

```
Error executing template "Designs/Swift/_parsed/Swift_Page.parsed.cshtml"
System.NullReferenceException: Objektreferencen er ikke indstillet til en forekomst af et objekt.
   ved DynamicWeb.Ecommerce.DynamicWebLiveIntegration.Connectors.WebServiceConnector.IsWebServiceConnectionAvailable(EndpointInfo endpoint)
   ved DynamicWeb.Ecommerce.DynamicWebLiveIntegration.TemplatesHelper.IsWebServiceConnectionAvailable()
   ved CompiledRazorTemplates.Dynamic.RazorEngine_ba66b157037046f093ac1e5a0e9413bc.Execute() i C:\WorkSpace\Goecker\Files\Templates\Designs\Swift\_parsed\Swift_Page.parsed.cshtml:linje 335
   ved RazorEngine.Templating.TemplateBase.RazorEngine.Templating.ITemplate.Run(ExecutionContext context, TextWriter writer)
   ved RazorEngine.Templating.RazorEngineService.RunCompile(ITemplateKey key, TextWriter writer, Type modelType, Object model, DynamicViewBag viewBag)
   ved RazorEngine.Templating.RazorEngineServiceExtensions.<>c__DisplayClass16_0.<RunCompile>_0(TextWriter writer)
   ved RazorEngine.Templating.RazorEngineServiceExtensions.WithWriter(Action`1 withWriter)
   ved DynamicWeb.Rendering.TemplateRenderingProvider.Render(Template template)
   ved DynamicWeb.Rendering.TemplateRenderingService.Render(Template template)
   ved DynamicWeb.Rendering.Template.RenderRazorTemplate()
```

```
1 @inherits DynamicWeb.Rendering.ViewModelTemplate<DynamicWeb.Frontend.PageView>()
2 @using System
3 @using DynamicWeb
4 @using DynamicWeb.Environment
5 @using DynamicWeb.Frontend
6 @using System.Web
7
8 @
9 {
10     string swiftVersion = ReadFile("~/Files/Templates/Designs/Swift/swift_version.txt");
11     bool renderAsResponsive = Model.Area.Item.GetSetting("DynamicRendering", "responsive").Equals("responsive", StringComparison.OrdinalIgnoreCase);
12     bool renderMobile = Pageview.Device == DynamicWeb.Frontend.Devices.DeviceType.Mobile || Pageview.Device == DynamicWeb.Frontend.Devices.DeviceType.Tablet;
13     string responsiveHeaderClassDesktop = string.Empty;
14     string responsiveHeaderClassMobile = string.Empty;
15     if (renderAsResponsive)
16     {
17         responsiveHeaderClassDesktop = "d-none d-xl-block";
18         responsiveHeaderClassMobile = "d-block d-xl-none";
19     }
20     var brandingPageId = Model.Area.Item.GetLink("BrandingPage") != null ? Model.Area.Item.GetLink("BrandingPage").PageId : 0;
21     var themePageId = Model.Area.Item.GetLink("ThemesPage") != null ? Model.Area.Item.GetLink("ThemesPage").PageId : 0;
22     string customHeaderInclude = Model.Area.Item.GetFile("CustomHeaderInclude") != null ? Model.Area.Item.GetFile("CustomHeaderInclude").Name : string.Empty;
23
24     var brandingPage = DynamicWeb.Services.Pages?.GetPage(brandingPageId) ?? null;
25     var themesParagraphLastChanged = Services.Paragraphs.GetParagraphsByPageId(themePageId).OrderByDescending(p => p.Audit.LastModifiedAt).FirstOrDefault();
26
27     var cssLastModified = brandingPage.Audit.LastModifiedAt > themesParagraphLastChanged.Audit.LastModifiedAt ? brandingPage.Audit.LastModifiedAt : themesParagraphLastChanged.Audit.LastModifiedAt;
28     var cssThemeAndBrandingStyleFileInfo = new System.IO.FileInfo(DynamicWeb.Core.SystemInformation.MapPath("~/Files/Templates/Designs/Swift/_parsed/Swift_styles/{Model.Area.ID}.min.css"));
29
30     // Schema.org details for PDP
31     string productId = !string.IsNullOrEmpty(DynamicWeb.Context.Current.Request.QueryString.Get("ProductID")) ? DynamicWeb.Context.Current.Request.QueryString.Get("ProductID") : "";
32     bool isProductDetailsPage = !string.IsNullOrEmpty(productId);
33     bool isArticlePage = Model.ItemType == "Swift_Article";
34     string schemaOrgType = string.Empty;
35
36     if (isProductDetailsPage)
37     {
38         schemaOrgType = "itemscope=\"" itype="https://schema.org/Product";
39     }
40
41     if (isArticlePage)
42     {
43         schemaOrgType = "itemscope=\"" itype="https://schema.org/Article";
44     }
45
46     if (!cssThemeAndBrandingStyleFileInfo.Exists || cssThemeAndBrandingStyleFileInfo.LastWriteTime < brandingPage.Audit.LastModifiedAt)
47     {
48         //Branding page has been saved or the file is missing. Rewrite the file to disc.
