sonar

demo1
1.0

css:Sonar way
java:Sonar way
js:Sonar way
web:Sonar way
xml:Sonar way
2023-06-27

目录

1. demo1

报告提供了项目指标的概要，显示了与项目质量相关的最重要的指标。如果需要获取更详细的信息，请登陆网站进一步查询。

报告的项目为demo1，生成时间为2023-06-27，使用的质量配置为 css:Sonar way java:Sonar way js:Sonar way web:Sonar way xml:Sonar way，共计 676条规则。

1.1. 概述

编码问题

Bug
222

可靠性修复工作
14h43min

漏洞
0

安全修复工作
0min

坏味道
1010

技术债务
41d18h37min

1232
问题

开启问题	1232
重开问题	0
确认问题	0
误判问题	0
不修复的问题	0
已解决的问题	0
已删除的问题	0
阻断	11
严重	87
主要	881
次要	240
提示	13

静态分析

项目规模

53196
代码行数

行数	67301
方法	2509
类	45
文件	342
目录	N/A
重复行(%)	57.1

复杂度

8922
复杂度

| 文件 | 47.2 |

注释(%)

10.1
注释(%)

| 注释行数 | 5952 |

动态分析

0.2
覆盖率(%)

4
单元测试数

代码覆盖率(%)	0.2
分支覆盖率(%)	0.0
单元测试失败数	0
单元测试错误数	0
单元测试忽略数	0
单元测试成功率(%)	100.0

1.2. 问题分析

违反最多的规则TOP10	
Variables should not be shadowed	195
Assignments should not be made from within sub-expressions	187
Ternary operators should not be nested	76
Method names should comply with a naming convention	56
Variables and functions should not be redeclared	54

Unnecessary imports should be removed	49
Non-empty statements should change control flow or have at least one side-effect	48
"<table>" tags should have a description	41
Attributes deprecated in HTML5 should not be used	35
Tables should have headers	35

违规最多的文件TOP5	
jquery-3.5.1.js	216
jquery-3.5.1.js	216
bootstrap.js	60
bootstrap.js	60
prettify.js	49

复杂度最高的文件TOP5	
jquery-3.5.1.js	2608
jquery-3.5.1.js	2608
bootstrap.js	629
bootstrap.js	629
bootstrap-datetimepicker.js	589

重复行最多的文件TOP5	
jquery-3.5.1.js	10859
jquery-3.5.1.js	10859
bootstrap.js	2574
bootstrap.js	2574
bootstrap-datetimepicker.js	1940

1.3. 问题详情

规则	Variables should not be shadowed

规则描述	Overriding or shadowing a variable declared in an outer scope can strongly impact the readability, and therefore the maintainability, of a piece of code. Further, it could lead maintainers to introduce bugs because they think they're using one variable but are really using another. See 　　CERT, DCL01-C. - Do not reuse variable names in subscopes 　　CERT, DCL51-J. - Do not shadow or obscure identifiers in subscopes

文件名称	违规行
index.js	125, 36, 528, 57, 26, 151
bootstrap.js	998, 1024, 1403, 1403, 1416
jquery-3.5.1.js	397, 414, 530, 532, 545, 570, 571, 574, 575, 579, 580, 628, 657, 664, 746, 755, 756, 957, 958, 1078, 1081, 1112, 1223, 1233, 1233, 1275, 1540, 1688, 1720, 1858, 1926, 2067, 2070, 2098, 2101, 2186, 2261, 2269, 2277, 2289, 2384, 2394, 2463, 2475, 2486, 2512, 2609, 2673, 2793, 2836, 2935, 2949, 2961, 2992, 3071, 3330, 3680, 3748, 4004, 4173, 4512, 4513, 4809, 5612, 6035, 6252, 6325, 7122, 7122, 7554, 7555, 7596, 7600, 8089, 8274, 8311, 8487, 8991, 9397, 10592, 10592, 10651, 10651
index.js	125, 36, 528, 57, 26, 151
bootstrap.js	998, 1024, 1403, 1403, 1416

jquery-3.5.1.js	397, 414, 530, 532, 545, 570, 571, 574, 575, 579, 580, 628, 657, 664, 746, 755, 756, 957, 958, 1078, 1081, 1112, 1223, 1233, 1233, 1275, 1540, 1688, 1720, 1858, 1926, 2067, 2070, 2098, 2101, 2186, 2261, 2269, 2277, 2289, 2384, 2394, 2463, 2475, 2486, 2512, 2609, 2673, 2793, 2836, 2935, 2949, 2961, 2992, 3071, 3330, 3680, 3748, 4004, 4173, 4512, 4513, 4809, 5612, 6035, 6252, 6325, 7122, 7122, 7554, 7555, 7596, 7600, 8089, 8274, 8311, 8487, 8991, 9397, 10592, 10592, 10651, 10651
prettify.js	415, 415, 474, 491, 500, 1085, 1466

规则	Assignments should not be made from within sub-expressions

规则描述	Assignments within sub-expressions are hard to spot and therefore make the code less readable. Ideally, sub-expressions should not have side-effects. Noncompliant Code Example

```
if (val = value() && check()) { // Noncompliant
  // ...
}
```

 Compliant Solution

```
val = value();
if (val && check()) {
  // ...
}
```

 Exceptions
The rule does not raise issues for the following patterns:

```
    assignments at declaration-level:  let a = b = 0;
    chained assignments:  a = b = c = 0;
    relational assignments:  (a = 0) != b
    sequential assignments:  a = 0, b = 1, c = 2
    assignments in lambda body:  () => a = 0
    conditional assignment idiom:  a || (a = 0)
    assignments in (do-)while conditions:  while (a = 0);
```

 See

   MITRE, CWE-481 - Assigning instead of Comparing
   CERT, EXP45-C. - Do not perform assignments in selection statements
   CERT, EXP51-J. - Do not perform assignments in conditional expressions

文件名称	违规行
index.js	67, 90
bootstrap-datetimepicker.js	62, 96, 1478
bootstrap.js	121, 148, 258, 384, 424, 485, 708, 724, 809, 838, 895, 1240, 1496, 1880, 1943, 2053, 2067, 2229, 2383, 2485, 2543

jquery-3.5.1.js	726, 783, 787, 821, 860, 863, 914, 1085, 1086, 1315, 1420, 1420, 1667, 1872, 2073, 2105, 2106, 2150, 2157, 2335, 2341, 2354, 2438, 2442, 2449, 2491, 2554, 2555, 2568, 2571, 2574, 2580, 2583, 2620, 2630, 2656, 2724, 2838, 2848, 2870, 2873, 2876, 2965, 2996, 3331, 3661, 3665, 4773, 5224, 5227, 5272, 5315, 5413, 5417, 6214, 6638, 6673, 7286, 7316, 8148, 8574, 8598, 8703, 9061, 9065, 9090, 10166
index.js	67, 90
bootstrap-datetimepicker.js	62, 96, 1478
bootstrap.js	121, 148, 258, 384, 424, 485, 708, 724, 809, 838, 895, 1240, 1496, 1880, 1943, 2053, 2067, 2229, 2383, 2485, 2543
jquery-3.5.1.js	726, 783, 787, 821, 860, 863, 914, 1085, 1086, 1315, 1420, 1420, 1667, 1872, 2073, 2105, 2106, 2150, 2157, 2335, 2341, 2354, 2438, 2442, 2449, 2491, 2554, 2555, 2568, 2571, 2574, 2580, 2583, 2620, 2630, 2656, 2724, 2838, 2848, 2870, 2873, 2876, 2965, 2996, 3331, 3661, 3665, 4773, 5224, 5227, 5272, 5315, 5413, 5417, 6214, 6638, 6673, 7286, 7316, 8148, 8574, 8598, 8703, 9061, 9065, 9090, 10166
prettify.js	455

规则	Ternary operators should not be nested

规则描述	Just because you can do something, doesn't mean you should, and that's the case with nested ternary operations. Nesting ternary operators results in the kind of code that may seem clear as day when you write it, but six months later will leave maintainers (or worse - future you) scratching their heads and cursing. Instead, err on the side of clarity, and use another line to express the nested operation as a separate statement. Noncompliant Code Example
	
	function getReadableStatus(job) {
	  return job.isRunning() ? "Running" : job.hasErrors() ? "Failed" : "Succeeded "; // Noncompliant
	}
	
	 Compliant Solution
	
	function getReadableStatus(job) {
	  if (job.isRunning()) {
	    return "Running";
	  }
	  return job.hasErrors() ? "Failed" : "Succeeded";
	}

文件名称	违规行
bootstrap-datetimepicker.js	96, 96, 98, 99, 115, 116, 985, 1744, 1788
bootstrap.js	1649, 1650, 1651, 1809, 1818, 1819
jquery-3.5.1.js	683, 1554, 1556, 1557, 1558, 1592, 1593, 1665, 1667, 1900, 1901, 1902, 1903, 1904, 1905, 2279, 2376, 2532, 2532, 2965, 9629
bootstrap-datetimepicker.js	96, 96, 98, 99, 115, 116, 985, 1744, 1788
bootstrap.js	1649, 1650, 1651, 1809, 1818, 1819
jquery-3.5.1.js	683, 1554, 1556, 1557, 1558, 1592, 1593, 1665, 1667, 1900, 1901, 1902, 1903, 1904, 1905, 2279, 2376, 2532, 2532, 2965, 9629
prettify.js	321, 1093, 1098, 1099

规则	Method names should comply with a naming convention

规则描述	Shared naming conventions allow teams to collaborate efficiently. This rule checks that all method names match a provided regular expression. Noncompliant Code Example With default provided regular expression ^[a-z][a-zA-Z0-9]*$: public int DoSomething(){...} Compliant Solution public int doSomething(){...} Exceptions Overriding methods are excluded. @Override public int Do_Something(){...}

文件名称	违规行
DemoApplicationTests.java	46, 60, 75, 92, 108, 124, 139, 156, 172, 188, 220, 236, 252, 268, 283, 300, 316, 333, 349, 366, 382, 398, 410, 427, 443, 459, 472, 489
DemoOptionTests.java	33, 50, 66, 83
DemoPaperAnswerTests.java	34, 50, 63, 80
DemoPaperTests.java	34, 50, 62, 79
DemoProjectTests.java	33, 49, 64, 81
DemoQuestionTests.java	33, 49, 64, 81
DemoQuestionnaireTests.java	49, 65, 97, 113
DemoUserTests.java	48, 62, 77, 94

规则	Variables and functions should not be redeclared

| 规则描述 | This rule checks that a declaration doesn't use a name that is already in use. Indeed, it is possible to use the same symbol multiple times as
either a variable or a function, but doing so is likely to confuse maintainers. Further it's possible that such reassignments are made in error, with
the developer not realizing that the value of the variable is overwritten by the new assignment.
 This rule also applies to function parameters.
 Noncompliant Code Example

var a = 'foo';
function a() {}   // Noncompliant
console.log(a);   // prints "foo"

function myFunc(arg) {
  var arg = "event"; // Noncompliant, argument value is lost
}

fun(); // prints "bar"

function fun() {
  console.log("foo");
}

fun(); // prints "bar"

function fun() { // Noncompliant
  console.log("bar");
}

fun(); // prints "bar"

 Compliant Solution

var a = 'foo';
function otherName() {}
console.log(a);

function myFunc(arg) {
  var newName = "event";
}

fun(); // prints "foo"

function fun() {
  print("foo");
}

fun(); // prints "foo"

function printBar() {
  print("bar");
}

printBar(); // prints "bar" |

| 文件名称 | 违规行 |

bootstrap-datetimepicker.js	60, 60, 834, 886, 1151, 1152, 1153, 1154, 1155, 1156, 1182, 1619, 1620, 1676, 1683, 1717, 1806, 60, 60, 834, 886, 1151, 1152, 1153, 1154, 1155, 1156, 1182, 1619, 1620, 1676, 1683, 1717, 1806
prettify.js	325, 450, 462, 463, 515, 520, 520, 521, 528, 537, 537, 544, 545, 565, 565, 566
sort.js	35, 37, 74, 80

规则	Unnecessary imports should be removed

规则描述	The imports part of a file should be handled by the Integrated Development Environment (IDE), not manually by the developer. Unused and useless imports should not occur if that is the case. Leaving them in reduces the code's readability, since their presence can be confusing.
	 Noncompliant Code Example
	
	package my.company;
	
	import java.lang.String; // Noncompliant; java.lang classes are always implicitly imported
	import my.company.SomeClass; // Noncompliant; same-package files are always implicitly imported
	import java.io.File; // Noncompliant; File is not used
	
	import my.company2.SomeType;
	import my.company2.SomeType; // Noncompliant; 'SomeType' is already imported
	
	class ExampleClass {
	
	  public String someString;
	  public SomeType something;
	
	}
	
	 Exceptions
	 Imports for types mentioned in comments, such as Javadocs, are ignored.

