

DotVVM v reálných aplikacích

Tomáš Herceg

Microsoft Most Valuable Professional
Microsoft Regional Director

CEO @ RIGANTI s.r.o.

tomas.herceg@riganti.cz

 @hercegtomas



riganti



Agenda

- Úvod
- Základní struktura line of business aplikace
- CRUD
- Generické viewmodely
- Vlastní komponenta
- Autentizace a autorizace
- Exception handling
- Modální okna
- Diagnostika a ladění výkonu

Proč DotVVM?

- Porodní bolesti .NET Core jsou z 90% vyřešeny
 - Začne se na něj přecházet ve velkém
- Snadné na naučení
- V RIGANTI se nám osvědčilo
 - Navrhli jsme jej, aby nám usnadnilo práci
 - Extension do Visual Studia velmi pomáhá

Aktuální stav

- Červenec 2016
 - DotVVM 1.0 – první stabilní verze
- Březen 2017
 - DotVVM 1.1 – přidána podpora ASP.NET Core
- Srpen 2017
 - DotVVM 1.1.5 – několik nových funkcí
- Q4 2017
 - DotVVM 1.2 – REST API bindingy, mnoho dalších funkcí

Kolik to stojí?

- Zdarma

- **DotVVM Framework**
 - jádro + 25 vestavěných komponent
- **DotVVM for Visual Studio**
 - základní IntelliSense
- **DotVVM Contrib**
 - komunitní komponenty
- **DotVVM Dynamic Data**
 - rozšíření DotVVM pro automatické generování uživatelského rozhraní

- Placené

- **Bootstrap for DotVVM**
 - komponenty pro snadné používání Bootstrapu
- **DotVVM Pro for Visual Studio**
 - kompletní IntelliSense a mnoho dalších funkcí
- **DotVVM Business Pack**
 - profesionální komponenty pro LOB aplikace

DotVVM Business Pack

- Profesionální sada komponent pro LOB aplikace
- První ostrý release na přelomu srpna a září
- Beta program (zdarma)
 - <https://www.dotvvm.com/landing/business-pack>
- Editor témat (od zítřka)
 - <https://themes.dotvvm.com>

.NET Summer Hackfest



- Akce pořádaná organizací **.NET Foundation**
 - Cílem je rozšířit open source komunitu
- DotVVM vybráno do druhé etapy
 - <https://github.com/riganti/dotvvm/blob/master/dotnetSummer.md>
- Přijďte si v pátek 11. srpna zaprogramovat!
 - <https://www.geekcore.cz/events/6085>
- Zapojte se do vývoje!

