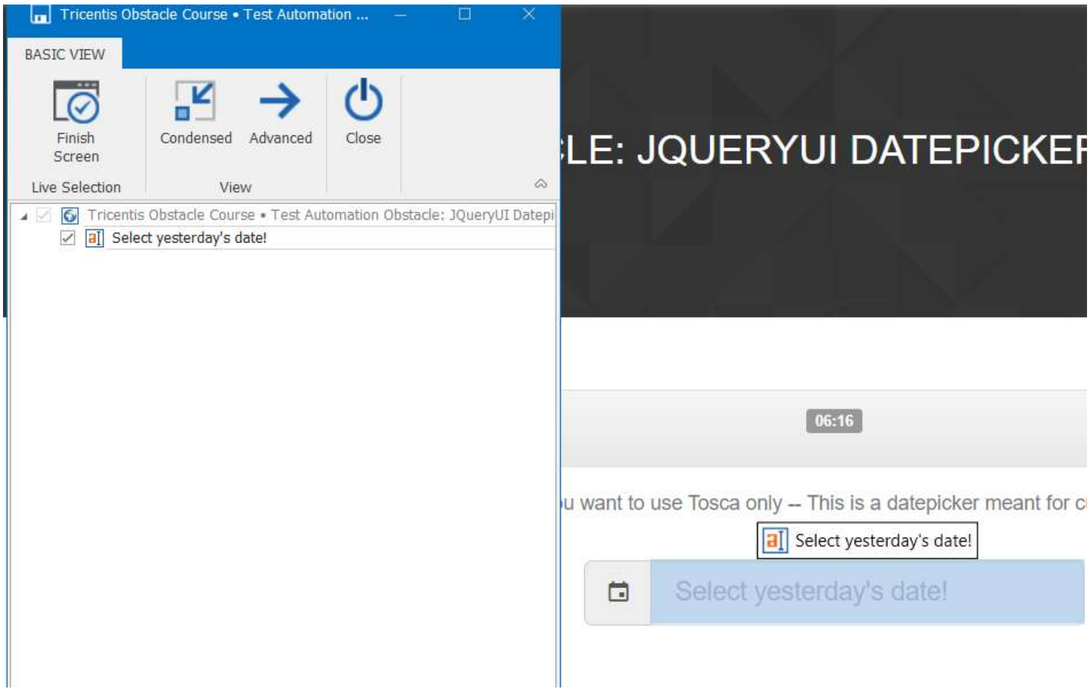
Link to obstacle

https://obstaclecourse.tricentis.com/Obstacles/Next?oid=54345

Initial Steering



Tricentis Obstacle Course • Test Automation Obstacle: JQueryUI Datepi

☑ Select yesterday's date!

LE: JQUERYUI DATEPICKEF

06:16

u want to use Tosca only -- This is a datepicker meant for c

Select yesterday's date!

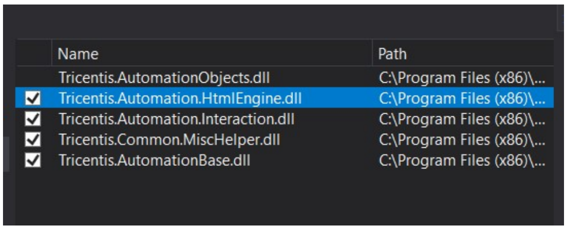
Select yesterday's date!

Hints

TBox API example:

https://documentation.tricentis.com/devcorner/1200/tboxapi/#topic35.html

References



	Name	Path
	Tricentis.AutomationObjects.dll	C:\Program Files (x86)\...
☑	Tricentis.Automation.HtmlEngine.dll	C:\Program Files (x86)\...
☑	Tricentis.Automation.Interaction.dll	C:\Program Files (x86)\...
☑	Tricentis.Common.MiscHelper.dll	C:\Program Files (x86)\...
☑	Tricentis.AutomationBase.dll	C:\Program Files (x86)\...

References for Adpater class

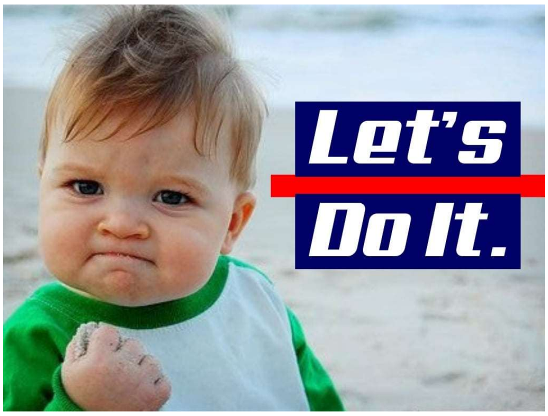
```
using Tricentis.Automation.Creation;
using Tricentis.Automation.Engines.Adapters;
using Tricentis.Automation.Engines.Adapters.Attributes;
using Tricentis.Automation.Engines.Adapters.Html.Generic;
using Tricentis.Automation.Engines.Technicals.Html;
```

