

Date Picker Automation Using Protractor

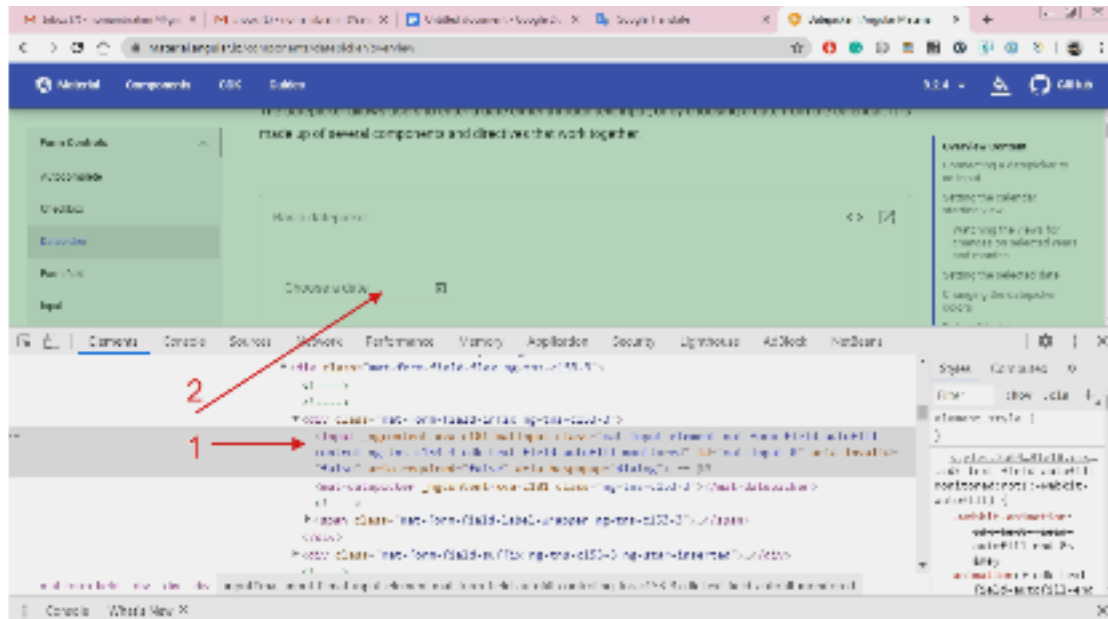
In this document I am going to illustrate various automation techniques to automate the date picker in Angular applications. I will be sharing 5 different ways for 5 different scenarios.

1. Input dates using sendKeys().
2. Input dates for current date.
3. Input dates for current month.
4. Input dates for next/previous month/year.
5. Input dates for any month and year.

Description:

1. Input dates using sendKeys()

As we know protractor gives us a simple way to send dates into an input field using sendKeys(). Method. For that we just need to grab the input element and then using sendKeys() method will do the work. For example, we need to send just a date to a date picker. The test scenario is like we need to save a record with a random date. The date could be any date. In this case using sendKeys() method is very convenient.



For this we just need to grab the input element by id, which is “mat-input-0”. On this element we can call the send keys method. We need to keep in mind sometimes the id can be dynamic and change every time the page gets loaded. In that case we need to apply a different approach to grab the input element.

```

it('It should send date using sendKeys Method', () => {
    let dateInput = element(by.id('mat-input-0'));
    dateInput.sendKeys('7/6/2022');
    browser.sleep(5000);
});

```

2. Input dates for current date.

Let's consider a scenario where we need to generate a report. And the report only generates according to the current date. Every time we want to generate the report we need to provide a current date. For that scenario simple sendKeys() with a manual static date won't be any good. So we need a Javascript function which helps us to generate a current date from the system whenever we call that function.

```

const getCurrentDate = require('./util')

describe('Testing Demo Date picker From Angular material Site', function(){
    it('Open the link', () => {
        browser.get('https://material.angular.io/components/datepicker/overview',5000);
        browser.sleep(5000);
    });
    it('It should select a date', () => {
        //Current day in the current month
        element(by.css('.cdk-overlay-container mat-month-view [aria-label="'+getCurrentDate+'"]')).click();
        browser.sleep(5000);
    });
});