Jak DotVVM funguje?

```
public class ContactFormViewModel
{
    public string Name { get; set; }
    public string Email { get; set; }

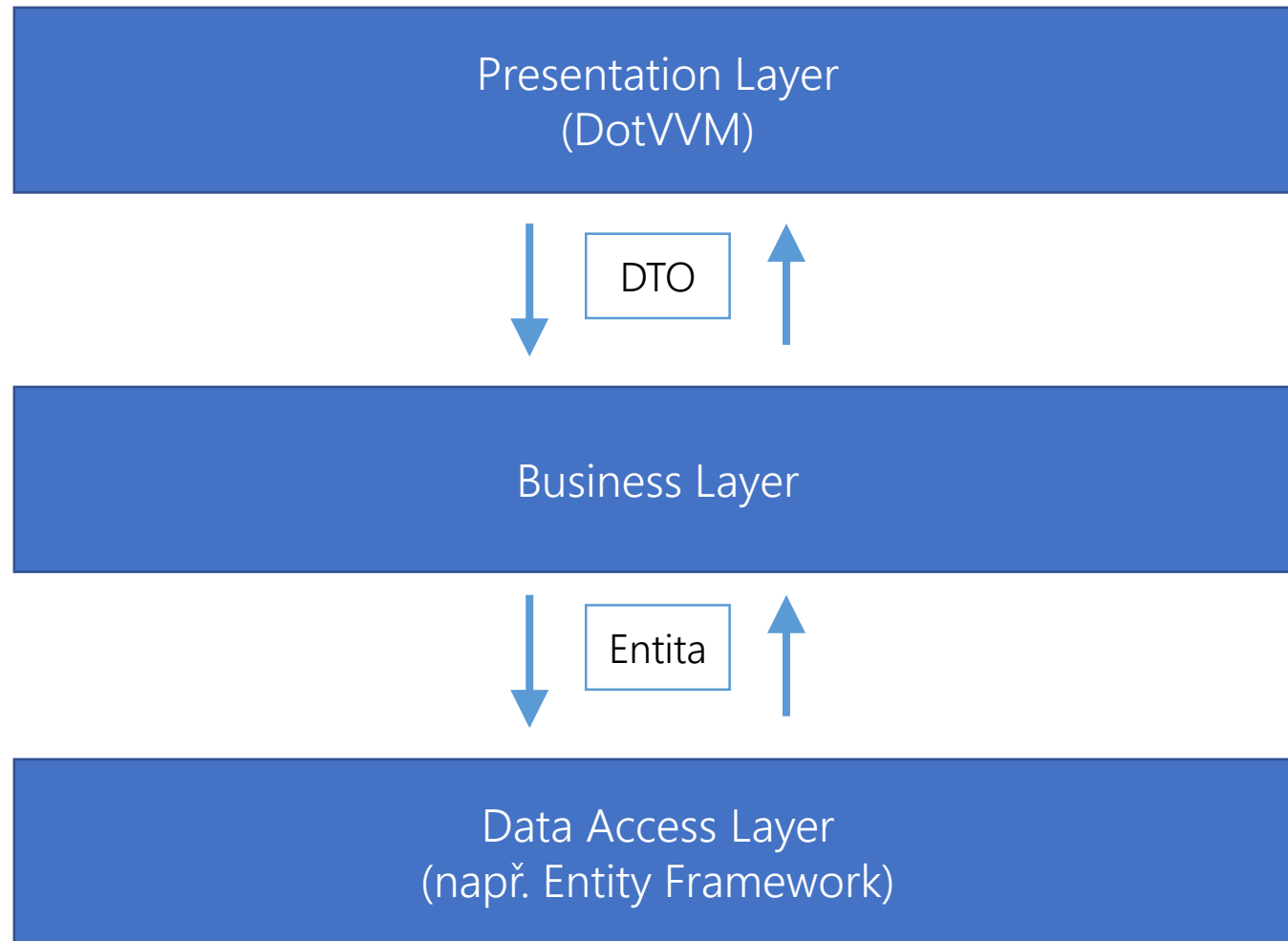
    public string Submit ()
    {
        ContactService.Submit(Name, Email);
    }
}
```

```
<div class="form-control">
    <dot:TextBox Text="{value: Name}" />
</div>
<div class="form-control">
    <dot:TextBox Text="{value: Email}" />
</div>
<div class="button-bar">
    <dot:Button Text="Submit"
        Click="{command: Submit()}" />
</div>
```


MVVM

- **View = HTML**
 - Serverové komponenty `<dot:Něco>`
 - Obousměrný data-binding `{value: NěcoZViewModelu}`
 - Direktivy `@viewModel` `MojeTřída`
- **ViewModel = C#**
 - Vlastnosti reprezentující stav view
 - Metody, které mohou být z view volány
- **Model**
 - Vrchní část business vrstvy: UI fasády a DTO

Architektura aplikace



Dependency Injection

- Základ pro jakoukoliv větší aplikaci
 - „Služby si nevytváříme sami, ale necháme si je podstrčit zvenku.“
- Jak na to?
 - Constructor Injection
 - Službu si vyžádáme v konstruktoru jako parametr
 - Property Injection
 - Službu zadeklarujeme jako public property
- Kontejner zajistí, že viewmodel dostane vše, co potřebuje
- Zároveň řeší životnost služeb a volá **Dispose**

Základní struktura aplikace

DEMO



riganti



Poznámky ke struktuře aplikace

- Business a Data Access Layers využívají různé knihovny
 - Entity Framework Core
 - AutoMapper
 - Riganti Utils Infrastructure
 - Castle Windsor
- Čím lépe nachystáte business vrstvu, tím lépe se bude UI psát

Jednoduchý CRUD

DEMO



riganti



Zásady pro práci s viewmodely

- ViewModely v sobě obsahují DTO objekt(y)
- DTO obsahují jen data
- DTO neobsahují nic, co by uživatel neměl vidět (PasswordHash)
- DTO by měla být jednoúčelová

Zásady pro práci s viewmodely

- ViewModel obsahuje „stav“ (`IsNewRecord`)
- ViewModel poskytuje UI „globální data“ (číselníky kategorií atd.)
- Struktura ViewModelu by měla odpovídat struktuře UI
 - Využívat vlastnost `DataContext`
 - Minimalizovat skoky do `_parent` atd.

