

Updated Exception enrichment functionality for added safety

👤 Mathias Kunto ⏰ October 13, 2025 🛡 EPiServer,

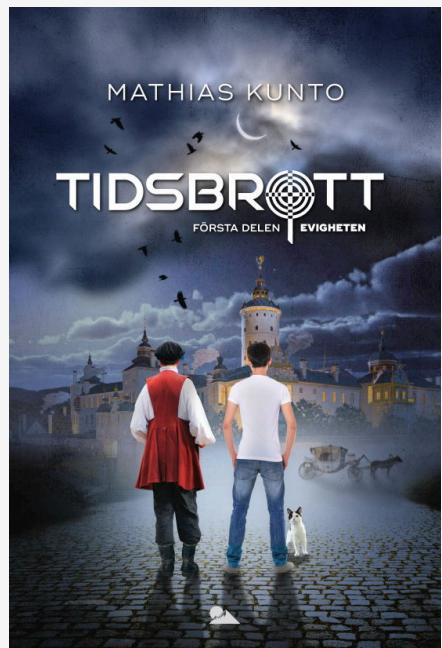
Optimizely 💬 No Comments

Here is an improved version of the Exception enrichment functionality described in the article [Enriching your exceptions with information from Episerver](#). Updates has been made in order to make it more robust in regards to usage of abstract classes that may in some cases throw NotImplementedException, causing the enrichment code to terminate.

```
1  public static class ExceptionExtensions
2  {
3      private static readonly ILogger _logger =
4          LogManager.GetLogger(typeof(ExceptionExtensions));
5
6      private static readonly
7          Injected<HttpContextBase> _context;
8
9      private const int MaxValueLength =
10         150;
11
12      private const string Enriched =
13         "OS_Extra";
```

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

Evigheten del 1: Tidsbrott



Första delen i serien [Evigheten](#), en historisk sci-fi för ungdomar. Provläs på [Smakprov.se](#) och köp ditt exemplar hos [Adlibris](#), [Bokus](#) eller [Akademibokhandeln](#). Du kan också låna den på ditt lokala bibliotek.

Tidsbrott finns som ljudbok hos bland andra [Storytel](#), [Nextory](#) och [Pustakatt](#).

[Cookies Settings](#)

[Accept All Cookies](#)

```

15     /// </summary>
16     /// <remarks>
17     /// Be really careful here so your
18     /// code does not result in exceptions.
19     /// That would be very bad for the
20     /// website.
21     /// </remarks>
22     public static void Enrich(this
23     Exception e)
24     {
25         if (e == null ||
26             e.Data.Contains(Enriched))
27         {
28             return;
29         }
30
31         try
32         {
33             EnrichUnsafe(e);
34         }
35         catch (Exception ex)
36         {
37             logger.Log(Level.Error,
38             DoubleFaultMessage, ex);
39         }
40
41         private static void
42         EnrichUnsafe(Exception e)
43         {
44             e.Data[Enriched] = string.Empty;
45
46             HttpContextBase context =
47             _context.Service;
48
49             if (context != null)
50             {
51                 UnsafeEnrichFrom(context.Request,
52                 e);
53
54                 UnsafeEnrichFrom(context.User, e);
55
56                 UnsafeEnrichFrom(context.Session,
57                 e);
58
59                 UnsafeEnrichFrom(context.Request?.Browser,
60                 e);
61             }
62             else
63             {
64                 e.Data["HttpContext"] = "null
65                 (possibly Optimizely scheduled job?)";
66             }
67         }
68     }
69 
```

vägen genom. Det finns inte ett kapitel, inte en sida då jag inte vill veta vad som händer härnäst." – Linnea Lindblom, [Recension hos NSD](#), 2020-01-08

Recent posts

[Updated Exception enrichment functionality for added safety](#)

[Allow only certain number of items in ContentAreas in Optimizely](#)