文件名称	违规行
PaperAnswerController.java	5
PaperController.java	5, 9
ProjectController.java	5, 6
QuestionnaireController.java	6, 16
ToPaperController.java	3, 4, 6

UserController.java	13, 15
ProjectEntityMapper.java	4
QuestionEntityMapper.java	4
QuestionnaireEntityMapper.java	9
UserEntityMapper.java	6, 9
Paper.java	3
ProjectService.java	3
QuestionnaireService.java	11
UserService.java	6, 11, 12
DemoPaperAnswerTests.java	5
DemoPaperTests.java	5
DemoQuestionnaireTests.java	4, 6, 9, 10, 11, 12, 13, 14, 25, 27, 29, 30
DemoUserTests.java	4, 6, 8, 9, 10, 11, 12, 13, 24, 26, 28, 29

规则	Non-empty statements should change control flow or have at least one side-effect

规则描述	Any statement (other than a null statement, which means a statement containing only a semicolon ;) which has no side effect and does
	not result in a change of control flow will normally indicate a programming error, and therefore should be refactored.
	 Noncompliant Code Example
	
	a == 1; // Noncompliant; was assignment intended?
	var msg = "Hello, "
	  "World!"; // Noncompliant; have we forgotten '+' operator on previous line?
	
	 See
	
	    MITRE, CWE-482 - Comparing instead of Assigning

文件名称	违规行
bootstrap-datetimepicker.js	1710
bootstrap.js	11, 28, 88, 184, 310, 448, 558, 653, 770, 936, 1017, 1134, 1295, 1973, 2097, 2270, 2347, 2426
jquery-3.5.1.js	733, 2197, 8208, 8218, 8221
bootstrap-datetimepicker.js	1710

bootstrap.js	11, 28, 88, 184, 310, 448, 558, 653, 770, 936, 1017, 1134, 1295, 1973, 2097, 2270, 2347, 2426
jquery-3.5.1.js	733, 2197, 8208, 8218, 8221

规则	"\<table\>" tags should have a description

| 规则描述 | In order to be accessible to visually impaired users, it is important that tables provides a description of its content before the data is accessed.
 The simplest way to do it, and also the one recommended by WCAG2 is to add a
 <caption> element inside the <table> .
 Other technics this rule accepts are:

   adding a concise description via   aria-label   or a href="https://www.w3.org/TR/wai-aria/#aria-labelledby"> aria-labelledby   attributes in the <table> .
   referencing a description element with an   aria-describedby attribute in the <table> .
   embedding the <table> inside a <figure> which also contains a <figcaption> .
   adding a summary attribute to the <table> tag. However note that this attribute has been deprecated in HTML5.

 See W3C WAI Web Accessibility Tutorials for more information.
 This rule raises an issue when a <table> has neither of the previously mentioned description mechanisms.
 Noncompliant Code Example

<table> <!-- Noncompliant -->
 ...
<table>

 Compliant Solution
 Adding a <caption> element.

<table>
  <caption>New York City Marathon Results 2013</caption>
  ...
</table>

 Adding an aria-describedby attribute.

<p id="mydesc">New York City Marathon Results 2013</p>
<table aria-describedby="mydesc">
  ...
</table>

 Embedding the table in a <figure> which also contains a <figcaption> .

<figure>
  <figcaption>New York City Marathon Results 2013</figcaption>
  <table>

  ...
  </table>
</figure>

 Adding a summary attribute. However note that this attribute has been deprecated in HTML5.

<table summary="New York City Marathon Results 2013">
  ...
</table> |

Exceptions
No issue will be raised on `<table>` used for layout purpose, i.e. when it contains a `role` attribute set to "presentation" or "none" . Note that using `<table>` for layout purpose is a bad practice.
No issue will be raised either on `<table>` containing an `aria-hidden` attribute set to "true" .
See

WCAG2, 1.3.1 - Info and Relationships
WCAG2, H39 - Using caption elements to associate data table captions with data tables

文件名称	违规行
index.html	27, 28, 31, 27, 28, 31
HttpResponseEntity.html	1
index.html	1
index.source.html	1
UUIDUtil.html	1
index.html	1
index.source.html	1
OptionController.html	1
PaperAnswerController.html	1
PaperController.html	1
ProjectController.html	1
QuestionController.html	1
QuestionnaireController.html	1
ShortUrlController.html	1
ToPaperController.html	1
UserController.html	1
index.html	1
index.source.html	1
UserEntity.html	1
index.html	1
index.source.html	1
OptionService.html	1
PaperAnswerService.html	1
PaperService.html	1
ProjectService.html	1
QuestionService.html	1
QuestionnaireService.html	1
UserService.html	1
index.html	1
index.source.html	1

Demo1Application.html	1
index.html	1
index.source.html	1
index.html	1
jacoco-sessions.html	1, 1

规则	Tables should have headers

规则描述	Assistive technologies, such as screen readers, use `<th>` headers to provide some context when users navigates a table. Without it the user gets rapidly lost in the flow of data.

Headers should be properly associated with the corresponding `<td>` cells by using either a scope attribute or
 headers and id attributes. See W3C WAI Web Accessibility
Tutorials for more information.
 This rule raises an issue whenever a `<table>` does not contain any `<th>` elements.
 Noncompliant Code Example

```
<table> <!-- Noncompliant -->
 <tr>
  <td>Name</td>
  <td>Age</td>
 </tr>
 <tr>
  <td>John Doe</td>
  <td>24</td>
 </tr>
 <tr>
  <td>Alice Doe</td>
  <td>54</td>
 </tr>
</table>
```

 Compliant Solution

```
<table>
 <tr>
  <th scope="col">Name</th>
  <th scope="col">Age</th>
 </tr>
 <tr>
  <td>John Doe</td>
  <td>24</td>
 </tr>
 <tr>
  <td>Alice Doe</td>
  <td>54</td>
 </tr>
</table>
```

 Exceptions
 No issue will be raised on `<table>` used for layout purpose, i.e. when it contains a role attribute set to
 "presentation" or "none" . Note that using `<table>` for layout purpose is a bad practice .
 No issue will be raised on `<table>` containing an aria-hidden attribute set to "true" .
 See

   WCAG2, 1.3.1 - Info
  and Relationships
   WCAG2, H51 - Using table markup to present tabular information

文件名称	违规行
HttpResponseEntity.html	1

index.html	1
index.source.html	1
UUIDUtil.html	1
index.html	1
index.source.html	1
OptionController.html	1
PaperAnswerController.html	1
PaperController.html	1
ProjectController.html	1
QuestionController.html	1
QuestionnaireController.html	1
ShortUrlController.html	1
ToPaperController.html	1
UserController.html	1
index.html	1
index.source.html	1
UserEntity.html	1
index.html	1
index.source.html	1
OptionService.html	1
PaperAnswerService.html	1
PaperService.html	1
ProjectService.html	1
QuestionService.html	1
QuestionnaireService.html	1
UserService.html	1
index.html	1
index.source.html	1
Demo1Application.html	1
index.html	1
index.source.html	1
index.html	1
jacoco-sessions.html	1, 1

规则	Attributes deprecated in HTML5 should not be used

规则描述	With the advent of HTML5, many old attributes were deprecated. To ensure the best user experience, deprecated attributes should not be used. This rule checks for the following deprecated attributes, where CSS should be used instead. Attributes

Attribute
Removed from

accept
form

align
caption , col , div , embed , h1-h6 , hr , iframe , img , input , legend , object , p , table , tbody , thead , tfoot , td , th , tr

alink
body

allowtransparency
iframe

archive
object

axis
td , th

background
body , table , thead , tbody , tfoot , tr , td , th

bgcolor
body , table , td , th , tr

border
img (border="0" allowed), object

bordercolor
table

cellpadding
table

cellspacing

table

char
col , tbody , thead , tfoot , td , th , tr

charoff
col , tbody , thead , tfoot , td , th , tr

charset
a , link

classid
object

clear
br

code
object

codebase
object

codetype
object

color
hr

compact
dl , ol , ul

coords
a

datafld
a , applet , button , div , fieldset , frame , iframe ,
img , input , label , legend , marquee , object , param ,
select , span , textarea

dataformatas
button , div , input , label , legend , marquee ,
object , option , select , span , table

datapagesize
table

datasrc
a , applet , button , div , frame , iframe , img ,
input , label , legend , marquee , object , option ,
select , span , table , textarea

declare
object

event
script

for
script

frame
table

frameborder
iframe

height
td , th

hspace
embed , iframe , img , input , object

ismap
input

langauge
script (language="javascript" , case insensitive, allowed)

link
body

lowsrc
img

marginbottom
body

marginheight
body , iframe

marginleft
body

marginright
body

margintop
body

marginwidth
body , iframe

methods
a , link

name
a (name="[a's element id]" allowed), embed , img , option

nohref
area

noshade
hr

nowrap
td , th

profile
head

rules
table

scheme
meta

scope
td

scrolling
iframe

shape
a

size
hr

standby
object

summary
table

target
link

text
body

type
li , param , ul

urn
a , link

usemap
input

valign
col , tbody , thead , tfoot , td , th , tr

valuetype
param

version
html

vlink
body

vspace
embed , iframe , img , input , object

width
col , hr , pre , table , td , th

See

W3C, Differences in HTML5
WHATWG, Obsolete Features

文件名称　　　　　　　　　　　　　　　　　　违规行

HttpResponseEntity.html	1
index.html	1
index.source.html	1
UUIDUtil.html	1
index.html	1
index.source.html	1
OptionController.html	1
PaperAnswerController.html	1
PaperController.html	1
ProjectController.html	1
QuestionController.html	1
QuestionnaireController.html	1
ShortUrlController.html	1
ToPaperController.html	1
UserController.html	1
index.html	1
index.source.html	1
UserEntity.html	1
index.html	1
index.source.html	1
OptionService.html	1
PaperAnswerService.html	1
PaperService.html	1
ProjectService.html	1
QuestionService.html	1
QuestionnaireService.html	1
UserService.html	1
index.html	1
index.source.html	1
Demo1Application.html	1
index.html	1
index.source.html	1
index.html	1
jacoco-sessions.html	1, 1

规则	Selectors should not be duplicated

规则描述	Duplication of selectors might indicate a copy-paste mistake. The rule detects the following kinds of duplications: within a list of selectors in a single rule set for duplicated selectors in different rule sets within a single stylesheet. Noncompliant Code Example .foo, .bar, .foo { ... } /* Noncompliant */ .class1 { ... } .class1 { ... } /* Noncompliant */ Compliant Solution .foo, .bar { ... } .class1 { ... } .class2 { ... }

文件名称	违规行
bootstrap.css	1080, 1084, 1099, 1112, 1115, 1146, 1529, 1558, 2349, 2589, 2595, 2612, 2620, 4143
demo.css	182, 285
bootstrap.css	1080, 1084, 1099, 1112, 1115, 1146, 1529, 1558, 2349, 2589, 2595, 2612, 2620, 4143
demo.css	182, 285

规则	Standard outputs should not be used directly to log anything

规则描述	When logging a message there are several important requirements which must be fulfilled:

	The user must be able to easily retrieve the logs
	The format of all logged message must be uniform to allow the user to easily read the log
	Logged data must actually be recorded
	Sensitive data must only be logged securely
	
	If a program directly writes to the standard outputs, there is absolutely no way to comply with those requirements. That's why defining and using a
	dedicated logger is highly recommended.
	Noncompliant Code Example
	
	System.out.println("My Message"); // Noncompliant
	
	Compliant Solution
	
	logger.log("My Message");
	
	See
	
	CERT, ERR02-J. - Prevent exceptions while logging data

文件名称	违规行
OptionController.java	42, 68, 94, 120
PaperAnswerController.java	42, 68, 94, 120
PaperController.java	44, 71, 98, 125
ProjectController.java	42, 68, 94, 120
QuestionController.java	42, 68, 94, 120
QuestionnaireController.java	44, 64, 90, 116, 142
UserController.java	41, 65, 89, 113, 137

规则	"<th>" tags should have "id" or "scope" attributes

| 规则描述 | Associating <table> headers, i.e. <th> elements, with their <td> cells enables screen
readers to announce the header prior to the data. This considerably increases the accessibility of tables to visually impaired users.
 There are two ways of doing it:

   Adding a scope attribute to <th> headers.
   Adding an id attribute to <th> headers and a headers attribute to every <td>
 element.