Zobecnění pomocí generiky

DEMO



riganti



Generické fasády a viewmodely

- Je dobré psát kód obecně a využívat generiku
- Zbavit se copy-paste a search-replace
- Bázové třídy a rozhraní lze používat i napříč projekty
- „Application Blocks“

Vlastní komponenty

DEMO



riganti



Typy komponent

- Markup Controls
 - Opakované použití kusu markupu na více stránkách
 - Lze doplnit code behind a parametrizovat
 - Jednoduché na použití
- Code-only Controls
 - Pro složitější hierarchie a lepší kontrolu nad strukturou komponenty
 - Nebát se jich
 - Inspirujte se existujícími komponentami
<https://github.com/riganti/dotvvm/tree/master/src/DotVVM.Framework/Controls>

Autentizace a autorizace

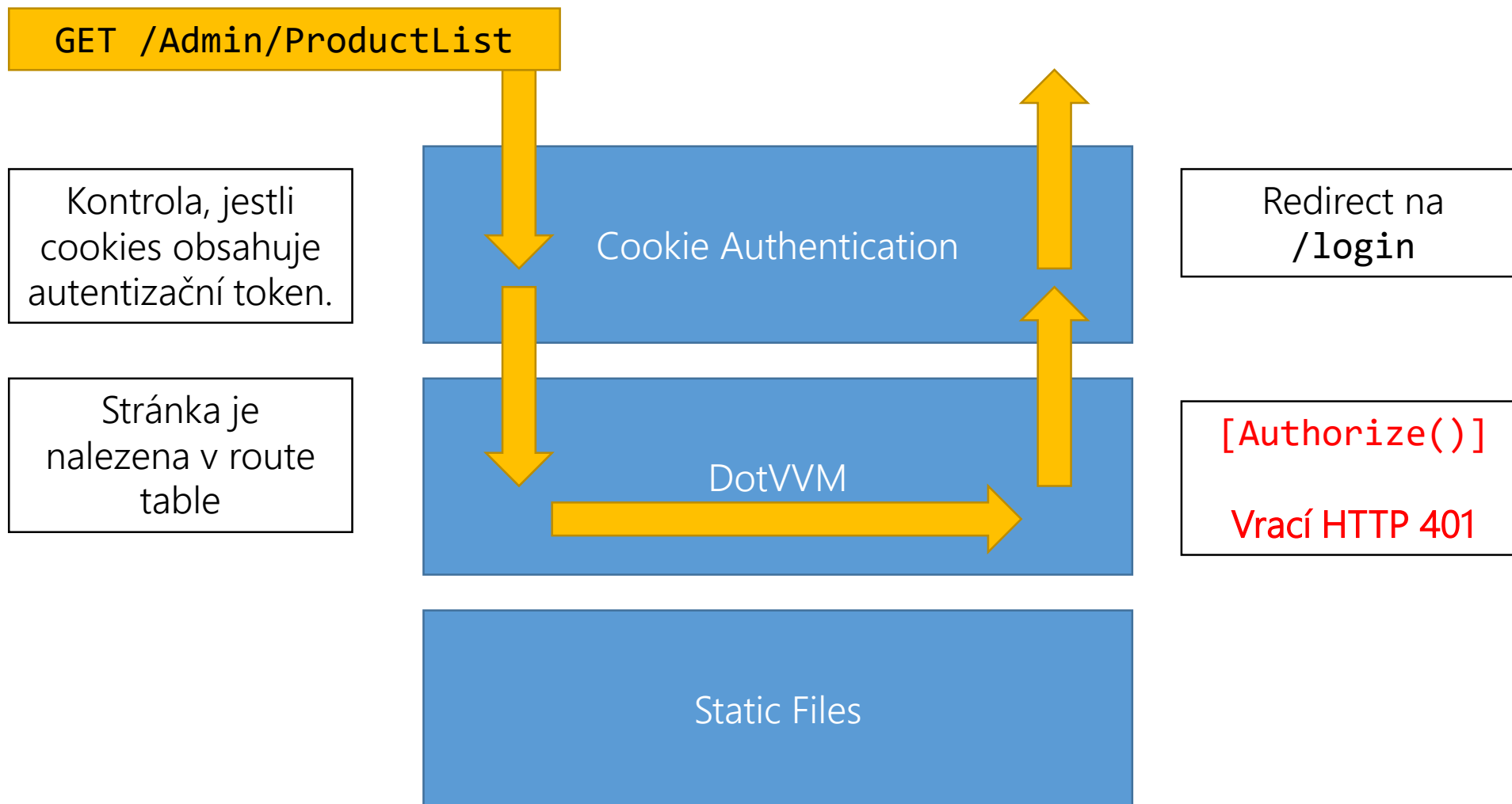
DEMO



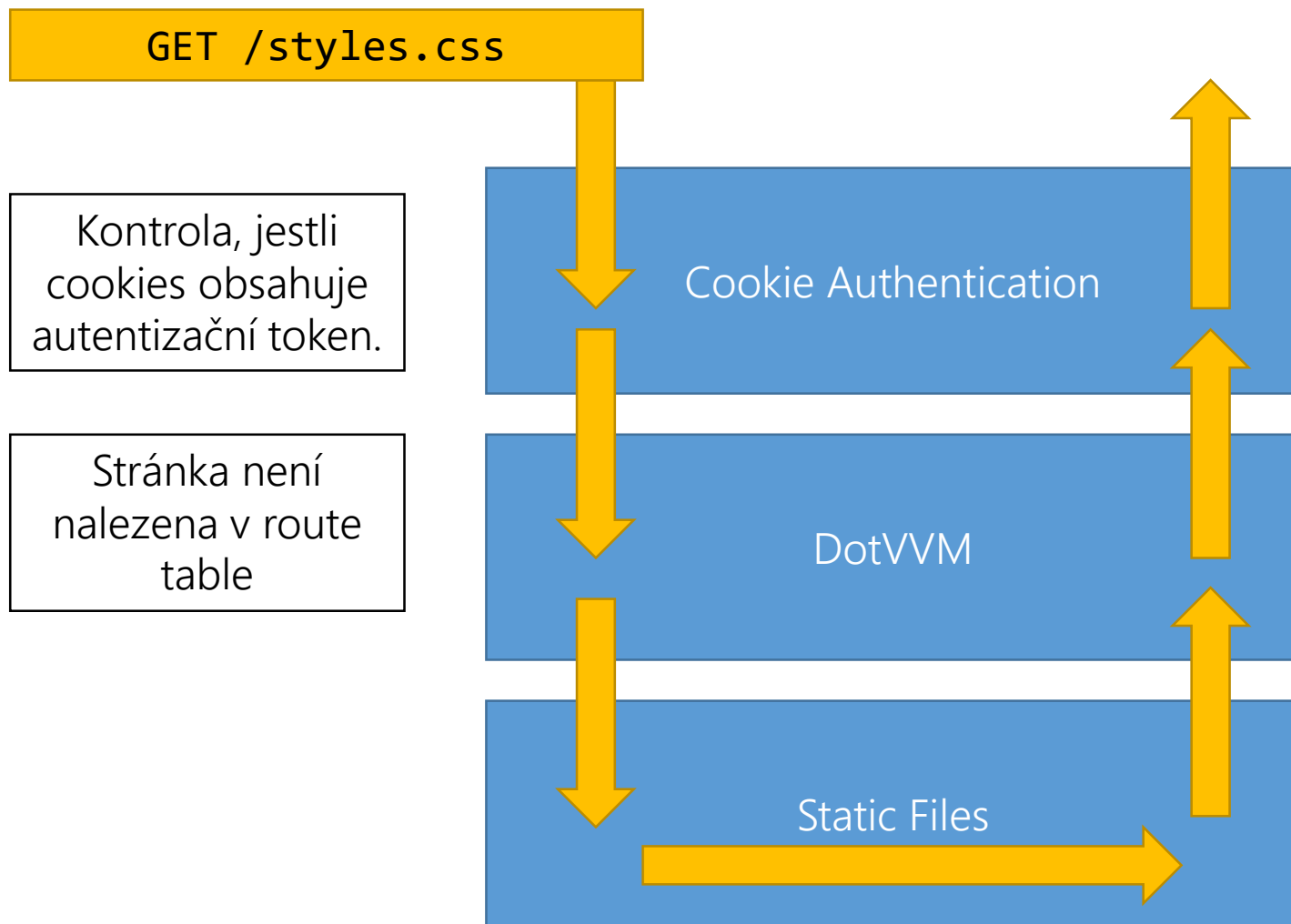
riganti



Middlewares v OWINu a ASP.NET Core



Middlewares v OWINu a ASP.NET Core



Autentizace

- Middleware se registruje **před DotVVM**
- Lze kombinovat (**AuthenticationType** pro rozlišení identit)
- Speciální chování DotVVM u redirectů
 - Při postbacku nelze vrátet HTTP 302
 - DotVVM provádí postbacky pomocí AJAXu a 302 nelze zachytit
 - Je nutné použít `DotvvmAuthenticationHelpers.ApplyRedirectResponse`
- <https://www.dotvvm.com/docs/tutorials/advanced-authentication-authorization/latest>

Autorizace

- Atribut `[Authorize]`
- Pozor na namespace: `DotVVM.Framework.Runtime.Filters`
- Možno specifikovat role nebo claimy
- Komponenty `AuthenticatedView`, `RoleView`, `ClaimView`

Exception handling

DEMO



riganti



Možnosti