49         if (brandingPageId > 0)
50         {
51             var brandingPageview = DynamicWeb.Frontend.PageView.GetPageviewByPageId(brandingPageId);
52             brandingPageview.Redirect = false;
53             brandingPageview.Output();
54         }
55     }
56
57     if (!cssThemeAndBrandingStyleFileInfo.Exists || cssThemeAndBrandingStyleFileInfo.LastWriteTime < themesParagraphLastChanged.Audit.LastModifiedAt)
58     {
59         //Branding page has been saved or the file is missing. Rewrite the file to disc.
60         if (themePageId > 0)
61         {
62             var themePageview = DynamicWeb.Frontend.PageView.GetPageviewByPageId(themePageId);
63             themePageview.Redirect = false;
64             themePageview.Output();
65         }
66     }
67
68     var cssStyleFileInfo = new System.IO.FileInfo(DynamicWeb.Core.SystemInformation.MapPath("~/Files/Templates/Designs/Swift/Assets/css/styles.css"));
69     var jsFileInfo = new System.IO.FileInfo(DynamicWeb.Core.SystemInformation.MapPath("~/Files/Templates/Designs/Swift/Assets/js/scripts.js"));
70
71     string masterTheme = !string.IsNullOrEmpty(Model.Area.Item.GetRawValueString("Theme")) ? " theme " + Model.Area.Item.GetRawValueString("Theme").Replace(" ", "").Trim().ToLower() : "";
72
73     string favicon = Model.Area.Item.GetFile("Favicon") != null ? Model.Area.Item.GetFile("Favicon").Path : "~/Files/Templates/Designs/Swift/Assets/Images/favicon.png";
74
75     string headerCssClass = "sticky-top";
76     bool movePageBehind = false;
77     string scrollInitScript = "Swift.Scroll.init(); AOS.init({offset: 240, duration: 600, delay: 100}); disable: window.matchMedia('(prefers-reduced-motion: reduce)');";
78     if (Pageview.Page.PropertyType != null)
79     {
80         headerCssClass = Pageview.Page.PropertyType["MoveThisPageBehindTheHeader"] != null ? Pageview.Page.PropertyType["MoveThisPageBehindTheHeader"].ToString() : "sticky-top";
81         movePageBehind = headerCssClass == "fixed-top" && !Pageview.IsVisualEditorMode ? true : false;
82     }
83     headerCssClass = headerCssClass == "" ? "sticky-top" : headerCssClass;
84     headerCssClass = Pageview.IsVisualEditorMode ? "" : headerCssClass;
85
86     string googleAnalyticsTrackingID = Model.Area.Item.GetSetting("GoogleAnalyticsTrackingID");
87     bool allowCookies = CookieManager.GetCookieOptInLevel() == 0 || !CookieManager.CookieOptInLevelExists ? false : true;
88
89     DynamicWeb.Context.Current.Response.AddHeader("link", $"{Files/Templates/Designs/Swift/Assets/css/styles.css}{cssStyleFileInfo.LastWriteTime.Ticks}; rel=preload; as=style;");
90     DynamicWeb.Context.Current.Response.AddHeader("link", $"{Files/Templates/Designs/Swift/_parsed/Swift_styles_{Model.Area.ID}.min.css}{cssLastModified.Ticks}; rel=preload; as=style;");
91     DynamicWeb.Context.Current.Response.AddHeader("link", $"{Files/Templates/Designs/Swift/Assets/js/aos.js}{jsFileInfo.LastWriteTime.Ticks}; rel=preload; as=script;");
92     DynamicWeb.Context.Current.Response.AddHeader("link", $"{Files/Templates/Designs/Swift/Assets/js/scripts.js}{jsFileInfo.LastWriteTime.Ticks}; rel=preload; as=script;");
93     //DynamicWeb.Context.Current.Response.Flush(); //This sends the headers where we are now in the rendering making the TTFB faster
94
95     SetMetaTags();
96
97     List<DynamicWeb.Content.Page> languages = new List<DynamicWeb.Content.Page>();
98
99     if (Pageview.Area.IsMaster)
100     {
101         languages.Add(Pageview.Page);
102         if (Pageview.Page.Languages != null)
103         {