What we will need:

- DatePicker adapter (with lots of helper functions)

We will do the implementation like the guided version did: step by step

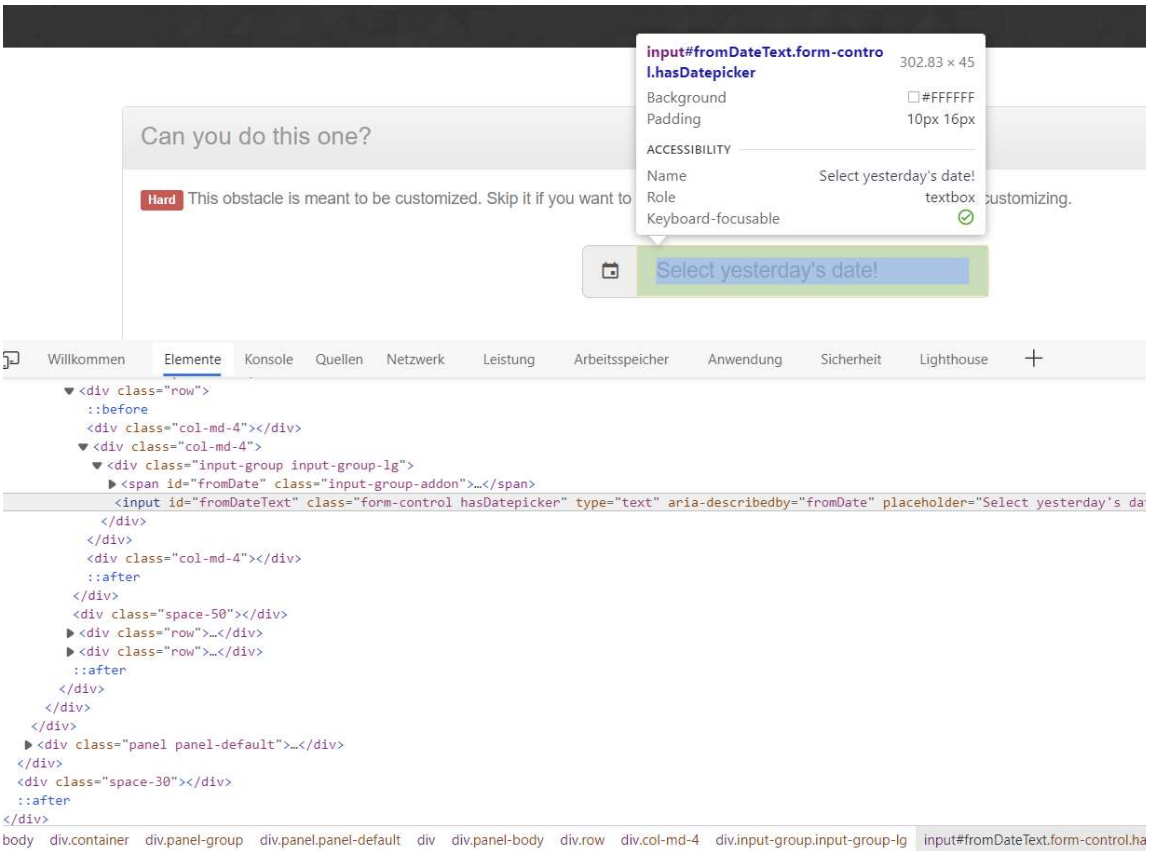
- Creation of the DatePicker Adapter
- Retrieving the date
- Select the date



Creation of the DatePicker Adapter

- Constructor
- Container property
- Open method
- (We will not need a method for closing the date picker, because the container will close after clicking a date)