- Globálně pomocí exception filteru
 - Nastavit `Context.IsCommandExceptionHandled = true`
- Případně filtr aplikovat jen tam, kde je třeba
- Ručně tam, kde je třeba, pomocí vlastní metody
 - Zavolá předaného delegáta
 - Ošetří výjimky
 - Pozor na zachytávání `DotvvmInterruptRequestExecutionException`

Modální dialogy

DEMO



riganti



Možná řešení

- DotVVM Framework komponentu pro modální dialog neobsahuje
- Komponenta **ModalDialog** dostupná v placených balících
 - Bootstrap for DotVVM
 - DotVVM Business Pack
- Lze samozřejmě použít vlastní řešení
 - **div** nastýlovaný pomocí CSS a zobrazovaný/skrývaný pomocí **Visible**
 - Ale má to kompromisy (ovládání klávesnicí atd.)

Ovládání ModalDialogu

- Vlastnost `IsDisplayed` bindovaná do viewmodelu
- Lze přepínat pomocí `staticCommand`
 - Nevyžaduje postback
- Doporučení
 - Obsah dialogu udělat jako markup control
 - Tělo dialogu má vlastní viewmodel
 - Komunikace pomocí eventů

Diagnostika aplikací

DEMO



riganti



Poznámky

- DotVVM je jako ORM
 - Značně usnadňuje práci
 - Pokud si nedáte pozor, může být neefektivní
- Typické problémy
 - Přenáší se mnohem více dat, než je potřeba
 - Dotazujeme se databáze víckrát, než je nutné
- Cíl je mít co nejlepší nástroje, aby k tomu nedocházelo
 - Application Insights
 - MiniProfiler

NuGet balíčky

- Application Insights
 - `DotVVM.Tracing.ApplicationInsights.Owin`
 - `DotVVM.Tracing.ApplicationInsights.AspNetCore`
 - `<dot:ApplicationInsightsJavascript />`
- MiniProfiler
 - `DotVVM.Tracing.MiniProfiler.Owin`