[Hiding and showing edit mode properties on same content type in Optimizely CMS depending on context](#)

[Changing Optimizely property return type yields MissingMethodException on getter](#)

[Object reference not set to an instance of an object in EPiServer.Shell.Web.Mvc.Http](#)

Recent comments

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

```

62     {
63         return !string.IsNullOrEmpty(value)
64         && value.Length > maxLength
65             ? value.Substring(0, maxLength) +
66                 "...";
67         : value;
68     }
69
70     private static void
71     UnsafeEnrichFrom(HttpRequestBase
72     request, Exception e)
73     {
74         if (request == null)
75         {
76             return;
77         }
78
79         e.Data["Request_Url"] =
80         GetValueOrDefault(() =>
81             request.Url?.ToString());
82         e.Data["Request_HttpMethod"] =
83         GetValueOrDefault(() =>
84             request.HttpMethod);
85         e.Data["Request_UserHostAddress"] =
86         GetValueOrDefault(() =>
87             request.UserHostAddress);
88         e.Data["Request_UserAgent"] =
89         GetValueOrDefault(() =>
90             request.UserAgent);
91         e.Data["Request.UrlReferrer"] =
92         GetValueOrDefault(() =>
93             request.UrlReferrer?.ToString());
94         e.Data["Request_ContentType"] =
95         GetValueOrDefault(() =>
96             request.ContentType);
97         e.Data["Request_ContentLength"] =
98         GetValueOrDefault(() =>
99             request.ContentLength);
100        e.Data["Request_IsSecureConnection"] =
101        GetValueOrDefault(() =>
102            request.IsSecureConnection);
103
104        // Headers
105        var headers = new StringBuilder();
106        foreach (string key in
107        request.Headers)
108        {
109            var value = request.Headers[key];
110            value = TruncateValue(value,
111                MaxValueLength);
112            headers.AppendLine($"    {key}:
113                {value}");
114        }
115    }
116}

```

wałdis iljuczonok
(@tech_fellow) on React
and EPiServer: Aggregated
output with
EPiServer.ContentDeliveryApi

Josef Ottosson on React
and EPiServer – Custom
rendering of EPiServer
block in XhtmlString field
without partial block
controller

Mathias Kunto on React
and EPiServer – Custom
rendering of EPiServer
block in XhtmlString field
without partial block
controller

Josef Ottosson on React
and EPiServer – Custom
rendering of EPiServer
block in XhtmlString field
without partial block
controller

```

98
99      // Form (not logging actual values
100     due to possible GDPR violation).
101     var form = new StringBuilder();
102     foreach (string key in request.Form)
103     {
104         var value = request.Form[key];
105         value = $"Length={value.Length}";
106         form.AppendLine($"{key}:{value}");
107     }
108
109     if (form.Length > 0)
110     {
111         form.Insert(0,
112         Environment.NewLine);
113         e.Data["Request_Form"] =
114         form.ToString();
115     }
116
117     // Cookies
118     var cookies = new StringBuilder();
119     foreach (string key in
120     request.Cookies)
121     {
122         var value =
123         request.Cookies[key] ?.Value;
124         value = TruncateValue(value,
125         MaxValueLength);
126         cookies.AppendLine($"{key}:{value}");
127     }
128
129     if (cookies.Length > 0)
130     {
131         cookies.Insert(0,
132         Environment.NewLine);
133         e.Data["Request_Cookies"] =
134         cookies.ToString();
135     }
136
137     // Files
138     var files = new StringBuilder();
139     foreach (string key in
140     request.Files)
141     {
142         var file = request.Files[key];
143         if (file != null)
144         {
145             files.AppendLine($"{key}: {file.FileName} ({file.ContentLength}
146             bytes)");