104             foreach (var language in Pageview.Page.Languages)
105             {
106                 languages.Add(language);
107             }
108         }
109     }
110     else
111     {
112         languages.Add(Pageview.Page.MasterPage);
113         if (Pageview.Page.MasterPage != null)
114         {
115             if (Pageview.Page.MasterPage.Languages != null)
116             {
117                 foreach (var language in Pageview.Page.MasterPage.Languages)
118                 {
119                     languages.Add(language);
120                 }
121             }
122         }
123     }
124
125     string siteLanguage = Pageview.Area.CultureInfo.Name;
126     Uri url = DynamicWeb.Context.Current.Request.Url;
127     string hostName = url.Host; // domain.com/da-dk or domain.com/en-us
128
129     var econCountries = DynamicWeb.Ecommerce.Services.Countries.GetCountries();
130     var econCurrencies = DynamicWeb.Ecommerce.Services.Currencies.GetAllCurrencies();
131
132     <doctype html>
133     <html lang="@Pageview.Area.CultureInfo.TwoLetterISOLanguageName">
134     <head>
135         <!-- @swiftVersion -->
136         @* Required meta tags *@
137         <meta charset="utf-8">
138         <meta name="viewport" content="height=device-height, width=device-width, initial-scale=1.0">
139         <link rel="preconnect" href="https://fonts.gstatic.com/" />
140
141         <link rel="shortcut icon" href="@favicon">
142         <link rel="apple-touch-icon" href="@Files/Templates/Designs/Swift/Assets/Images/logo_transparent.png">
143
144         @Model.MetaTags
145
146         @
147         @* Languages meta data *@
148         foreach (var language in languages)
149         {
150             if (language?.Area != null)
151             {
152                 if (language != null && language.Published && language.Active && language.Area.Active && language.Area.Published && language.Area.ID != DynamicWeb.Frontend.PageView.Current().AreaID)
153                 {
154                 }
155             }
156         }
157     }
158 }
```

```

154         if (!string.IsNullOrEmpty(language.Area.DomainLock))
155         {
156             hostName = language.Area.DomainLock; //dk.domain.com or dk-domain.dk
157         }
158         string friendlyUrl = Dynamicweb.Frontend.SearchEngineFriendlyURLs.GetFriendlyUrl(language.ID);
159         string href = $"{url.Scheme}://{hostName}{friendlyUrl}";
160
161         <link rel="alternate" hreflang="@language.Area.CultureInfo.Name.ToLower()" href="@href" />
162     }
163 }
164 }
165 }
166
167 <title>@Model.Title</title>
168 @* Bootstrap + Swift stylesheet *@
169 <link href="/Files/Templates/Designs/Swift/Assets/css/styles.css?@cssStyleFileInfo.LastWriteTime.Ticks" rel="stylesheet" media="all" type="text/css">
170 @* Branding and Themes min stylesheet *@
171 <link href="/Files/Templates/Designs/Swift/_parsed/Swift_css/Swift_styles_@(@Model.Area.ID).min.css?@cssLastModified.Ticks" rel="stylesheet" media="all" type="text/css" data-last-modified-content="@cssLastModified">
172 <script src="/Files/Templates/Designs/Swift/Assets/js/aos.js?@jsFileInfo.LastWriteTime.Ticks" defer></script>
173 <script src="/Files/Templates/Designs/Swift/Assets/js/scripts.js?@jsFileInfo.LastWriteTime.Ticks" defer></script>
174
175
176 @* Global site tag (gtag.js) - Google Analytics *@
177 @if (!string.IsNullOrEmptySpace(googleAnalyticsTrackingID) && allowCookies)
178 {
179     <script src="https://www.googletagmanager.com/gtag/js?id=@googleAnalyticsTrackingID" async></script>
180     <script>
181         window.dataLayer = window.dataLayer || [];
