http://2a246.yeg.rac.sh/')];
                        case 2:
                            _a.sent();
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.id('outlined-ma
                        case 3:
                            userNameElement = _a.sent();
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.id('outlined-ad
                        case 4:
                            passwordElement = _a.sent();
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.className('MuiB
                        case 5:
                            loginButton = _a.sent();
                            return [4 /*yield*/, userNameElement.sendKeys('user')];
                        case 6:
                            _a.sent();
                            return [4 /*yield*/, loginButton.click()];
                        case 7:
                            _a.sent();
                            // quick wait to ensure element has been updated
                            sleep(1);
                            return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
                        case 8:
                            errorElement = _a.sent();
                            return [4 /*yield*/, errorElement.getText()];
                        case 9:
                            errorMessage = _a.sent();
                            assert.strictEqual(errorMessage, 'Please enter username and password.', 'Empty 
                            return [4 /*yield*/, passwordElement.sendKeys('pass')];
                        case 10:
                            _a.sent();
                            return [4 /*yield*/, loginButton.click()];
                        case 11:
                            _a.sent();
                            // quick wait to ensure element loaded with error message
                            //driver.manage().setTimeouts({implicit:2000});
                            sleep(1);
                            return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
                        case 12:
                            errorElement = _a.sent();
                            _a.sent();
                            return [4 /*yield*/, userNameElement.sendKeys(selenium_webdriver_1.Key.DELETE)]
                        case 15:
                            _a.sent();
                            return [4 /*yield*/, passwordElement.sendKeys(selenium_webdriver_1.Key.CONTROL
                        case 16:
                            _a.sent();
                            return [4 /*yield*/, passwordElement.sendKeys(selenium_webdriver_1.Key.DELETE)]
                        case 17:
                            _a.sent();
                            return [4 /*yield*/, userNameElement.sendKeys(username)];
                        case 18:
                            _a.sent();
                            return [4 /*yield*/, passwordElement.sendKeys(password)];
                        case 19:
                            _a.sent();
                            return [4 /*yield*/, loginButton.click()];
                        case 20:
                            _a.sent();
                            // make sure elements in dashboard located before continuing
                            return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
                        case 21:
                            // make sure elements in dashboard located before continuing
                            _a.sent();
                            return [4 /*yield*/, driver.getCurrentUrl()];
                        case 22:
                            currentUrl = _a.sent();
                            // testing that logging in takes the user to the dashboard
                            assert.strictEqual('http://2a246.yeg.rac.sh/dashboard', currentUrl, 'Should be
                            return [2 /*return*/];
                    }
                });
            });
```

✓ Create a team and match 12.337s

```
return __awaiter(void 0, void 0, void 0, function () {
                var driver, userNameElement, passwordElement, loginButton, teamsButton, currentURL, teamPag
                return __generator(this, function (_a) {
                    switch (_a.label) {
                        case 0: return [4 /*yield*/, new selenium_webdriver_1.Builder().forBrowser('chrome'
                        case 1:
                            driver = _a.sent();
                            return [4 /*yield*/, driver.get('http://2a246.yeg.rac.sh/')];
                        case 2:
                            _a.sent();
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.id('outlined-ma
                        case 3:
                            userNameElement = _a.sent();
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.id('outlined-ad
                        case 4:
                            passwordElement = _a.sent();
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.className('MuiB
    // log in
    return [4 /*yield*/, loginButton.click()];
case 8:
    // log in
    _a.sent();
    // wait until expected component available as sometimes loads slowly here
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
case 9:
    // wait until expected component available as sometimes loads slowly here
    _a.sent();
