

LinksPlatform's Platform.RegularExpressions.Transformer.HasuraSQLSimplifier Class Library

1.1 ./csharp/Platform.RegularExpressions.Transformer.HasuraSQLSimplifier/HasuraSQLSimplifierTransformer.cs

```

1 using System.Collections.Generic;
2 using System.Linq;
3 using System.Text.RegularExpressions;
4
5 #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
6
7 namespace Platform.RegularExpressions.Transformer.HasuraSQLSimplifier
8 {
9     /// <summary>
10     /// <para>
11     /// Represents the hasura sql simplifier transformer.
12     /// </para>
13     /// <para></para>
14     /// </summary>
15     /// <seealso cref="TextTransformer"/>
16     public class HasuraSQLSimplifierTransformer : TextTransformer
17     {
18         /// <summary>
19         /// <para>
20         /// The to list.
21         /// </para>
22         /// <para></para>
23         /// </summary>
24         public static readonly IList<ISubstitutionRule> DefaultRules = new List<SubstitutionRule>
25         {
26             // HTML clean up
27             (new Regex(@"<span class=""[^"""]*">(<[^>]*)</span>"), "$1", 0),
28             // ('describe')
29             // 'describe'
30             (new Regex(@"\[\\s\\n]*('['']+')\[\\s\\n]*\\)"), "$1", int.MaxValue),
31             // AND ('true' AND 'true')
32             //
33             (new Regex(@"\[\\s\\n]*AND\[\\s\\n]*\[\\s\\n]*'true' \[\\s\\n]*AND\[\\s\\n]*'true' \[\\s\\n]*\\)"), "",
34             → 0),
35             // AND ('true')
36             //
37             (new Regex(@"\[\\s\\n]*AND\[\\s\\n]*'true'"), "", 0),
38             // ::
39             // ::
40             (new Regex(@"\[\\s]*::\[\\s]*"), "::", 0),
41             // ('describe'::text)
42             // 'describe'::text
43             (new Regex(@"\[\\s\\n]*('['']+')::text \[\\s\\n]*\\)"), "$1", 0),
44             // ("_0__be_0_nodes"."target_id")
45             // "_0__be_0_nodes"."target_id"
46             (new Regex(@"\[\\s\\n]*(""[^"""]*"") \[\\s\\n]*\\. \[\\s\\n]*(""[^"""]*"") \[\\s\\n]*\\)"), "$1.$2",
47             → 0),
48             // ("public"."nodes"."_id")
49             // "public"."nodes"."_id"
50             (new Regex(@"\[\\s\\n]*(""[^"""]*"") \[\\s\\n]*\\. \[\\s\\n]*(""[^"""]*"") \[\\s\\n]*\\. \[\\s\\n]*(""[^"""]*"") \[\\s\\n]*\\)"), "$1.$2.$3",
51             → 0),
52             // LIMIT\n\t\t\t\t\t1
53             // LIMIT 1
54             (new Regex(@"(LIMIT) \[\\s\\n]*([\\d+])"), "$1 $2", 0),
55             // ("_0__be_0_nodes"."type" = 'describe'::text)
56             // "_0__be_0_nodes"."type" = 'describe'::text
57             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
58             → int.MaxValue),
59             // (EXISTS (...))
60             // EXISTS (...)
61             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2", int.MaxValue),
62             // ((EXISTS (...)))
63             // EXISTS (...)
64             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
65             → int.MaxValue),
66             // ((EXISTS (...)))
67             // EXISTS (...)
68             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
69             → int.MaxValue),
70             // ((EXISTS (...)))
71             // EXISTS (...)
72             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
73             → int.MaxValue),
74             // ((EXISTS (...)))
75             // EXISTS (...)
76             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
77             → int.MaxValue),
78             // ((EXISTS (...)))
79             // EXISTS (...)
80             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
81             → int.MaxValue),
82             // ((EXISTS (...)))
83             // EXISTS (...)
84             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
85             → int.MaxValue),
86             // ((EXISTS (...)))
87             // EXISTS (...)
88             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
89             → int.MaxValue),
90             // ((EXISTS (...)))
91             // EXISTS (...)
92             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
93             → int.MaxValue),
94             // ((EXISTS (...)))
95             // EXISTS (...)
96             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
97             → int.MaxValue),
98             // ((EXISTS (...)))
99             // EXISTS (...)
100             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
101             → int.MaxValue),
102             // ((EXISTS (...)))
103             // EXISTS (...)
104             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
105             → int.MaxValue),
106             // ((EXISTS (...)))
107             // EXISTS (...)
108             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
109             → int.MaxValue),
110             // ((EXISTS (...)))
111             // EXISTS (...)
112             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
113             → int.MaxValue),
114             // ((EXISTS (...)))
115             // EXISTS (...)
116             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
117             → int.MaxValue),
118             // ((EXISTS (...)))
119             // EXISTS (...)
120             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
121             → int.MaxValue),
122             // ((EXISTS (...)))
123             // EXISTS (...)
124             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
125             → int.MaxValue),
126             // ((EXISTS (...)))
127             // EXISTS (...)
128             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
129             → int.MaxValue),
130             // ((EXISTS (...)))
131             // EXISTS (...)
132             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
133             → int.MaxValue),
134             // ((EXISTS (...)))
135             // EXISTS (...)
136             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
137             → int.MaxValue),
138             // ((EXISTS (...)))
139             // EXISTS (...)
140             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
141             → int.MaxValue),
142             // ((EXISTS (...)))
143             // EXISTS (...)
144             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
145             → int.MaxValue),
146             // ((EXISTS (...)))
147             // EXISTS (...)
148             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
149             → int.MaxValue),
150             // ((EXISTS (...)))
151             // EXISTS (...)
152             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
153             → int.MaxValue),
154             // ((EXISTS (...)))
155             // EXISTS (...)
156             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
157             → int.MaxValue),
158             // ((EXISTS (...)))
159             // EXISTS (...)
160             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
161             → int.MaxValue),
162             // ((EXISTS (...)))
163             // EXISTS (...)
164             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
165             → int.MaxValue),
166             // ((EXISTS (...)))
167             // EXISTS (...)
168             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
169             → int.MaxValue),
170             // ((EXISTS (...)))
171             // EXISTS (...)
172             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
173             → int.MaxValue),
174             // ((EXISTS (...)))
175             // EXISTS (...)
176             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
177             → int.MaxValue),
178             // ((EXISTS (...)))
179             // EXISTS (...)
180             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
181             → int.MaxValue),
182             // ((EXISTS (...)))
183             // EXISTS (...)
184             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
185             → int.MaxValue),
186             // ((EXISTS (...)))
187             // EXISTS (...)
188             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
189             → int.MaxValue),
190             // ((EXISTS (...)))
191             // EXISTS (...)
192             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
193             → int.MaxValue),
194             // ((EXISTS (...)))
195             // EXISTS (...)
196             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
197             → int.MaxValue),
198             // ((EXISTS (...)))
199             // EXISTS (...)
200             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
201             → int.MaxValue),
202             // ((EXISTS (...)))
203             // EXISTS (...)
204             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
205             → int.MaxValue),
206             // ((EXISTS (...)))
207             // EXISTS (...)
208             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
209             → int.MaxValue),
210             // ((EXISTS (...)))
211             // EXISTS (...)
212             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
213             → int.MaxValue),
214             // ((EXISTS (...)))
215             // EXISTS (...)
216             (new Regex(@"(\\W) \[\\s\\n]*((?!SELECT) [^\\s\\n() [^()]*?) \[\\s\\n]*\\)"), "$1$2",
217             → int.MaxValue),
218             // ((EXISTS (...)))
219             // EXISTS
```

```

69         public HasuraSQLSimplifierTransformer()
70             : base(DefaultRules)
71         {
72         }
73     }
74 }