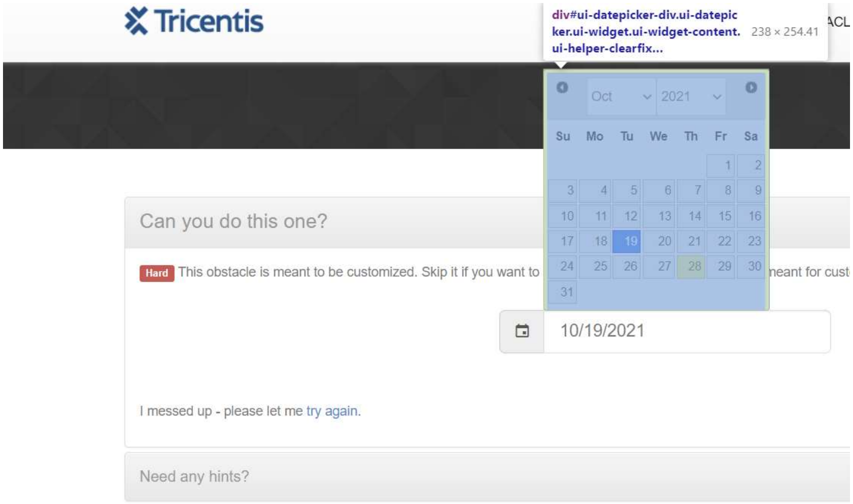
Input field



This obstacle is meant to be customized. Skip it if you want to customizing.

```
    ▼<div class="row">
       ::before
       <div class="col-md-4"></div>
     ▼<div class="col-md-4">
       ▼<div class="input-group input-group-lg">
         ▶<span id="fromDate" class="input-group-addon">…</span>
           <input id="fromDateText" class="form-control hasDatepicker" type="text" aria-describedby="fromDate" placeholder="Select yesterday's da
         </div>
       </div>
       <div class="col-md-4"></div>
       ::after
     </div>
     <div class="space-50"></div>
    ▶<div class="row">…</div>
    ▶<div class="row">…</div>
     ::after
   </div>
  </div>
 </div>
▶<div class="panel panel-default">…</div>
</div>
<div class="space-30"></div>
::after
</div>
```

<div> Container



div#ui-datepicker-div.ui-datepicker.ui-widget.ui-widget-content.ui-helper-clearfix... 238 × 254.41

Oct	2021					
Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

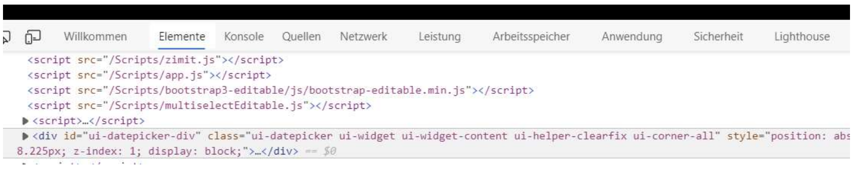
Can you do this one?

Hard This obstacle is meant to be customized. Skip it if you want to … neant for cust

