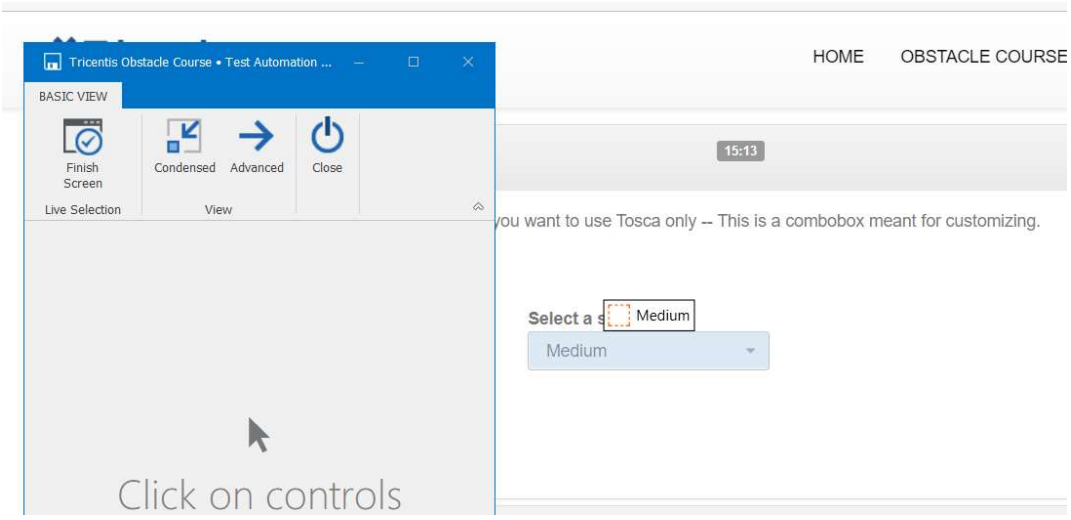
Link to obstacle

https://obstaclecourse.tricentis.com/Obstacles/Next?oid=17771

References for Adapter Classes

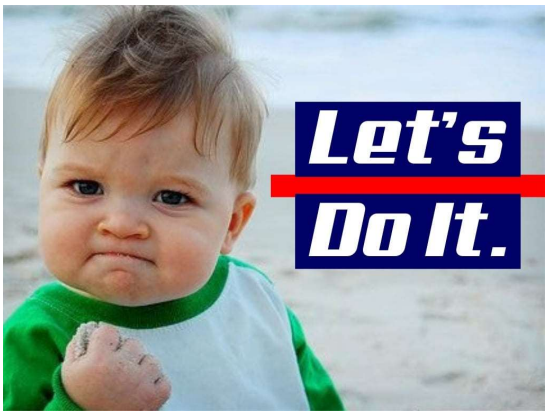
```
using Tricentis.Automation.Creation;
using Tricentis.Automation.Engines.Adapters.Attributes;
using Tricentis.Automation.Engines.Adapters.Html.Generic;
using Tricentis.Automation.Engines.Adapters.Lists;
using Tricentis.Automation.Engines.Technicals.Html;
```

Initial Steering



What we will need:

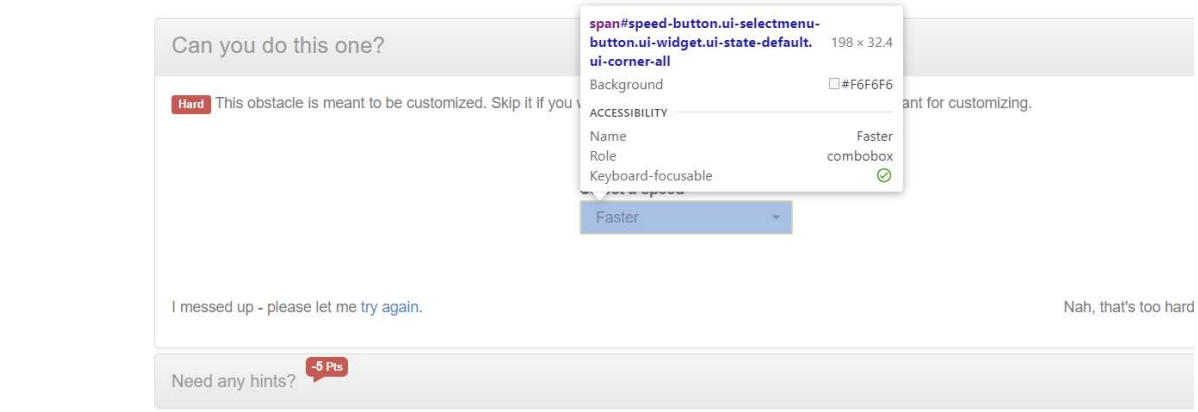
- Combobox Adapter
- Listitem Adapter
- Combox Adapter controller



