


# Razvoj primijenjene programske potpore

---

## 13. Izrada programske dokumentacije

Video upute: <https://bit.ly/3kull2c>

# Dokumentacija programskog koda

- XML dokumentacijski komentari
  - omogućavaju pisanje dokumentacije koja se odnosi na programski kod
  - označavaju se trostrukom kosom crtom : ///
  - dodaju se ispred programskog bloka na koji se odnose
  - sadrže posebne XML oznake koje opisuju dokumentirani kod
  - Primjer:  WebServices\WebApi\WebApi\Util\Extensions\ExceptionExtensions.cs

```
/// <summary>
/// Razred sa proširenjima za iznimke
/// </summary>
public static class ExceptionExtensions {
    /// <summary>
    /// Vraća sadržaj poruka cijele hijerarhije neke iznimke.
    /// </summary>
    /// <param name="exc">Iznimka čija se hijerarhija poruka
    ///dohvaća </param>
    /// <returns>String s porukama svih unutarnjih iznimki.
    /// Poruka svake iznimmke dodana je u novi redak</returns>
    public static string CompleteExceptionMessage(this Exception exc ...
```

# Xml datoteka s programskom dokumentacijom

- /doc opcija prevoditelja prilikom prevođenja uključuje traženje XML komentara u kodu te kreiranje dokumentacijske XML datoteke
    - Alternativno: kontekstni izbornik projekta :  
Project properties \ Build \ XML documentation file
      - po želji promijeniti lokaciju generirane datoteke
      - Bolja varijanta – dodati sljedeći sadržaj u odgovarajući csproj
- ```
<GenerateDocumentationFile>true</GenerateDocumentationFile>
```
- Prevođenje proizvede *NazivAsemblija.XML*
    - dodavanjem reference na neki dll, uzima se i istoimena xml datoteka (ako postoji) – služi za prikaz dokumentacijskih komentara
  - Služi kao pomoć prilikom programiranja
    - Nudi opise razreda i postupaka iz referenciranog projekta


# XML oznake

- XML oznake za komentiranje programskog koda
  - `<summary>` - sažetak opisa razreda ili člana
  - `<remarks>` - nadopunjuje opis o razredu ili članu
  - `<param>` - opis parametra metode
  - `<paramref>` - oznaka da se opis odnosi na parametar
  - `<returns>` - povratne vrijednosti metode
  - `<value>` - opis vrijednosti svojstva
  - `<exception>` - iznimke koje mogu biti bačene u opisanom kodu.  
Sadrži referencu na iznimku.
  - `<typeparam>` - opis generičkog razreda ili metode
  - `<typeparamref>` - tip parametra kod generičkih razreda ili metoda

# XML oznake (nastavak)

- `<c>` - tekst unutar oznake je kod
  - `<code>` - tekst unutar oznake je višeretkovni kod
  - `<example>` - primjer
  - `<include>` - uključivanje vanjske datoteke s XML dokumentacijom
  - `<list>` - lista, nabrojanje elemenata
  - `<para>` - paragraf teksta
  - `<see>` - omogućava stvaranje linka
  - `<seealso>` - link u sekciji *See also*
  - ...
- 
- Za detaljniji opis oznaka pogledati:
    - <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/xml/doc/recommended-tags>

# Primjer generirane XML datoteke

- Primjer:   
WebServices\...WebApi\bin  
\Debug\net6.0\  
WebApi.dll

```
WebServices.xml
13 </returns></returns>
14 </member>
15 <member name="T:WebServices.Controllers.JTable.LookupController">
16 <summary>
17 Lookup controller prilagođen za jTable
18 </summary>
19 </member>
20 <member name="T:WebServices.Controllers.MjestoController">
21 <summary>
22 Web API servis za rad s mjestima
23 </summary>
24 </member>
25 <member name="M:WebServices.Controllers.MjestoController.Count(System.String)">
26 <summary>
27 vraća broj svih mjesta filtriran prema nazivu mjesta
28 </summary>
29 <param name="filter">Opcionalni filter za naziv mjesta</param>
30 <returns></returns>
31 </member>
32 <member name="M:WebServices.Controllers.MjestoController.GetAll(WebServices)">
33 <summary>
34 Dohvat mjesta (opcionalno filtrirano po nazivu mjesta).
35 Broj mjesta, poredak, početna pozicija određeni s loadParams.
36 </summary>
37 <param name="loadParams">Postavke za straničenje i filter</param>
38 <returns></returns>
39 </member>
40 <member name="M:WebServices.Controllers.MjestoController.Get(System.Int32)">
41 <summary>
42 vraća grad čiji je IdMjesta jednak vrijednosti parametra id
43 </summary>
44 <param name="id">IdMjesta</param>
45 <returns></returns>
46 </member>
47 <member name="M:WebServices.Controllers.MjestoController.Delete(System.Int32)">
48 <summary>
49 Brisanje mjesta određenog s id
50 </summary>
51 <param name="id">Vrijednost primarnog ključa (Id mjesta)</param>
52 <returns></returns>
53 <response code="204">Ako je mjesto uspješno obrisano</response>
54 <response code="404">Ako mjesto s poslanim id-om ne postoji</response>
```

