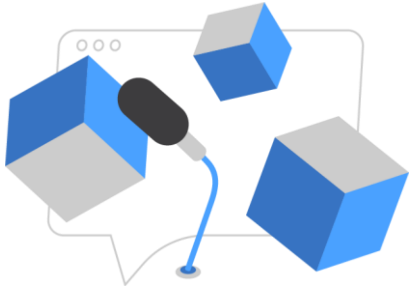


DOTVVM
COMMUNITY MEETUP

APR 22, 2021

DotVVM 3.0 and beyond

Tomas Herceg

@hercegtomas

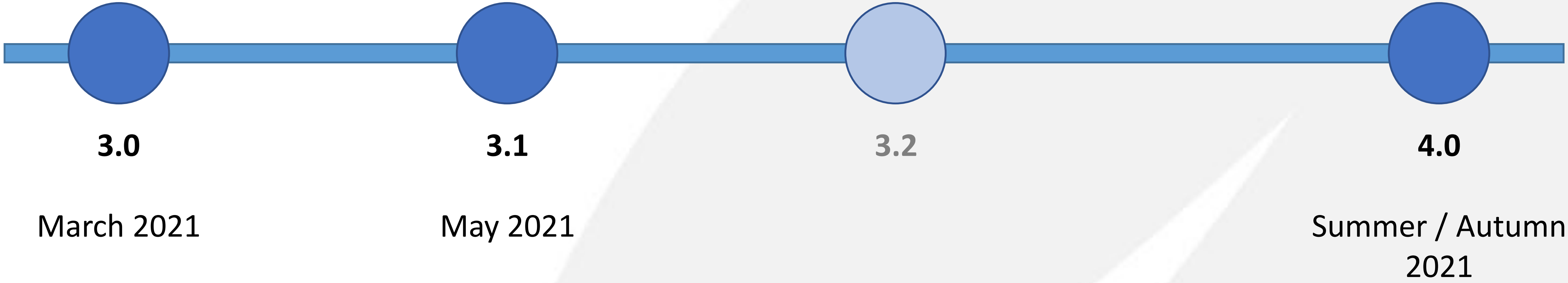
BUILD new web apps

with
MVVM approach

MODERNIZE Web Forms apps

and migrate them
to .NET Core

Timeline



3.0

March 2021

3.1

May 2021

3.2

4.0

Summer / Autumn 2021

What’s new in DotVVM 3.0

DotVVM 3.0 Released in March

- Hotfix 3.0.1
  - Released on April 11
- Hotfix 3.0.2
  - Will be released this week

- Mostly fixing breaking changes discovered when upgrading our own apps
- Release notes
  - https://github.com/riganti/dotvvm/releases/tag/v3.0

JS Directive

- Allows to interact with DotVVM in JavaScript code

- Use cases
  - Client-side changes to the viewmodel
  - Calling custom APIs
  - Integration with 3rd party components and libraries

- `@js FeatureSamples_Resources_Incrementer`

DOTVVM

JS Directive

- JavaScript file must **export default function**
  - Receives the context argument
    - `context.namedCommands.MethodName`
      - Allows to call DotVVM commands and static commands
      -
        ```
        <dot:NamedCommand Name="MyMethod"
                          Command="{command: ...}" />
        ```
  - Should return object with methods
    - They can be called from DotVVM
      - `{staticCommand: _js.Invoke("MethodInModule", args...)}`

DEMO

JS Directive

Typed viewmodels & Immutable state

- State is persisted in a plain JS object
  - Immutable – every change replaces parts of the object

  - `dotvvm.state` – returns the current state snapshot
  - `dotvvm.patchState(…)` – preferred way to update the state

- Knockout observables are just a façade
- Assignments from JS are validated
  - Value can be coerced to the correct type, or you get exception

DEMO

Typed viewmodels

LINQ and lambda expressions

- Lambda expressions in data-bindings
  - No type inference yet – you need to specify argument types

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

- Where and Select
- Other LINQ methods will be in 3.1
  - https://github.com/riganti/dotvvm/pull/965