 - `DotVVM.Tracing.MiniProfiler.AspNetCore`
 - `<dot:MiniProfilerWidget />`

```
4 {  
5     1 reference | exyi, 342 days ago | 1 author, 1 change  
6     public class ButtonViewModel : DotvvmViewModelBase  
7     {  
8         1 reference | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live  
9         public string Title { get; set; }  
10  
11        1 reference | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live  
12        public int Count { get; set; }  
13  
14        0 references | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live  
15        public ButtonViewModel()  
16        {  
17            Title = "Hello from DotVVM!";  
18        }  
19  
20        0 references | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live  
21        public void IncreaseCount()  
22        {  
23            Count++;  
24        }  
25    }  
26 }
```

110 %
Encoding: Western European (Windows) Content type: CSharp Classification: None Selection: 302 Roles: PRIMARYDOCUMENT

DotVVM Diagnostics Tool Window

Microsoft Edge

Status	Type	URL	Time	Size	Request Headers	Tree View	JSON View
200	Get	/ControlSamples/Button/ButtonOnClick	335 ms	1 kB	Request ViewModel	Count 2 Int64	
200	Command	/ControlSamples/Button/ButtonOnClick	94 ms	234 B	Response Headers	Title Hello from DotVVM! String	
200	Get	/ControlSamples/Button/Button	1 ms	1 kB	Response ViewModel Diff		
200	Command	/ControlSamples/Button/Button	15 ms	252 B			
200	Command	/ControlSamples/Button/Button	0 ms	252 B			
200	Command	/ControlSamples/Button/Button	1 ms	252 B			