    return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.xpath("//a[@hre
case 10:
    teamsButton = _a.sent();
    return [4 /*yield*/, teamsButton.click()];
case 11:
    _a.sent();
    return [4 /*yield*/, driver.getCurrentUrl()];
case 12:
    currentURL = _a.sent();
    // testing to see taken to teams page
    assert.strictEqual(currentURL, 'http://2a246.yeg.rac.sh/teams', 'Teams page loa
    return [4 /*yield*/, driver.findElements(selenium_webdriver_1.By.className('Mui
case 13:
    teamPageButtons = _a.sent();
    addTeamButton = teamPageButtons[teamPageButtons.length - 2];
    return [4 /*yield*/, addTeamButton.click()];
case 14:
    _a.sent();
    sleep(1);
    return [4 /*yield*/, driver.getCurrentUrl()];
case 15:
    currentURL = _a.sent();
    // should be on create team page now
    assert.strictEqual(currentURL, 'http://2a246.yeg.rac.sh/create-team', 'Should n
    return [4 /*yield*/, driver.findElements(selenium_webdriver_1.By.xpath("//input
case 16:
    textBoxesAddTeam = _a.sent();
    randName = Math.floor(Math.random() * 100000000);
    return [4 /*yield*/, driver.findElements(selenium_webdriver_1.By.className('Mui
case 17:
    createTeamButtons = _a.sent();
    return [4 /*yield*/, textBoxesAddTeam[0].sendKeys(randName)];
case 18:
    _a.sent();
    // populating players
    return [4 /*yield*/, textBoxesAddTeam[1].sendKeys('Joey')];
case 19:
    // populating players
    _a.sent();
    return [4 /*yield*/, textBoxesAddTeam[2].sendKeys('Banks')];
case 20:
    _a.sent();
    return [4 /*yield*/, textBoxesAddTeam[3].sendKeys(1)];
case 21:
    _a.sent();
    return [4 /*yield*/, createTeamButtons[0].click()];
case 22:
```

```
                        case 25:
                            _a.sent();
                            return [4 /*yield*/, createTeamButtons[0].click()];
                        case 26:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[1].sendKeys('Mr')];
                        case 27:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[2].sendKeys('Moo')];
                        case 28:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[3].sendKeys(3)];
                        case 29:
                            _a.sent();
                            return [4 /*yield*/, createTeamButtons[0].click()];
                        case 30:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[1].sendKeys('GoodOl')];
                        case 31:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[2].sendKeys('Lou')];
                        case 32:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[3].sendKeys(4)];
                        case 33:
                            _a.sent();
                            return [4 /*yield*/, createTeamButtons[0].click()];
                        case 34:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[1].sendKeys('Red')];
                        case 35:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[2].sendKeys('Hot')];
                        case 36:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[3].sendKeys(5)];
                        case 37:
                            _a.sent();
                            return [4 /*yield*/, createTeamButtons[0].click()];
                        case 38:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[1].sendKeys('Mr')];
                        case 39:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[2].sendKeys('Robot')];
                        case 40:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[3].sendKeys(6)];
                        case 41:
                            _a.sent();
                            return [4 /*yield*/, createTeamButtons[0].click()];
                        case 42:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[1].sendKeys('Slow')];
                        case 43:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[2].sendKeys('Poke')];
                            return [4 /*yield*/, textBoxesAddTeam[1].sendKeys('No')];
                        case 47:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[2].sendKeys('Joke')];
                        case 48:
                            _a.sent();
                            return [4 /*yield*/, textBoxesAddTeam[3].sendKeys(8)];
                        case 49:
                            _a.sent();
                            return [4 /*yield*/, createTeamButtons[0].click()];
                        case 50:
                            _a.sent();
                            // create the team
                            return [4 /*yield*/, createTeamButtons[1].click()];
                        case 51:
                            // create the team
                            _a.sent();
                            // back to dashboard , just used get to simplify, but createTeamButton[2] would
                            return [4 /*yield*/, driver.get('http://2a246.yeg.rac.sh/dashboard')];
                        case 52:
                            // back to dashboard , just used get to simplify, but createTeamButton[2] would
                            _a.sent();
                            // make a match with our new team
                            return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
                        case 53:
                            // make a match with our new team
                            _a.sent();
                            return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementsLocated(sel
                        case 54:
                            clickablesWithSecondMenu = _a.sent();
                            clickablesWithSecondMenu[0].click();
                            sleep(1);
                            return [4 /*yield*/, driver.getCurrentUrl()];
                        case 55:
                            currentURL = _a.sent();
                            assert.strictEqual(currentURL, 'http://2a246.yeg.rac.sh/create-match', 'Correct
                            return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementsLocated(sel
                        case 56:
                            teamsAvailable = _a.sent();
                            return [4 /*yield*/, teamsAvailable[1].click()];
                        case 57:
                            _a.sent();
                            // give opponent team name
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.id("standard-ba
                        case 58:
                            // give opponent team name
                            _a.sent();
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.className("MuiB
                        case 59:
                        // create the team
                        return [4 /*yield*/, (_a.sent()).click()];
                        case 60:
                            // create the team
                            _a.sent();
                            return [2 /*return*/];
                    }
```

```
return __awaiter(void 0, void 0, void 0, function () {
                var driver, userNameElement, passwordElement, loginButton, matchToRecord, currentURL, check
                return __generator(this, function (_a) {
                    switch (_a.label) {
                        case 0: return [4 /*yield*/, new selenium_webdriver_1.Builder().forBrowser('chrome'
                        case 1:
                            driver = _a.sent();
                            return [4 /*yield*/, driver.get('http://2a246.yeg.rac.sh/')];
                        case 2:
                            _a.sent();
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.id('outlined-ma
                        case 3:
                            userNameElement = _a.sent();
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.id('outlined-ad
                        case 4:
                            passwordElement = _a.sent();
                            return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.className('MuiB
                        case 5:
                            loginButton = _a.sent();
                            return [4 /*yield*/, userNameElement.sendKeys(username)];
                        case 6:
                            _a.sent();
                            return [4 /*yield*/, passwordElement.sendKeys(password)];
                        case 7:
                            _a.sent();
                            // log in
                            return [4 /*yield*/, loginButton.click()];
                        case 8:
                            // log in
                            _a.sent();
                            return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
                        case 9:
                        // wait until element on dashboard available then click it
                        return [4 /*yield*/, (_a.sent()).click()];
                        case 10:
                            // wait until element on dashboard available then click it
                            _a.sent();
                            return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
                        case 11:
                            matchToRecord = _a.sent();
                            return [4 /*yield*/, driver.getCurrentUrl()];
                        case 12:
                            currentURL = _a.sent();
                            assert.strictEqual('http://2a246.yeg.rac.sh/matches/upcoming', currentURL, "Cli
                            // click on first match to be recorded, will be available from last test
                            return [4 /*yield*/, matchToRecord.click()];
                        case 13:
                            // click on first match to be recorded, will be available from last test
                            _a.sent();
                            return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementsLocated(sel
                        case 14:
                            checkBoxesForLineup = _a.sent();
                            return [4 /*yield*/, driver.getCurrentUrl()];
                        case 15:
                            currentURL2 = _a.sent();
                            assert.strictEqual('http://2a246.yeg.rac.sh/match/lineup', currentURL2, "Clicki
case 18:
    i++;
    return [3 /*break*/, 16];
case 19: return [4 /*yield*/, driver.findElements(selenium_webdriver_1.By.className
case 20:
    matchLineupButtons = _a.sent();
    // click next button
    return [4 /*yield*/, matchLineupButtons[1].click()];
case 21:
    // click next button
