

DOTVVM
COMMUNITY MEETUP
MAY 27, 2021

Web Components Hot reload

Tomas Herceg

[@hercegtomas](#)

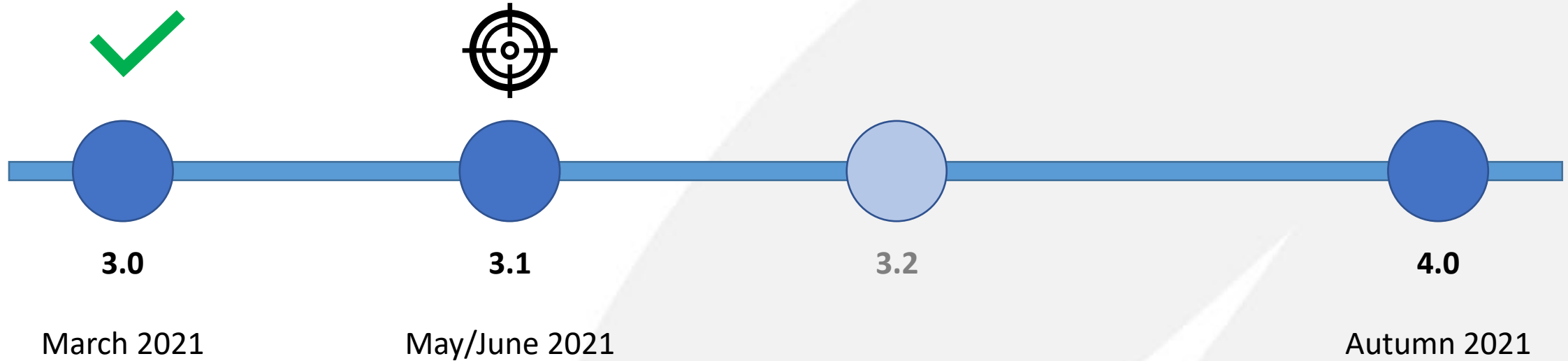
BUILD
new web apps

with
MVVM approach

MODERNIZE
Web Forms apps

and migrate them
to .NET Core

Timeline



What's coming in DotVVM 3.1



LINQ methods in bindings

- Present in DotVVM 3.0

- Select, Where

- Added in DotVVM 3.1

- All, Any, Concat, Count, Distinct

- FirstOrDefault, LastOrDefault

- Max, Min

- OrderBy, OrderByDescending

- Skip, Take



List and Dictionary in bindings

🌿 List<T>

- 🌿 Add, AddRange
- 🌿 Clear
- 🌿 Insert, InsertRange
- 🌿 RemoveAt, RemoveRange, RemoveAll
- 🌿 Reverse
- 🌿 Custom extensions: AddOrUpdate, RemoveFirst, RemoveLast

🌿 Dictionary<K, V>

- 🌿 [...]
- 🌿 ContainsKey
- 🌿 Remove
- 🌿 Clear



Syntax enhancements in bindings

- Type inference in lambdas

- `{value: Customers.Where(c => ...)}`

- String interpolation

- `{value: $"Hello {Name}"}`



Other improvements

- 🌿 FileUpload without hidden iframe
- 🌿 AbortSignal for commands and staticCommand
- 🌿 Refactoring and redesign of DotVVM CLI
 - 🌿 We need it to improve IntelliSense experience and to ship Selenium Generator



Hot reload in DotVVM apps



Hot reload experience

- 🍃 Speeds up developer inner loop
 - 🍃 Change code
 - 🍃 Recompile
 - 🍃 Refresh the page in the browser
- 🍃 Previously, it took 10-20 secs based on the app size
- 🍃 Less than 1 sec with hot reload



How it works

🍃 .NET 6 Preview 4

- 🍃 Changes in code applied to the running process **without restart**
- 🍃 Method body is replaced with new version

🍃 DotVVM.Diagnostics.ViewHotReload package

- 🍃 Uses SignalR to detect changes in DotHTML files
- 🍃 Refreshes the page automatically
 - 🍃 While keeping the viewmodel state

🍃 <https://github.com/riganti/dotvvm-utils-hot-reload>

- 🍃 You need Visual Studio 16.11 Preview 1 & .NET 6 Preview 4



DEMO

Hot reload in DotVVM



Web Components



What is a Web Component?

- 🌿 API to define custom HTML elements
- 🌿 Shadow DOM
 - 🌿 Allows the browser to render something else for such elements

```
<script type="module" src="..."></script>
```

```
<fluent-select>  
  <fluent-option>One</fluent-option>  
  <fluent-option disabled>Two</fluent-option>  
</fluent-select>
```



Advantages

- ▀ Anyone can implement support for custom elements which live directly in the browser
 - ▀ You can set attributes and modify the child elements
 - ▀ Based on that, the component changes its Shadow DOM
- ▀ Framework-agnostic – works everywhere
 - ▀ Including DotVVM!



Fluent UI

- 🍃 Components used in Azure, Office 365 and other Microsoft apps
- 🍃 Originally written in React
- 🍃 Now MS is working on the Web Components version
 - 🍃 <https://github.com/microsoft/fluentui/tree/master/packages/web-components>



Fluent UI in DotVVM

- ✦ Import the script and just use the custom elements
- ✦ Data-binding to the controls works
 - ✦ `<fluent-button`
 `Text="{value: ButtonText}"`
 `Events.Click="{command: Click()}" />`
- ✦ Data-binding from the controls needs custom binding
 - ✦ `Fast.Bind-attributeName="{value: Property}"`



Easier way to write controls

- DotVVM 4.0 will introduce a new way to build controls
 - Ideal for wrappers over existing controls or HTML elements
- Inherit from `CompositeControl`
- Implement a static method called `GetContents`
 - Parameters are emitted as DotVVM properties
- <https://github.com/tomasherceg/dotvvm-webcomponent-experiment>



DEMO

Fluent UI in DotVVM



Resources

🍃 Homepage
dotvvm.com

🍃 Documentation
dotvvm.com/docs

🍃 GitHub repo
github.com/riganti/dotvvm



Q&A

