

Create a Scrolling.js file in the test\specs

```
JS wdio.conf.js ● JS calculatorApp.js JS scrolling.js JS firstTest.js
JS wdio.conf.js > [⌘] config > [⌘] specs
1   exports.config = {
22     // of the config file unless it's absolute.
23     //
24     specs: [
25       // ToDo: define location for spec files here
26       "test/specs/scrolling.js"
27     ],
28     // Patterns to exclude.
```

```
JS wdio.conf.js ● JS calculatorApp.js JS scrolling.js ● JS firstTest.js
JS wdio.conf.js > [⌘] config > [⌘] capabilities > [⌘] "appium:appActivity"
1   exports.config = {
54     capabilities: [{
56       platformName: "Android",
57       "appium:platformVersion": "14",
58       "appium:udid": "emulator-5554",
59       "appium:automationName": "UiAutomator2",
60       "appium:appPackage": "io.appium.android.apis",
61       "appium:appActivity": "io.appium.android.apis.ApiDemos"
62     }],
63   },
```

```
{
  "platformName": "Android",
  "appium:platformVersion": "14",
  "appium:deviceName": "appDevice",
  "appium:automationName": "UiAutomator2",
  "appium:appPackage": "io.appium.android.apis",
  "appium:appActivity": "io.appium.android.apis.ApiDemos"
}
```

describe('Scrolling on app', () => {

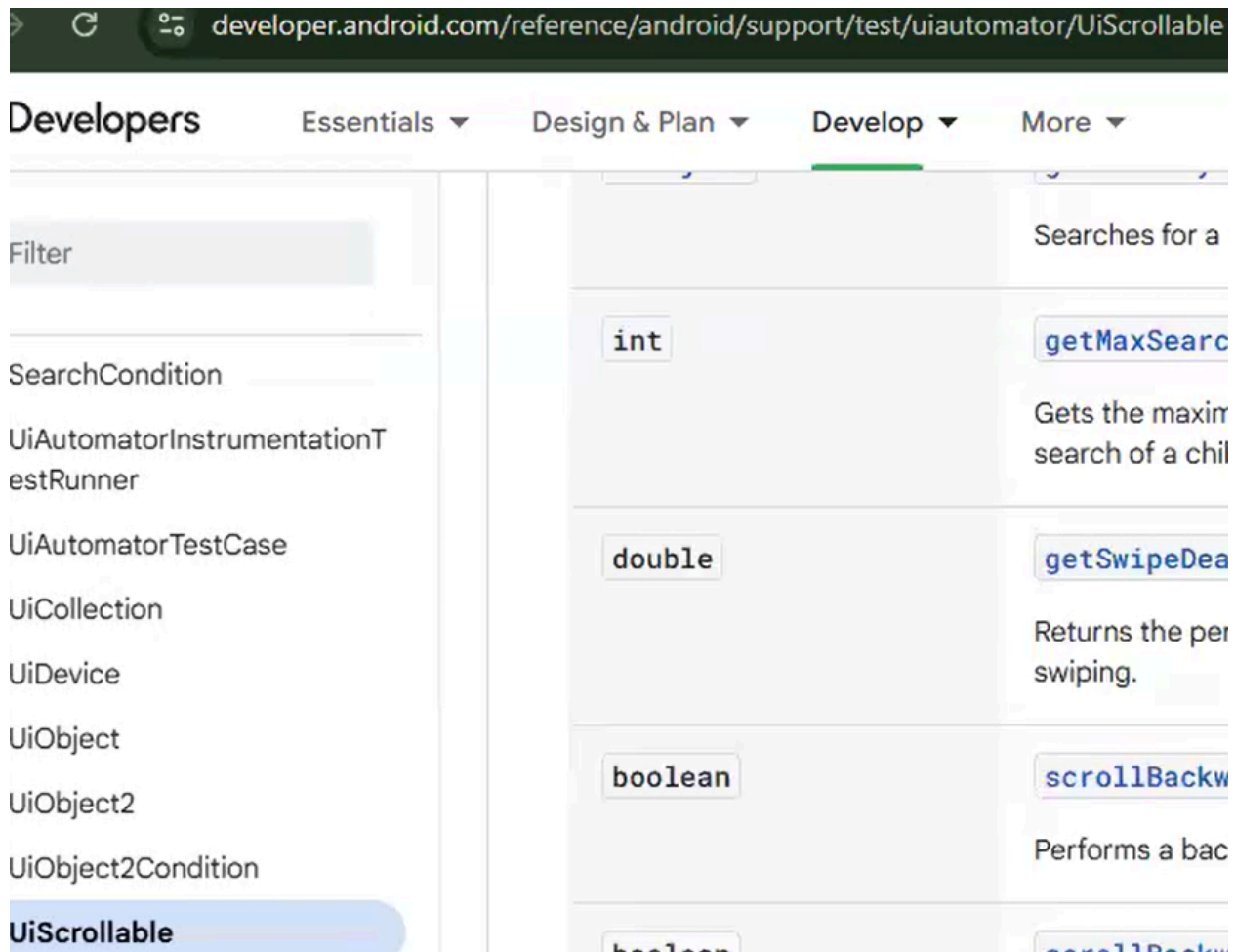
```

    it('Scrolling to end', async () => {
      await browser.pause(2000)
      await $('~Views').click()
      await browser.pause(2000)
    })
  })
})