Other improvements

- Complete rewrite of the TypeScript part of DotVVM
  - ES6 modules, `rollup` for bundling, polyfills for IE 11…
- Refactoring of the postback pipeline
- New events in `dotvvm.events` with more information

How to try

- **DotVVM 3.0** is released
- **DotVVM Business Pack 3.0 Preview 3** compatible
  - Final version will be released soon

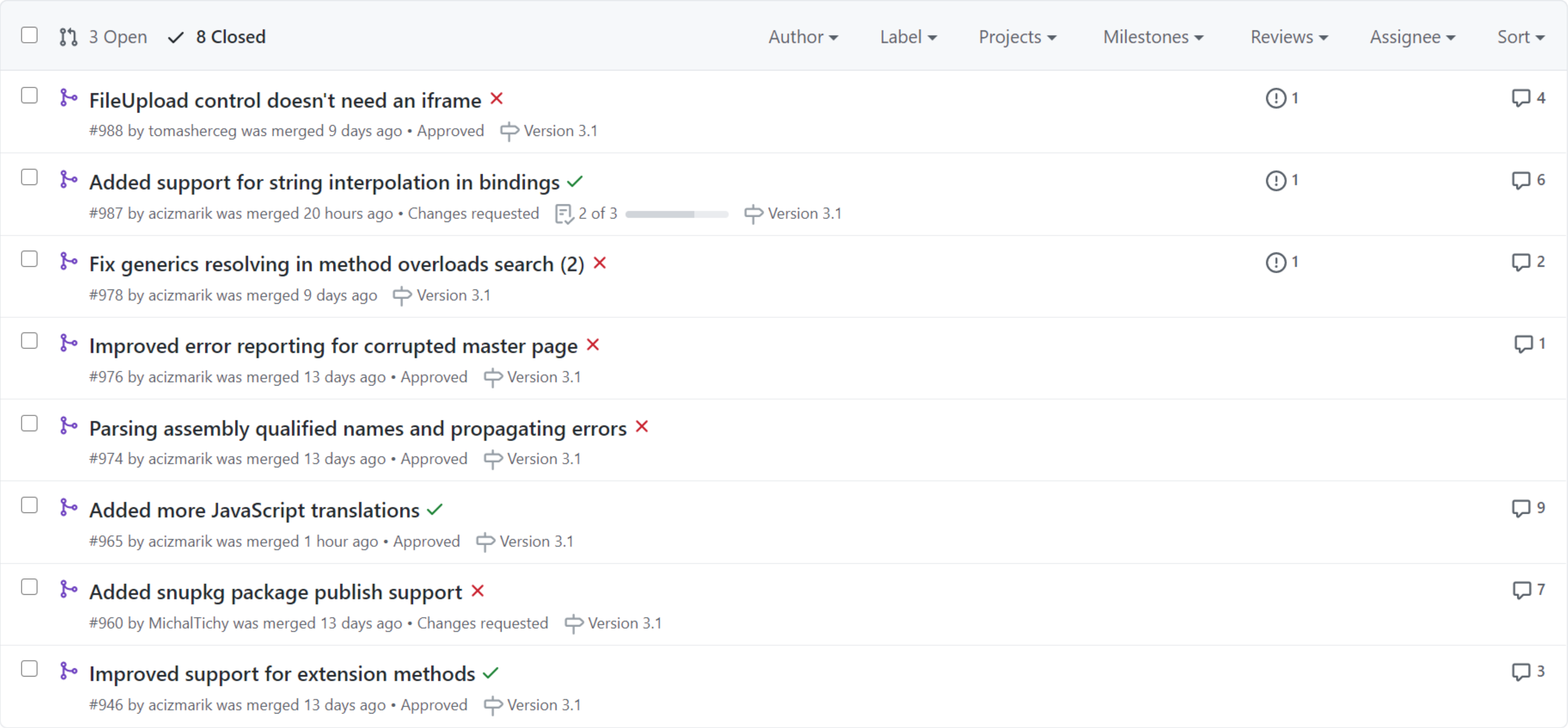
- Updated JS directive sample
  - https://github.com/riganti/dotvvm-samples-js-integration