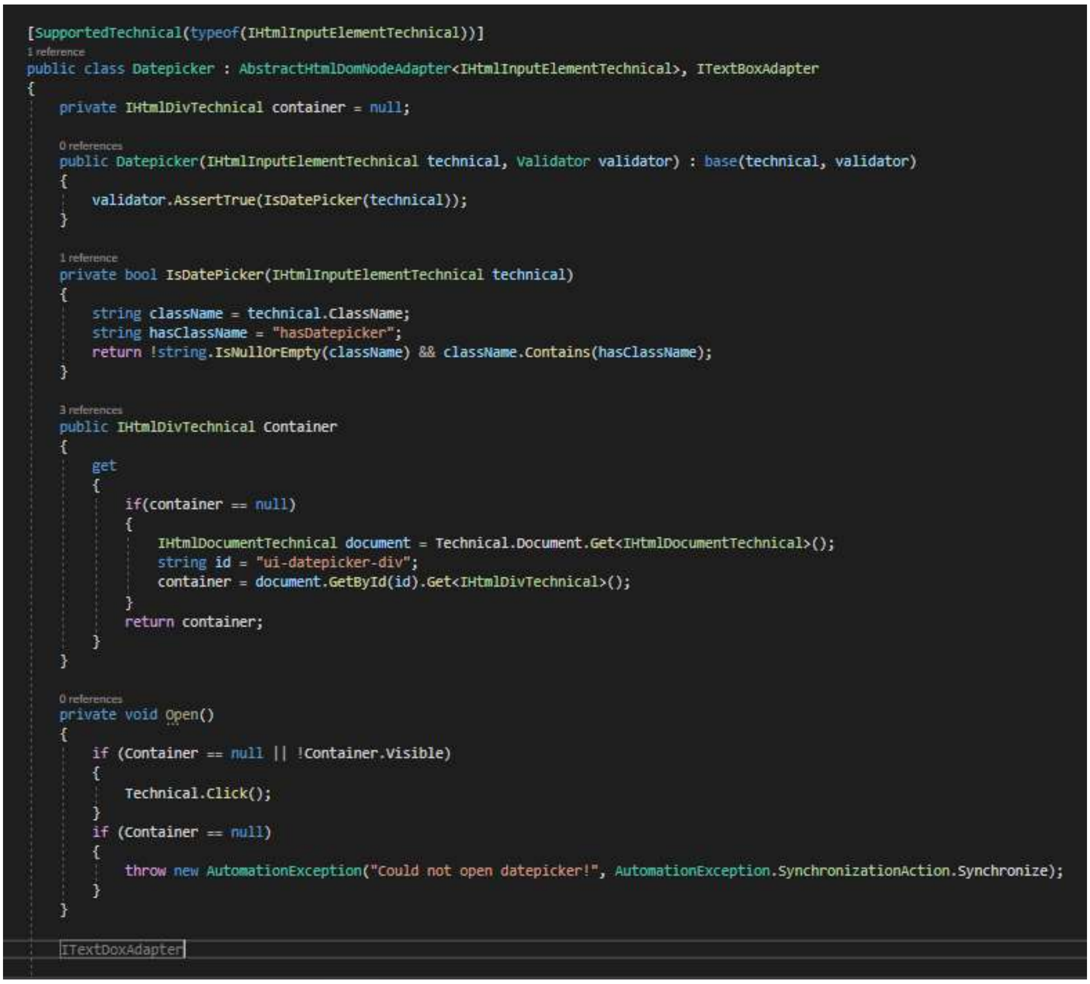
10/19/2021

I messed up - please let me try again.

Need any hints?



```
  <script src="/Scripts/zimit.js"></script>
  <script src="/Scripts/app.js"></script>
  <script src="/Scripts/bootstrap3-editable/js/bootstrap-editable.min.js"></script>
  <script src="/Scripts/multiselectEditable.js"></script>
 ▶<script>…</script>
 ▶<div id="ui-datepicker-div" class="ui-datepicker ui-widget ui-widget-content ui-helper-clearfix ui-corner-all" style="position: abs
8.225px; z-index: 1; display: block;">…</div> == $0
```



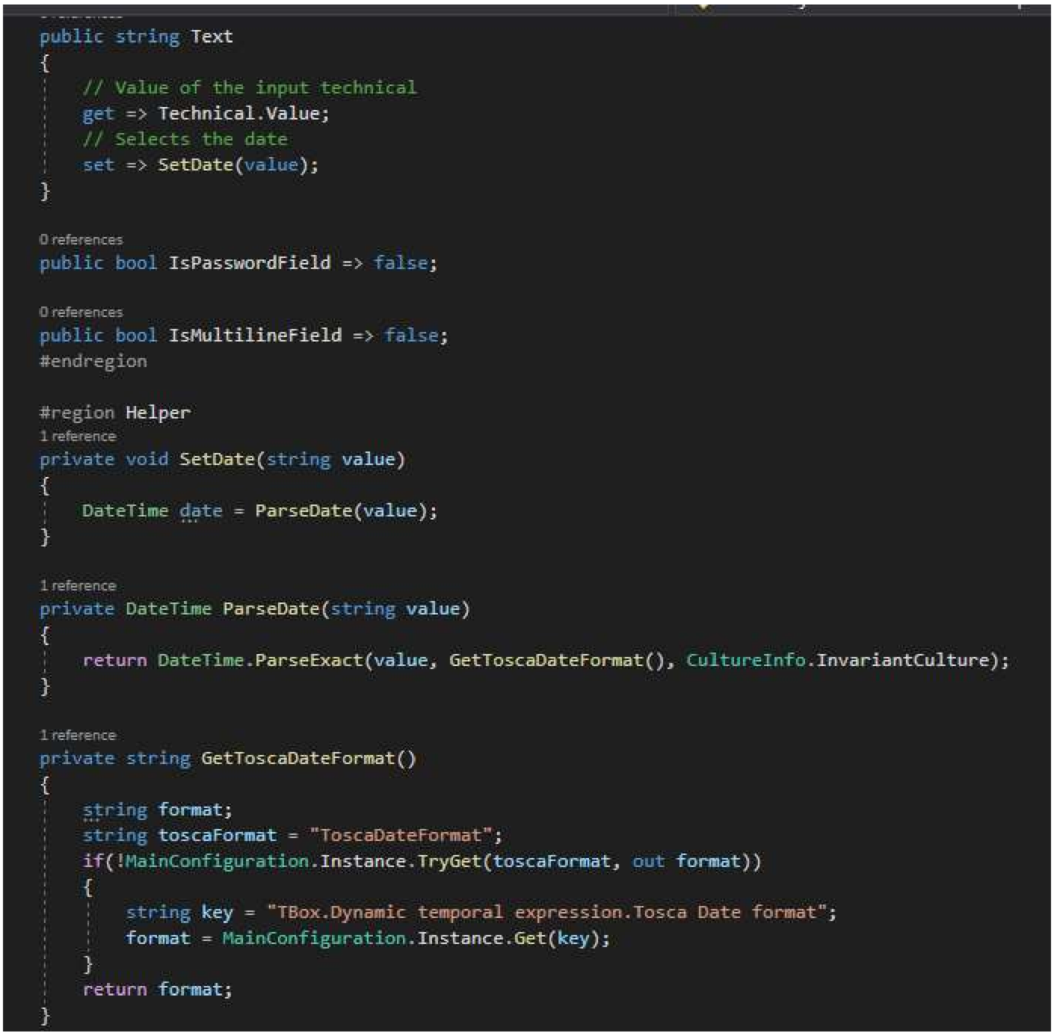
```
[SupportedTechnical(typeof(IHtmlInputElementTechnical))]
1 reference
public class Datepicker : AbstractHtmlDomNodeAdapter<IHtmlInputElementTechnical>, ITextBoxAdapter
{
    private IHtmlDivTechnical container = null;

    0 references
    public Datepicker(IHtmlInputElementTechnical technical, Validator validator) : base(technical, validator)
    {
        validator.AssertTrue(IsDatePicker(technical));
    }

    1 reference
    private bool IsDatePicker(IHtmlInputElementTechnical technical)
    {
        string className = technical.ClassName;
        string hasClassName = "hasDatepicker";
        return !string.IsNullOrEmpty(className) && className.Contains(hasClassName);
    }

    3 references
    public IHtmlDivTechnical Container
    {
        get
        {
            if(container == null)
            {
                IHtmlDocumentTechnical document = Technical.Document.Get<IHtmlDocumentTechnical>();
                string id = "ui-datepicker-div";
                container = document.GetById(id).Get<IHtmlDivTechnical>();
            }
            return container;
        }
    }

    0 references
    private void Open()
    {
        if (Container == null || !Container.Visible)
        {
            Technical.Click();
        }
        if (Container == null)
        {
            throw new AutomationException("Could not open datepicker!", AutomationException.SynchronizationAction.Synchronize);
        }
    }

    ITextDoxAdapter
```