```

1.2 ./csharp/Platform.RegularExpressions.Transformer.HasuraSQLSimplifier.Tests/HasuraSQLSimplifierTransformerTests.cs

[illegible]

```

72         )
73         AND ('true')
74     )
75 )
76 AND (
77     ('true')
78     AND ('true')
79 )
80 )
81 )
82 )
83 )
84 )
85 ) AS ""_1_root.base""
86 LIMIT
87     1
88 ) AS ""_3_root"";
89
90     var expected = @"SELECT
91 coalesce(json_agg("""root""), '[]') AS ""root""
92 FROM
93 (
94     SELECT
95         row_to_json(
96             (
97                 SELECT
98                     ""_2_e""
99                 FROM
100                 (
101                     SELECT
102                         ""_1_root.base"".""id"" AS ""id""
103                     ) AS ""_2_e""
104                 )
105             ) AS ""root""
106         FROM
107         (
108             SELECT
109                 *
110             FROM
111                 ""public"".""nodes""
112             WHERE
113                 ""public"".""nodes"".""type"" = 'auth_token'::text
114                 AND EXISTS (
115                     SELECT
116                         1
117                     FROM
118                         ""public"".""nodes"" AS ""_0__be_0_nodes""
119                     WHERE
120                         ""_0__be_0_nodes"".""_source_id"" = ""public"".""nodes"".""_id""
121                         AND ""_0__be_0_nodes"".""type"" = 'describe'::text
122                         AND ""_0__be_0_nodes"".""target_id"" = 'X-Hasura-User-Id'::text
123                     )
124                 ) AS ""_1_root.base""
125             LIMIT 1
126         ) AS ""_3_root"";
127     var transformer = new HasuraSQLSimplifierTransformer();
128     var actual = transformer.Transform(original);
129     Assert.Equal(expected, actual);
130 }
131 }
132 }

```

Index

- ./csharp/Platform.RegularExpressions.Transformer.HasuraSQLSimplifier.Tests/HasuraSQLSimplifierTransformerTests.cs, 2
- ./csharp/Platform.RegularExpressions.Transformer.HasuraSQLSimplifier/HasuraSQLSimplifierTransformer.cs, 1