DotVVM Virtual Meetup

Building Custom Controls

https://gitter.im/riganti/dotvvm-meetups

Who are we?

Standa
@exyi
core framework magic

Tomáš
@tomasherceg
docs, framework

Milan
@Mylan719
VS extension magic

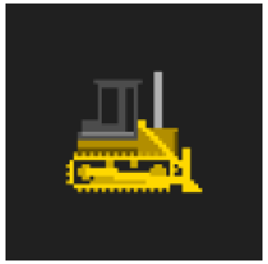
Michal
@MichalTichy
Bootstrap for DotVVM

Ladislav
@quigamdev
framework, extension

Michal
@mrnustik
Business Pack

Martin
@martindybal
Business Pack

Maroš
@marosjanota
Business Pack themes


Adam
@ENgateman
academy, framework

+ many others

We want to talk with you

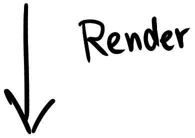
We want to hear your feedback

Questions

- Feel free to unmute yourself and ask us anything
- Gitter room during the meetup
  - https://gitter.im/riganti/dotvvm-meetups
- Reach out to us any time

Building Custom Controls

```
<dot:Repeater WrapperTagName="ul" DataSource={value: Routes }>
    <li><a href={value: Url}>{{value: RouteName}}</a></li>
</dot:Repeater>
<div>Number of Tests: {{value: Routes.Count}}</div>
```



```
<ul data-bind="foreach: { 'data': Routes }">
    <li><a data-bind="attr: { 'href': Url }"><!-- ko text: RouteName --><!-- /ko --></a></li>
</ul>
<div>Number of Tests: <!-- ko text: (Routes() || {}).length --><!-- /ko --></div>
```

```
<dot:Repeater WrapperTagName="ul" DataSource={value: Routes }>
    <li><a href={value: Url}>{{value: RouteName}}</a></li>
</dot:Repeater>
<div>Number of Tests: {{value: Routes.Count}}</div>
```

(View Compiler & Builder) ↓

Dotvvm Control tree

Render method

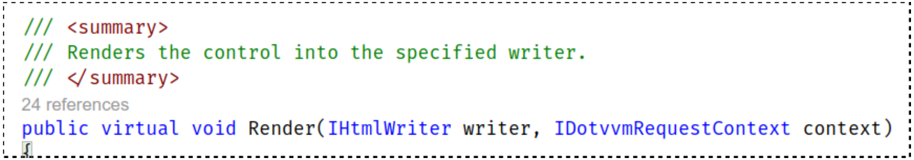
↓

IHtmlWriter instructions

↓

```
<ul data-bind="foreach: { 'data': Routes }">
    <li><a data-bind="attr: { 'href': Url }"><!-- ko text: RouteName --><!-- /ko --></a></li>
</ul>
<div>Number of Tests: <!-- ko text: (Routes() || {}).length --><!-- /ko --></div>
```

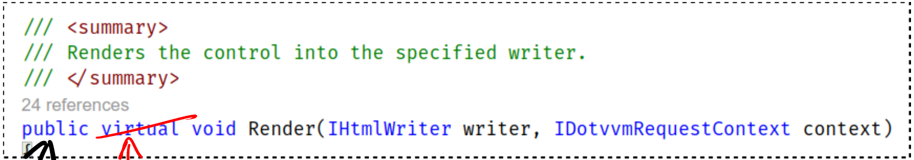
Dotvvm Control



```
/// <summary>
/// Renders the control into the specified writer.
/// </summary>
24 references
public virtual void Render(IHtmlWriter writer, IDotvvmRequestContext context)
{
```

•

Dotvvm Control

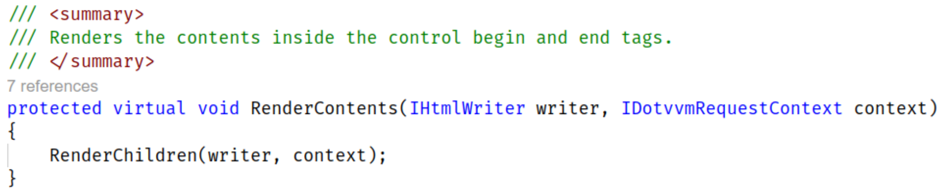


```
/// <summary>
/// Renders the control into the specified writer.
/// </summary>
24 references
public ~~virtual~~ void Render(IHtmlWriter writer, IDotvvmRequestContext context)
{
```

Please don't

This method should be called

Dotvvm Control

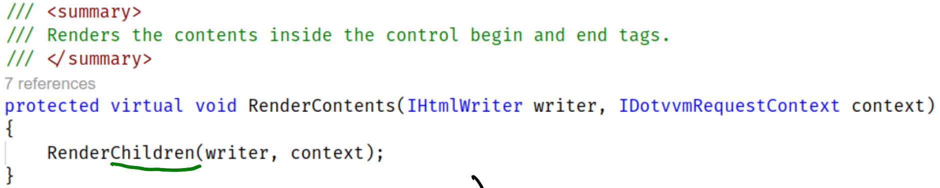


```
/// <summary>
/// Renders the contents inside the control begin and end tags.
/// </summary>
7 references
protected virtual void RenderContents(IHtmlWriter writer, IDotvvmRequestContext context)
{
    RenderChildren(writer, context);
}
```