182         function gtag() { window.dataLayer.push(arguments); }
183         gtag('js', new Date());
184         gtag('config', '@googleAnalyticsTrackingID');
185     </script>
186 }
187
188 @if (!string.IsNullOrEmptySpace(customHeaderInclude))
189 {
190     @RenderPartial($"Components/Custom/{customHeaderInclude}")
191 }
192 </head>
193 <body class="brand @(masterTheme)" id="page@(@Model.ID)" onload="@scrollInitScript">
194 @if (renderAsResponsive || !renderMobile)
195 {
196     <header class="page-header @headerCssClass top-0@(responsiveHeaderClassDesktop)" id="page-header-desktop">
197         @if (@@Model.Area.Item.GetLink("HeaderDesktop") != null)
198         {
199             @RenderGrid(@@Model.Area.Item.GetLink("HeaderDesktop").PageId)
200         }
201     </header>
202 }
203
204 @if ((renderAsResponsive || renderMobile))
205 {
206     <header class="page-header @headerCssClass top-0@(responsiveHeaderClassMobile)" id="page-header-mobile">
207         @if (@@Model.Area.Item.GetLink("HeaderMobile") != null)
208         {
209             @RenderGrid(@@Model.Area.Item.GetLink("HeaderMobile").PageId)
210         }
211     </header>
212 }
213
214 <main id="content" @(schemaOrgType)>
215 @inherits Dynamicweb.Rendering.ViewModelTemplate<Dynamicweb.Frontend.PageViewModel>
216 @using System
217 @using Dynamicweb.Ecommerce.ProductCatalog
218
219 @if (
220     bool isArticlePage = Model.ItemType == "Swift_Article";
221     string schemaOrgProp = string.Empty;
222     if (isArticlePage)
223     {
224         schemaOrgProp = "Itemprop=\"articleBody\"";
225     }
226
227     string theme = "";
228     string gridContent = "";
229
230     if (Model.PropertyItem != null)
231     {
232         theme = !string.IsNullOrEmpty(Model.PropertyItem.GetRawValueString("Theme")) ? " theme " + Model.PropertyItem.GetRawValueString("Theme").Replace(" ", "").Trim().ToLower() : "";
233     }
234
235     if (Model.Item != null)
236     {
237         gridContent = Model.Grid("Grid", "Grid", "default:true;sort:1", "Page");
238     }
239
240     bool doNotRenderPage = false;
241
242     //Check if we are on the product detail page, and if there is data to render
243     ProductViewModel product = new ProductViewModel();
244     if (Dynamicweb.Context.Current.Items.Contains("ProductDetails"))
245     {
246         product = (ProductViewModel)Dynamicweb.Context.Current.Items["ProductDetails"];
247         if (string.IsNullOrEmpty(product.Id)) {
248             doNotRenderPage = true;
249         }
250     }
251
252     //Render the page
253     if (!doNotRenderPage) {
254         string itemIdentifier = Model?.Item?.SystemName != null ? "item_" + Model.Item.SystemName.ToLower() : "item_swift_page";
255         if (theme != "")
256         {
257             <div class="@theme item_@itemIdentifier" @schemaOrgProp>
258                 @gridContent
259             </div>
260         }
261         else
262         {
263             <div class="item_@itemIdentifier" @schemaOrgProp>
264                 @gridContent
265             </div>
266         }
267     }
268     else {
269         <div class="container">
270             <div class="alert alert-info" role="alert">@Translate("Sorry. There is nothing to view here")</div>
271         </div>
272     }
273
274     if (!Model.IsCurrentUserAllowed)
275     {
276         int signInPage = GetPageIdByNavigationTag("SignInPage");
277         int dashboardPage = GetPageIdByNavigationTag("/MyAccountDashboardPage");
278
279         if (!Pageview.IsVisualEditMode)
280         {
281             {
