

































-  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 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

Track uses of "TODO" tags

Analyze your code

-  Code Smell
-  Info 
-  cwe

TODO tags are commonly used to mark places where some more code is required, but which the developer wants to implement later.

Sometimes the developer will not have the time or will simply forget to get back to that tag.

This rule is meant to track those tags and to ensure that they do not go unnoticed.

Noncompliant Code Example

```
<!-- TODO Drop this dependency -->
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-lang3</artifactId>
  <version>3.8.1</version>
</dependency>
```

See

- [MITRE, CWE-546](#) - Suspicious Comment

Available In:

sonarlint  | sonarcloud  | sonarqube 