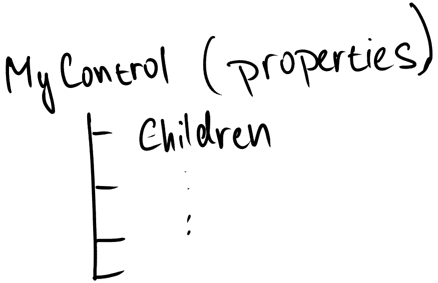
DEMO

- Render
- IHtmlWriter

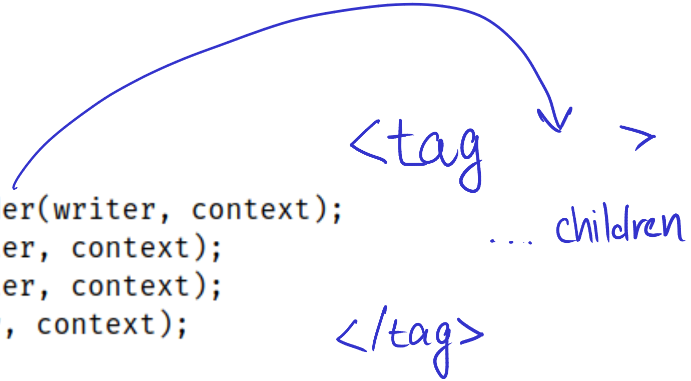
Dotvvm Control



```
/// <summary>
/// Renders the contents inside the control begin and end tags.
/// </summary>
7 references
protected virtual void RenderContents(IHtmlWriter writer, IDotvvmRequestContext context)
{
    RenderChildren(writer, context);
}
```



```
AddAttributesToRender(writer, context);
RenderBeginTag(writer, context);
RenderContents(writer, context);
RenderEndTag(writer, context);
```



set DataContext
if IncludeInPage

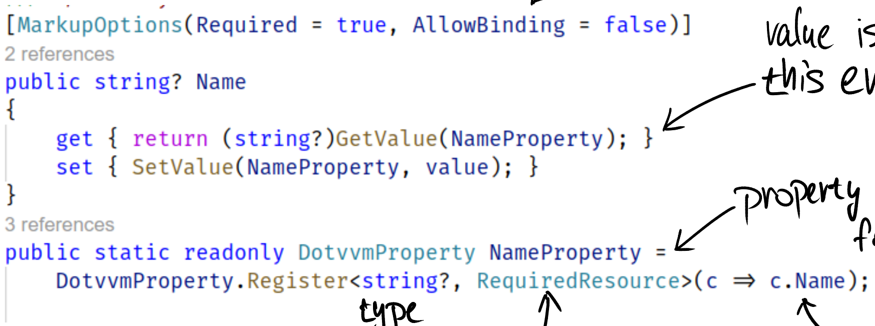
<tag >

```
AddAttributesToRender(writer, context);
RenderBeginTag(writer, context);
RenderContents(writer, context);
RenderEndTag(writer, context);
```

.... children

</tag>

disable value bindings



```
[MarkupOptions(Required = true, AllowBinding = false)]
2 references
public string? Name
{
    get { return (string?)GetValue(NameProperty); }
    set { SetValue(NameProperty, value); }
}
3 references
public static readonly DotvvmProperty NameProperty =
    DotvvmProperty.Register<string?, RequiredResource>(c => c.Name);
```

value is stored in properties

this evaluates bindings

property descriptor for DotVVM

type

the declaring control needed to get full name

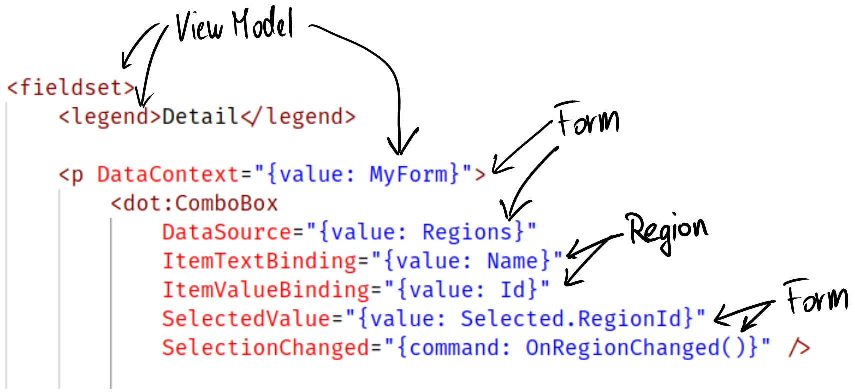
just for property name

might also be "Name"