DotVVM Diagnostics Tool Window Output Immediate Window Command Window Exception Settings Breakpoints Error List Autos Watch 1

Solution Explorer

- SeleniumGenerator
- Solution Items
- Tests
 - DotVVM.Framework.PerfTests
 - DotVVM.Framework.Tests.AspCore
 - DotVVM.Framework.Tests.Common
 - DotVVM.Framework.Tests.Javascript
 - DotVVM.Framework.Tests.Owin
 - DotVVM.Samples.Tests
 - DotVVM.Samples.Tests.CompletenessChecker
- Tools
- DotVVM.Compiler
- DotVVM.Compiler.Light
- DotVVM.Core
- DotVVM.Framework
 - Dependencies
 - Properties
 - Binding
 - Compilation
 - Configuration
 - Controls
 - DependencyInjection
 - DotvvmBuilderExtensions.cs
 - DotvvmOptions.cs
 - DotVVMServiceCollectionExtensions.cs
 - IDotvvmOptions.cs
 - RequireDependencyInjectionAttribute.cs
 - Diagnostics
 - Models
 - DiagnosticsInformation.cs
 - HttpHeaderItem.cs
 - RequestDiagnostics.cs
 - ResponseDiagnostics.cs
 - DiagnosticsInformationSender.cs
 - DiagnosticsRenderer.cs
 - DiagnosticsRequestTracer.cs

```

4  {
5  1 reference | exyi, 342 days ago | 1 author, 1 change
6  public class ButtonViewModel : DotvvmViewModelBase
7  {
8  1 reference | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live
9  public string Title { get; set; }
10
11 1 reference | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live
12 public int Count { get; set; }
13
14 0 references | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live
15 public ButtonViewModel()
16 {
17     Title = "Hello from DotVVM!";
18 }
19
20 0 references | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live
21 public void IncreaseCount()
22 {
23     Count++;
24 }
25 }
    
```

Encoding: Western European (Windows) Content type: CSharp Classification: None Selection: 302 Roles: PRIMARYDOCUMENT

DotVVM Diagnostics Tool Window

Microsoft Edge

Status	Type	URL	Time	Size	Request Headers	Request ViewModel	Response Headers	Response ViewModel Diff
200	Get	/ControlSamples/Button/ButtonOnClick	335 ms	1 kB				
200	Command	/ControlSamples/Button/ButtonOnClick	94 ms	234 B				
200	Get	/ControlSamples/Button/Button	1 ms	1 kB				
200	Command	/ControlSamples/Button/Button	15 ms	252 B				
200	Command	/ControlSamples/Button/Button	0 ms	252 B				
200	Command	/ControlSamples/Button/Button	1 ms	252 B				

Tree View JSON View

```

Schema: <No Schema Selected>
1  {
2  2 "Count": 3,
3  3 "CountOptions": {
4  4   "type": "int32"
5  5 }
6  6 }
    
```

Encoding: Western European (Windows) Content type: JSON Classification: None

Solution Explorer

Search Solution Explorer

- SeleniumGenerator
- Solution Items
- Tests
 - DotVVM.Framework.PerfTests
 - DotVVM.Framework.Tests.AspNetCore
 - DotVVM.Framework.Tests.Common
 - DotVVM.Framework.Tests.Javascript
 - DotVVM.Framework.Tests.Owin
 - DotVVM.Samples.Tests
 - DotVVM.Samples.Tests.CompletenessChecker
- Tools
- DotVVM.Compiler
- DotVVM.Compiler.Light
- DotVVM.Core
- DotVVM.Framework
 - Dependencies
 - Properties
 - Binding
 - Compilation
 - Configuration
 - Controls
 - DependencyInjection
 - DotvvmBuilderExtensions.cs
 - DotvvmOptions.cs
 - DotVVMServiceCollectionExtensions.cs
 - IDotvvmOptions.cs
 - RequireDependencyInjectionAttribute.cs
 - Diagnostics
 - Models
 - DiagnosticsInformation.cs
 - HttpHeaderItem.cs
 - RequestDiagnostics.cs
 - ResponseDiagnostics.cs
 - DiagnosticsInformationSender.cs
 - DiagnosticsRenderer.cs
 - DiagnosticsRequestTracer.cs

```

4 {
5     1 reference | exyi, 342 days ago | 1 author, 1 change
6     public class ButtonViewModel : DotvvmViewModelBase
7     {
8         1 reference | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live
9         public string Title { get; set; }
10
11        1 reference | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live
12        public int Count { get; set; }
13
14        0 references | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live
15        public ButtonViewModel()
16        {
17            Title = "Hello from DotVVM!";
18        }
19
20        0 references | exyi, 342 days ago | 1 author, 1 change | 0 exceptions, ? live
21        public void IncreaseCount()
22        {
23            Count++;
24        }
25    }