    _a.sent();
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
case 22:
// wait for start button to load and click it to start timer
return [4 /*yield*/, (_a.sent()).click()
    // assert we are on loading screen now that elements loaded
];
case 23:
    // wait for start button to load and click it to start timer
    _a.sent();
    return [4 /*yield*/, driver.getCurrentUrl()];
case 24:
    currentURL3 = _a.sent();
    assert.strictEqual(currentURL3, 'http://2a246.yeg.rac.sh/match/recording', 'Che
    // sleep for a second to make sure timer has incremented
    sleep(1);
    return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.className('Time
case 25:
    timer = _a.sent();
    return [4 /*yield*/, timer.getText()];
case 26:
    timerTime = _a.sent();
    assert.notStrictEqual(timerTime, '00 : 00 : 00 : 00', 'Ensure timer has started
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
case 27:
// pause button should now be present, click it and end the half
return [4 /*yield*/, (_a.sent()).click()];
case 28:
    // pause button should now be present, click it and end the half
    _a.sent();
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
case 29:
// now click end half
return [4 /*yield*/, (_a.sent()).click()];
case 30:
    // now click end half
    _a.sent();
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
case 31:
// start second half and end it, buttons have been changing need to locate again
// clicks start to start second half
return [4 /*yield*/, (_a.sent()).click()];
case 32:
    // start second half and end it, buttons have been changing need to locate agai
    // clicks start to start second half
    _a.sent();
    //find period display and assert it is now the second half
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
case 35:
// click pause
return [4 /*yield*/, (_a.sent()).click()];
case 36:
    // click pause
    _a.sent();
    // click end game
    sleep(0.5);
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
case 37: return [4 /*yield*/, (_a.sent()).click()];
case 38:
    _a.sent();
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
case 39:
// we will now be on the dashboard page, lets open the stats for the game played and
// click stats by match icon
return [4 /*yield*/, (_a.sent()).click()];
case 40:
    // we will now be on the dashboard page, lets open the stats for the game played
    // click stats by match icon
    _a.sent();
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
case 41:
// select first match
return [4 /*yield*/, (_a.sent()).click()];
case 42:
    // select first match
    _a.sent();
    // locating that table object is present
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementsLocated(sel
case 43:
    // locating that table object is present
    _a.sent();
    // giving time for tables to load
    sleep(1);
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementsLocated(sel
case 44:
    tablesWithIds = _a.sent();
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
case 45:
    possessionTimeTable = _a.sent();
    return [4 /*yield*/, driver.wait(selenium_webdriver_1.until.elementLocated(sele
case 46:
    onForGoalTable = _a.sent();
    return [4 /*yield*/, possessionTimeTable.getText()];
case 47:
    possessionTimeTableText = _a.sent();
    // make sure possession time table first one present
    assert.strictEqual(possessionTimeTableText, 'Possession Time', 'Possession time
    return [4 /*yield*/, tablesWithIds[0].getText()];
case 48:
    numberOfTouches = _a.sent();
    return [4 /*yield*/, tablesWithIds[1].getText()];
case 49:
    timeOnField = _a.sent();
    return [4 /*yield*/, tablesWithIds[2].getText()];
```

```
                console.log('plus mins', plusMinus);
                // assert table headers equal what they should
                //number of touches
                assert.strictEqual(numberOfTouches, 'Number of Touches', 'Correct header for Nu
                // time on field
                assert.strictEqual(timeOnField, 'Time on Field', 'Correct header for Time on Fi
                // plus minus
                assert.strictEqual(plusMinus, 'Plus Minus', 'Correct header for Plus Minus Tabl
                assert.strictEqual(onForGoal, 'Lineup for Goals', 'Correct header for Lineup Du
                return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.className("btn
            case 52:
            //log out of app by clicking log out button
            return [4 /*yield*/, (_a.sent()).click()];
            case 53:
                //log out of app by clicking log out button
                _a.sent();
                //should be brought back to main page
                // check if signup element available to ensure page is back to root
                return [4 /*yield*/, driver.findElement(selenium_webdriver_1.By.xpath("//a[@hre
            case 54:
                //should be brought back to main page
                // check if signup element available to ensure page is back to root
                _a.sent();
                return [4 /*yield*/, driver.getCurrentUrl()];
            case 55:
                currentURL4 = _a.sent();
                assert.strictEqual(currentURL4, 'http://2a246.yeg.rac.sh/');
                return [2 /*return*/];
        }
    });
});
```