 It is recommended to add scope attributes to <th> headers whenever possible. Use <th id="...">
and <td headers="..."> only when <th scope="..."> is not capable of associating cells to their headers. This
happens for very complex tables which have headers splitting the data in multiple subtables. See a
href="https://www.w3.org/WAI/tutorials/tables/tips/">W3C WAI Web Accessibility Tutorials for more information.
 Note that complex tables can often be split into multiple smaller tables, which improves the user experience.
 This rule raises an issue when a <th> element has neither id nor scope attributes set.
 Noncompliant Code Example |

```
<table border="1">
  <caption>Contact Information</caption>
  <tr>
   <td></td>
   <th>Name</th>                          <!-- Non-Compliant
-->
   <th>Phone#</th>                        <!-- Non-Compliant
-->
   <th>City</th>                        <!-- Non-Compliant --
>
  </tr>
  <tr>
   <td>1.</td>
   <th>Joel Garner</th>                  <!-- Non-Compliant
-->
   <td>412-212-5421</td>
   <td>Pittsburgh</td>
  </tr>
  <tr>
   <td>2.</td>
   <th>Clive Lloyd</th>                  <!-- Non-Compliant
-->
   <td>410-306-1420</td>
   <td>Baltimore</td>
  </tr>
</table>
```

 Compliant Solution

```
<table border="1">
  <caption>Contact Information</caption>
  <tr>
   <td></td>
   <th scope="col">Name</th>                 <!-- Compliant
```

```
-->
    <th scope="col">Phone#</th>                    <!-- Compliant
-->
    <th scope="col">City</th>                      <!-- Compliant --
>
  </tr>
  <tr>
    <td>1.</td>
    <th scope="row">Joel Garner</th>               <!--
Compliant -->
    <td>412-212-5421</td>
    <td>Pittsburgh</td>
  </tr>
  <tr>
    <td>2.</td>
    <th scope="row">Clive Lloyd</th>               <!--
Compliant -->
    <td>410-306-1420</td>
    <td>Baltimore</td>
  </tr>
</table>

 or:

<table border="1">
  <caption>Contact Information</caption>
  <tr>
    <td></td>
    <th id="name">Name</th>                     <!-- Compliant
-->
    <th id="phone">Phone#</th>                  <!-- Compliant
-->
    <th id="city">City</th>                     <!-- Compliant -->
  </tr>
  <tr>
    <td>1.</td>
    <th id="person1" headers="name">Joel Garner</th>      <!--
Compliant -->
    <td headers="phone person1">412-212-5421</td>
    <td headers="city person1">Pittsburgh</td>
  </tr>
  <tr>
    <td>2.</td>
    <th id="person2" headers="name">Clive Lloyd</th>      <!--
Compliant -->
    <td headers="phone person2">410-306-1420</td>
    <td headers="city person2">Baltimore</td>
  </tr>
</table>

 See

    WCAG2, 1.3.1  - Info
  and Relationships
    WCAG2, H43  - Using id and headers attributes to associate
data cells with header
  cells in data tables
```

文件名称		违规行

index.html	30, 31, 32, 33, 31, 32, 33, 34, 34, 35, 36, 37, 38, 39, 30, 31, 32, 33, 31, 32, 33, 34, 34, 35, 36, 37, 38, 39

规则	Cognitive Complexity of functions should not be too high
规则描述	Cognitive Complexity is a measure of how hard the control flow of a function is to understand. Functions with high Cognitive Complexity will be difficult to maintain. See Cognitive Complexity

文件名称	违规行
index.js	36, 37, 56
bootstrap-datetimepicker.js	36
bootstrap.js	1112, 1467, 1603, 1973, 2458
jquery-3.5.1.js	40
index.js	36, 37, 56
bootstrap-datetimepicker.js	36
bootstrap.js	1112, 1467, 1603, 1973, 2458
jquery-3.5.1.js	40
prettify.js	308, 354, 675, 809, 1068, 1378

规则	Local variables should not be declared and then immediately returned or thrown

规则描述	Declaring a variable only to immediately return or throw it is a bad practice.
	Some developers argue that the practice improves code readability, because it enables them to explicitly name what is being returned. However, this variable is an internal implementation detail that is not exposed to the callers of the method. The method name should be sufficient for callers to know exactly what will be returned.
	Noncompliant Code Example
	
	public long computeDurationInMilliseconds() {
	  long duration = (((hours * 60) + minutes) * 60 + seconds) * 1000 ;
	  return duration;
	}
	
	public void doSomething() {
	  RuntimeException myException = new RuntimeException();
	  throw myException;
	}
	
	Compliant Solution
	
	public long computeDurationInMilliseconds() {
	  return (((hours * 60) + minutes) * 60 + seconds) * 1000 ;
	}
	
	public void doSomething() {
	  throw new RuntimeException();
	}

文件名称	违规行
OptionService.java	22, 46, 56
PaperAnswerService.java	23, 47, 57
PaperService.java	22, 46, 56
ProjectService.java	23, 48, 58
QuestionService.java	22, 46, 56
QuestionnaireService.java	25, 35, 58, 68
UserService.java	23, 31, 53, 61

规则	String literals should not be duplicated

规则描述	Duplicated string literals make the process of refactoring error-prone, since you must be sure to update all occurrences.
	On the other hand, constants can be referenced from many places, but only need to be updated in a single place.
	Noncompliant Code Example
	With the default threshold of 3:
	
	public void run() {
	  prepare("action1"); // Noncompliant - "action1" is duplicated 3 times
	  execute("action1");
	  release("action1");
	}
	
	@SuppressWarning("all") // Compliant - annotations are excluded
	private void method1() { /* ... */ }
	@SuppressWarning("all")
	private void method2() { /* ... */ }
	
	public String method3(String a) {
	  System.out.println("'" + a + "'"); // Compliant - literal "'" has less than 5 characters and is excluded
	  return ""; // Compliant - literal "" has less than 5 characters and is excluded
	}
	
	 Compliant Solution
	
	private static final String ACTION_1 = "action1"; // Compliant
	
	public void run() {
	  prepare(ACTION_1); // Compliant
	  execute(ACTION_1);
	  release(ACTION_1);
	}
	
	 Exceptions
	To prevent generating some false-positives, literals having less than 5 characters are excluded.

文件名称	违规行
DemoApplicationTests.java	54, 62, 63, 64, 238
DemoOptionTests.java	43, 52, 54
DemoPaperAnswerTests.java	43, 53
DemoPaperTests.java	43, 53
DemoProjectTests.java	42, 51, 52
DemoQuestionTests.java	42, 51, 52
DemoQuestionnaireTests.java	58, 68
DemoUserTests.java	56, 64, 65

规则	Sections of code should not be commented out

规则描述	Programmers should not comment out code as it bloats programs and reduces readability. Unused code should be deleted and can be retrieved from source control history if required.
文件名称	违规行
index.js	27, 92
bootstrap-datetimepicker.js	1220, 1228, 1236, 1443
jquery-3.5.1.js	3924, 3939, 3943, 3955, 3968
index.js	27, 92
bootstrap-datetimepicker.js	1220, 1228, 1236, 1443
jquery-3.5.1.js	3924, 3939, 3943, 3955, 3968

规则	Image, area and button with image tags should have an "alt" attribute

| 规则描述 | The alt attribute provides a textual alternative to an image. It is used whenever the actual image cannot be rendered. Common reasons for that include:

The image can no longer be found
Visually impaired users using a screen reader software
Images loading is disabled, to reduce data consumption on mobile phones

It is also very important to not set an alt attribute to a non-informative value. For example `<img ... alt="logo">` is useless as it doesn't give any information to the user. In this case, as for any other decorative image, it is better to use a CSS background image instead of an `<img>` tag. If using CSS background-image is not possible, an empty alt="" is tolerated. See Exceptions bellow.
This rule raises an issue when

an `<input type="image">` tag or an `<area>` tag have no alt attribute or their
alt attribute has an empty string value.
an `<img>` tag has no alt attribute.

Noncompliant Code Example

`<img src="foo.png" />` `<!-- Noncompliant -->`
`<input type="image" src="bar.png" />` `<!-- Noncompliant -->`
`<input type="image" src="bar.png" alt="" />` `<!-- Noncompliant -->`

`<img src="house.gif" usemap="#map1"`
`  alt="rooms of the house." />`
`<map id="map1" name="map1">`
`  <area shape="rect" coords="0,0,42,42"`
`  href="bedroom.html"/>` `<!-- Noncompliant -->`
`  <area shape="rect" coords="0,0,21,21"`
`  href="lounge.html" alt=""/>` `<!-- Noncompliant -->`
`</map>`

Compliant Solution

`<img src="foo.png" alt="Some textual description of foo.png" />`
`<input type="image" src="bar.png" alt="Textual description of bar.png" />`

`<img src="house.gif" usemap="#map1"`
`  alt="rooms of the house." />`
`<map id="map1" name="map1">`
`  <area shape="rect" coords="0,0,42,42"`
`  href="bedroom.html" alt="Bedroom" />`
`  <area shape="rect" coords="0,0,21,21"`
`  href="lounge.html" alt="Lounge"/>`
`</map>`

Exceptions
`<img>` tags with empty string alt="" attributes won't raise any issue. However this technic should be used in two cases only:
When the image is decorative and it is not possible to use a CSS background image. For example, when the decorative `<img>` is |

generated via javascript with a source image coming from a
database, it is better to use an <img alt=""> tag rather than
generate
CSS code.

```
<li *ngFor="let image of images">
   <img [src]="image" alt="">
</li>
```

 When the image is not decorative but it's alt text would repeat a
nearby text. For example, images contained in links should not
duplicate the link's text in their alt attribute, as it would make the
screen reader repeat the text twice.

```
<a href="flowers.html">
   <img src="tulip.gif" alt="" />
   A blooming tulip
</a>
```

 In all other cases you should use CSS background images.
 See W3C WAI Web Accessibility Tutorials for
more
information.
 See

    WCAG2, H24 - Providing text alternatives for the area elements
of image maps
    WCAG2, H36 - Using alt attributes on images used as submit
buttons
    WCAG2, H37 - Using alt attributes on img elements
    WCAG2, H67 - Using null alt text and no title attribute on img
elements for images
 that AT should ignore
    WCAG2, H2 - Combining adjacent image and text links for the
same resource
    WCAG2, 1.1.1 - Non-text Content
    WCAG2, 2.4.4 - Link Purpose (In Context)
    WCAG2, 2.4.9 - Link Purpose (Link Only)

文件名称	违规行
index.html	5, 11, 15, 20, 41, 53, 65, 18
demo_index.html	40
index.html	5, 11, 15, 20, 41, 53, 65, 18
demo_index.html	40

规则	Source files should not have any duplicated blocks
规则描述	An issue is created on a file as soon as there is at least one block of duplicated code on this file

文件名称	违规行
OptionController.java	N/A
PaperAnswerController.java	N/A

PaperController.java	N/A
ProjectController.java	N/A
QuestionController.java	N/A
QuestionnaireController.java	N/A
UserController.java	N/A
OptionEntity.java	N/A
TempOptionEntity.java	N/A
DemoApplicationTests.java	N/A
DemoOptionTests.java	N/A
DemoPaperAnswerTests.java	N/A
DemoPaperTests.java	N/A
DemoProjectTests.java	N/A
DemoQuestionTests.java	N/A
DemoQuestionnaireTests.java	N/A
DemoUserTests.java	N/A

规则	Unused assignments should be removed

| 规则描述 | A dead store happens when a local variable is assigned a value that is not read by any subsequent instruction. Calculating or retrieving a value
only to then overwrite it or throw it away, could indicate a serious error in the code. Even if it's not an error, it is at best a waste of resources.
Therefore all calculated values should be used.
 Noncompliant Code Example

i = a + b; // Noncompliant; calculation result not used before value is overwritten
i = compute();

 Compliant Solution

i = a + b;
i += compute();

 Exceptions
 This rule ignores initializations to -1, 0, 1, null , undefined , [], {}, true , false and
 "" . Variables that start with an underscore (e.g. ' _unused ') are ignored.
 This rule also ignores variables declared with object destructuring using rest syntax (used to exclude some properties from object):

let {a, b, ...rest} = obj; // 'a' and 'b' are ok
doSomething(rest);

let [x1, x2, x3] = arr;   // but 'x1' is noncompliant, as omitting syntax can be used: "let [, x2, x3] = arr;"
doSomething(x2, x3);

 See

    MITRE, CWE-563 - Assignment to Variable without Use ('Unused Variable')
    CERT, MSC13-C. - Detect and remove unused values
    CERT, MSC56-J. - Detect and remove superfluous code and values |

文件名称	违规行
index.js	65
jquery-3.5.1.js	363, 947, 947, 1971, 1974, 2010
index.js	65
jquery-3.5.1.js	363, 947, 947, 1971, 1974, 2010
prettify.js	836, 1388, 1481

规则	Properties should not be duplicated

规则描述	CSS allows duplicate property names but only the last instance of a duplicated name determines the actual value that will be used for it.
	Therefore, changing values of other occurrences of a duplicated name will have no effect and may cause misunderstandings and bugs.
	 This rule ignores $sass , @less , and var(--custom-property) variable syntaxes.
	 Noncompliant Code Example
	
	a {
	　color: pink;
	　background: orange;
	　color: orange
	}
	
	 Compliant Solution
	
	a {
	　color: pink;
	　background: orange
	}

文件名称	违规行
bootstrap-datetimepicker.css	200, 252, 351
bootstrap-theme.css	74, 345, 372
bootstrap.css	57, 2674
bootstrap-datetimepicker.css	200, 252, 351
bootstrap-theme.css	74, 345, 372
bootstrap.css	57, 2674

规则	Loop counters should not be assigned to from within the loop body

| 规则描述 | Loop counters should not be modified in the body of the loop. However other loop control variables representing logical values may be modified in
the loop, for example a flag to indicate that something has been completed, which is then tested in the for statement.
 Noncompliant Code Example

var names = ["Jack", "Jim", "", "John"];
for (var i = 0; i < names.length; i++) {
  if (!names[i]) {
    i = names.length;　　　　　　　　// Non-Compliant
  } else {
    console.log(names[i]);
  }
}

 Compliant Solution

var names = ["Jack", "Jim", "", "John"];
for (var name of names) {
  if (!name) {
    break;　　　　　　　　// Compliant
  } else {
    console.log(name);
  }
}|

文件名称	违规行
jquery-3.5.1.js	2639, 3493, 3500, 7478, 7521, 7901, 2639, 3493, 3500, 7478, 7521, 7901
prettify.js	429, 719

规则	Functions should not be empty

| 规则描述 | There are several reasons for a function not to have a function body:

It is an unintentional omission, and should be fixed to prevent an unexpected behavior in production.
It is not yet, or never will be, supported. In this case an exception should be thrown in languages where that mechanism is available.
The method is an intentionally-blank override. In this case a nested comment should explain the reason for the blank override.