```

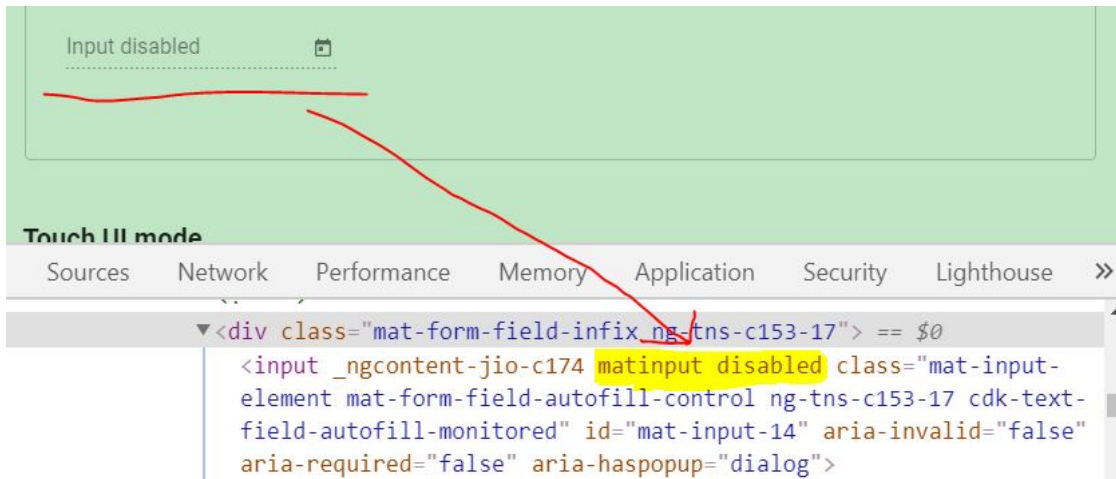
```

1  getCurrentDate = () =>{
2      const myDate = new Date().toLocaleDateString('en-UK',{year:'numeric',month:'long',day:'numeric'});
3      console.log('The date is: ',myDate);
4      return myDate;
5  }
6  module.exports = getCurrentDate();

```

3. Input dates for current month.

Sometimes we can be encountered with a scenario where input element for date is disable like the following example.



In that case we can not perform `sendKeys()` method on this input field. So we need to walk through with a new approach. In that case we need to find the date picker element icon in the DOM, click on that. Then a calendar will open. We can just then provide the date by following.

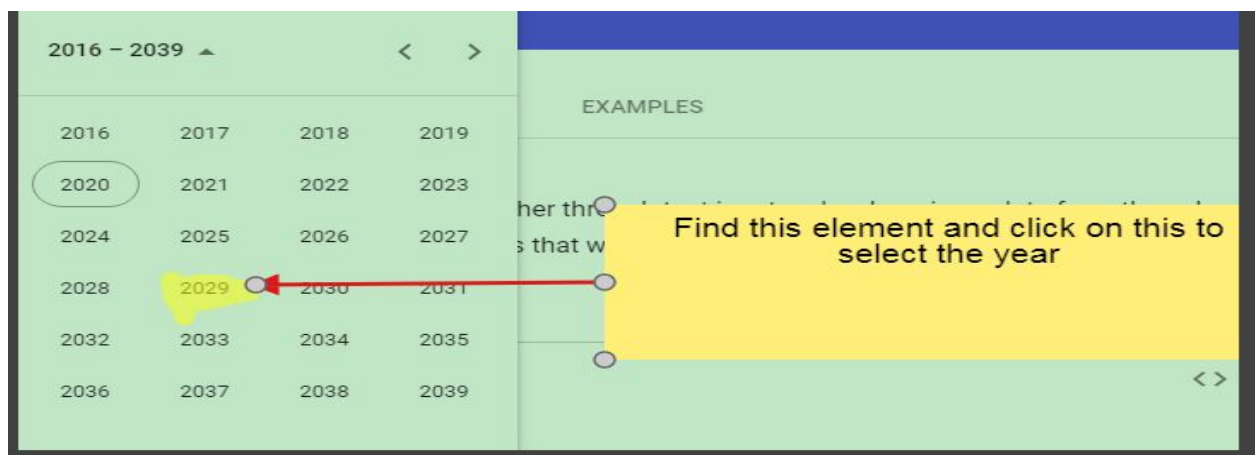
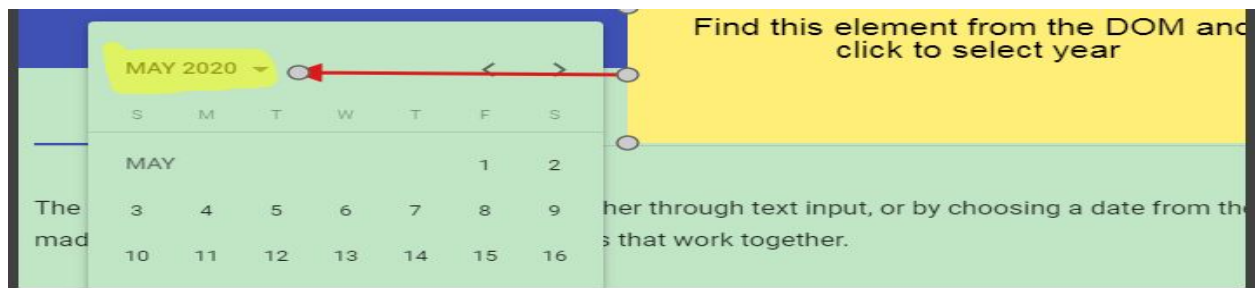
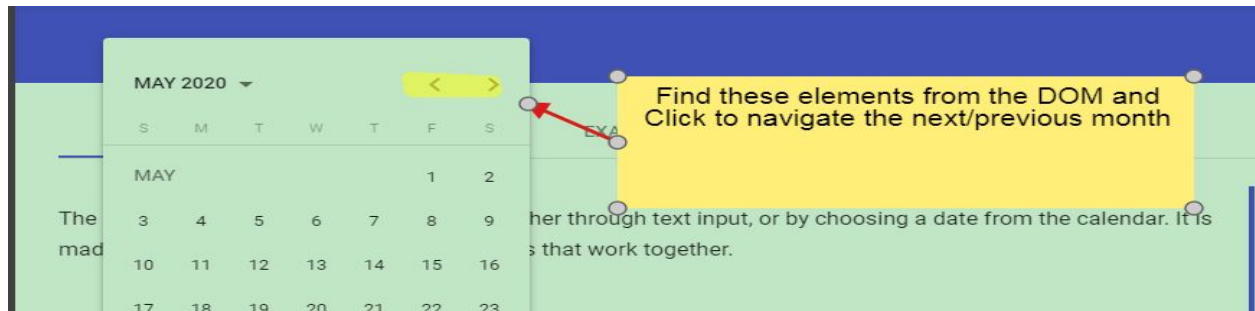
```
describe('Testing Demo Date picker From Angular material Site', function() {
  it('Open the link', () => {
    browser.get('https://material.angular.io/components/datepicker/overview', 5000);
    browser.sleep(5000);
  });
  it('It should click on datepicker', () => {
    let icons = element.all(by.className('mat-datepicker-toggle-default-icon ng-star-inserted'));
    console.log(getCurrentDate());
    icons.get(0).click();
    browser.sleep(5000);
  });
  it('It should select a date', () => {
    // Any day in the current month
    element(by.css('.cdk-overlay-container mat-month-view [aria-label="May 6, 2020"]')).click();
    browser.sleep(5000);
  });
});
```

