NitroNet for Sitecore.

ZÜRICH, 28. NOVEMBER 2016

Nitro Net

Fabian Geiger.

```
@using Glass.Mapper
@using Glass.Mapper.Sc
@using Glass.Mapper.Sc.Web.Mvc
@using Sitecore.Common
@using System.Text.RegularExpressions;
@model RenderingViewModel<LeadTextModel>

@{ bool isPageEditing = Html.Glass().IsInEditingMode; }
@{ bool isTemplateNotNull = @Model.Data.Template != null; }
@{ Regex emptyTextInRichtext = new Regex("^(|(<br>)|(<br\\/>)|(<br \\/>)|(<br>)|(<p><\\/p>)|(<p> <\\/p>)|( ))$"); }
@{ Regex paragraphRegex = new Regex("^(<p>).*(</p>)"); }
@{ Match paragraphMatchLeadText = paragraphRegex.Match(isTemplateNotNull ? @Model.Data.Template.Lead_TextField : string.Empty);}
@{ Match paragraphMatchMoreInformation = paragraphRegex.Match(isTemplateNotNull ? @Model.Data.Template.More_InformationField : string.Empty);}
@{ Match emptyTextMatch = emptyTextInRichtext.Match(isTemplateNotNull ? @Model.Data.Template.More_InformationField : string.Empty);}

@if (Model.Data != null && (Model.Data.Template.Lead_ImageField.HasValue() || Model.Data.Template.Lead_HeadlineField.HasValue() || Model.Data.Template.Lead_TextField.HasValue() || isPageEditing))
{
    var reg = Model.Data.IsProductPage ? "\u00AE" : string.Empty;

    <div class="m-leadtext-product m-leadtext-product--expandable js-track-page-component" data-t-name="LeadTextProduct" data-t-decorator="expandable" data-track-component-id="@Model.RenderingId" data-t
        <div class="m-leadtext-product__inner">
            @if (Model.Data.Template.Lead_ImageField.HasValue() || isPageEditing)
            {
                <div class="m-leadtext-product__placeholder col-sm-4 col-md-3 col-xs-12">
                    <div data-t-name="Image" class="m-image">
                        <div class="m-image__inner">
                            <figure class="m-image__picture">
                                @Html.Glass().Editable(Model.Data.Template, x => x.Lead_ImageField)
                            </figure>
                        </div>
                    </div>
                </div>
            }

            @if (Model.Data.Template.Lead_HeadlineField.HasValue() || Model.Data.Template.Lead_TextField.HasValue() || isPageEditing)
            {
                <div class="m-leadtext-product__contentarea @(Model.Data.Template.Lead_ImageField.HasValue() ? " col-sm-8 col-md-9" : "") col-xs-12">
                    <h1 class="m-leadtext-product__headline" itemprop="headline"><span>@Html.Glass().Editable(Model.Data.Template, x => x.Lead_HeadlineField)</span><sup class="m-leadtext-product__regist

                    @if (Model.Data.IsProductPage && Model.Data.ProductType != null)
                    {
                        using (Html.BeginEditFrame(Model.Data.ProductType.Id.ToID(), Model.Data.ProductTypeEditFrame))
                        {
                            <h3 class="m-leadtext-product__subheadline">
                                @if (string.IsNullOrEmpty(Model.Data.SubHeadline) && isPageEditing)
                                {
                                    @Model.Data.NoProductTypeSet
                                }
                                else
```

Agenda.

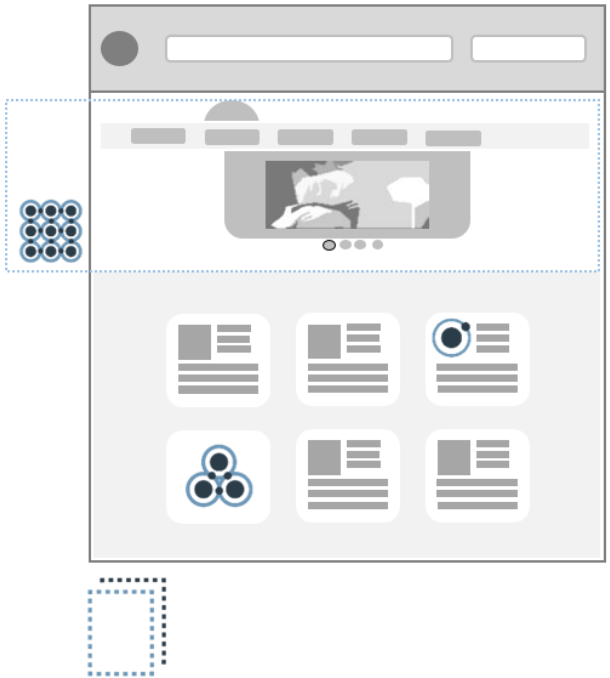
FIRST	**Nitro – What is that?**
SECOND	**NitroNet for Sitecore.**
THIRD	**Demo.**
FOURTH	**Outlook. Future.**
FIFTH	**Questions. Discussion.**

FIRST

Nitro – What is that?