Noncompliant Code Example

```
function foo() {
}

var foo = () => {};
```

Compliant Solution

```
function foo() {
    // This is intentional
}

var foo = () => {
    do_something();
};
``` |
|---|---|

文件名称	违规行
index.js	283, 524, 542
jquery-3.5.1.js	337, 2309, 9732, 9871
index.js	283, 524, 542
jquery-3.5.1.js	337, 2309, 9732, 9871

规则	"for of" should be used with Iterables
规则描述	If you have an iterable, such as an array, set, or list, your best option for looping through its values is the for of syntax. Use a counter, and ... well you'll get the right behavior, but your code just isn't as clean or clear.
Noncompliant Code Example

```
const arr = [4, 3, 2, 1];

for (let i = 0; i < arr.length; i++) {  // Noncompliant
  console.log(arr[i]);
}
```

Compliant Solution

```
const arr = [4, 3, 2, 1];

for (let value of arr) {
  console.log(value);
}
``` |

文件名称	违规行
index.js	245
bootstrap-datetimepicker.js	1599
index.js	245
bootstrap-datetimepicker.js	1599
prettify.js	208, 450, 462, 1383, 1466
sort.js	66, 80, 96, 106

规则	Track uses of "TODO" tags

规则描述	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
	
	function doSomething() {
	  // TODO
	}
	
	 See
	
	    MITRE, CWE-546 - Suspicious Comment

文件名称	违规行
bootstrap.js	1759
jquery-3.5.1.js	790, 804, 4394, 4462
bootstrap.js	1759
jquery-3.5.1.js	790, 804, 4394, 4462
prettify.js	217, 550, 1025

规则	Local variables should not be declared and then immediately returned or thrown

规则描述	Declaring a variable only to immediately return or throw it is a bad practice.
	 Some developers argue that the practice improves code readability, because it enables them to explicitly name what is being returned. However, this
	variable is an internal implementation detail that is not exposed to the callers of the method. The method name should be sufficient for callers to
	know exactly what will be returned.
	 Noncompliant Code Example
	
	function computeDurationInMilliseconds() {
	  var duration = (((hours * 60) + minutes) * 60 + seconds) * 1000 ;
	  return duration;
	}
	
	 Compliant Solution
	
	function computeDurationInMilliseconds() {
	  return (((hours * 60) + minutes) * 60 + seconds) * 1000 ;
	}

文件名称	违规行
index.js	164, 231, 298, 331, 422
storage.js	9
index.js	164, 231, 298, 331, 422
storage.js	9

规则	"" and "" tags should be used

规则描述	The / and / <i> tags have exactly the same effect in most web browsers, but there is a fundamental difference between them: and have a semantic meaning whereas and <i> only convey styling information like CSS. While can have simply no effect on a some devices with limited display or when a screen reader software is used by a blind person, will: Display the text bold in normal browsers Speak with lower tone when using a screen reader such as Jaws Consequently: in order to convey semantics, the and <i> tags shall never be used, in order to convey styling information, the and <i> should be avoided and CSS should be used instead. Noncompliant Code Example <i>car</i> <!-- Noncompliant --> train <!-- Noncompliant --> Compliant Solution car train Exceptions This rule is relaxed in case of icon fonts usage. <i class="..." aria-hidden="true" /> <!-- Compliant icon fonts usage -->

文件名称	违规行
index.html	26, 30, 34, 38, 42
500.html	13
index.html	26, 30, 34, 38, 42
500.html	13

规则	Nested blocks of code should not be left empty
规则描述	Most of the time a block of code is empty when a piece of code is really missing. So such empty block must be either filled or removed. Noncompliant Code Example for (var i = 0; i < length; i++) {} // Empty on purpose or missing piece of code ? Exceptions When a block contains a comment, this block is not considered to be empty. Moreover catch blocks are ignored.

文件名称	违规行
index.js	185, 187, 272, 274
jquery-3.5.1.js	749, 3331
index.js	185, 187, 272, 274
jquery-3.5.1.js	749, 3331

规则	Functions should always return the same type
规则描述	Unlike strongly typed languages, JavaScript does not enforce a return type on a function. This means that different paths through a function can return different types of values, which can be very confusing to the user and significantly harder to maintain. Noncompliant Code Example

function foo(a) { // Noncompliant
  if (a === 1) {
    return true;
  }
  return 3;
}

 Compliant Solution

function foo(a) {
  if (a === 1) {
    return true;
  }
  return false;
}

 Exceptions
 Functions returning this are ignored.

function foo() {
  // ...
  return this;
}

Functions returning expressions having type any are ignored. |

文件名称	违规行
bootstrap-datetimepicker.js	54
bootstrap.js	34, 407, 412, 417
jquery-3.5.1.js	7543
bootstrap-datetimepicker.js	54
bootstrap.js	34, 407, 412, 417
jquery-3.5.1.js	7543

规则	Variables should be declared explicitly

规则描述	JavaScript variable scope can be particularly difficult to understand and get right. The situation gets even worse when you consider the accidental creation of global variables, which is what happens when you declare a variable inside a function or the for clause of a for-loop without using the let , const or var keywords. let and const were introduced in ECMAScript 2015, and are now the preferred keywords for variable declaration. Noncompliant Code Example

	`function f(){`
	`  i = 1;        // Noncompliant; i is global`
	
	`  for (j = 0; j < array.length; j++) {  // Noncompliant; j is global now too`
	`    // ...`
	`  }`
	`}`
	
	Compliant Solution
	
	`function f(){`
	`  var i = 1;`
	
	`  for (let j = 0; j < array.length; j++) {`
	`    // ...`
	`  }`
	`}`

文件名称	违规行
index.js	8, 12, 58, 295, 8, 12, 58, 295
sort.js	67, 95, 105

规则	Font declarations should contain at least one generic font family

规则描述	If none of the font names defined in a font or font-family declaration are available on the browser of the user, the browser will display the text using its default font. It's recommended to always define a generic font family for each declaration of font or font-family to get a less degraded situation than relying on the default browser font. All browsers should implement a list of generic font matching these families: Serif , Sans-serif , cursive , fantasy , Monospace .
	Noncompliant Code Example

	`a {`
	`  font-family: Helvetica, Arial, Verdana, Tahoma; /* Noncompliant; there is no generic font family in the list */`
	`}`

	Compliant Solution

	`a {`
	`  font-family: Helvetica, Arial, Verdana, Tahoma, sans-serif;`
	`}`

	See
	CSS Specification - Generic font families

文件名称	违规行
bootstrap.css	275, 6612
demo.css	12
iconfont.css	9
style.css	15
bootstrap.css	275, 6612
demo.css	12
iconfont.css	9
style.css	15

规则	Sections of code should not be commented out
规则描述	Programmers should not comment out code as it bloats programs and reduces readability. Unused code should be deleted and can be retrieved from source control history if required.

文件名称	违规行
index.html	11, 12, 11, 12, 43, 11, 12, 11, 12, 43

规则	A conditionally executed single line should be denoted by indentation

规则描述	In the absence of enclosing curly braces, the line immediately after a conditional is the one that is conditionally executed. By both convention and good practice, such lines are indented. In the absence of both curly braces and indentation the intent of the original programmer is entirely unclear and perhaps not actually what is executed. Additionally, such code is highly likely to be confusing to maintainers. Noncompliant Code Example
	
	if (condition) // Noncompliant
	doTheThing();
	
	doTheOtherThing();
	somethingElseEntirely();
	
	foo();
	
	 Compliant Solution
	
	if (condition)
	  doTheThing();
	
	doTheOtherThing();
	somethingElseEntirely();
	
	foo();

文件名称	**违规行**
index.js	12, 161, 268, 12, 161, 268

规则	Sections of code should not be commented out
规则描述	Programmers should not comment out code as it bloats programs and reduces readability. Unused code should be deleted and can be retrieved from source control history if required.
文件名称	**违规行**
ShortUrlController.java	22, 28
DemoApplicationTests.java	203, 503
DemoProjectTests.java	95
DemoQuestionnaireTests.java	80

规则	Unused local variables and functions should be removed

规则描述	If a local variable or a local function is declared but not used, it is dead code and should be removed. Doing so will improve maintainability
	because developers will not wonder what the variable or function is used for.
	 Noncompliant Code Example
	
	function numberOfMinutes(hours) {
	  var seconds = 0; // seconds is never used
	  return hours * 60;
	}
	
	 Compliant Solution
	
	function numberOfMinutes(hours) {
	  return hours * 60;
	}

文件名称	违规行
index.js	223, 65, 223, 65
prettify.js	836

规则	Functions should not have identical implementations

规则描述	When two functions have the same implementation, either it was a mistake - something else was intended - or the duplication was intentional, but
	may be confusing to maintainers. In the latter case, the code should be refactored.
	 Noncompliant Code Example
	
	function calculateCode() {
	  doTheThing();
	  doOtherThing();
	  return code;
	}
	
	function getName() { // Noncompliant
	  doTheThing();
	  doOtherThing();
	  return code;
	}
	
	 Compliant Solution
	
	function calculateCode() {
	  doTheThing();
	  doOtherThing();
	  return code;
	}
	
	function getName() {
	  return calculateCode();
	}
	
	 Exceptions
	Functions with fewer than 3 lines are ignored.

文件名称	违规行
index.js	82
jquery-3.5.1.js	10177
index.js	82
jquery-3.5.1.js	10177

规则	Multiline blocks should be enclosed in curly braces

规则描述	Curly braces can be omitted from a one-line block, such as with an if statement or for loop, but doing so can be misleading and induce bugs. This rule raises an issue when the whitespacing of the lines after a one line block indicates an intent to include those lines in the block, but the omission of curly braces means the lines will be unconditionally executed once. Note that this rule considers tab characters to be equivalent to 1 space. If you mix spaces and tabs you will sometimes see issues in code which looks fine in your editor but is confusing when you change the size of tabs. Noncompliant Code Example

```
if (condition)
  firstActionInBlock();
  secondAction();  // Noncompliant; executed unconditionally
thirdAction();

if (condition) firstActionInBlock(); secondAction();  // Noncompliant; secondAction executed unconditionally

if (condition) firstActionInBlock();  // Noncompliant
  secondAction();  // Executed unconditionally

if (condition); secondAction();  // Noncompliant; secondAction executed unconditionally

let str = undefined;
for (let i = 0; i < array.length; i++)
  str = array[i];
  doTheThing(str);  // Noncompliant; executed only on last array element
```

 Compliant Solution

```
if (condition) {
  firstActionInBlock();
  secondAction();
}
thirdAction();

let str = undefined;
for (let i = 0; i < array.length; i++) {
  str = array[i];
  doTheThing(str);
}
```

 See

     MITRE, CWE-483 - Incorrect Block Delimitation

文件名称	违规行
index.js	9, 38, 9, 38

规则	Extra semicolons should be removed

规则描述	Extra semicolons (;) are usually introduced by mistake, for example because:
	
	    It was meant to be replaced by an actual statement, but this was forgotten.
	    There was a typo which lead the semicolon to be doubled, i.e. ;; .
	    There was a misunderstanding about where semicolons are required or useful.
	
	 Noncompliant Code Example
	
	var x = 1;; // Noncompliant
	
	function foo() {
	}; // Noncompliant
	
	 Compliant Solution
	
	var x = 1;
	
	function foo() {
	}
	
	 See
	
	    CERT, MSC12-C. - Detect and remove code that has no effect or is never executed
	
	    CERT, MSC51-J. - Do not place a semicolon immediately following an if, for, or while
	  condition
	    CERT, EXP15-C. - Do not place a semicolon on the same line as an if, for, or while
	  statement

文件名称	违规行
bootstrap-datetimepicker.zh-CN.js	5
jquery-3.5.1.js	3029
bootstrap-datetimepicker.zh-CN.js	5
jquery-3.5.1.js	3029

规则	A "while" loop should be used instead of a "for" loop

规则描述	When only the condition expression is defined in a for loop, and the initialization and increment expressions are missing, a
	 while loop should be used instead to increase readability.
	 Noncompliant Code Example
	
	for (;condition;) { /*...*/ }
	
	 Compliant Solution
	
	while (condition) { /*...*/ }

文件名称	违规行
jquery-3.5.1.js	2282, 2290, 2282, 2290

规则	CSS files should not be empty
规则描述	This rule raises an issue when a CSS file is empty (ie: containing only spaces).
文件名称	违规行
index.css	1, 1, 1, 1

规则	Arguments to built-in functions should match documented types

规则描述	The types of the arguments to built-in functions are specified in the JavaScript language specifications. Calls to these functions should conform to the documented types, otherwise the result will most likely not be what was expected (e.g.: the call would always return false). Noncompliant Code Example
	
	const isTooSmall = Math.abs(x < 0.0042);
	
	 Compliant Solution
	
	const isTooSmall = Math.abs(x) < 0.0042;

文件名称	违规行
jquery-3.5.1.js	7810, 7810

规则	Duplicated font names should be removed

规则描述	Having duplicated font names doesn't help to read the font declaration and may be an indicator the author of the line was not sure how to configure it. This rule raises an issue when font or font-family properties contain a duplicated font name. This rule ignores $sass , @less , and var(--custom-property) variable syntaxes. Noncompliant Code Example
	
	a {
	  font-family: 'Georgia', Georgia, serif; /* Noncompliant; 'Georgia' is duplicated */
	}
	