282                 if (signInPage != 0)
283                 {
284                     if (signInPage != Model.ID) {
285                         Dynamicweb.Context.Current.Response.Redirect("/Default.aspx?ID=" + signInPage);
286                     } else {
287                         if (dashboardPage != 0) {
288                             Dynamicweb.Context.Current.Response.Redirect("/Default.aspx?ID=" + dashboardPage);
289                         } else {
290                             Dynamicweb.Context.Current.Response.Redirect("/");
291                         }
292                     }
293                 }
294                 else
295                 {
296                     <div class="alert alert-dark m-0" role="alert">
297                         <span>@Translate("You do not have access to this page")</span>
298                     </div>
299                 }
300             }
301             else
302             {
303                 <div class="alert alert-dark m-0" role="alert">
304                     <span>@Translate("To work on this page, you must be signed in, in the frontend")</span>
305                 </div>
306             }
307         }
308     }
309
310 </main>
311
312 @if (renderAsResponsive || !renderMobile)
313 {
314     <footer class="page-footer d-none d-lg-block" id="page-footer-desktop">
315         @if (@@Model.Area.Item.GetLink("FooterDesktop") != null)
316         {
317             @RenderGrid(@@Model.Area.Item.GetLink("FooterDesktop").PageId)
318         }
319     </footer>
320 }
321
322 @if (renderAsResponsive || renderMobile)

```

```

322 {
323     <footer class="page-footer d-block d-lg-none" id="page-footer-mobile">
324         @if (@Model.Area.Item.GetLink("FooterMobile") != null)
325         {
326             @RenderGrid(@Model.Area.Item.GetLink("FooterMobile").PageId)
327         }
328     </footer>
329 }
330
331 @* Render any offcanvas menu here *@
332 @RenderSnippet("offcanvas")
333
334 @({
335     bool isErpConnectionDown = DynamicWeb.ECommerce.DynamicWebLiveIntegration.TemplatesHelper.IsWebServiceConnectionAvailable();
336     if (isErpConnectionDown && Model.Area.Item.GetBoolean("ShowErpDownMessage"))
337     {
338         @RenderNoErpConnectionMessage()
339     }
340 })
341
342 @* Language selector modal *@
343 @if (languages.Count > 1 || ecomCountries.Count > 1 || ecomCurrencies.Count() > 1)
344 {
345     <div class="modal fade" id="PreferencesModal" tabindex="-1" aria-hidden="true">
346         <div class="modal-dialog modal-dialog-centered modal-sm" id="PreferencesModalContent">
347             @* The content here comes from an external request *@
348         </div>
349     </div>
350 }
351
352 @* Favorite toast *@
353 <div aria-live="polite" aria-atomic="true">
354     <div class="position-fixed bottom-0 end-0 p-3" style="z-index: 11">
355         <div id="favoriteNotificationToast" class="toast" role="alert" aria-live="assertive" aria-atomic="true">
356             <div class="toast-header">
357                 <strong class="me-auto">@Translate("Favorite list updated")</strong>
358                 <button type="button" class="btn-close" data-bs-dismiss="toast" aria-label="Close"></button>
359             </div>
360             <div class="toast-body d-flex gap-3">
361                 <div id="favoriteNotificationToast_Image"></div>
362                 <div id="favoriteNotificationToast_Text"></div>
363             </div>
364         </div>
365     </div>
366
367 @* Modal for dynamic content *@
368 <div class="modal fade js-product" id="DynamicModal" tabindex="-1" aria-hidden="true">