Value binding

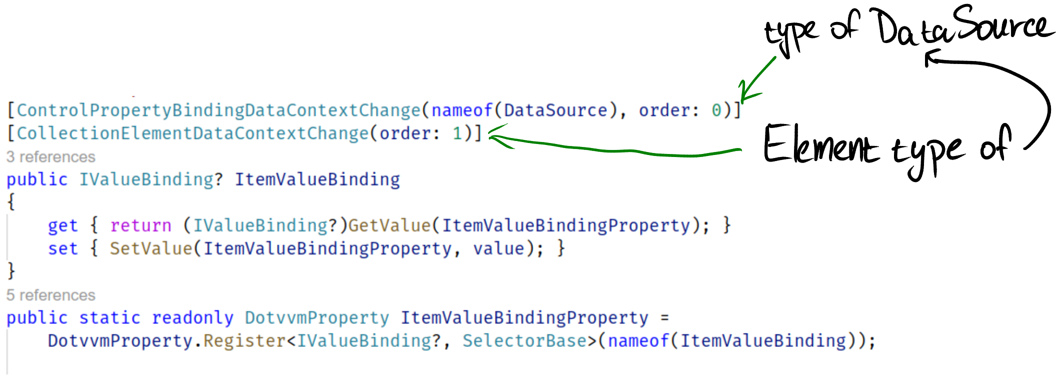
- Evaluate → html
- to JS expression → knockout binding

Data Context Type



```
<fieldset>
    <legend>Detail</legend>

    <p DataContext="{value: MyForm}">
        <dot:ComboBox
            DataSource="{value: Regions}"
            ItemTextBinding="{value: Name}"
            ItemValueBinding="{value: Id}"
            SelectedValue="{value: Selected.RegionId}"
            SelectionChanged="{command: OnRegionChanged()}" />
```

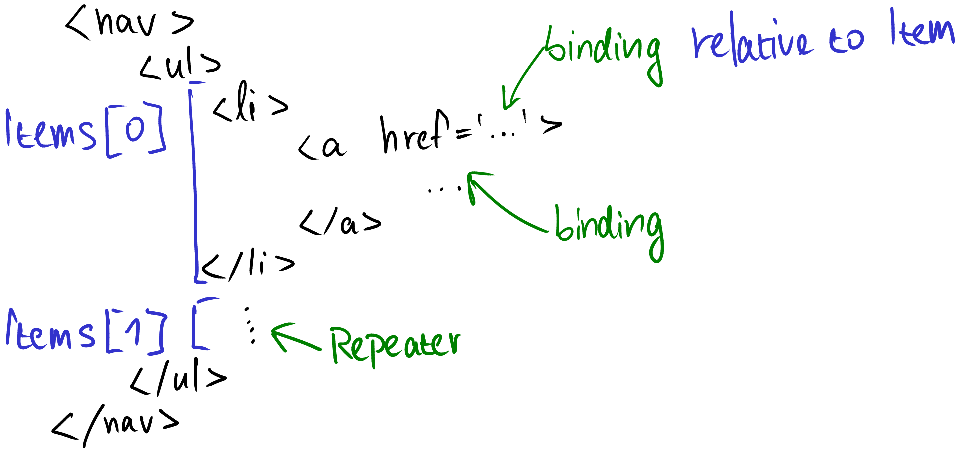


```
[ControlPropertyBindingDataContextChange(nameof(DataSource), order: 0)]
[CollectionElementDataContextChange(order: 1)]
3 references
public IValueBinding? ItemValueBinding
{
    get { return (IValueBinding?)GetValue(ItemValueBindingProperty); }
    set { SetValue(ItemValueBindingProperty, value); }
}
5 references
public static readonly DotvvmProperty ItemValueBindingProperty =
    DotvvmProperty.Register<IValueBinding?, SelectorBase>(nameof(ItemValueBinding));
```

Demo: menu

```
<nav>
  <ul>
    <li>
      <a href='...'>        ↙ binding
            ...             ↖ binding
      </a>
    </li>
     ⋮   ← Repeater
  </ul>
</nav>
```

Demo: menu



```
<nav>
  <ul>
    <li>
      <a href='...'>
              ...
      </a>
    </li>
    ⋮
  </ul>
</nav>
```

Build control tree

Init

Load

PreRender

Render

Build control tree

Init

← view model is loaded

Load

PreRender

Render

don't touch others, they may be already redered

Build control tree

Init

← view model is loaded

Load

Comand is executed →

PreRender

Render

dont touch others, they may be already rederd command binding - execute method & id

found by
- id
- data context

in the control tree after Load

Please fill out a short survey

https://bit.ly/3bdhHWn

DOTVVM

VIRTUAL CONFERENCE

APRIL 29-30

DOTVVM