

Link to obstacle:

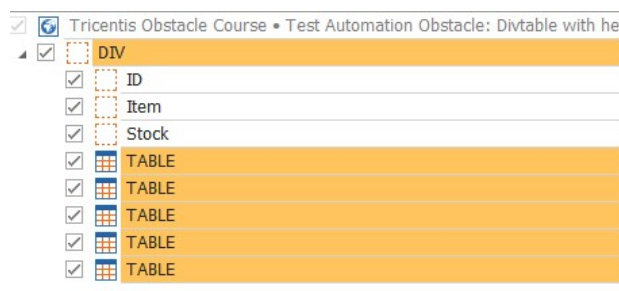
<https://obstaclecourse.tricentis.com/Obstacles/Next?oid=30300>



to be customized. Skip it if you want to use Tosca only – This is a table meant for customizing Tosca

ID	Item	Stock
1099	Apple ?	26
1412	Banana ?	15
2312	Peach ?	40
2567	Watermelon ?	5
3030	Yam ?	3

Initial Steering



Hints

In case of this table, some of the cells are td elements – so they are standard HTML table cells. No additional adapter has to be created for those – the platform handles these already for you. Just make sure that your controllers return them.

References

Name	Path
Tricentis.AutomationObjects.dll	C:\Program Files (x86)\...
<input checked="" type="checkbox"/> Tricentis.Automation.HtmlEngine.dll	C:\Program Files (x86)\...
<input checked="" type="checkbox"/> Tricentis.Automation.Interaction.dll	C:\Program Files (x86)\...
<input checked="" type="checkbox"/> Tricentis.Common.MiscHelper.dll	C:\Program Files (x86)\...
<input checked="" type="checkbox"/> Tricentis.AutomationBase.dll	C:\Program Files (x86)\...

References for Adapter classes

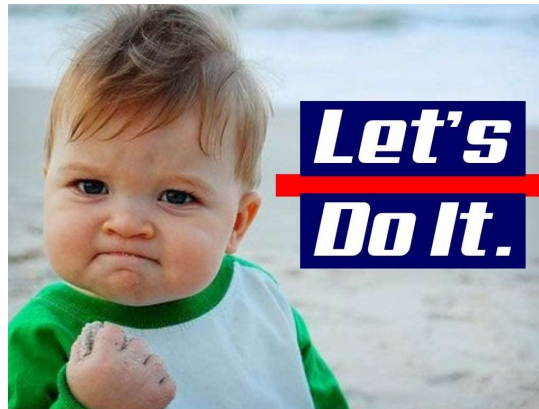
```
using Tricentis.Automation.Creation;  
using Tricentis.Automation.Engines.Adapters;  
using Tricentis.Automation.Engines.Adapters.Attributes;  
using Tricentis.Automation.Engines.Adapters.Html.Generic;  
using Tricentis.Automation.Engines.Technicals.Html;
```

References for Controller Classes

```
using System.Collections.Generic;  
using Tricentis.Automation.AutomationInstructions.TestActions;  
using Tricentis.Automation.AutomationInstructions.TestActions.Associations;  
using Tricentis.Automation.Creation;  
using Tricentis.Automation.Engines.Adapters.Controllers;  
using Tricentis.Automation.Engines.Representations.Attributes;
```

What we will need:

- Table adapter + controller
- Row adapter + controller
- Cell adapter





Adapter classes are the easiest part here, so we start with them

TableAdapter

```
// This class also implements the ITableAdapter, telling the framework that this
// Adapter works as a table. This allows the framework
// to determine that this Adapter works for the according Representation
```

Hard This obstacle is meant to be customized. Skip it if you want to use Tosca only -- This is a table meant for customizing Tosca

div.z-grid.z-content	Item	Stock
1099	Apple ?	26
1412	Banana ?	15
2312	Peach ?	40
2567	Watermelon ?	5
3030	Yam ?	3

I messed up - please let me try again.

Nah, that's too hard. T

```
Willkommen Elemente Konsole Quellen Netzwerk Leistung Arbeitsspeicher Anwendung Sicherheit Lighthouse +
-5 Pts
<div class="col-md-8"> </div>
  <div class="z-grid z-next-gen">...</div>
  <div class="z-grid z-next-gen">
    <div class="z-grid z-content"> == $0
      <table class="z-grid z-style1-row">...</table>
      <table class="z-grid z-style2-row">...</table>
      <table class="z-grid z-style1-row">...</table>
      <table class="z-grid z-style2-row">...</table>
      <table class="z-grid z-style1-row">...</table>
    </div>
  </div>
</div>
<div class="col-md-2"> </div>
::after
```

Do not get fooled by <div>col-md-8 !

```
MLTable FruitHTMLTable.Html.Adapter.DivTableAdapter LoadSt
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;

namespace FruitHTMLTable.Html.Adapter
{
    [SupportedTechnical(typeof(IHtmlDivTechnical))]
    public class DivTableAdapter : AbstractHtmlDomNodeAdapter<IHtmlDivTechnical>, ITableAdapter
    {
        public DivTableAdapter(IHtmlDivTechnical technical, Validator validator) : base(technical, validator)
        {
            validator.AssertTrue(() => !string.IsNullOrEmpty(technical.InnerText) && technical.ClassName.Equals("z-grid z-content"));
        }

        public LoadStrategy LoadStrategy => LoadStrategy.Default;

        public override string DefaultName => "Tutti Frutti Table";
    }
}
```

The table adapter controller will later deal with the question:
How are we getting the rows which are <div> and <tr>?