369     <div class="modal-dialog modal-dialog-centered modal-md">
370         <div class="modal-content theme light" id="DynamicModalContent">
371             @* The content here comes from an external request *@
372         </div>
373     </div>
374 </div>
375
376 @* Offcanvas for dynamic content *@
377 <div class="offcanvas offcanvas-end theme light" tabindex="-1" id="DynamicOffcanvas" style="width: 30rem">
378     @* The content here comes from an external request *@
379 </div>
380 </body>
381 </html>
382 @functions {
383     void SetMetaTags()
384     {
385         //Verification Tokens
386         string siteVerificationGoogle = Model.Area.Item.GetString("Google_Site_Verification") != null ? Model.Area.Item.GetString("Google_Site_Verification") : "";
387         //string siteVerificationYandex = Model.Area.Item.GetString("Yandex_Verification") != null ? Model.Area.Item.GetString("Yandex_Verification") : "";
388         //string siteVerificationMS = Model.Area.Item.GetString("Msvalidate_01") != null ? Model.Area.Item.GetString("Msvalidate_01") : "";
389         //string siteVerificationAlexa = Model.Area.Item.GetString("AlexaVerifyID") != null ? Model.Area.Item.GetString("AlexaVerifyID") : "";
390         //string siteVerificationInterest = Model.Area.Item.GetString("P_domain_verify") != null ? Model.Area.Item.GetString("P_domain_verify") : "";
391         //string siteVerificationNorton = Model.Area.Item.GetString("Norton_safeweb_site_verification") != null ? Model.Area.Item.GetString("Norton_safeweb_site_verification") : "";
392
393         //Generic Site Values
394         string openGraphFacebookAppID = Model.Area.Item.GetString("Fb_app_id") != null ? Model.Area.Item.GetString("Fb_app_id") : "";
395         string openGraphType = Model.Area.Item.GetString("Open_Graph_Type") != null ? Model.Area.Item.GetString("Open_Graph_Type") : "";
396         string openGraphSiteName = Model.Area.Item.GetString("Open_Graph_Site_Name") != null ? Model.Area.Item.GetString("Open_Graph_Site_Name") : "";
397
398         string twitterCardSite = Model.Area.Item.GetString("Twitter_Site") != null ? Model.Area.Item.GetString("Twitter_Site") : "";
399
400         //Page specific values
401         string openGraphSiteTitle = Model.Area.Item.GetString("Open_Graph_Title") != null ? Model.Area.Item.GetString("Open_Graph_Title") : "";
402         FileViewModel openGraphImage = Model.Area.Item.GetFile("Open_Graph_Image");
403         string openGraphImageALT = Model.Area.Item.GetString("Open_Graph_Image_ALT") != null ? Model.Area.Item.GetString("Open_Graph_Image_ALT") : "";
404         string openGraphDescription = Model.Area.Item.GetString("Open_Graph_Description") != null ? Model.Area.Item.GetString("Open_Graph_Description") : "";
405
406         string twitterCardURL = Model.Area.Item.GetString("Twitter_URL") != null ? Model.Area.Item.GetString("Twitter_URL") : "";
407         string twitterCardTitle = Model.Area.Item.GetString("Twitter_Title") != null ? Model.Area.Item.GetString("Twitter_Title") : "";
408         string twitterCardDescription = Model.Area.Item.GetString("Twitter_Description") != null ? Model.Area.Item.GetString("Twitter_Description") : "";
409         FileViewModel twitterCardImage = Model.Area.Item.GetFile("Twitter_Image");
410         string twitterCardImageALT = Model.Area.Item.GetString("Twitter_Image_ALT") != null ? Model.Area.Item.GetString("Twitter_Image_ALT") : "";
411
412         if (Istring.IsNullOrEmpty(siteVerificationGoogle))
413         {
414             Pageview.Meta.AddTag("google-site-verification", siteVerificationGoogle);