# Alat za generiranje programske dokumentacije

- Xml datoteka korisna unutar razvojnog sučelja, ali nije prikladna za čitanje
- Za izradu programske dokumentacije namijenjene za čitanje koristi se DocFx
  - Generira HTML sadržaj na osnovi programske dokumentacije
  - Omogućava dodavanje dodatnih sadržaja u dokumentaciju
- Instalacija DocFx-a
  - <https://dotnet.github.io/docfx/>
  - Preporuka: dodati docfx u putanju
  - Alternativa: Unutar Visual Studio 2022 instalirati NuGet paket Microsoft.DocAsCode.App
- Napomena: podržava podskup oznaka iz xml dokumenacije
  - [http://dotnet.github.io/docfx/spec/triple\\_slash\\_comments\\_spec.html](http://dotnet.github.io/docfx/spec/triple_slash_comments_spec.html)

The screenshot shows the DocFx-generated HTML documentation for the `PageLinkTagHelper` class. The interface includes a sidebar with a search bar and a tree view of the project structure. The main content area displays the class name, a description, an example of HTML output, and instructions on how to use the tag helper in a Razor view.

**Programska dokumentacija**

Programska dokumentacija / Firma.Mvc.TagHelpers / PageLinkTagHelper

Enter here to filter...

- + Firma.Mvc
- + Firma.Mvc.Controllers
- + Firma.Mvc.Controllers.Auto Complete
- + Firma.Mvc.Extensions
- + Firma.Mvc.Models
- Firma.Mvc.TagHelpers
  - PageLinkTagHelper
- + Firma.Mvc.Util.Logging
- + Firma.Mvc.ViewComponents
- + Firma.Mvc.ViewModels

## Class PageLinkTagHelper

Tag helper za kreiranje vlastitih poveznica na stranice u rezultatu nekog upra HTML oznake `div` Primjer upotrebe

```
<div page-info="@Model.PagingInfo" page-action="Index" page-class="btn" page-class-normal="btn-default" page-class-selected="pull-right"></div>
```

U datoteku `_ViewImports.cshtml` potrebno dodati `@addTagHelper *, Microsoft.DocAsCode.TagHelpers` te u pogled uključiti vlastitu javascript datoteku `gotopage.js`


Inheritance

- System.Object
  - PageLinkTagHelper

Namespace: `Firma.Mvc.TagHelpers`

Assembly: `Firma.Mvc.dll`

# Neki primjeri korištenja DocFX-a iz prakse (1)



Documentation

Montelektro.MVVM

Montelektro.Util

CME.Data

CME.Dal

CME.StartupWorkflow

Search

Documentation / How to develop new form?

Enter here to filter...

Introduction

Configuration files

navigation.json

How to develop new form?

+ Table dependencies

Startup workflow

```
public PLCTypesRepository(IDatabaseWrapper dbWrapper)
{
    this.db = dbWrapper;
    dependencyChecker = new DependencyRepository(dbWrapper);
}
```

## ActionResponse as a result

An easy way to catch an exception and return proper ActionResponse is to create anonymous function/action and call ActionResponse extension method

```
public ActionResponse InsertPlcType(PLCTypes type)
{
    Action func = () =>
    {
        string updateQuery = "INSERT INTO dbo.cme_PLCTypes (Name,Description) OUTPUT Inserted.Id V
alues (?,?)"; //ODBC does not support named parameters
        using (var command = db.Database.GetSqlCommand(updateQuery))
        {
            db.Database.AddInParameter(command, "@Name", DbType.String, type.Name);
            db.Database.AddInParameter(command, "@Description", DbType.String, type.Description);
            db.Database.ExecuteNonQuery(command);
        }
    };
    return func.ActionResponse();
}
```

IN THIS ARTICLE

Basic steps

Create repository

Data Transfer Obj

Repository interfa

> Repository imple

> Repository imple

ActionResponse a

Data for combob

Saving changes

Register repositor

Create View Mode

Choosing appropri



Enter here to filter...