RowAdapter

```
// This class represents (or interacts with) the row. The class implements the
// ITableRowAdapter, allowing the framework to identify this Adapter as the row.
```

Hard This obstacle is meant to be customized. Skin it if you want to use Tosca only -- This is a table meant for customizing Tosca

div.z-grid.z-header 728.66 × 42

ID	Item	Stock
1099	Apple ?	26
1412	Banana ?	15
2312	Peach ?	40
2567	Watermelon ?	5
3030	Yam ?	3

I messed up - please let me try again. Nah, that's too f

-5 Pts

Willkommen Elemente Konsole Quellen Netzwerk Leistung Arbeitsspeicher Anwendung Sicherheit Lighthouse +

```
</style>
<div class="row"></div>
<div class="row">
  ::before
  <div class="col-md-2"> </div>
  <div class="col-md-8">
    <div class="z-grid z-next-gen">
      <div class="z-grid z-header"></div> == $0
    </div>
    <div class="z-grid z-next-gen"></div>
  </div>
  <div class="col-md-2"> </div>
  ::after
</div>
<div class="space-50"></div>
```

```
3
4
5 namespace FruitHTMLTable.Html.Adapter
6 {
7     /// <summary>
8     /// Header row is a <div></div>, all other rows are table <tr></tr>
9     /// </summary>
10    [SupportedTechnical(typeof(IHtmlDivTechnical))]
11    4 references
12    public class RowAdapter : AbstractHtmlDomNodeAdapter<IHtmlDivTechnical>, ITableRowAdapter
13    {
14        0 references
15        public RowAdapter(IHtmlDivTechnical technical, Validator validator) : base(technical, validator)
16        {
17            validator.AssertTrue(() => technical.ClassName.Contains("z-header"));
18        }
19    }
20 }
```

Clarify the representation: header row

CellAdapter

```
// This class represents (or interacts with) the cell. The class implements the
// ITableCellAdapter, allowing the framework to identify this Adapter as a cell.
```

Hard This obstacle is meant to be customized. Skip it if you want to use Tosca only – This is a table meant for customizing Tosca

div.z-grid.z-column 242.55 × 41

ID	Item	Stock
1099	Apple ?	26
1412	Banana ?	15
2312	Peach ?	40
2567	Watermelon ?	5
3030	Yam ?	3

I messed up - please let me try again. Nah, that's too han

-5 Pts

Willkommen Elemente Konsole Quellen Netzwerk Leistung Arbeitsspeicher Anwendung Sicherheit Lighthouse +

```
<div class="row">...</div>
▼ <div class="row">
  ::before
  <div class="col-md-2"> </div>
  ▼ <div class="col-md-8">
    ▼ <div class="z-grid z-next-gen">
      ▼ <div class="z-grid z-header">
        <div class="z-grid z-column">ID</div> == $0
        <div class="z-grid z-column">Item</div>
        <div class="z-grid z-column">Stock</div>
      </div>
    </div>
  </div>
</div>
```

```
namespace FruitHTMLTable.Html.Adapter
{
    [SupportedTechnical(typeof(HtmlElementTechnical))]
    1 reference
    public class CellAdapter : AbstractHtmlDomNodeAdapter<IHtmlDivTechnical>, ITableCellAdapter
    {
        0 references
        public CellAdapter(IHtmlDivTechnical technical, Validator validator) : base(technical, validator)
        {
            validator.AssertTrue(() => technical.ClassName.Contains("z-column"));
        }

        0 references
        public int ColSpan => 1;

        0 references
        public int RowSpan => 1;

        0 references
        public string Text => Technical.InnerText;
    }
}
```

TableAdapterController

Tell the platform how to move from the table to the rows.

```
0 references
protected override IEnumerable<IAssociation> ResolveAssociation(DescendantsBusinessAssociation businessAssociation)
{
    yield return new TechnicalAssociation("All");
}
0 references
protected override IEnumerable<IAssociation> ResolveAssociation(ColumnsBusinessAssociation businessAssociation)
{
    throw new NotSupportedException();
}
0 references
protected override IEnumerable<IAssociation> ResolveAssociation(RowsBusinessAssociation businessAssociation)
{
    yield return new AlgorithmicAssociation("Rows");
}
0 references
protected override IEnumerable<ITechnical> SearchTechnicals(IAgorithmicAssociation algorithmicAssociation)
{
    return (algorithmicAssociation.AlgorithmName != "Rows") ? base.SearchTechnicals(algorithmicAssociation) : GetRows();
}
1 reference
private IEnumerable<ITechnical> GetRows()
{
    return ContextAdapter.Technical.ParentNode.Get<IHtmlDivTechnical>()
        .ParentNode.Get<IHtmlDivTechnical>()
        .All.Get<ITechnical>();
}
#endregion
```