```

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

```

    files.ToString();
}
}

private static void
UnsafeEnrichFrom(IPrincipal principal,
Exception e)
{
    if (principal?.Identity == null)
    {
        return;
    }

    if
(principal.Identity.IsAuthenticated)
{
    e.Data["AuthenticatedUser"] = $""
{principal.Identity.Name}
(AuthenticationType:
{principal.Identity.AuthenticationType})";
}
else
{
    e.Data["AuthenticatedUser"] =
$"User not authenticated.";
}
}

private static void
UnsafeEnrichFrom(HttpSessionStateBase
session, Exception e)
{
    if (session == null)
    {
        return;
    }

    e.Data["Session_SessionID"] =
GetValueOrDefault(() =>
session.SessionID);

    var sessionVars = new
StringBuilder();
    foreach (string key in session)
    {
        var value =
session[key]?.ToString();
        value = TruncateValue(value,
MaxValueLength);
        sessionVars.AppendLine($" {key}:
{value}");
    }
}

```

By clicking “Accept All Cookies”, you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

```

186     }
187
188
189     private static void
190     UnsafeEnrichFrom(HttpBrowserCapabilitiesBase
191     browser, Exception e)
192     {
193         if (browser == null)
194         {
195             return;
196         }
197
198         e.Data["Browser_Type"] =
199         GetValueOrDefault(() => browser.Type);
200         e.Data["Browser_Browser"] =
201         GetValueOrDefault(() =>
202         browser.Browser);
203         e.Data["Browser_Version"] =
204         GetValueOrDefault(() =>
205         browser.Version);
206         e.Data["Browser_Platform"] =
207         GetValueOrDefault(() =>
208         browser.Platform);
209         e.Data["Browser_MobileDeviceModel"] =
210         GetValueOrDefault(() =>
211         browser.MobileDeviceModel);
212     }
213
214     private static void
215     UnsafeEnrichWithOptimizelyData(Exception
216     e)
217     {
218         e.Data["StartPage_ID"] =
219         ContentReference.StartPage?.ID.ToString(CultureInfo.InvariantCulture)
220         ?? string.Empty;
221
222         PageData page = null;
223         try
224         {
225             page =
226             ServiceLocator.Current.GetInstance<IPageRouteHelper>
227             ()?.Page;
228         }
229         catch (Exception ex)
230         {
231             e.Data["CurrentPage"] =
232             $"Exception: {ex.Message}.";
233         }
234         if (page != null)
235         {
236             e.Data["CurrentPage"] = $"
237             {page.PageName} (ID:{page.PageLink}) .";
238         }
239     }
240 
```

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

```

225     }
226     catch (Exception ex)
227     {
228         e.Data["CurrentContent"] =
229             $"Exception: {ex.Message}." ;
230         if (content != null)
231         {
232             e.Data["CurrentContent"] = $""
233                 {content.Name} (ID:
234                 {content.ContentLink}) .";
235         }
236     }
237     /// <summary>
238     /// Used to safely get values from
239     /// properties.
240     /// </summary>
241     /// <typeparam name="T">Type of value
242     /// to get.</typeparam>
243     /// <param name="getterFunc">Func that
244     /// calls the getter. Example: () =>
245     /// request.HttpMethod</param>
246     /// <returns>The value to get, or the
247     /// type default if not able to get the
248     /// value.</returns>
249     /// <remarks>
250     /// When using abstract classes such
251     /// as HttpContextBase there are instances
252     /// when property getters
253     /// are not implemented (such as when
254     /// mocking requests for testing or running
255     /// scheduled jobs).
256     /// We need a safe way of getting
257     /// values that do not cause
258     /// NotImplementedException to be thrown.
259     /// </remarks>
260     private static T GetValueOrDefault<T>
261     (Func<T> getterFunc)
262     {
263         try
264         {
265             return getterFunc();
266         }
267         catch
268         {
269             return default;
270         }
271     }
272 }
```

By clicking “Accept All Cookies”, you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.



Neoleaf AB in Stockholm.

Mathias Kunto's blog Copyright © 2025.

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.