	 Compliant Solution
	
	a {
	  font-family: Georgia, serif;
	}

文件名称	违规行

bootstrap.css	112, 112

规则	Special identifiers should not be bound or assigned

规则描述	JavaScript has special identifiers that, while not reserved, still should not be used as identifiers. They include:
	
	eval - evaluates a string as JavaScript code
	arguments - used to access function arguments through indexed properties.
	undefined - returned for values and properties that have not yet been assigned
	NaN - Not a Number; returned when math functions fail.
	Infinity - when a number exceeds the upper limit of the floating point numbers
	
	These words should not be bound or assigned, because doing so would overwrite the original definitions of these identifiers. What's more, assigning
	or binding some of these names will generate an error in JavaScript strict mode code.
	Noncompliant Code Example
	
	`eval = 17; // Noncompliant`
	`arguments++; // Noncompliant`
	`++eval; // Noncompliant`
	`var obj = { set p(arguments) { } }; // Noncompliant`
	`var eval; // Noncompliant`
	`try { } catch (arguments) { } // Noncompliant`
	`function x(eval) { } // Noncompliant`
	`function arguments() { } // Noncompliant`
	`var y = function eval() { }; // Noncompliant`
	`var f = new Function("arguments", "return 17;"); // Noncompliant`
	
	`function fun() {`
	`  if (arguments.length == 0) { // Compliant`
	`    // do something`
	`  }`
	`}`
	
	Compliant Solution
	
	`result = 17;`
	`args++;`
	`++result;`
	`var obj = { set p(arg) { } };`
	`var result;`
	`try { } catch (args) { }`
	`function x(arg) { }`
	`function args() { }`
	`var y = function fun() { };`
	`var f = new Function("args", "return 17;");`
	
	`function fun() {`
	`  if (arguments.length == 0) {`
	`    // do something`
	`  }`
	`}`

文件名称	违规行
bootstrap-datetimepicker.js	36, 36

规则	Functions should not be defined inside loops

规则描述	Defining a function inside of a loop can yield unexpected results. Such a function keeps references to the variables which are defined in outer scopes. All function instances created inside the loop therefore see the same values for these variables, which is probably not expected.
	 Noncompliant Code Example
	
	var funs = [];
	for (var i = 0; i < 13; i++) {
	  funs[i] = function() { // Non-Compliant
	    return i;
	  };
	}
	console.log(funs[0]()); // 13 instead of 0
	console.log(funs[1]()); // 13 instead of 1
	console.log(funs[2]()); // 13 instead of 2
	console.log(funs[3]()); // 13 instead of 3
	...

文件名称	违规行
jquery-3.5.1.js	7469, 7469

规则	Array indexes should be numeric

规则描述	Associative arrays allow you to store values in an array with either numeric or named indexes. But creating and populating an object is just as easy as an array, and more reliable if you need named members.
	 Noncompliant Code Example
	
	let arr = [];
	arr[0] = 'a';
	arr['name'] = 'bob'; // Noncompliant
	arr[1] = 'foo';
	
	 Compliant Solution
	
	let obj = {
	  name: 'bob',
	  arr: ['a', 'foo']
	};

文件名称	违规行
jquery-3.5.1.js	2758, 2758

规则	"for" loop increment clauses should modify the loops' counters

规则描述	It can be extremely confusing when a for loop's counter is incremented outside of its increment clause. In such cases, the increment
	should be moved to the loop's increment clause if at all possible.
	 Noncompliant Code Example
	
	for (i = 0; i < 10; j++) { // Noncompliant
	  // ...
	  i++;
	}
	
	 Compliant Solution
	
	for (i = 0; i < 10; i++, j++) {
	  // ...
	}
	
	 Or
	
	for (i = 0; i < 10; i++) {
	  // ...
	  j++;
	}

文件名称	违规行
jquery-3.5.1.js	3491, 3491

规则	Collection and array contents should be used

规则描述	When a collection is populated but its contents are never used, then it is surely some kind of mistake. Either refactoring has rendered the
	collection moot, or an access is missing.
	 This rule raises an issue when no methods are called on a collection other than those that add or remove values.
	 Noncompliant Code Example
	
	function getLength(a, b, c) {
	  const strings = []; // Noncompliant
	  strings.push(a);
	  strings.push(b);
	  strings.push(c);
	
	  return a.length + b.length + c.length;
	}
	
	 Compliant Solution
	
	function getLength(a, b, c) {
	  return a.length + b.length + c.length;
	}

文件名称	违规行
index.js	15, 15

规则	Functions should not be called both with and without "new"

规则描述	Constructor functions, which create new object instances, must only be called with new . Non-constructor functions must not. Mixing
	these two usages could lead to unexpected results at runtime.
	 Noncompliant Code Example
	
	function getNum() {
	  return 5;
	}
	
	function Num(numeric, alphabetic) {
	  this.numeric = numeric;
	  this.alphabetic = alphabetic;
	}
	
	var myFirstNum = getNum();
	var my2ndNum = new getNum(); // Noncompliant. An empty object is returned, NOT 5
	
	var myNumObj1 = new Num();
	var myNumObj2 = Num(); // Noncompliant. undefined is returned, NOT an object

文件名称	违规行
bootstrap-datetimepicker.js	743, 743

规则	Strict equality operators should not be used with dissimilar types

规则描述	Comparing dissimilar types using the strict equality operators `===` and `!==` will always return the same value, respectively `false` and `true`, because no type conversion is done before the comparison. Thus, such comparisons can only be bugs.
	 Noncompliant Code Example
	
	var a = 8;
	var b = "8";
	
	if (a === b) { // Noncompliant; always false
	 // ...
	}
	
	 Compliant Solution
	
	var a = 8;
	var b = "8";
	
	if (a == b) {
	 // ...
	}
	
	 or
	
	var a = 8;
	var b = "8";
	
	if (a === Number(b)) {
	 // ...
	}

文件名称	违规行
jquery-3.5.1.js	2746, 2746

规则	Function parameters, caught exceptions and foreach variables' initial values should not be ignored

规则描述	While it is technically correct to assign to parameters from within function bodies, it reduces code readability because developers won't be able to tell whether the original parameter or some temporary variable is being accessed without going through the whole function. Moreover, some developers might also expect assignments of function parameters to be visible to callers, which is not the case, and this lack of visibility could confuse them. Instead, all parameters, caught exceptions, and foreach parameters should be treated as constants. Noncompliant Code Example
	
	function MyClass(name, strings) {
	  name = foo; // Noncompliant
	
	  for (var str of strings) {
	    str = ""; // Noncompliant
	  }
	}

文件名称	违规行
jquery-3.5.1.js	2707, 2707

规则	"<html>" element should have a language attribute

| 规则描述 | The <html> element should provide the lang and/or xml:lang attribute in order to identify the default language of a document.
 It enables assistive technologies, such as screen readers, to provide a comfortable reading experience by adapting the pronunciation and accent to the language. It also helps braille translation software, telling it to switch the control codes for accented characters for instance.
 Other benefits of marking the language include:

    assisting user agents in providing dictionary definitions or helping users benefit from translation tools.
    improving search engine ranking .

 Both the lang and the xml:lang attributes can take only one value.

 Noncompliant Code Example

<!DOCTYPE html>
<html> <!-- Noncompliant -->
    <head>
         <title>A page written in english</title>
         <meta content="text/html; charset=utf-8" />
    </head>

    <body>   ...
    </body>
</html>

 Compliant Solution

<!DOCTYPE html>
<html lang="en">
    <head>
         <title>A page written in english</title>
         <meta content="text/html; charset=utf-8" />
    </head>

    <body>   ...
    </body>
</html>

<!DOCTYPE html>
<html lang="en" xml:lang="en">
    <head>
         <title>A page written in english</title>
         <meta content="text/html; charset=utf-8" /> |

    </head>

    <body>   ...   </body>
</html>

 See

    WCAG2, H57 - Using language attributes on the html element
WCAG2, 3.1.1 - Language of Page

文件名称	违规行
demo_index.html	2, 2

规则	Variables should not be self-assigned

规则描述	There is no reason to re-assign a variable to itself. Either this statement is redundant and should be removed, or the re-assignment is a mistake and some other value or variable was intended for the assignment instead.
	 Noncompliant Code Example
	
	function setName(name) {
	    name = name;
	}
	
	 Compliant Solution
	
	function setName(name) {
	    this.name = name;
	}
	
	 See
	
	   CERT, MSC12-C. - Detect and remove code that has no effect or is never executed

文件名称	违规行
jquery-3.5.1.js	9535, 9535

规则	Media features should be valid

规则描述	The W3C specifications define the valid media features. Only the official and browser-specific media features should be used to get the expected impact in the final rendering. Noncompliant Code Example @media screen and (unknown: 1000px) { .. } Compliant Solution @media screen and (width: 1000px) { .. } See CSS Specification - @media rule

文件名称	违规行
bootstrap.css	6332, 6332

规则	Strings and Boxed types should be compared using "equals()"

规则描述	It's almost always a mistake to compare two instances of java.lang.String or boxed types like java.lang.Integer using reference equality == or != , because it is not comparing actual value but locations in memory.
	 Noncompliant Code Example
	
	String firstName = getFirstName(); // String overrides equals
	String lastName = getLastName();
	
	if (firstName == lastName) { ... }; // Non-compliant; false even if the strings have the same value
	
	 Compliant Solution
	
	String firstName = getFirstName();
	String lastName = getLastName();
	
	if (firstName != null && firstName.equals(lastName)) { ... };
	
	 See
	
	    MITRE, CWE-595 - Comparison of Object References Instead of Object Contents
	    MITRE, CWE-597 - Use of Wrong Operator in String Comparison
	    CERT, EXP03-J. - Do not use the equality operators when comparing values of boxed
	     primitives
	    CERT, EXP50-J. - Do not confuse abstract object equality with reference equality

文件名称	违规行
PaperController.java	61

规则	Field names should comply with a naming convention

规则描述	Sharing some naming conventions is a key point to make it possible for a team to efficiently collaborate. This rule allows to check that field names match a provided regular expression. Noncompliant Code Example With the default regular expression ^[a-z][a-zA-Z0-9]*$:
	
	class MyClass {
	  private int my_field;
	}
	
	 Compliant Solution
	
	class MyClass {
	  private int myField;
	}

文件名称	违规行
ShortUrlController.java	14

规则	Empty arrays and collections should be returned instead of null

规则描述	Returning null instead of an actual array or collection forces callers of the method to explicitly test for nullity, making them more complex and less readable. Moreover, in many cases, null is used as a synonym for empty. Noncompliant Code Example

```java
public static List<Result> getResults() {
  return null;                     // Noncompliant
}

public static Result[] getResults() {
  return null;                     // Noncompliant
}

public static void main(String[] args) {
  Result[] results = getResults();

  if (results != null) {              // Nullity test required to prevent NPE
    for (Result result: results) {
      /* ... */
    }
  }
}
```

 Compliant Solution

```java
public static List<Result> getResults() {
  return Collections.emptyList();        // Compliant
}

public static Result[] getResults() {
  return new Result[0];
}

public static void main(String[] args) {
  for (Result result: getResults()) {
    /* ... */
  }
}
```

 See

    CERT, MSC19-C. - For functions that return an array, prefer returning an empty array over a null value
    CERT, MET55-J. - Return an empty array or collection instead of a null value for methods that return an array or collection

文件名称	违规行
UUIDUtil.java	21

规则	"Random" objects should be reused

规则描述	Creating a new Random object each time a random value is needed is inefficient and may produce numbers which are not random depending on the JDK. For better efficiency and randomness, create a single Random , then store, and reuse it. The Random() constructor tries to set the seed with a distinct value every time. However there is no guarantee that the seed will be random or even uniformly distributed. Some JDK will use the current time as seed, which makes the generated numbers not random at all. This rule finds cases where a new Random is created each time a method is invoked and assigned to a local random variable. Noncompliant Code Example

```
public void doSomethingCommon() {
  Random rand = new Random();  // Noncompliant; new instance created with each invocation
  int rValue = rand.nextInt();
  //...
```

Compliant Solution

```
private Random rand = SecureRandom.getInstanceStrong();  // SecureRandom is preferred to Random

public void doSomethingCommon() {
  int rValue = this.rand.nextInt();
  //...
```

Exceptions
A class which uses a Random in its constructor or in a static main function and nowhere else will be ignored by this rule.
See

   OWASP Top 10 2017 Category A6 - Security Misconfiguration

文件名称	违规行
ShortUrlController.java	44

规则	Labels should not be used

规则描述	Labels are not commonly used, and many developers do not understand how they work. Moreover, their usage makes the control flow harder to follow, which reduces the code's readability. Noncompliant Code Example

```
myLabel: {
  let x = doSomething();
  if (x > 0) {
    break myLabel;
  }
  doSomethingElse();
}
```

 Compliant Solution

```
let x = doSomething();
if (x <= 0) {
  doSomethingElse();
}
```


文件名称	违规行
prettify.js	705

规则	A conditionally executed single line should be denoted by indentation
规则描述	In the absence of enclosing curly braces, the line immediately after a conditional is the one that is conditionally executed. By both convention and good practice, such lines are indented. In the absence of both curly braces and indentation the intent of the original programmer is entirely unclear and perhaps not actually what is executed. Additionally, such code is highly likely to be confusing to maintainers. Noncompliant Code Example

```
if (condition)  // Noncompliant
doTheThing();

doTheOtherThing();
somethingElseEntirely();

foo();
```

 Compliant Solution

```
if (condition)
  doTheThing();

doTheOtherThing();
somethingElseEntirely();

foo();
```

文件名称	违规行

UserService.java	45

规则	Utility classes should not have public constructors

规则描述	Utility classes, which are collections of static members, are not meant to be instantiated. Even abstract utility classes, which can be extended, should not have public constructors. Java adds an implicit public constructor to every class which does not define at least one explicitly. Hence, at least one non-public constructor should be defined. Noncompliant Code Example
	
	class StringUtils { // Noncompliant
	
	  public static String concatenate(String s1, String s2) {
	    return s1 + s2;
	  }
	
	}
	
	 Compliant Solution
	
	class StringUtils { // Compliant
	
	  private StringUtils() {
	    throw new IllegalStateException("Utility class");
	  }
	
	  public static String concatenate(String s1, String s2) {
	    return s1 + s2;
	  }
	
	}
	
	 Exceptions
	 When class contains public static void main(String[] args) method it is not considered as utility class and will be ignored by this rule.

