

































-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML
- XML














XML static code analysis

Unique rules to find Bugs and Code Smells in your XML code

- All rules 36
-  Vulnerability 6
-  Bug 5
-  Security Hotspot 9
-  Code Smell 16

Tags ▼

Search by name... 🔍

Track uses of "TODO" tags		Code Smell
EJB interceptor exclusions should be declared as annotations		Code Smell
Track uses of disallowed dependencies		Code Smell
Newlines should follow each element		Code Smell
XML parser failure		Code Smell
Track breaches of an XPath rule		Code Smell
Lines should not be too long		Code Smell
pom elements should be in the recommended order		Code Smell
Artifact ids should follow a naming convention		Code Smell
Group ids should follow a naming convention		Code Smell
"action" mappings should not have too many "forward" entries		Code Smell
Source code should be indented consistently		Code Smell
Tabulation characters should not be used		Code Smell

XML parser failure

Analyze your code

-  Code Smell
-  Major 
-  suspicious

When the XML parser fails, it is possible to record the failure as a violation on the file. This way, not only it is possible to track the number of files that do not parse but also to easily find out why they do not parse.

Available In:

sonarlint 

sonarcloud 

sonarqube 