## - Montelektro.MVVM

ChangedValue  
 ChangedValue.ChangeState  
 CustomCommandInvoker  
 CustomObservableCollection<T>  
 EditableCollection<T>  
 EditableObject  
 IChangeLogIdentifiable  
 Messenger  
 ObservableObject  
 RelayCommand

## + Montelektro.MVVM. Extensions

# Class CustomCommandInvoker

Simple WPF TriggerAction to bind events into commands Does not define CommandParameters in order to pass additional parameter If ever needed code for adding command parameter could be implemented as in <http://putridparrot.com/blog/a-simple-wpf-triggeraction-to-turn-events-into-commands/>

## Inheritance

↳ System.Object  
   ↳ CustomCommandInvoker

**Namespace:** Montelektro.MVVM

**Assembly:** Montelektro.MVVM.dll

## Syntax

```
public sealed class CustomCommandInvoker : TriggerAction<DependencyObject>
```

## Examples

Example of use: setting Interaction Trigger (EventTrigger) that calls a Command when an Event occurs.

```

<TextBlock Text="{Binding Name}" Tag="#FF0000">
  <i:Interaction.Triggers>
    <i:EventTrigger EventName = "ContextMenuOpening">
      <mvvm:CustomCommandInvoker Command = "{Binding Path=DataContext.ContextMenuOpen
ingCommand, RelativeSource={RelativeSource AncestorType={x:Type UserControl}}}" />
    </i:EventTrigger>
    <i:EventTrigger EventName = "ContextMenuClosing">
      <mvvm:CustomCommandInvoker Command = "{Binding Path=DataContext.ContextMenuClos
ingCommand, RelativeSource={RelativeSource AncestorType={x:Type UserControl}}}" />
    </i:EventTrigger>
  </i:Interaction.Triggers>
  ...

```

## Fields

## CommandProperty

Comand dependency property that allows (from WPF xaml file) binding an command that need to be invoked (e.g. when an event occurs)

## Declaration

```
public static readonly DependencyProperty CommandProperty
```

## Field Value

Type	Description
System.Windows.DependencyProperty	

# Neki primjeri korištenja DocFX-a iz prakse (2)

# Generiranje programske dokumentacije

- Upute koje slijede prilagođene su upute originalnih uputa  
[https://dotnet.github.io/docfx/tutorial/docfx\\_getting\\_started.html](https://dotnet.github.io/docfx/tutorial/docfx_getting_started.html)
  - Preduvjet: Instaliran DocFx, dodan u putanju i uspješno kompilirana web-aplikacija s napisanom dokumentacijom
- U mapi u kojoj se nalazi sln datoteka (paralelno s mapom web-aplikacije) iz naredbenog retka pokrenuti
  - stvara kostur projekta za programsku dokumentaciju
  - Alternativa: Visual Studio 2022:  
Projekt tipa ClassLibrary + NuGet paket  
*Microsoft.DocAsCode.App*

```
docfx init -q -o Docs
```

FER-RPPP > examples > WebServices > WebApi



Name	Da
.vs	4.1
ConsoleClientApps	4.1
WebApi	8.1
WebApi.sln	4.1

WebServices > WebApi > Docs

Name










api
apidoc
articles
images
src
.gitignore
docfx.json
index.md
toc.yml

# Uređivanje početne stranice dokumentacije

- Docfx predstavlja strukturu projekta za izradu projektne dokumentacije
  - Mapa api će naknadno sadržavati informacije o razredima iz web-aplikacije
  - Mapa articles predstavlja proizvoljne dokumente koji se mogu uključiti u programsku dokumentaciju
    - Izmijeniti sadržaj dodatne dokumentacije po uzoru na  WebServices \ WebApi \ Docs \ articles \ index.md i toc.yml
- Toc.yml datoteke predstavljaju strukturu pojedinog dokumenta
- Sadržaj stranice piše se koristeći Markdown
  - ekstenzija md
  - <https://en.wikipedia.org/wiki/Markdown>
- Uključiti web-aplikaciju u dokumentaciju
  - Vidi sljedeći slajd
- Izmijeniti index.md i toc.yml po uzoru na  WebServices \ WebApi \ Docs \ index.md i toc.yml

WebServices > WebApi > Docs

Name

 api  
 apidoc  
 articles  
 images  
 src  
 .gitignore  
 docfx.json  
 index.md  
 toc.yml

# Uključivanje web-aplikacije u izradu dokumentacije

WebServices > WebApi >

- Primjer  WebServices\WebApi\Docs\docfx.json

```
"metadata": [  
  {  
    "src": [  
      {  
        "files": [  
        "src/**/*.csproj"  
        ],  
      }  
    ],  
    "dest": "api"  
  }  
], ...
```

```
"metadata": [  
  {  
    "src": [  
      {  
        "files": ["*.csproj"],  
        "src": "../WebApi",  
      }  
    ],  
    "dest": "api"  
  }  
],
```

Name

- .vs
- ConsoleClientApps
- Docs
- WebApi
- WebApi.sln

# Generiranje dokumentacije

- U mapi Docs pokrenuti `docfx`
  - Generira podmapu `_site` s html stranicama generiranim na osnovi markdown datoteka
- Za pregled dokumentacije pokrenuti `docfx serve _site`
  - Privremeni web server s dokumentacijom na adresi <http://localhost:8080>
- ...ili otvoriti `index.html` u pregledniku
  - Ako `docfx.json` za template ima `statictoc` umjesto `default`

```
"template": [ "statictoc"],
```

