

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

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

```

1 using Xunit;
2
3 namespace Platform.RegularExpressions.Transformer.HasuraSQLSimplifier.Tests
4 {
5     public class HasuraSQLSimplifierTransformerTests
6     {
7         [Fact]
8         public void EmptyLineTest()
9         {
10             // This test can help to test basic problems with regular expressions like incorrect
11             ↪ syntax

```

```

11     var transformer = new HasuraSQLSimplifierTransformer();
12     var actualResult = transformer.Transform("");
13     Assert.Equal("", actualResult);
14 }
15
16 [Fact]
17 public void BasicRequestTest()
18 {
19     var original = @"SELECT
20 coalesce(json_agg("root"), '[]') AS "root"
21 FROM
22 (
23     SELECT
24         row_to_json(
25             (
26                 SELECT
27                     "_2_e"
28                 FROM
29                 (
30                     SELECT
31                         "_1_root.base"."id" AS "id"
32                     ) AS "_2_e"
33             )
34         ) AS "root"
35 FROM
36     (
37         SELECT
38             *
39         FROM
40             "public"."nodes"
41         WHERE
42             (
43                 (
44                     ("public"."nodes"."type") = (('auth_token') :: text)
45                 )
46                 AND (
47                     EXISTS (
48                         SELECT
49                             1
50                         FROM
51                             "public"."nodes" AS "_0__be_0_nodes"
52                         WHERE
53                             (
54                                 (
55                                     (
56                                         ("_0__be_0_nodes"."_source_id") = ("public"."nodes"."_id")
57                                     )
58                                     AND ('true')
59                                 )
60                                 AND (
61                                     (
62                                         (
63                                             ("_0__be_0_nodes"."type") = (('describe') :: text))
64                                             AND ('true')
65                                         )
66                                         AND (
67                                             (
68                                                 (
69                                                     ("_0__be_0_nodes"."target_id") = (('X-Hasura-User-Id') :: text)
70                                                 )
71                                                 AND ('true')
72                                             )
73                                             AND ('true')
74                                         )
75                                         )
76                                         AND (
77                                             ('true')
78                                             AND ('true')
79                                         )
80                                     )
81                                 )
82                             )
83                         )
84                     ) AS "_1_root.base"
85                 LIMIT
86                 1
87             ) AS "_3_root";
88
89     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, 1
./csharp/Platform.RegularExpressions.Transformer.HasuraSQLSimplifier/HasuraSQLSimplifierTransformer.cs, 1