```

In the command terminal of Visual Studio:

```
> npx wdio
```



```

describe('Scrolling on app', () => {
  it('Scrolling to end', async () => {
    await browser.pause(2000)
    await $('~Views').click()
    await browser.pause(2000)
  })
})

```

```

        await $('android = new UiScrollable(new
UiSelector().scrollable(true)).scrollToEnd(1, 5)')
        await browser.pause(2000)
        const ele = await $('~WebView')
        expect(ele).toExist()
    })
    it('Scrolling to a Text', async() => {
        await browser.pause(2000)
        await $('~Views').click()
        await browser.pause(2000)
        // await $('android = new UiScrollable(new
// UiSelector().scrollable(true)).scrollIntoView(new
//UiSelector.text("Tabs"))')
        await $('android = new UiScrollable(new
UiSelector().scrollable(true)).scrollTextIntoView("Tabs"
)')

        // await $('~Tabs').scrollIntoView()
        await browser.pause(2000)

    })
    it.only('Scrolling to an Element', async() => {
        await browser.pause(2000)
        await $('~Graphics').click()
        await browser.pause(2000)
        // await $('android = new UiScrollable(new
UiSelector().scrollable(true)).scrollIntoView(new
UiSelector.text("Touch Paint"))')
        await browser.pause(2000)

    })
})

```

In the gitbash terminal type in VS code:

```
> npx wdio
```

<https://developer.android.com/reference/androidx/test/ui/automator/UiSelector#summary>

Create a webApp.js in a test\specs:

```
describe('Automating a Web App', () => {
  it('Search on Google', async() => {
    await browser.pause(2000)
    await
$( '//android.widget.TextView[@content-desc="Chrome"]' ).click()
    await browser.pause(2000)
    const logo = await
$( '//android.widget.FrameLayout[@resource-id="com.android.chrome:id/search_provider_logo"]' )
    expect(logo).toExist()
    // Adding a element in search box of google app
    await
$( 'android.widget.EditText' ).addValue("selenium")
    await browser.keys('Enter')
    //await $( 'android.widget.CheckBox' ).click()
    await browser.pause(3000)
    const result = await $( "//div[@class='PZPZlf
']")
    expect(result).toExist()
  })
})
```

```
JS wdio.conf.js • JS calculatorApp.js JS scrolling.js
JS wdio.conf.js > [🔗] config > 🔑 specs
1 exports.config = {
22 // of the config file unless it's
23 //
24 specs: [
25 // ToDo: define location for
26 "test/specs/webApp.js"
27 ],
28 // Patterns to exclude.
29 exclude: [
30 // 'path/to/excluded/files'
```

```
JS wdio.conf.js • JS calculatorApp.js JS scrolling.js JS webApp.js • JS firstTest.js
JS wdio.conf.js > [🔗] config > 🔑 capabilities
1 exports.config = {
54 capabilities: [{
55 // capabilities for local Appium web tests on an Android Emu
56 platformName: "Android",
57 "appium:platformVersion": "14",
58 "appium:udid": "emulator-5554",
59 "appium:automationName": "UiAutomator2",
60 // "appium:appPackage": "io.appium.android.apis",
61 // "appium:appActivity": "io.appium.android.apis.ApiDemos"
62
63 },
```

JS wdio.conf.js • JS calculatorApp.js JS scrolling.js JS wd

JS wdio.conf.js > [🔍] config > 🔑 capabilities > 🔑 "appium:clearSystemFiles"

```
1  exports.config = {
54    capabilities: [{
56      platformName: "Android",
57      "appium:platformVersion": "14",
58      "appium:udid": "emulator-5554",
59      "appium:automationName": "UiAutomator2",
60      "appium:noReset": false,
61      "appium:fullContextList": true
62      "appium:clearSystemFiles": true
63
64      // "appium:appPackage": "io.appium.android
65      // "appium:appActivity": "io.appium.andro
66
67    }],
```

JS wdio.conf.js × JS calculatorApp.js JS scrolling.js JS wd

JS wdio.conf.js > [🔍] config > 🔑 capabilities > 🔑 "appium:noReset"

```
1  exports.config = {
54    capabilities: [{
57      "appium:platformVersion": "14",
58      "appium:udid": "emulator-5554",
59      "appium:automationName": "UiAutomator2",
60      "appium:noReset": true
61
62      // "appium:appPackage": "io.appium.android
63      // "appium:appActivity": "io.appium.andro
64
65    }],
```


JSON Representation

```
{  
  "platformName": "Android",  
  "appium:platformVersion": "14",  
  "appium:deviceName": "appDevice",  
  "appium:automationName": "UiAutomator2"  
}
```

Run the following Command in terminal:

> npx wdio