Retrieve the date

Interface ITextBoxAdapter wants us to implement:

- Text property
- IsPasswordField property
- IsMultilineField property

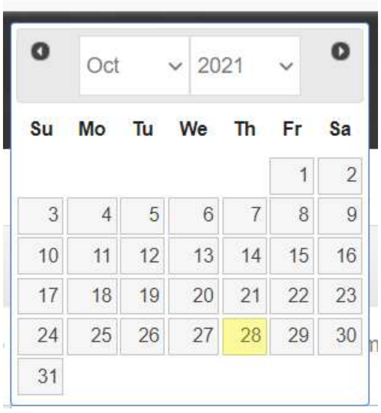
The code for SetDate() was taken from the guided exercise 😊



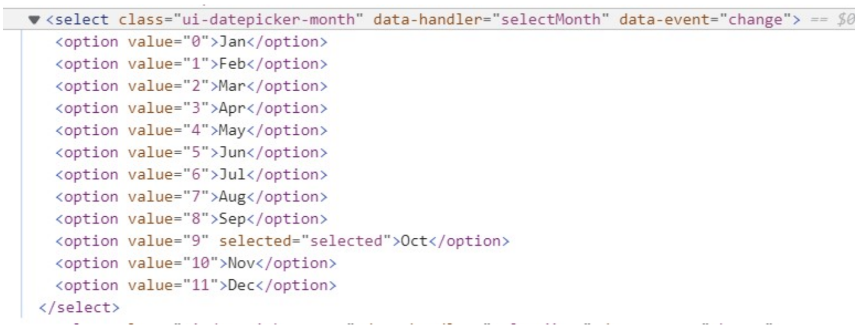
```
public string Text
{
    // Value of the input technical
    get => Technical.Value;
    // Selects the date
    set => SetDate(value);
}

0 references
public bool IsPasswordField => false;

0 references
public bool IsMultilineField => false;
#endregion

#region Helper
1 reference
private void SetDate(string value)
{
    DateTime date = ParseDate(value);
}

1 reference
private DateTime ParseDate(string value)
{
    return DateTime.ParseExact(value, GetToscaDateFormat(), CultureInfo.InvariantCulture);
}

1 reference
private string GetToscaDateFormat()
{
    string format;
    string toscaFormat = "ToscaDateFormat";
    if(!MainConfiguration.Instance.TryGet(toscaFormat, out format))
    {
        string key = "TBox.Dynamic temporal expression.Tosca Date format";
        format = MainConfiguration.Instance.Get(key);
    }
    return format;
}
```