ComboBoxAdapter

Analyzing the Combo Box

Box itself is a element




```
namespace SpeedCombobox.Html.Adapter
{
    [SupportedTechnical(typeof(IHtmlSpanTechnical))]
    4 references
    public class SpanComboboxAdapter : AbstractHtmlDomNodeAdapter<IHtmlSpanTechnical>, IComboBoxAdapter
    {
        0 references
        public SpanComboboxAdapter(IHtmlSpanTechnical technical, Validator validator) : base(technical, validator)
        {
            validator.AssertTrue(() => technical.ClassName.Contains("ui-selectmenu-button"));
        }

        0 references
        public override string DefaultName => Technical.Id;

        /// <summary>
        /// Open the menu with the dropdown button
        /// </summary>
        1 reference
        public void OpenComboBox()
        {
            string buttonClassName = "ui-icon ui-icon-triangle-1-s";
            IHtmlSpanTechnical boxButton = Technical.Children.Get<IHtmlSpanTechnical>().SingleOrDefault(button => button.ClassName.Equals(buttonClassName));
            if (boxButton != null) boxButton.Click();
        }
    }
}
```

OpenComboBox() will help us later (see Controller section).
Defining it here makes it easy to find the control we want to click in order to open the box.

ListItemAdapter

Analyzing the list items

Items are stored in an ul element / items:

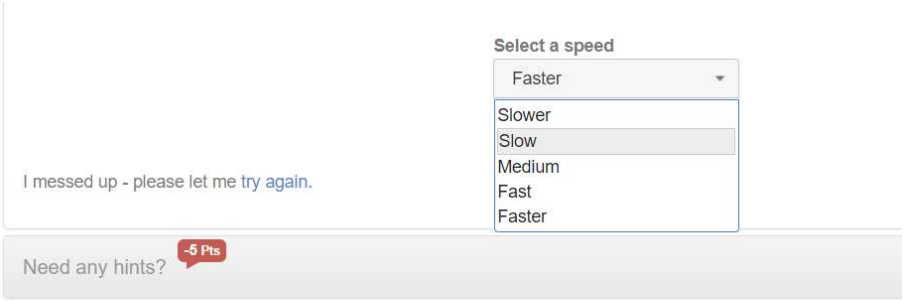
```
▼ <div class="ui-selectmenu-menu ui-front ui-selectmenu-open" style="top: 551px; left: 542px;">
  ▼ <ul aria-hidden="false" aria-labelledby="speed-button" id="speed-menu" class="ui-menu ui-widget ui-widget-content ui-corner-bottom" role="listbox" tabindex=
  "0" aria-activedescendant="ui-id-6" aria-disabled="false" style="width: 198px;">
...   ▶ <li class="ui-menu-item" id="ui-id-2" tabindex="-1" role="option">…</li> == $0
    ▶ <li class="ui-menu-item" id="ui-id-3" tabindex="-1" role="option">…</li>
    ▶ <li class="ui-menu-item" id="ui-id-4" tabindex="-1" role="option">…</li>
    ▶ <li class="ui-menu-item" id="ui-id-5" tabindex="-1" role="option">…</li>
    ▶ <li class="ui-menu-item ui-state-focus" id="ui-id-6" tabindex="-1" role="option">…</li>
  </ul>
</div>
```

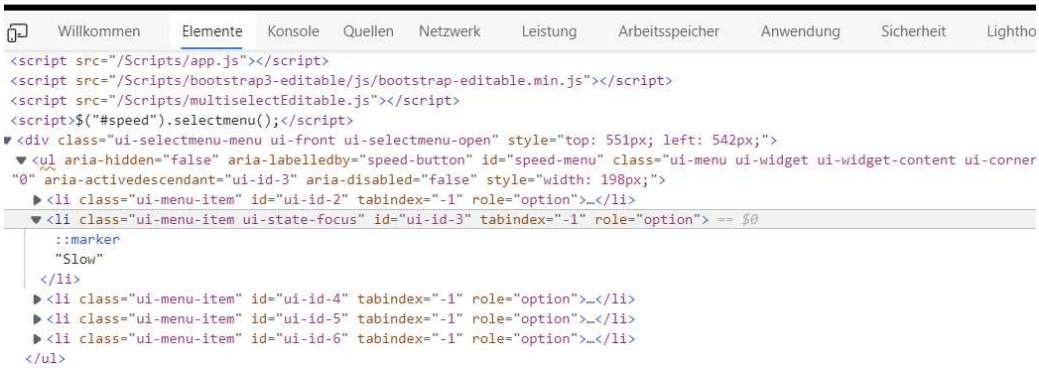
IListAdapter interface wants us to implement:

Text

The InnerText property of the <li> item

Selected





An item is selected, when…

1. you hover with the mouse over it (see the change of the <li> class name to ‚ui-menu item ui-state-focus' when hovering over the item „Slow"?);
2. you click the focused item (mouse click)

```
    0 references
    public bool Selected
    {
        get => Technical.ClassName.Contains("ui-state-focus");
        set
        {
            if (value)
            {
                Technical.FireEvent("mouseover");
                Technical.Click();
            }
        }
    }

    1 reference
    public string Text => Technical.InnerText;
}
```