文件名称	违规行
UUIDUtil.java	5

规则	Unused "private" methods should be removed

规则描述	private methods that are never executed are dead code: unnecessary, inoperative code that should be removed. Cleaning out dead code
	decreases the size of the maintained codebase, making it easier to understand the program and preventing bugs from being introduced.
	 Note that this rule does not take reflection into account, which means that issues will be raised on private methods that are only accessed using the reflection API.
	 Noncompliant Code Example
	
	public class Foo implements Serializable
	{
	  private Foo(){} //Compliant, private empty constructor intentionally used to prevent any direct instantiation of a class.
	  public static void doSomething(){
	    Foo foo = new Foo();
	    ...
	  }
	  private void unusedPrivateMethod(){...}
	  private void writeObject(ObjectOutputStream s){...} //Compliant, relates to the java serialization mechanism
	  private void readObject(ObjectInputStream in){...} //Compliant, relates to the java serialization mechanism
	}
	
	 Compliant Solution
	
	public class Foo implements Serializable
	{
	  private Foo(){} //Compliant, private empty constructor intentionally used to prevent any direct instantiation of a class.
	  public static void doSomething(){
	    Foo foo = new Foo();
	    ...
	  }
	
	  private void writeObject(ObjectOutputStream s){...} //Compliant, relates to the java serialization mechanism
	
	  private void readObject(ObjectInputStream in){...} //Compliant, relates to the java serialization mechanism
	}
	
	 Exceptions
	 This rule doesn't raise any issue on annotated methods.

文件名称	违规行
ShortUrlController.java	43

1.4. 质量配置

质量配置	css:Sonar way Bug:15 坏味道:8	
规则	类型	违规级别
Single line comment syntax should not be used	Bug	阻断
Color definitions should be valid	Bug	阻断

CSS properties should be valid	Bug	阻断
Units should be valid	Bug	阻断
Shorthand properties that override related longhand properties should be avoided	Bug	严重
"linear-gradient" directions should be valid	Bug	严重
Expressions within "calc" should be valid	Bug	严重
Selectors should be known	Bug	严重
"at-rules" should be valid	Bug	主要
Media features should be valid	Bug	主要
Font declarations should contain at least one generic font family	Bug	主要
"!important" should not be used on "keyframes"	Bug	主要
Pseudo-element selectors should be valid	Bug	主要
Properties should not be duplicated	Bug	主要
Pseudo-class selectors should be valid	Bug	主要
Selectors should not be duplicated	坏味道	主要
CSS files should not be empty	坏味道	主要
Duplicated font names should be removed	坏味道	主要
Strings should not contain new lines	坏味道	主要
Empty blocks should be removed	坏味道	主要
Multi-line comments should not be empty	坏味道	次要
Extra semicolons should be removed	坏味道	次要
Duplicate imports should be removed	坏味道	次要

质量配置	java:Sonar way　Bug:140　漏洞:27　坏味道:256	
规则	类型	违规级别
Methods should not call same-class methods with incompatible "@Transactional" values	Bug	阻断
Methods "wait(...)", "notify()" and "notifyAll()" should not be called on Thread instances	Bug	阻断
Files opened in append mode should not be used with ObjectOutputStream	Bug	阻断
"PreparedStatement" and "ResultSet" methods should be called with valid indices	Bug	阻断
"wait(...)" should be used instead of "Thread.sleep(...)" when a lock is held	Bug	阻断
Printf-style format strings should not lead to unexpected behavior at runtime	Bug	阻断
"@SpringBootApplication" and "@ComponentScan" should not be used in the default package	Bug	阻断
"@Controller" classes that use "@SessionAttributes" must call "setComplete" on their "SessionStatus" objects	Bug	阻断
Loops should not be infinite	Bug	阻断
"wait" should not be called when multiple locks are held	Bug	阻断

Double-checked locking should not be used	Bug	阻断
Resources should be closed	Bug	阻断
Regular expressions should be syntactically valid	Bug	严重
Locks should be released	Bug	严重
Jump statements should not occur in "finally" blocks	Bug	严重
"Random" objects should be reused	Bug	严重
"super.finalize()" should be called at the end of "Object.finalize()" implementations	Bug	严重
Dependencies should not have "system" scope	Bug	严重
Assertions comparing incompatible types should not be made	Bug	严重
Assertion methods should not be used within the try block of a try-catch catching an Error	Bug	严重
The signature of "finalize()" should match that of "Object.finalize()"	Bug	严重
Only one method invocation is expected when testing checked exceptions	Bug	严重
"runFinalizersOnExit" should not be called	Bug	严重
Regex boundaries should not be used in a way that can never be matched	Bug	严重
"ScheduledThreadPoolExecutor" should not have 0 core threads	Bug	严重
Regex patterns following a possessive quantifier should not always fail	Bug	严重
Hibernate should not update database schemas	Bug	严重
Zero should not be a possible denominator	Bug	严重
Back references in regular expressions should only refer to capturing groups that are matched before the reference	Bug	严重
Regex lookahead assertions should not be contradictory	Bug	严重
JUnit5 inner test classes should be annotated with @Nested	Bug	严重
Map "computeIfAbsent()" and "computeIfPresent()" should not be used to add "null" values.	Bug	严重
Getters and setters should access the expected fields	Bug	严重
"toString()" and "clone()" methods should not return null	Bug	主要
Servlets should not have mutable instance fields	Bug	主要
Value-based classes should not be used for locking	Bug	主要
Alternatives in regular expressions should be grouped when used with anchors	Bug	主要
Regex alternatives should not be redundant	Bug	主要
Conditionally executed code should be reachable	Bug	主要
Overrides should match their parent class methods in synchronization	Bug	主要

"DefaultMessageListenerContainer" instances should not drop messages during restarts	Bug	主要
Reflection should not be used to check non-runtime annotations	Bug	主要
"SingleConnectionFactory" instances should be set to "reconnectOnException"	Bug	主要
"hashCode" and "toString" should not be called on array instances	Bug	主要
Collections should not be passed as arguments to their own methods	Bug	主要
Case insensitive Unicode regular expressions should enable the "UNICODE_CASE" flag	Bug	主要
"BigDecimal(double)" should not be used	Bug	主要
Non-public methods should not be "@Transactional"	Bug	主要
Assertions should not compare an object to itself	Bug	主要
Invalid "Date" values should not be used	Bug	主要
Non-serializable classes should not be written	Bug	主要
Blocks should be synchronized on "private final" fields	Bug	主要
Unicode Grapheme Clusters should be avoided inside regex character classes	Bug	主要
Optional value should only be accessed after calling isPresent()	Bug	主要
AssertJ configuration should be applied	Bug	主要
Return values from functions without side effects should not be ignored	Bug	主要
".equals()" should not be used to test the values of "Atomic" classes	Bug	主要
"notifyAll" should be used	Bug	主要
Non-serializable objects should not be stored in "HttpSession" objects	Bug	主要
AssertJ methods setting the assertion context should come before an assertion	Bug	主要
The Object.finalize() method should not be called	Bug	主要
Assertions should not be used in production code	Bug	主要
Tests method should not be annotated with competing annotations	Bug	主要
InputSteam.read() implementation should not return a signed byte	Bug	主要
"InterruptedException" should not be ignored	Bug	主要
Silly equality checks should not be made	Bug	主要
Dissimilar primitive wrappers should not be used with the ternary operator without explicit casting	Bug	主要
"wait", "notify" and "notifyAll" should only be called when a lock is obviously held on an object	Bug	主要
"Double.longBitsToDouble" should not be used for "int"	Bug	主要
Regular expressions should not overflow the stack	Bug	主要

Values should not be uselessly incremented	Bug	主要
Silly String operations should not be made	Bug	主要
Null pointers should not be dereferenced	Bug	主要
Expressions used in "assert" should not produce side effects	Bug	主要
Classes extending java.lang.Thread should override the "run" method	Bug	主要
Loop conditions should be true at least once	Bug	主要
A "for" loop update clause should move the counter in the right direction	Bug	主要
Intermediate Stream methods should not be left unused	Bug	主要
Variables should not be self-assigned	Bug	主要
Consumed Stream pipelines should not be reused	Bug	主要
Inappropriate regular expressions should not be used	Bug	主要
"=+" should not be used instead of "+="	Bug	主要
Loops with at most one iteration should be refactored	Bug	主要
Classes should not be compared by name	Bug	主要
Identical expressions should not be used on both sides of a binary operator	Bug	主要
JUnit5 test classes and methods should not be silently ignored	Bug	主要
"Thread.run()" should not be called directly	Bug	主要
"null" should not be used with "Optional"	Bug	主要
"read" and "readLine" return values should be used	Bug	主要
Strings and Boxed types should be compared using "equals()"	Bug	主要
Methods should not be named "tostring", "hashcode" or "equal"	Bug	主要
Non-thread-safe fields should not be static	Bug	主要
Getters and setters should be synchronized in pairs	Bug	主要
Unary prefix operators should not be repeated	Bug	主要
DateTimeFormatters should not use mismatched year and week numbers	Bug	主要
"StringBuilder" and "StringBuffer" should not be instantiated with a character	Bug	主要
Week Year ("YYYY") should not be used for date formatting	Bug	主要
"equals" method overrides should accept "Object" parameters	Bug	主要
Exceptions should not be created without being thrown	Bug	主要
Collection sizes and array length comparisons should make sense	Bug	主要
"ThreadLocal" variables should be cleaned up when no longer used	Bug	主要

Synchronization should not be done on instances of value-based classes	Bug	主要
Related "if/else if" statements should not have the same condition	Bug	主要
All branches in a conditional structure should not have exactly the same implementation	Bug	主要
The regex escape sequence \cX should only be used with characters in the @-_ range	Bug	主要
"Iterator.hasNext()" should not call "Iterator.next()"	Bug	主要
"String" calls should not go beyond their bounds	Bug	主要
Raw byte values should not be used in bitwise operations in combination with shifts	Bug	主要
Custom serialization method signatures should meet requirements	Bug	主要
"Externalizable" classes should have no-arguments constructors	Bug	主要
"iterator" should not return "this"	Bug	主要
Child class methods named for parent class methods should be overrides	Bug	主要
Inappropriate "Collection" calls should not be made	Bug	主要
"compareTo" should not be overloaded	Bug	主要
AssertJ assertions with "Consumer" arguments should contain assertion inside consumers	Bug	主要
"volatile" variables should not be used with compound operators	Bug	主要
Map values should not be replaced unconditionally	Bug	主要
"getClass" should not be used for synchronization	Bug	主要
Assignment of lazy-initialized members should be the last step with double-checked locking	Bug	主要
Min and max used in combination should not always return the same value	Bug	主要
"compareTo" results should not be checked for specific values	Bug	次要
Regex repetition pattern's body should not match the empty String	Bug	次要
AssertJ assertions "allMatch" and "doesNotContains" should also test for emptiness	Bug	次要
Double Brace Initialization should not be used	Bug	次要
Boxing and unboxing should not be immediately reversed	Bug	次要
"Iterator.next()" methods should throw "NoSuchElementException"	Bug	次要
"@NonNull" values should not be set to null	Bug	次要
Neither "Math.abs" nor negation should be used on numbers that could be "MIN_VALUE"	Bug	次要
The value returned from a stream read should be checked	Bug	次要