Select the date

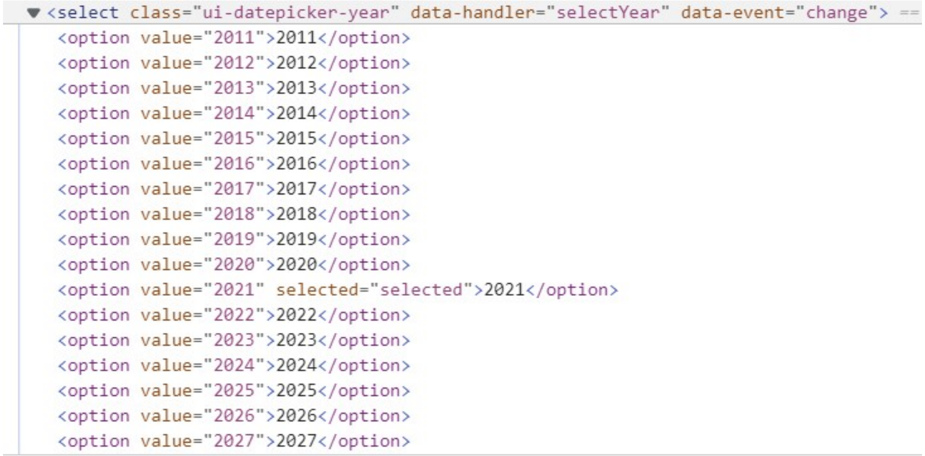
- Month and year are dropdown menus
- Month and year are options in a <select> element



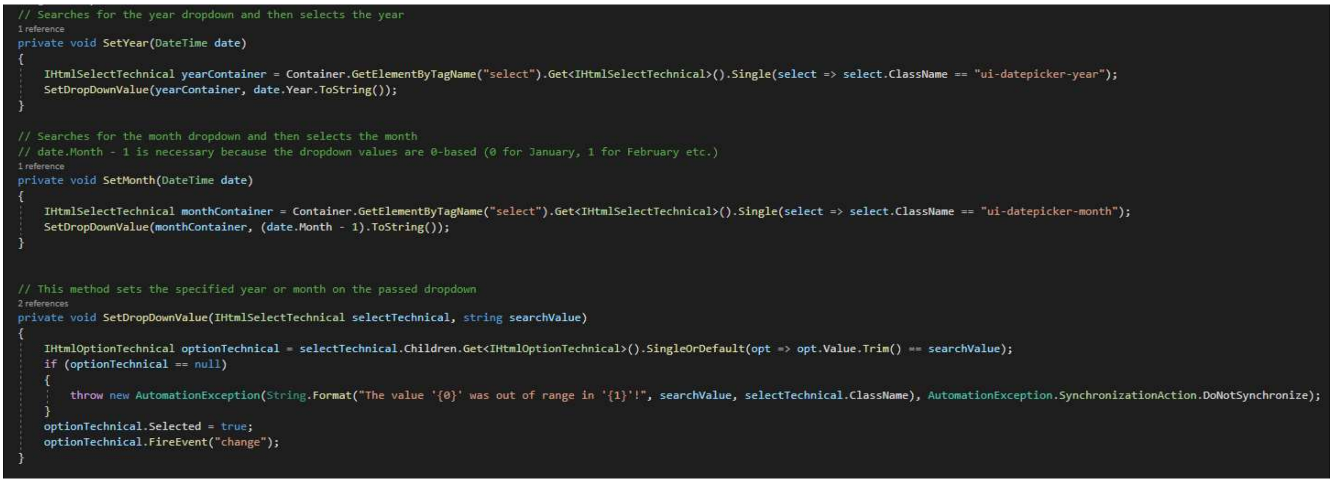
Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						



```
▼<select class="ui-datepicker-month" data-handler="selectMonth" data-event="change"> == $0
    <option value="0">Jan</option>
    <option value="1">Feb</option>
    <option value="2">Mar</option>
    <option value="3">Apr</option>
    <option value="4">May</option>
    <option value="5">Jun</option>
    <option value="6">Jul</option>
    <option value="7">Aug</option>
    <option value="8">Sep</option>
    <option value="9" selected="selected">Oct</option>
    <option value="10">Nov</option>
    <option value="11">Dec</option>
  </select>
```