Node.js Application
Yeoman Generator
Handlebars.js
Terrific.js
Atomic Design
Open Source

Nitro. Atomic Design.



Atom

Kleinste wiederverwendbare Einheit.

z.B. ein Bild

Molekül

Element, welches mit einem oder mehreren Atomen interagiert

z.B. ein Teaser

Organismus

Element, welches mit einem oder mehreren Moleküle interagiert

z.B. ein Seiten-Header

Template

Seitenvorlage, welche eine spezifische Dokumentenstruktur definiert

z.B. eine Content Seite

Nitro. Handlebars.

- Logicless template engine
- Extension of Mustache
- Custom helper expressions

```
<ul>
    {{#each links}}
        {{#if navigationTitle}}
            <li>
                <a href="{{href}}">{{navigationTitle}}</a>
            </li>
        {{/if}}
    {{/each}}
</ul>
```

```
{
    "links": [
        {
            "href": "/",
            "navigationTitle": "External Link"
        },
        {
            "href": "/",
            "navigationTitle": "Another External Link"
        },
        {
            "href": "/",
            "navigationTitle": "Third External Link"
        }
    ]
}
```

Nitro. Handlebars. Custom helpers.

- **Component**

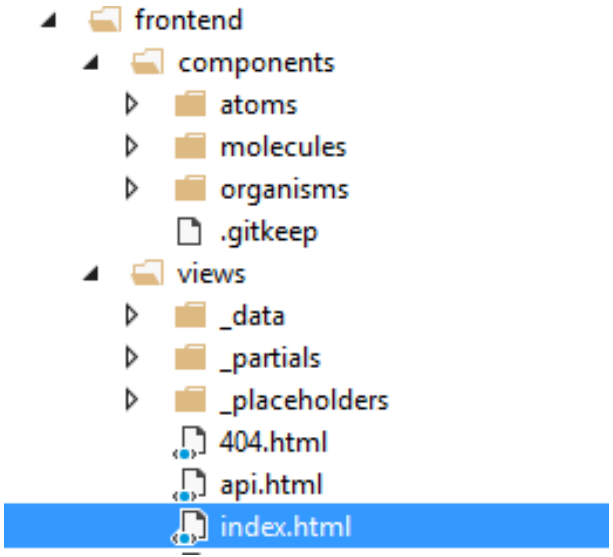
```
{{component name="Headline"}}
```

- **Placeholder**

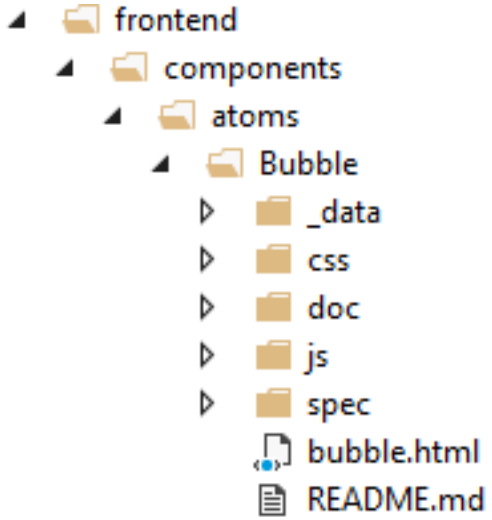
```
{{placeholder name="ContentArea"}}
```

Nitro. Folder structure.

Frontend Folder: Components and Views



Atom, Molekule, Organism: Folders



SECOND

NitroNet for Sitecore.

NitroNet. Overview.

- View engine for ASP.NET MVC
- Handlebars parser
- Placeholders in Sitecore
- Support for IoC frameworks
- Open source

NitroNet. Model.

textimage.html

```
<div class="m-text-image">
    {{#if image}}
        <figure class="m-text-image__figure">
            {{component name='Image'}}
        </figure>
    {{/if}}
    <div class="a-richtext m-text-image__text">
        {{{html}}}
    </div>
    {{component name='Button' data='ButtonData'}}
</div>
```

image.html

```
<picture class="a-image">
    <img class="a-image__img"
         src="{{df}}"
         alt="{{ alt }}"
</picture>
```

C# Klassen

```
public class TextImageModel
{
    public ImageModel Image { get; set; }

    public string Html { get; set; }

    public ButtonModel ButtonData { get; set; }
}

public class ImageModel
{
    public string Df { get; set; }

    public string Alt { get; set; }
}
```