Method parameters, caught exceptions and foreach variables' initial values should not be ignored	Bug	次要
"equals(Object obj)" and "hashCode()" should be overridden in pairs	Bug	次要
"Serializable" inner classes of non-serializable classes should be "static"	Bug	次要
Math operands should be cast before assignment	Bug	次要
Ints and longs should not be shifted by zero or more than their number of bits-1	Bug	次要
"compareTo" should not return "Integer.MIN_VALUE"	Bug	次要
The non-serializable super class of a "Serializable" class should have a no-argument constructor	Bug	次要
"toArray" should be passed an array of the proper type	Bug	次要
Non-primitive fields should not be "volatile"	Bug	次要
"equals(Object obj)" should test argument type	Bug	次要
Return values should not be ignored when they contain the operation status code	Bug	次要
A secure password should be used when connecting to a database	漏洞	阻断
XML parsers should not be vulnerable to XXE attacks	漏洞	阻断
Default EJB interceptors should be declared in "ejb-jar.xml"	漏洞	阻断
Struts validation forms should have unique names	漏洞	阻断
Cipher Block Chaining IV's should be unpredictable	漏洞	严重
Persistent entities should not be used as arguments of "@RequestMapping" methods	漏洞	严重
Defined filters should be used	漏洞	严重
JWT should be signed and verified with strong cipher algorithms	漏洞	严重
Cipher algorithms should be robust	漏洞	严重
Encryption algorithms should be used with secure mode and padding scheme	漏洞	严重
A new session should be created during user authentication	漏洞	严重
Weak SSL/TLS protocols should not be used	漏洞	严重
Cryptographic keys should be robust	漏洞	严重
"HttpServletRequest.getRequestedSessionId()" should not be used	漏洞	严重
LDAP connections should be authenticated	漏洞	严重
Server hostnames should be verified during SSL/TLS connections	漏洞	严重
"HttpSecurity" URL patterns should be correctly ordered	漏洞	严重
Basic authentication should not be used	漏洞	严重

Server certificates should be verified during SSL/TLS connections	漏洞	严重
Passwords should not be stored in plain-text or with a fast hashing algorithm	漏洞	严重
"SecureRandom" seeds should not be predictable	漏洞	严重
Insecure temporary file creation methods should not be used	漏洞	严重
Hashes should include an unpredictable salt	漏洞	严重
Authorizations should be based on strong decisions	漏洞	主要
OpenSAML2 should be configured to prevent authentication bypass	漏洞	主要
"ActiveMQConnectionFactory" should not be vulnerable to malicious code deserialization	漏洞	次要
Exceptions should not be thrown from servlet methods	漏洞	次要
Tests should include assertions	坏味道	阻断
Child class fields should not shadow parent class fields	坏味道	阻断
Assertions should be complete	坏味道	阻断
"clone" should not be overridden	坏味道	阻断
"switch" statements should not contain non-case labels	坏味道	阻断
Methods returns should not be invariant	坏味道	阻断
Silly bit operations should not be performed	坏味道	阻断
Switch cases should end with an unconditional "break" statement	坏味道	阻断
Methods and field names should not be the same or differ only by capitalization	坏味道	阻断
JUnit test cases should call super methods	坏味道	阻断
TestCases should contain tests	坏味道	阻断
"ThreadGroup" should not be used	坏味道	阻断
Future keywords should not be used as names	坏味道	阻断
Short-circuit logic should be used in boolean contexts	坏味道	阻断
"default" clauses should be last	坏味道	严重
Whitespace and control characters in literals should be explicit	坏味道	严重
IllegalMonitorStateException should not be caught	坏味道	严重
Cognitive Complexity of methods should not be too high	坏味道	严重
The Object.finalize() method should not be overridden	坏味道	严重
Package declaration should match source file directory	坏味道	严重
Null should not be returned from a "Boolean" method	坏味道	严重
String offset-based methods should be preferred for finding substrings from offsets	坏味道	严重

Instance methods should not write to "static" fields	坏味道	严重
"indexOf" checks should not be for positive numbers	坏味道	严重
Factory method injection should be used in "@Configuration" classes	坏味道	严重
Empty lines should not be tested with regex MULTILINE flag	坏味道	严重
Mocking all non-private methods of a class should be avoided	坏味道	严重
"Object.finalize()" should remain protected (versus public) when overriding	坏味道	严重
"Cloneables" should implement "clone"	坏味道	严重
"Object.wait(...)" and "Condition.await(...)" should be called inside a "while" loop	坏味道	严重
Methods should not be empty	坏味道	严重
"equals" method parameters should not be marked "@Nonnull"	坏味道	严重
Classes should not access their own subclasses during initialization	坏味道	严重
Exceptions should not be thrown in finally blocks	坏味道	严重
Method overrides should not change contracts	坏味道	严重
"for" loop increment clauses should modify the loops' counters	坏味道	严重
Constants should not be defined in interfaces	坏味道	严重
Generic wildcard types should not be used in return types	坏味道	严重
Execution of the Garbage Collector should be triggered only by the JVM	坏味道	严重
Derived exceptions should not hide their parents' catch blocks	坏味道	严重
Methods setUp() and tearDown() should be correctly annotated starting with JUnit4	坏味道	严重
Conditionals should start on new lines	坏味道	严重
A conditionally executed single line should be denoted by indentation	坏味道	严重
Class members annotated with "@VisibleForTesting" should not be accessed from production code	坏味道	严重
Fields in a "Serializable" class should either be transient or serializable	坏味道	严重
"switch" statements should have "default" clauses	坏味道	严重
JUnit assertions should not be used in "run" methods	坏味道	严重
"readResolve" methods should be inheritable	坏味道	严重
Constant names should comply with a naming convention	坏味道	严重
"static" base class members should not be accessed via derived types	坏味道	严重
String literals should not be duplicated	坏味道	严重

Class names should not shadow interfaces or superclasses	坏味道	严重
"String#replace" should be preferred to "String#replaceAll"	坏味道	严重
Try-with-resources should be used	坏味道	严重
Source files should not have any duplicated blocks	坏味道	主要
Regexes containing characters subject to normalization should use the CANON_EQ flag	坏味道	主要
Boolean expressions should not be gratuitous	坏味道	主要
Similar tests should be grouped in a single Parameterized test	坏味道	主要
Track uses of "FIXME" tags	坏味道	主要
Tests should be stable	坏味道	主要
"@Deprecated" code marked for removal should never be used	坏味道	主要
Parameters should be passed in the correct order	坏味道	主要
Unused "private" methods should be removed	坏味道	主要
"ResultSet.isLast()" should not be used	坏味道	主要
"URL.hashCode" and "URL.equals" should be avoided	坏味道	主要
Names of regular expressions named groups should be used	坏味道	主要
Try-catch blocks should not be nested	坏味道	主要
Character classes in regular expressions should not contain the same character twice	坏味道	主要
Synchronized classes Vector, Hashtable, Stack and StringBuffer should not be used	坏味道	主要
Redundant pairs of parentheses should be removed	坏味道	主要
"Lock" objects should not be "synchronized"	坏味道	主要
Multiline blocks should be enclosed in curly braces	坏味道	主要
Classes with only "static" methods should not be instantiated	坏味道	主要
Labels should not be used	坏味道	主要
"static" members should be accessed statically	坏味道	主要
Utility classes should not have public constructors	坏味道	主要
Unused type parameters should be removed	坏味道	主要
Assertion arguments should be passed in the correct order	坏味道	主要
Local variables should not shadow class fields	坏味道	主要
AssertJ "assertThatThrownBy" should not be used alone	坏味道	主要
"switch" statements should not have too many "case" clauses	坏味道	主要
Regular expressions should not be too complicated	坏味道	主要
Deprecated elements should have both the annotation and the Javadoc tag	坏味道	主要

Assignments should not be made from within sub-expressions	坏味道	主要
Test methods should not contain too many assertions	坏味道	主要
Inner class calls to super class methods should be unambiguous	坏味道	主要
Ternary operators should not be nested	坏味道	主要
'List.remove()' should not be used in ascending 'for' loops	坏味道	主要
Exception testing via JUnit ExpectedException rule should not be mixed with other assertions	坏味道	主要
Only one method invocation is expected when testing runtime exceptions	坏味道	主要
Nullness of parameters should be guaranteed	坏味道	主要
Unused method parameters should be removed	坏味道	主要
Only static class initializers should be used	坏味道	主要
Vararg method arguments should not be confusing	坏味道	主要
Unused "private" fields should be removed	坏味道	主要
Collapsible "if" statements should be merged	坏味道	主要
Unused labels should be removed	坏味道	主要
JUnit assertTrue/assertFalse should be simplified to the corresponding dedicated assertion	坏味道	主要
Whitespace for text block indent should be consistent	坏味道	主要
Throwable and Error should not be caught	坏味道	主要
Printf-style format strings should be used correctly	坏味道	主要
"Integer.toHexString" should not be used to build hexadecimal strings	坏味道	主要
Constructors of an "abstract" class should not be declared "public"	坏味道	主要
Constructors should not be used to instantiate "String", "BigInteger", "BigDecimal" and primitive-wrapper classes	坏味道	主要
Enumeration should not be implemented	坏味道	主要
Empty arrays and collections should be returned instead of null	坏味道	主要
Objects should not be created only to "getClass"	坏味道	主要
Primitives should not be boxed just for "String" conversion	坏味道	主要
Exceptions should be either logged or rethrown but not both	坏味道	主要
"@Override" should be used on overriding and implementing methods	坏味道	主要
"Preconditions" and logging arguments should not require evaluation	坏味道	主要
"entrySet()" should be iterated when both the key and value are needed	坏味道	主要
"Class.forName()" should not load JDBC 4.0+ drivers	坏味道	主要

Two branches in a conditional structure should not have exactly the same implementation	坏味道	主要
"Map.get" and value test should be replaced with single method call	坏味道	主要
"Arrays.stream" should be used for primitive arrays	坏味道	主要
"@RequestMapping" methods should be "public"	坏味道	主要
Non-constructor methods should not have the same name as the enclosing class	坏味道	主要
"readObject" should not be "synchronized"	坏味道	主要
"Threads" should not be used where "Runnables" are expected	坏味道	主要
Java features should be preferred to Guava	坏味道	主要
"Stream.peek" should be used with caution	坏味道	主要
Unused "private" classes should be removed	坏味道	主要
Raw types should not be used	坏味道	主要
A field should not duplicate the name of its containing class	坏味道	主要
Single-character alternations in regular expressions should be replaced with character classes	坏味道	主要
String multiline concatenation should be replaced with Text Blocks	坏味道	主要
Reluctant quantifiers in regular expressions should be followed by an expression that can't match the empty string	坏味道	主要
Unused assignments should be removed	坏味道	主要
"DateUtils.truncate" from Apache Commons Lang library should not be used	坏味道	主要
"Thread.sleep" should not be used in tests	坏味道	主要
Sections of code should not be commented out	坏味道	主要
"for" loop stop conditions should be invariant	坏味道	主要
JUnit4 @Ignored and JUnit5 @Disabled annotations should be used to disable tests and should provide a rationale	坏味道	主要
Anonymous inner classes containing only one method should become lambdas	坏味道	主要
"Object.wait(...)" should never be called on objects that implement "java.util.concurrent.locks.Condition"	坏味道	主要
Inheritance tree of classes should not be too deep	坏味道	主要
Generic exceptions should never be thrown	坏味道	主要
Silly math should not be performed	坏味道	主要
Standard outputs should not be used directly to log anything	坏味道	主要
Methods should not have too many parameters	坏味道	主要
Nested blocks of code should not be left empty	坏味道	主要
"writeObject" should not be the only "synchronized" code in a class	坏味道	主要

Classes named like "Exception" should extend "Exception" or a subclass	坏味道	主要
Reflection should not be used to increase accessibility of classes, methods, or fields	坏味道	主要
Static fields should not be updated in constructors	坏味道	主要
Exception types should not be tested using "instanceof" in catch blocks	坏味道	主要
Classes from "sun.*" packages should not be used	坏味道	主要
"java.nio.Files#delete" should be preferred	坏味道	主要
Assignments should not be redundant	坏味道	主要
"else" statements should be clearly matched with an "if"	坏味道	主要
Operator "instanceof" should be used instead of "A.class.isInstance()"	坏味道	主要
Methods should not have identical implementations	坏味道	主要
Restricted Identifiers should not be used as Identifiers	坏味道	主要
Asserts should not be used to check the parameters of a public method	坏味道	主要
Consecutive AssertJ "assertThat" statements should be chained	坏味道	次要
"throws" declarations should not be superfluous	坏味道	次要
Character classes should be preferred over reluctant quantifiers in regular expressions	坏味道	次要
A "while" loop should be used instead of a "for" loop	坏味道	次要
"Collections.EMPTY_LIST", "EMPTY_MAP", and "EMPTY_SET" should not be used	坏味道	次要
Chained AssertJ assertions should be simplified to the corresponding dedicated assertion	坏味道	次要
Empty statements should be removed	坏味道	次要
Return of boolean expressions should not be wrapped into an "if-then-else" statement	坏味道	次要
Loggers should be named for their enclosing classes	坏味道	次要
Local variables should not be declared and then immediately returned or thrown	坏味道	次要
Boolean literals should not be redundant	坏味道	次要
Modifiers should be declared in the correct order	坏味道	次要
Deprecated "${pom}" properties should not be used	坏味道	次要
Unnecessary imports should be removed	坏味道	次要
Unused local variables should be removed	坏味道	次要
Exception testing via JUnit @Test annotation should be avoided	坏味道	次要
Catches should be combined	坏味道	次要
Mutable fields should not be "public static"	坏味道	次要
Null checks should not be used with "instanceof"	坏味道	次要

