Escali Extensions

on XML Prague

Thursday, 09 February 2017

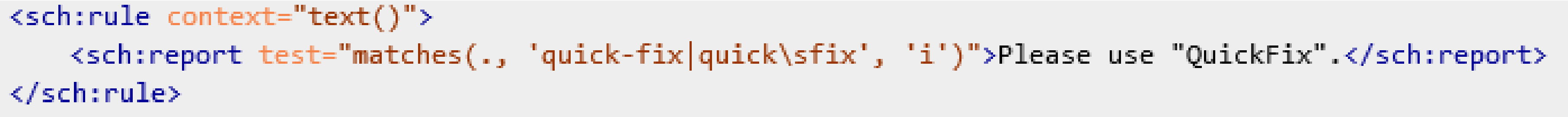
Nico Kutscherauer

Overview

- Escali
  - Schematron implementation based on XSLT
  - Similar to Skeleton
  - Only for XSLT 2.0
- Including a QuickFix implementation
  - Published on escali.schematron-quickfix.com (JS)
  - www.schematron-quickfix.com/downloads.html (XProc)
- Provides Schematron extensions
  - 6 elements
  - 6 attributes
  - Own namespace: `http://www.escali.schematron-quickfix.com/`
  - Read more on: www.schematron-quickfix.com//escali/escali-ext_en.html

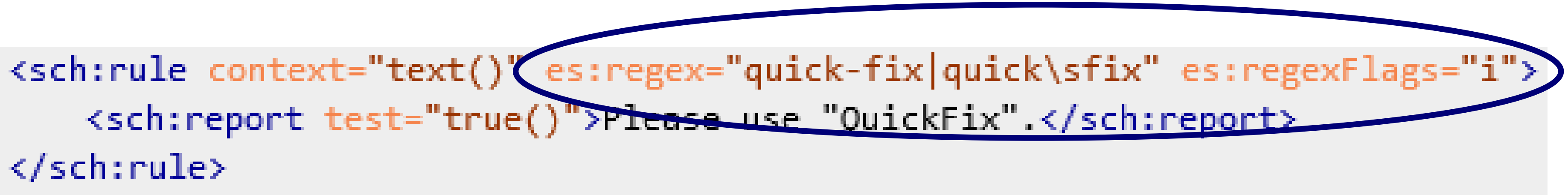
Regex

- Schematron is based on XPath/XSLT
  - Full power and Restrictions of XPath
  - **Schematron error location needs to be a node**
- Possible check:



```
<sch:rule context="text()">
    <sch:report test="matches(., 'quick-fix|quick\sfix', 'i')">Please use "QuickFix".</sch:report>
</sch:rule>
```

  - What if there are multiple errors in one text node?
  - How to find such text phrases in large paragraphs?
- Let‘s match with RegEx:



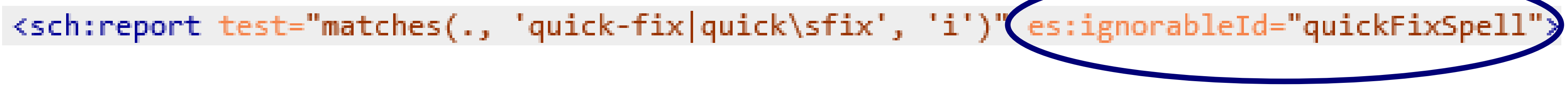
```
<sch:rule context="text()" es:regex="quick-fix|quick\sfix" es:regexFlags="i">
    <sch:report test="true()">Please use "QuickFix".</sch:report>
</sch:rule>
```

Regex

- Context inside of the rule
  - A xs:string – the text phrase
  - Build-in variable $es:current to get the node context
- Not for element as context
  - Mixed content is not feasible
- QuickFix implementation is tricky
  - Not ready yet

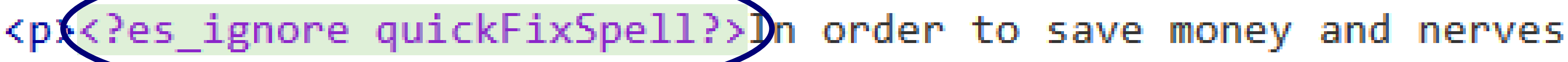
Ignore Logic

- Add a ignorableId attribute to your test:



```
<sch:report test="matches(., 'quick-fix|quick\sfix', 'i')" es:ignorableId="quickFixSpell">
```

- Each node with a specific PI will be ignored:



```
<p><?es_ignore quickFixSpell?>In order to save money and nerves
```

- The Escali creates automatically QuickFixes to ignore

Other Escali Extensions

- Namespaces:
  - Default namespace for XPath
- Import/include
  - Import similar to xsl:import
- Pattern matching
  - Use case: If multiple rules match to the same node
- Phases
  - Reuse phases, make patterns always active