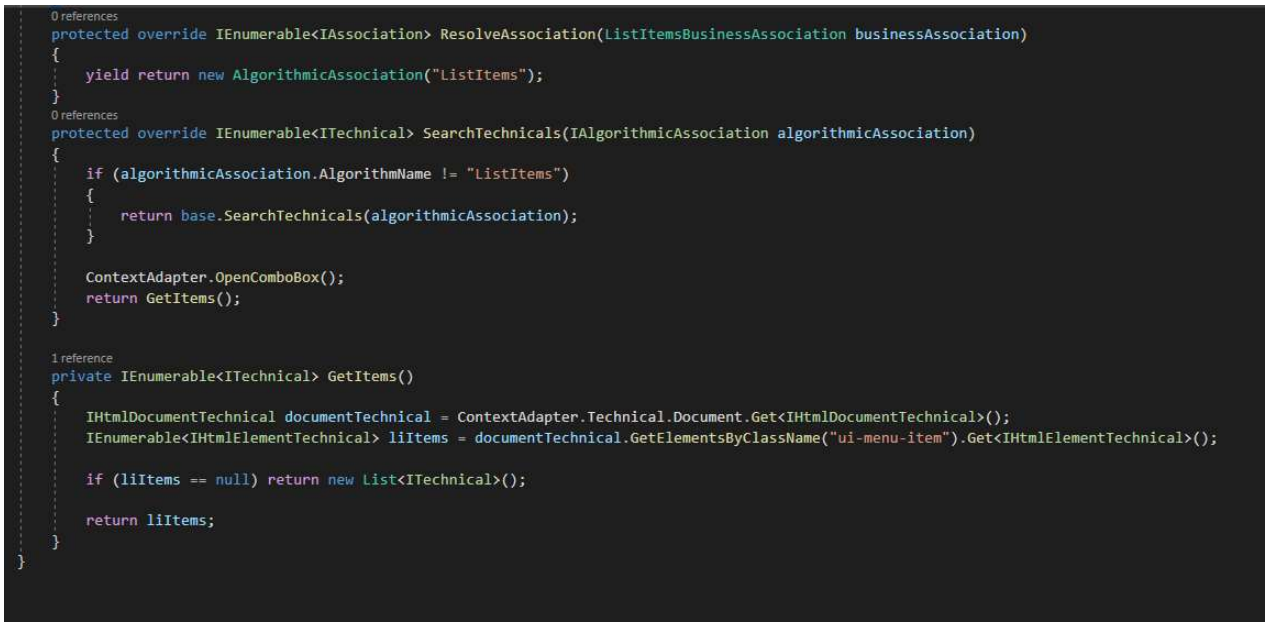
Controller

Finding the items

ListItem is the only interesting part we have to implement, rest is like the guided example here.

We have to open the combobox first in order to collect all <li> items.

ContextAdapter.OpenComboBox() (from the combobox adapter class) will do this for us:



```
    0 references
    protected override IEnumerable<IAssociation> ResolveAssociation(ListItemsBusinessAssociation businessAssociation)
    {
        yield return new AlgorithmicAssociation("ListItems");
    }
    0 references
    protected override IEnumerable<ITechnical> SearchTechnicals(IAlgorithmicAssociation algorithmicAssociation)
    {
        if (algorithmicAssociation.AlgorithmName != "ListItems")
        {
            return base.SearchTechnicals(algorithmicAssociation);
        }

        ContextAdapter.OpenComboBox();
        return GetItems();
    }

    1 reference
    private IEnumerable<ITechnical> GetItems()
    {
        IHtmlDocumentTechnical documentTechnical = ContextAdapter.Technical.Document.Get<IHtmlDocumentTechnical>();
        IEnumerable<IHtmlElementTechnical> liItems = documentTechnical.GetElementsByClassName("ui-menu-item").Get<IHtmlElementTechnical>();

        if (liItems == null) return new List<ITechnical>();

        return liItems;
    }
}
```

Tosca Commander

Module after steering the combobox:

Name	Image	TechnicalID	TypeInfoDescription	ValueRange
Modules				
Standard modules				
Buffer				
SPEED				
speed-button				Slower;Slow;Medium;Fast;Faster

Test Case

Name	Value	ActionMode	DataType	WorkState
Exercise 1b | Select Faster				COMPLETED
OpenUrl				
Url	https://obstaclecourse.tricentis.com/Obstacles/Next?oid=17771	Input	String	
UseActiveTab		Input	String	
WebDriverBrowserArguments			String	
SPEED				
speed-button	Faster	Input	String	

ScratchBook

Details | Test configuration

Name	Value	ActionMode	Loginfo	StartTime	Duration	Detail
ScratchBook					00:02.386	
OpenUrl				25.10.21 1...	00:00.038	
SPEED				25.10.21 1...	00:02.348	
speed-button				25.10.21 1...	00:00.104