NitroNet. Controller.

```
public class TextImageController : Controller
{
    public ActionResult Index()
    {
        var model = new TextImageModel();

        // Fill model.

        return View("textimage", model);
    }
}
```

Renders textimage.html and all its sub components

NitroNet. View.

```
<!DOCTYPE html>
<html>
<head>
    {{component name="Head"}}
    <link rel="stylesheet" href="/assets/css/bootstrap/bootstrap.min.css" />
    <link rel="stylesheet" href="/assets/css/custom.css" />
</head>
<body>
    <header class="l-header">
        <img src="/assets/img/nitronet_logo.png" />
        {{component name="Headline"}}
    </header>
    <div class="l-container">
        {{placeholder name="ContentArea"}}
    </div>
</body>
</html>
```

Static Sitecore renderings

Sitecore Placeholder
with Key «ContentArea»

NitroNet. Why?

- **Reuse of Handlebars templates in ASP.NET. No copy & paste**
- **Backend developers don’t need to worry about HTML anymore (mostly)**
- **Frontend code is written in a multi-platform language**
- **Identical data schemas in backend and frontend**
- **Real logicless views**

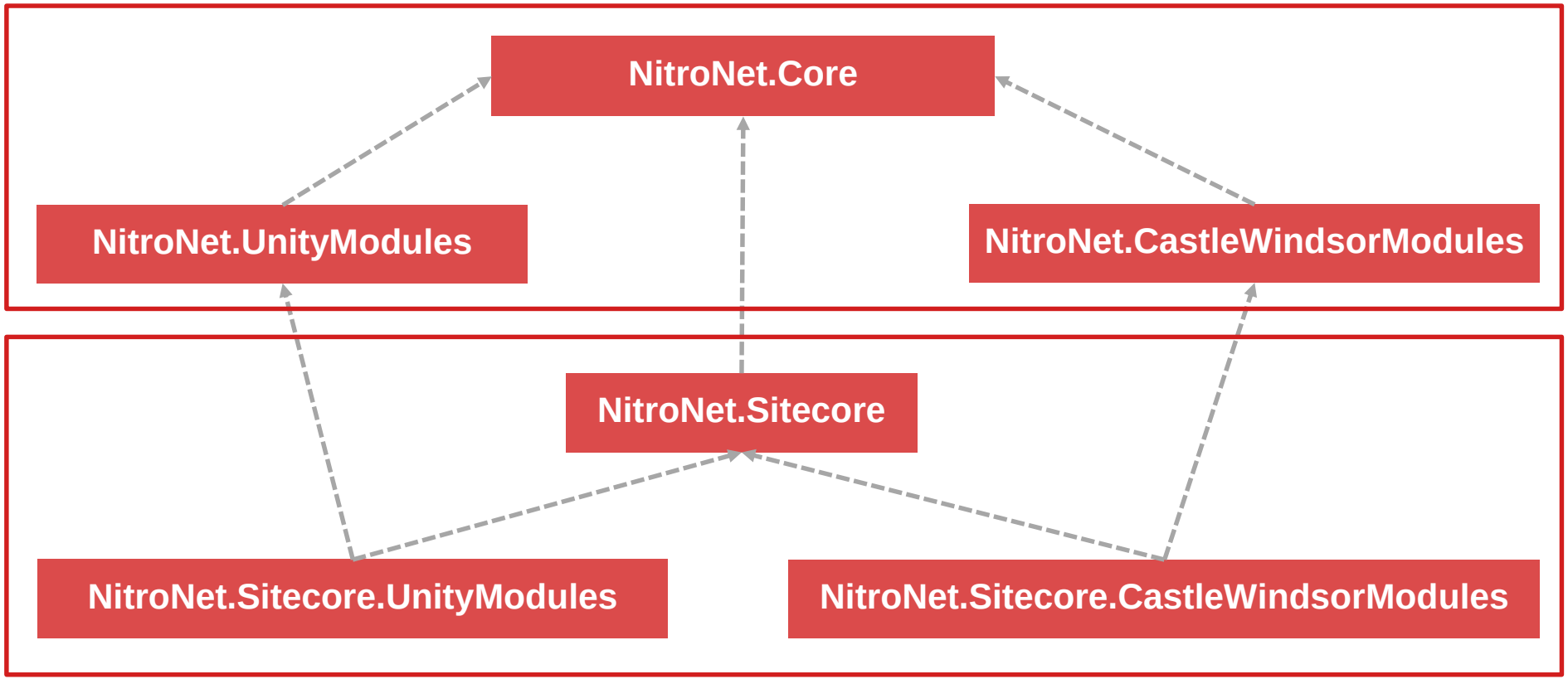
Handlebars.

```
<div class="m-leadtext-product{{#if modifier}} m-leadtext-product--{{modifier}}{{/if}}" data-t-name="LeadText
    <div class="m-leadtext-product__inner">
        {{#if image}}
            <div class="m-leadtext-product__placeholder col-md-3 col-xs-12">
                {{component name='Image' template='image-leadtext' data='image-australia'}}
            </div>
        {{/if}}

        <div class="m-leadtext-product__contentarea col-md-9 col-xs-12">
            <h1 class="m-leadtext-product__headline"><span>{{headline}}</span>{{#if registrationMark}}<sup cl
            {{#if subheadline}}<h3 class="m-leadtext-product__subheadline">{{subheadline}}</h3>{{/if}}

            <p class="leadtext">{{copy}}</p>

            {{#if copyMore}}
                <div class="m-leadtext-product__copy-container">
                    <p class="leadtext">{{copyMore}}</p>
                </div>
                <a href="#" class="m-leadtext-product__cta-button l-shadow" data-label-less="{{ctabutton.labe
            {{/if}}
        </div>
    </div>
</div>
```