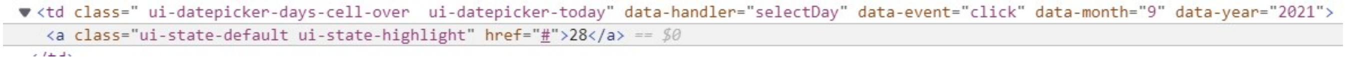
```
▼<select class="ui-datepicker-year" data-handler="selectYear" data-event="change"> ==
    <option value="2011">2011</option>
    <option value="2012">2012</option>
    <option value="2013">2013</option>
    <option value="2014">2014</option>
    <option value="2015">2015</option>
    <option value="2016">2016</option>
    <option value="2017">2017</option>
    <option value="2018">2018</option>
    <option value="2019">2019</option>
    <option value="2020">2020</option>
    <option value="2021" selected="selected">2021</option>
    <option value="2022">2022</option>
    <option value="2023">2023</option>
    <option value="2024">2024</option>
    <option value="2025">2025</option>
    <option value="2026">2026</option>
    <option value="2027">2027</option>
```



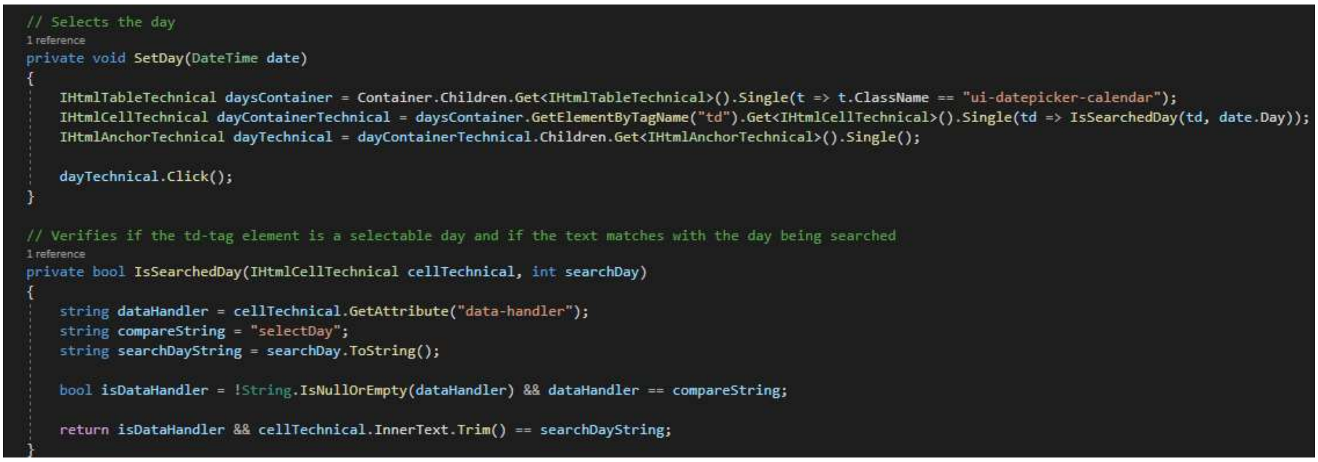
```
// Searches for the year dropdown and then selects the year
1 reference
private void SetYear(DateTime date)
{
    IHtmlSelectTechnical yearContainer = Container.GetElementByTagName("select").Get<IHtmlSelectTechnical>().Single(select => select.ClassName == "ui-datepicker-year");
    SetDropDownValue(yearContainer, date.Year.ToString());
}

// Searches for the month dropdown and then selects the month
// date.Month - 1 is necessary because the dropdown values are 0-based (0 for January, 1 for February etc.)
1 reference
private void SetMonth(DateTime date)
{
    IHtmlSelectTechnical monthContainer = Container.GetElementByTagName("select").Get<IHtmlSelectTechnical>().Single(select => select.ClassName == "ui-datepicker-month");
    SetDropDownValue(monthContainer, (date.Month - 1).ToString());
}


// This method sets the specified year or month on the passed dropdown
2 references
private void SetDropDownValue(IHtmlSelectTechnical selectTechnical, string searchValue)
{
    IHtmlOptionTechnical optionTechnical = selectTechnical.Children.Get<IHtmlOptionTechnical>().SingleOrDefault(opt => opt.Value.Trim() == searchValue);
    if (optionTechnical == null)
    {
        throw new AutomationException(String.Format("The value '{0}' was out of range in '{1}'!", searchValue, selectTechnical.ClassName), AutomationException.SynchronizationAction.DoNotSynchronize);
    }
    optionTechnical.Selected = true;
    optionTechnical.FireEvent("change");
}
```