DotVVM 3.1

Plan

- Type inference in lambdas
  - `{value: Customers.Where(c => …)}`
- More translatable LINQ methods
  - And not only LINQ methods
- String interpolation
- `FileUpload` without `iframe`

Already merged



□ 3 Open ✓ 8 Closed — Author ▾ Label ▾ Projects ▾ Milestones ▾ Reviews ▾ Assignee ▾ Sort ▾

□ FileUpload control doesn't need an iframe ✗ — ⓘ 1 — 💬 4
#988 by tomasherceg was merged 9 days ago • Approved — Version 3.1

□ Added support for string interpolation in bindings ✓ — ⓘ 1 — 💬 6
#987 by acizmarik was merged 20 hours ago • Changes requested — 2 of 3 — Version 3.1

□ Fix generics resolving in method overloads search (2) ✗ — ⓘ 1 — 💬 2
#978 by acizmarik was merged 9 days ago — Version 3.1

□ Improved error reporting for corrupted master page ✗ — 💬 1
#976 by acizmarik was merged 13 days ago • Approved — Version 3.1

□ Parsing assembly qualified names and propagating errors ✗
#974 by acizmarik was merged 13 days ago • Approved — Version 3.1

□ Added more JavaScript translations ✓ — 💬 9
#965 by acizmarik was merged 1 hour ago • Approved — Version 3.1

□ Added snupkg package publish support ✗ — 💬 7
#960 by MichalTichy was merged 13 days ago • Changes requested — Version 3.1

□ Improved support for extension methods ✓ — 💬 3
#946 by acizmarik was merged 13 days ago • Approved — Version 3.1

Areas for DotVVM 4.0

Easier way to write controls

- Markup controls can‘t do everything
- Code-only controls are hard to maintain

- General problems
  - Passing templates in controls
  - Passing bindings in controls
  - Creating binding expressions at runtime

Easier way to write controls

- Simplify building of control hierarchies
  - „composite“ controls
  - https://github.com/riganti/dotvvm/pull/535
- Get rid of DotVVM property declarations

  -
    ```
    @controlProperty Title string = "default value",
               Required = true, AllowBinding = false
    ```
- `<dot:TemplateHost Template="{resource: _control.ItemTemplate}" />`
- Better API for building binding expressions

Integration with other frameworks

- Host React components in DotVVM page
  - Maybe export React component from JS module
  - Something better than **`ReactBridge`** in DotVVM Contrib
    - https://github.com/riganti/dotvvm-contrib/blob/master/Controls/ReactBridge/readme.md

- Support for Web components

```
<fast-card>
  <h2>{{value: Title}}</h2>
  <fast-button appearance="accent"
               Click="{command: ...}">Visit</fast-button>
</fast-card>
```

DOTVVM

Static command improvements

- Static commands are powerful
  - Can solve problems of classic `{command: …}` binding
  - Can modify viewmodel without calling server
  - Can call server if needed

- But they lack some features
  - Action & exception filters
  - Validation

DOTVVM

GridViewDataSet enhancements

- `GridViewDataSet` is designed for command bindings

- Can be loaded with static command, but
  - Not very comfortable

    -
      ```
      {staticCommand: Data = _service.LoadData(
                                  data.SortingOptions,
                                  data.PagingOptions)}
      ```
  - `GridView` and `DataPager` controls need to be customized not to make classic commands

GridViewDataSet enhancements

- Pluggable paging and sorting options
  - Paging with page index & page size
  - Paging with next page token (a.k.a. load more...)
  - Sorting with multiple criteria
- Pluggable templates for GridView and DataPager
  - Different controls, customizable command/static command
- DataSetLoader component
  - Specifies expression to obtain new data
- Possible enhancements: infinite scrolling, virtualization

Support for new C# features

- New constructs cannot be used in the viewmodel
  - Record types
  - Nullable reference types
  - …
- Hot reload for DotVVM
  - Partially implemented
  - https://github.com/riganti/dotvvm-utils-hot-reload

Resources

- Homepage
  dotvvm.com
- Documentation
  dotvvm.com/docs
- GitHub repo
  github.com/riganti/dotvvm

DOTVVM

Q&A