Grab the `aria-label` attribute.

```
<td role="gridcell" class="mat-calendar-body-cell mat-focus-indicator ng-star-inserted"
tabindex="-1" aria-label="May 6, 2020" aria-selected="false" style="width: 14.2857%;
padding-top: 7.14286%; padding-bottom: 7.14286%;">
```

4. Input dates for next or previous month.

What if we want a date from next month/previous month/next year/previous year. Look at the following image.



The code for this:

```

describe('Testing Demo Date picker From Angular material Site', function(){
  it('Open the link', () => {
    browser.get('https://material.angular.io/components/datepicker/overview',5000);
    browser.sleep(5000);
  });

  it('It should click on datepicker', () => {
    let icons = element.all(by.className('mat-datepicker-toggle-default-icon ng-star-inserted'))
    icons.get(0).click();
    browser.sleep(5000);
  });

  it('should Click on the year header', () => {
    element(by.css('.cdk-overlay-container .mat-datepicker-content .mat-calendar .mat-calendar-content .mat-calendar-table td[aria-label="Choose month and year"]')).click();
    browser.sleep(5000);
  });

  it('It should select a Year', () => {
    let year = 2022;
    element(by.css('.cdk-overlay-container .mat-datepicker-content .mat-calendar .mat-calendar-content .mat-calendar-table td[aria-label="'+year+"]')).click();
    browser.sleep(5000);
  });

  it('it should select a month', () => {
    element(by.xpath('//div[contains(text(),"MAR")]')).click();
    browser.sleep(500);
  });

  it('It should select a day', () => {
    element(by.xpath('//div[contains(text(),"17")]')).click();
    browser.sleep(5000);
  });
});

```

5. Input dates for any month and year.

I have created a method where we can input the elements and the date we need to select. The method will dynamically find that date for me. This method can be upgraded more for more convenient and easy use. I am working on it.

```

let datePicker = function(){
  this.getDateFromAnyInput = function (dateElement, dateValue, yearButton) {
    var splitDate = dateValue.split("/");
    var year = splitDate[2];
    var month = splitDate[1];
    var day = splitDate[0];
    //var monthYear =
    const monthNames = ["January", "February", "March", "April", "May", "June",
      "July", "August", "September", "October", "November",
      "December"];
    var monthName = monthNames[month-1];

    dateElement.click()
    .then(() => yearButton.click())
    .then(() =>
    element(by.css('.cdk-overlay-container .mat-datepicker-content .mat-calendar .mat-calendar-content .mat-calendar-table td[aria-label="'+year+"]')).click()
    .then(() =>
    element(by.css('.cdk-overlay-container .mat-datepicker-content .mat-calendar .mat-calendar-content .mat-calendar-table td[aria-label="'+monthName+" "+year+"]')).click()
    .then(() =>
    element(by.css('.cdk-overlay-container .mat-datepicker-content .mat-calendar .mat-calendar-content .mat-calendar-table td[aria-label="'+monthName+" "+day+" "+year+"]')).click()
    )
    )
  }
}
module.exports = new datePicker();

```

There is another way to put the date. That is to execute JavaScript. Like the following.

```
browser.executeScript("document.getElementById('datepicker').value='02/11/2019'");
```

But I found it not stable and convenient to use.

I have tested my code on the following website.

[Material.angular.io](https://material.angular.io)

You can find the code in my github repo. The link is:

https://github.com/Noman46/Protractor_Practice/tree/master/DatePicker