We make use of the *ResolveAssociation(DescendantBusinessAssociation)* to get *all* the rows of the table:

- Contextadapter is our table
=> z-content node in the html
- Parentnode of Contextadapter
=> z-next-gen node in the html
- Parentnode of z-next-gen
=> col-md-8 node
- All.Get<ITechnical>() grants us access to all elements below the col-md-8 node

```
::before
<div class="col-md-2"> </div>
▼ <div class="col-md-8">
  ▼ <div class="z-grid z-next-gen">
    ▶ <div class="z-grid z-header">...</div>
  </div>
  ▼ <div class="z-grid z-next-gen">
    ▼ <div class="z-grid z-content"> == $0
      ▶ <table class="z-grid z-style1-row">...</table>
      ▶ <table class="z-grid z-style2-row">...</table>
      ▶ <table class="z-grid z-style1-row">...</table>
      ▶ <table class="z-grid z-style2-row">...</table>
      ▶ <table class="z-grid z-style1-row">...</table>
    </div>
  </div>
```


RowAdapterController

Tell the platform how to get from row to cells.

We only have to deal with the header since the table rows already „know“ how to access their cells.

```
/// </summary>
[SupportedAdapter(typeof(RowAdapter))]
1 reference
public class RowAdapterController : TableRowContextAdapterController<RowAdapter>
{
    0 references
    public RowAdapterController(RowAdapter contextAdapter, ISearchQuery query, Validator validator) : base(contextAdapter, query, validator)
    {
    }

    0 references
    protected override IEnumerable<IAssociation> ResolveAssociation(DescendantsBusinessAssociation businessAssociation)
    {
        yield return new TechnicalAssociation("All");
    }

    0 references
    protected override IEnumerable<IAssociation> ResolveAssociation(CellsBusinessAssociation businessAssociation)
    {
        yield return new TechnicalAssociation("Children");
    }

    0 references
    protected override IEnumerable<IAssociation> ResolveAssociation(ChildrenBusinessAssociation businessAssociation)
    {
        yield return new TechnicalAssociation("Children");
    }

    0 references
    protected override IEnumerable<IAssociation> ResolveAssociation(ParentBusinessAssociation businessAssociation)
    {
        yield return new TechnicalAssociation("ParentNode");
    }
}
```

The cells are the direct children of z-header:

```
<div class="row">
  ::before
  <div class="col-md-2"> </div>
  <div class="col-md-8">
    <div class="z-grid z-next-gen">
      <div class="z-grid z-header">
        <div class="z-grid z-column">ID</div> == $0
        <div class="z-grid z-column">Item</div>
        <div class="z-grid z-column">Stock</div>
      </div>
    </div>
    <div class="z-grid z-next-gen">...</div>
  </div>
  <div class="col-md-2"> </div>
  ::after
</div>
```

Tosca Commander

XScan

The XScan interface displays a list of elements on the left, including '04', 'Obstacle #30300', 'Tutti Frutti Table' (selected), 'try again', 'next one', 'Need any hints?', 'www.tricentis.com', 'support.tricentis.com', 'manuals', 'feedback forum', 'https://www.facebook.com/TRICENTIS', and 'https://twitter.com/Tricentis'. The 'Content View' window is open, showing a table with the following data:

ID	Item	Stock
1099	Apple	26
1412	Banana	15
2312	Peach	40
2567	Watermelon	5
3030	Yam	3

Module after steering the table:

The module structure shows 'TuttiFrutti' as the parent module. Under it, 'Tutti Frutti Ta...' is listed, which contains sub-elements '<Row>' and '<Col>'.

Test Case

Exercise 2b Verify Banana ID					PLANNED
OpenUrl					
Url	https://obstaclecourse.tricentis.com/Obstacles/Next?oid=30300	Input	String		
UseActiveTab		Input	String		
WebDriverBrowserArguments			String		
TuttiFrutti					
Tutti Frutti Table					
\$2		Select	String		
ID	1412	Verify	String		
<Cell>			String		
<Row>			String		
<Col>			String		

ScratchBook

Details

Test configuration

	Name	Value	ActionMode	Loginfo	StartTime
▶	ScratchBook				
▶	OpenUrl				25.10.21 1...
▶	TuttiFrutti				25.10.21 1...
▶	Tutti Frutti Table				25.10.21 1...
▶	\$2				25.10.21 1...
▶	ID			Verification was success...	25.10.21 1...
				Verification was successful. Expected value == "1412" Actual value: "1412"	