Day

Example for 2021.10.28

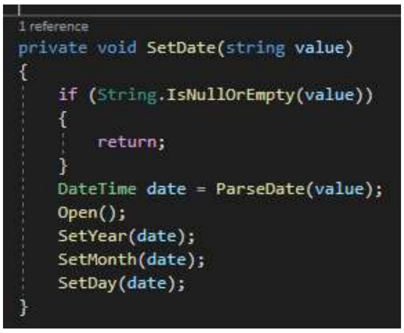


```
▼<td class=" ui-datepicker-days-cell-over  ui-datepicker-today" data-handler="selectDay" data-event="click" data-month="9" data-year="2021">
    <a class="ui-state-default ui-state-highlight" href="#">28</a> == $0
```



```
// Selects the day
1 reference
private void SetDay(DateTime date)
{
    IHtmlTableTechnical daysContainer = Container.Children.Get<IHtmlTableTechnical>().Single(t => t.ClassName == "ui-datepicker-calendar");
    IHtmlCellTechnical dayContainerTechnical = daysContainer.GetElementByTagName("td").Get<IHtmlCellTechnical>().Single(td => IsSearchedDay(td, date.Day));
    IHtmlAnchorTechnical dayTechnical = dayContainerTechnical.Children.Get<IHtmlAnchorTechnical>().Single();

    dayTechnical.Click();
}

// Verifies if the td-tag element is a selectable day and if the text matches with the day being searched
1 reference
private bool IsSearchedDay(IHtmlCellTechnical cellTechnical, int searchDay)
{
    string dataHandler = cellTechnical.GetAttribute("data-handler");
    string compareString = "selectDay";
    string searchDayString = searchDay.ToString();

    bool isDataHandler = !String.IsNullOrEmpty(dataHandler) && dataHandler == compareString;

    return isDataHandler && cellTechnical.InnerText.Trim() == searchDayString;
}
```

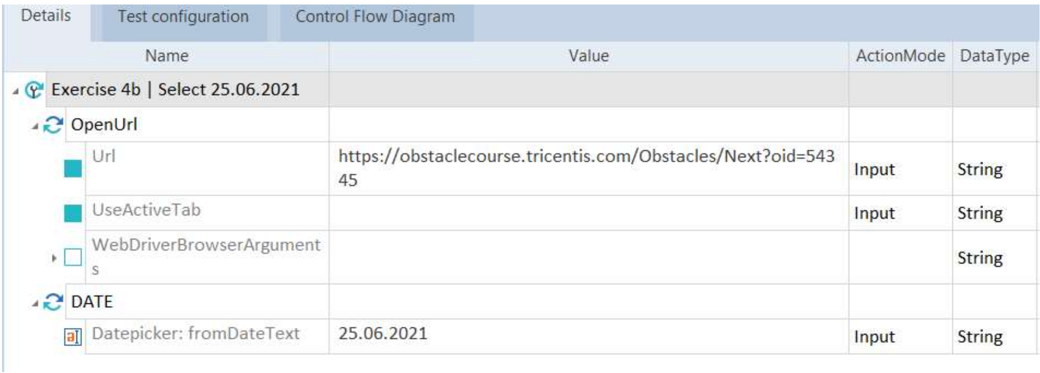
Update code for SetDate()



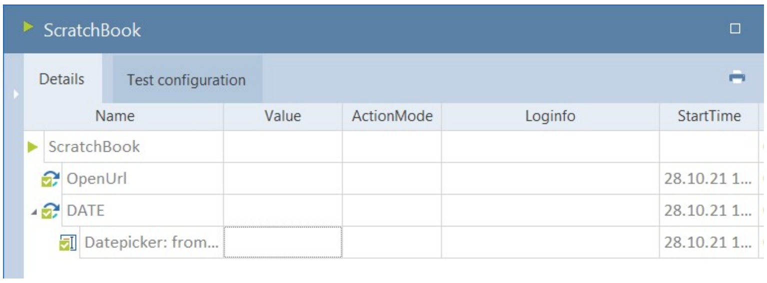
```
1 reference
private void SetDate(string value)
{
    if (String.IsNullOrEmpty(value))
    {
        return;
    }
    DateTime date = ParseDate(value);
    Open();
    SetYear(date);
    SetMonth(date);
    SetDay(date);
}
```

Tosca Commander

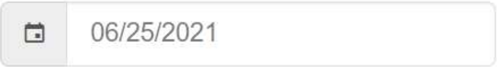
Test Case and run



Name	Value	ActionMode	DataType
Exercise 4b | Select 25.06.2021			
OpenUrl			
Url	https://obstaclecourse.tricentis.com/Obstacles/Next?oid=54345	Input	String
UseActiveTab		Input	String
WebDriverBrowserArguments			String
DATE			
Datepicker: fromDateText	25.06.2021	Input	String



Name	Value	ActionMode	Loginfo	StartTime
ScratchBook				
OpenUrl				28.10.21 1...
DATE				28.10.21 1...
Datepicker: from...				28.10.21 1...



06/25/2021