Razor.

```
@using Glass.Mapper
@using Glass.Mapper.Sc
@using Glass.Mapper.Sc.Web.Mvc
@using Sitecore.Common
@using System.Text.RegularExpressions;
@model BASE.FE.Components.Model.General.RenderingViewModel<BASE.FE.Components.Model.Content.LeadTextModel>

@{ bool isPageEditing = Html.Glass().IsInEditingMode; }
@{ bool isTemplateNotNull = @Model.Data.Template != null; }
@{ Regex emptyTextInRichtext = new Regex("^(|(<br>)|(<br\\/>)|(<br \\/>)|(<br>)|(<p><\\/p>)|(<p> <\\/p
@{ Regex paragraphRegex = new Regex("^(<p>).*(</p>)"); }
@{ Match paragraphMatchLeadText = paragraphRegex.Match(isTemplateNotNull ? @Model.Data.Template.Lead_TextFi
@{ Match paragraphMatchMoreInformation = paragraphRegex.Match(isTemplateNotNull ? @Model.Data.Template.More
@{ Match emptyTextMatch = emptyTextInRichtext.Match(isTemplateNotNull ? @Model.Data.Template.More_Informati

@if (Model.Data != null && (Model.Data.Template.Lead_ImageField.HasValue() || Model.Data.Template.Lead_Head
{
    var reg = Model.Data.IsProductPage ? "\u00AE" : string.Empty;

    <div class="m-leadtext-product m-leadtext-product--expandable js-track-page-component" data-t-name="Lea
        <div class="m-leadtext-product__inner">
            @if (Model.Data.Template.Lead_ImageField.HasValue() || isPageEditing)
            {
                <div class="m-leadtext-product__placeholder col-sm-3 col-xs-12">
                    @{ Html.RenderPartial("/Layout/Sublayouts/FE_Components/Content/Image/ImageLeadText.csh
                </div>
            }

            @if (Model.Data.Template.Lead_HeadlineField.HasValue() || Model.Data.Template.Lead_TextField.Ha
            {
                <div class="m-leadtext-product__contentarea @(Model.Data.Template.Lead_ImageField.HasValue(
                    <h1 class="m-leadtext-product__headline" itemprop="headline"><span>@Html.Glass().Editab

                    @if (Model.Data.IsProductPage && Model.Data.ProductType != null)
                    {
                        using (Html.BeginEditFrame(Model.Data.ProductType.Id.ToID(), Model.Data.ProductType
                        {
                            <h3 class="m-leadtext-product__subheadline">
                                @if (string.IsNullOrEmpty(Model.Data.SubHeadline) && isPageEditing)
                                {
                                    @Model.Data.NoProductTypeSet
                                }
                                else
                                {
                                    @Model.Data.SubHeadline
                                }
                            </h3>
                        }
                    }
```

NitroNet. Nuget.



THIRD

Demo.

FOURTH

Outlook. Future.

Outlook. Future.

- Release 1.1.0 (hopefully) in December
  - Bugfixes
  - NitroNet for standard ASP.NET MVC applications
- Further bugfixing/improvements
  - e.g. Support of native partials in handlebars: {{< mypartial}}
- Keep up with new Sitecore versions
  - e.g. IoC with Microsoft.Extensions.DependencyInjection
- Keep up with new Nitro versions
  - e.g. Model generator based on schemas

GitHub. NuGet. Twitter. YouTube.

- **http://www.nitronet.io/**
- **https://github.com/namics/NitroNet**
- **https://www.nuget.org/packages?q=nitronet**
- **https://twitter.com/hashtag/NitroNet**
- **https://www.youtube.com/watch?v=_nnH6cszfQI**
  (by Oliver Eisenhut)

- **Check it out and help improving NitroNet!**

FIFTH

Questions. Discussion.

Thank you.