```

Encoding: Western European (Windows) Content type: CSharp Classification: None Selection: 302 Roles: PRIMARYDOCUMENT

DotVVM Diagnostics Tool Window

Status	Type	URL	Time	Size	Request Headers	Response Headers	Response ViewModel Diff
200	Get	/ControlSamples/Button/ButtonOnClick	335 ms	1 kB			
200	Command	/ControlSamples/Button/ButtonOnClick	94 ms	234 B			
200	Get	/ControlSamples/Button/Button	1 ms	1 kB			
200	Command	/ControlSamples/Button/Button	15 ms	252 B			
200	Command	/ControlSamples/Button/Button	0 ms	252 B			
200	Command	/ControlSamples/Button/Button	1 ms	252 B			

Content Negotiation
 Accept: */*
 Accept-Encoding: gzip, deflate
 Accept-Language: en-US,en;q=0.7,cs;q=0.3

Connection Management
 Connection: Keep-Alive

Cookies
 Cookie: dotvvm_sid_localhost-16018=CfDJ8DYEvhcXkSFAoQgQ6a%2Frta1V

Message Body
 Content-Length: 527
 Content-Type: application/json

Request Context

Caching

Other

Solution Explorer

Search Solution Explorer

- ▶ SeleniumGenerator
- ▶ Solution Items
- ▶ Tests
 - ▶ DotVVM.Framework.PerfTests
 - ▶ DotVVM.Framework.Tests.AspCore
 - ▶ DotVVM.Framework.Tests.Common
 - ▶ DotVVM.Framework.Tests.Javascript
 - ▶ DotVVM.Framework.Tests.Owin
 - ▶ DotVVM.Samples.Tests
 - ▶ DotVVM.Samples.Tests.CompletenessChecker
- ▶ Tools
- ▶ DotVVM.Compiler
- ▶ DotVVM.Compiler.Light
- ▶ DotVVM.Core
- ▶ DotVVM.Framework
 - ▶ Dependencies
 - ▶ Properties
 - ▶ Binding
 - ▶ Compilation
 - ▶ Configuration
 - ▶ Controls
 - ▶ DependencyInjection
 - ▶ DotvvmBuilderExtensions.cs
 - ▶ DotvvmOptions.cs
 - ▶ DotVVMServiceCollectionExtensions.cs
 - ▶ IDotvvmOptions.cs
 - ▶ RequireDependencyInjectionAttribute.cs
 - ▶ Diagnostics
 - ▶ Models
 - ▶ DiagnosticsInformation.cs
 - ▶ HttpHeadersItem.cs
 - ▶ RequestDiagnostics.cs
 - ▶ ResponseDiagnostics.cs
 - ▶ DiagnosticsInformationSender.cs
 - ▶ DiagnosticsRenderer.cs
 - ▶ DiagnosticsRequestTracer.cs

Závěrem



riganti



Chcete přejít na DotVVM? Pomůžeme vám!

- Vývoj konkrétních komponent
- Návrh infrastruktury na základě Riganti Utils Infrastructure
 - <https://github.com/riganti/infrastructure>
- Konzultace a školení
- Placený support (přítel na Skypu)
- Kontaktujte nás na info@dotnetcollege.cz

Zdroje a materiály

- Ukázková aplikace
 - <https://github.com/riganti/dotvvm-samples-northwind>
- Dokumentace
 - <https://www.dotvvm.com/docs>
- Gitter chat
 - <https://gitter.im/riganti/dotvvm>
- Výuková videa v češtině
 - <https://www.youtube.com/channel/UCPD-YWullwgSgoX5ff2SWcA>

www.dotvvm.com

@dotvvm

Tomáš Herceg

Microsoft MVP (ASP.NET)

Microsoft Regional Director

CEO @ RIGANTI s.r.o.



riganti