415         }
416
417         if (Istring.IsNullOrEmpty(openGraphFacebookAppID))
418         {
419             Pageview.Meta.AddTag("fb:app_id", openGraphFacebookAppID);
420         }
421
422         if (Istring.IsNullOrEmpty(openGraphType))
423         {
424             Pageview.Meta.AddTag("og:type", openGraphType);
425         }
426
427         if (Istring.IsNullOrEmpty(openGraphSiteName))
428         {
429             Pageview.Meta.AddTag("og:site_name", openGraphSiteName);
430         }
431
432         if (Istring.IsNullOrEmpty(Model.Title))
433         {
434             Pageview.Meta.AddTag("og:title", Model.Title);
435         }
436         else
437         {
438             Pageview.Meta.AddTag("og:title", openGraphSiteTitle);
439         }
440     }
441
442     if (Istring.IsNullOrEmpty(DynamicWeb.Context.Current.Request.QueryString["ProductID"]))
443     {
444         if (Istring.IsNullOrEmpty(Model.Description))
445         {
446             Pageview.Meta.AddTag("og:description", Model.Description);
447         }
448         else
449         {
450             Pageview.Meta.AddTag("og:description", openGraphDescription);
451         }
452         if (openGraphImage != null)
453         {
454             Pageview.Meta.AddTag("og:image", openGraphImage.Path);
455         }
456
457         if (Istring.IsNullOrEmpty(openGraphImageALT))
458         {
459             Pageview.Meta.AddTag("og:image:alt", openGraphImageALT);
460         }
461         if (Istring.IsNullOrEmpty(twitterCardDescription))
462         {
463             Pageview.Meta.AddTag("twitter:description", twitterCardDescription);
464         }
465         if (twitterCardImage != null)
466         {
467             Pageview.Meta.AddTag("twitter:image", twitterCardImage.Path);
468         }
469         if (Istring.IsNullOrEmpty(twitterCardImageALT))
470         {
471             Pageview.Meta.AddTag("twitter:image:alt", twitterCardImageALT);
472         }
473         if (Istring.IsNullOrEmpty(twitterCardSite))
474         {
475             Pageview.Meta.AddTag("twitter:site", twitterCardSite);
476         }
477         if (Istring.IsNullOrEmpty(twitterCardURL))
478         {
479             Pageview.Meta.AddTag("twitter:url", twitterCardURL);
480         }
481         if (Istring.IsNullOrEmpty(twitterCardTitle))
482         {
483             Pageview.Meta.AddTag("twitter:title", twitterCardTitle);
484         }
485     }
486 }
487
488 }
489

```

```
490         Pageview.Meta.AddTag("twitter:title", twitterCardTitle);
491     }
492 }
493 }
494
495 @helper RenderNoErpConnectionMessage()
496 {
497     string erpDownMessageTheme = !string.IsNullOrWhiteSpace(Model.Area.Item.GetRawValueString("ErpDownMessageTheme")) ? " theme " + Model.Area.Item.GetRawValueString("ErpDownMessageTheme").Replace(" ", "").Trim().ToLower() : "theme light";
498
499     <div class="position-fixed bottom-0 end-0 p-3" style="z-index: 1040">
500         <div class="toast fade show border-0 @erpDownMessageTheme" role="alert" aria-live="assertive" aria-atomic="true">
501             <div class="toast-header">
502                 <strong class="me-auto">@Translate("Connection down")</strong>
503                 <button type="button" class="btn-close" data-bs-dismiss="toast" aria-label="Close"></button>
504             </div>
505             <div class="toast-body">
506                 @Translate("We are experiencing some connectivity issues. Not all features may be available to you.")
507             </div>
508         </div>
509     </div>
510 }
511 }
```