Boxed "Boolean" should be avoided in boolean expressions	坏味道	次要
Methods of "Random" that return floating point values should not be used in random integer generation	坏味道	次要
"@CheckForNull" or "@Nullable" should not be used on primitive types	坏味道	次要
Public constants and fields initialized at declaration should be "static final" rather than merely "final"	坏味道	次要
Simple string literal should be used for single line strings	坏味道	次要
Escape sequences should not be used in text blocks	坏味道	次要
Overriding methods should do more than simply call the same method in the super class	坏味道	次要
Static non-final field names should comply with a naming convention	坏味道	次要
Classes that override "clone" should be "Cloneable" and call "super.clone()"	坏味道	次要
Primitive wrappers should not be instantiated only for "toString" or "compareTo" calls	坏味道	次要
Case insensitive string comparisons should be made without intermediate upper or lower casing	坏味道	次要
Test classes should comply with a naming convention	坏味道	次要
Collection.isEmpty() should be used to test for emptiness	坏味道	次要
String.valueOf() should not be appended to a String	坏味道	次要
Exception classes should be immutable	坏味道	次要
Parsing should be used to convert "Strings" to primitives	坏味道	次要
"switch" statements should have at least 3 "case" clauses	坏味道	次要
Multiple variables should not be declared on the same line	坏味道	次要
"read(byte[],int,int)" should be overridden	坏味道	次要
"@Deprecated" code should not be used	坏味道	次要
Strings should not be concatenated using '+' in a loop	坏味道	次要
Maps with keys that are enum values should be replaced with EnumMap	坏味道	次要
"catch" clauses should do more than rethrow	坏味道	次要
Nested "enum"s should not be declared static	坏味道	次要
"equals(Object obj)" should be overridden along with the "compareTo(T obj)" method	坏味道	次要
Private fields only used as local variables in methods should become local variables	坏味道	次要
Arrays should not be created for varargs parameters	坏味道	次要

Class variable fields should not have public accessibility	坏味道	次要
Methods should not return constants	坏味道	次要
The default unnamed package should not be used	坏味道	次要
Type parameters should not shadow other type parameters	坏味道	次要
Declarations should use Java collection interfaces such as "List" rather than specific implementation classes such as "LinkedList"	坏味道	次要
"public static" fields should be constant	坏味道	次要
An iteration on a Collection should be performed on the type handled by the Collection	坏味道	次要
"StandardCharsets" constants should be preferred	坏味道	次要
Jump statements should not be redundant	坏味道	次要
"close()" calls should not be redundant	坏味道	次要
Boolean checks should not be inverted	坏味道	次要
"indexOf" checks should use a start position	坏味道	次要
Redundant casts should not be used	坏味道	次要
"ThreadLocal.withInitial" should be preferred	坏味道	次要
Abstract classes without fields should be converted to interfaces	坏味道	次要
"toString()" should never be called on a String object	坏味道	次要
Lambdas should be replaced with method references	坏味道	次要
Parentheses should be removed from a single lambda input parameter when its type is inferred	坏味道	次要
Call to Mockito method "verify", "when" or "given" should be simplified	坏味道	次要
JUnit rules should be used	坏味道	次要
Annotation repetitions should not be wrapped	坏味道	次要
Loops should not contain more than a single "break" or "continue" statement	坏味道	次要
Lambdas containing only one statement should not nest this statement in a block	坏味道	次要
Abstract methods should not be redundant	坏味道	次要
"private" methods called only by inner classes should be moved to those classes	坏味道	次要
Fields in non-serializable classes should not be "transient"	坏味道	次要
Composed "@RequestMapping" variants should be preferred	坏味道	次要
Package names should comply with a naming convention	坏味道	次要
Interface names should comply with a naming convention	坏味道	次要
Field names should comply with a naming convention	坏味道	次要

Local variable and method parameter names should comply with a naming convention	坏味道	次要
Type parameter names should comply with a naming convention	坏味道	次要
"write(byte[],int,int)" should be overridden	坏味道	次要
Nested code blocks should not be used	坏味道	次要
Array designators "[]" should be on the type, not the variable	坏味道	次要
URIs should not be hardcoded	坏味道	次要
"finalize" should not set fields to "null"	坏味道	次要
Arrays should not be copied using loops	坏味道	次要
Array designators "[]" should be located after the type in method signatures	坏味道	次要
Subclasses that add fields should override "equals"	坏味道	次要
Class names should comply with a naming convention	坏味道	次要
Method names should comply with a naming convention	坏味道	次要
The diamond operator ("<>") should be used	坏味道	次要
Switch arrow labels should not use redundant keywords	坏味道	次要
Text blocks should not be used in complex expressions	坏味道	次要
"Stream" call chains should be simplified when possible	坏味道	次要
Functional Interfaces should be as specialised as possible	坏味道	次要
"enum" fields should not be publicly mutable	坏味道	次要
Packages containing only "package-info.java" should be removed	坏味道	次要
Classes should not be empty	坏味道	次要
Track uses of "TODO" tags	坏味道	提示
Deprecated code should be removed	坏味道	提示
JUnit5 test classes and methods should have default package visibility	坏味道	提示
Comma-separated labels should be used in Switch with colon case	坏味道	提示
Local-Variable Type Inference should be used	坏味道	提示

质量配置	js:Sonar way Bug:44 漏洞:11 坏味道:74	
规则	**类型**	**违规级别**
Callbacks of array methods should have return statements	Bug	阻断
Loops should not be infinite	Bug	阻断
Jump statements should not occur in "finally" blocks	Bug	严重

"in" should not be used with primitive types	Bug	严重
Function calls should not pass extra arguments	Bug	严重
"Symbol" should not be used as a constructor	Bug	严重
A compare function should be provided when using "Array.prototype.sort()"	Bug	严重
Results of "in" and "instanceof" should be negated rather than operands	Bug	严重
"super()" should be invoked appropriately	Bug	严重
Getters and setters should access the expected fields	Bug	严重
Destructuring patterns should not be empty	Bug	主要
Return values from functions without side effects should not be ignored	Bug	主要
"NaN" should not be used in comparisons	Bug	主要
Generators should "yield" something	Bug	主要
Related "if/else if" statements should not have the same condition	Bug	主要
The output of functions that don't return anything should not be used	Bug	主要
All branches in a conditional structure should not have exactly the same implementation	Bug	主要
Values should not be uselessly incremented	Bug	主要
Special identifiers should not be bound or assigned	Bug	主要
Properties of variables with "null" or "undefined" values should not be accessed	Bug	主要
Objects should not be created to be dropped immediately without being used	Bug	主要
A "for" loop update clause should move the counter in the right direction	Bug	主要
Function argument names should be unique	Bug	主要
Variables should not be self-assigned	Bug	主要
Property names should not be duplicated within a class or object literal	Bug	主要
Non-empty statements should change control flow or have at least one side-effect	Bug	主要
Non-existent operators '=+', '=-' and '=!' should not be used	Bug	主要
"new" operators should be used with functions	Bug	主要
Loops with at most one iteration should be refactored	Bug	主要
All code should be reachable	Bug	主要
Identical expressions should not be used on both sides of a binary operator	Bug	主要
Bitwise operators should not be used in boolean contexts	Bug	主要
Strict equality operators should not be used with dissimilar types	Bug	主要
Setters should not return values	Bug	主要

Promise rejections should not be caught by 'try' block	Bug	主要
Comma and logical OR operators should not be used in switch cases	Bug	主要
Attempts should not be made to update "const" variables	Bug	主要
Collection elements should not be replaced unconditionally	Bug	主要
Errors should not be created without being thrown	Bug	主要
Collection sizes and array length comparisons should make sense	Bug	主要
"delete" should be used only with object properties	Bug	次要
Function parameters, caught exceptions and foreach variables' initial values should not be ignored	Bug	次要
"with" statements should not be used	Bug	次要
Empty collections should not be accessed or iterated	Bug	次要
XML parsers should not be vulnerable to XXE attacks	漏洞	阻断
File uploads should be restricted	漏洞	严重
Origins should be verified during cross-origin communications	漏洞	严重
JWT should be signed and verified with strong cipher algorithms	漏洞	严重
Cipher algorithms should be robust	漏洞	严重
Encryption algorithms should be used with secure mode and padding scheme	漏洞	严重
A new session should be created during user authentication	漏洞	严重
Weak SSL/TLS protocols should not be used	漏洞	严重
Cryptographic keys should be robust	漏洞	严重
Server hostnames should be verified during SSL/TLS connections	漏洞	严重
Server certificates should be verified during SSL/TLS connections	漏洞	严重
Variables should be declared explicitly	坏味道	阻断
Switch cases should end with an unconditional "break" statement	坏味道	阻断
"future reserved words" should not be used as identifiers	坏味道	阻断
"switch" statements should not contain non-case labels	坏味道	阻断
Function returns should not be invariant	坏味道	阻断
Octal values should not be used	坏味道	阻断
"default" clauses should be last	坏味道	严重
Cognitive Complexity of functions should not be too high	坏味道	严重
Functions should not be empty	坏味道	严重

"await" should only be used with promises	坏味道	严重
"for" loop increment clauses should modify the loops' counters	坏味道	严重
Loop counters should not be assigned to from within the loop body	坏味道	严重
Conditionals should start on new lines	坏味道	严重
A conditionally executed single line should be denoted by indentation	坏味道	严重
"void" should not be used	坏味道	严重
Equality operators should not be used in "for" loop termination conditions	坏味道	严重
Boolean expressions should not be gratuitous	坏味道	主要
Track uses of "FIXME" tags	坏味道	主要
Parameters should be passed in the correct order	坏味道	主要
Functions should not be called both with and without "new"	坏味道	主要
Comma operator should not be used	坏味道	主要
"arguments.caller" and "arguments.callee" should not be used	坏味道	主要
Template literals should not be nested	坏味道	主要
Redundant pairs of parentheses should be removed	坏味道	主要
Labels should not be used	坏味道	主要
Multiline blocks should be enclosed in curly braces	坏味道	主要
Variables should not be shadowed	坏味道	主要
"switch" statements should not have too many "case" clauses	坏味道	主要
Assignments should not be made from within sub-expressions	坏味道	主要
"indexOf" checks should not be for positive numbers	坏味道	主要
Shorthand promises should be used	坏味道	主要
Arguments to built-in functions should match documented types	坏味道	主要
Ternary operators should not be nested	坏味道	主要
Unused assignments should be removed	坏味道	主要
Functions should always return the same type	坏味道	主要
Array indexes should be numeric	坏味道	主要
Literals should not be thrown	坏味道	主要
Sections of code should not be commented out	坏味道	主要
Variables and functions should not be redeclared	坏味道	主要
Functions should not have too many parameters	坏味道	主要
Nested blocks of code should not be left empty	坏味道	主要
"delete" should not be used on arrays	坏味道	主要
Function parameters with default values should be last	坏味道	主要
Functions should not be defined inside loops	坏味道	主要

Two branches in a conditional structure should not have exactly the same implementation	坏味道	主要
Array-mutating methods should not be used misleadingly	坏味道	主要
Assignments should not be redundant	坏味道	主要
"in" should not be used on arrays	坏味道	主要
Functions should not have identical implementations	坏味道	主要
Only "while", "do", "for" and "switch" statements should be labelled	坏味道	主要
Sparse arrays should not be declared	坏味道	主要
Collection and array contents should be used	坏味道	主要
Boolean checks should not be inverted	坏味道	次要
Default export names and file names should match	坏味道	次要
A "while" loop should be used instead of a "for" loop	坏味道	次要
Function call arguments should not start on new lines	坏味道	次要
Extra semicolons should be removed	坏味道	次要
Boolean literals should not be used in comparisons	坏味道	次要
Return of boolean expressions should not be wrapped into an "if-then-else" statement	坏味道	次要
Local variables should not be declared and then immediately returned or thrown	坏味道	次要
Unnecessary imports should be removed	坏味道	次要
Unused local variables and functions should be removed	坏味道	次要
"await" should not be used redundantly	坏味道	次要
Wrapper objects should not be used for primitive types	坏味道	次要
"switch" statements should have at least 3 "case" clauses	坏味道	次要
"for of" should be used with Iterables	坏味道	次要
The global "this" object should not be used	坏味道	次要
Deprecated APIs should not be used	坏味道	次要
Multiline string literals should not be used	坏味道	次要
Class names should comply with a naming convention	坏味道	次要
"catch" clauses should do more than rethrow	坏味道	次要
Imports from the same modules should be merged	坏味道	次要
Jump statements should not be redundant	坏味道	次要
Track uses of "TODO" tags	坏味道	提示

质量配置	web:Sonar way　　Bug:17　坏味道:10

规则	类型	违规级别
Tables used for layout should not include semantic markup	Bug	严重
Table cells should reference their headers	Bug	严重
<script>...</script> elements should not be nested	Bug	主要
Tables should have headers	Bug	主要
"<title>" should be present in all pages	Bug	主要
"<html>" element should have a language attribute	Bug	主要
"<!DOCTYPE>" declarations should appear before "<html>" tags	Bug	主要
Elements deprecated in HTML5 should not be used	Bug	主要
"<th>" tags should have "id" or "scope" attributes	Bug	主要
"<frames>" should have a "title" attribute	Bug	次要
Server-side image maps ("ismap" attribute) should not be used	Bug	次要
"<table>" tags should have a description	Bug	次要
"<strong>" and "<em>" tags should be used	Bug	次要
Image, area and button with image tags should have an "alt" attribute	Bug	次要
Flash animations should be embedded using both "<object>" and "<embed>"	Bug	次要
"<fieldset>" tags should contain a "<legend>"	Bug	次要
" " and "<dt>" item tags should be in " ", " " or "<dl>" container tags	Bug	次要
Track uses of "FIXME" tags	坏味道	主要
"aria-label" or "aria-labelledby" attributes should be used to differentiate similar elements	坏味道	主要
HTML "<table>" should not be used for layout purposes	坏味道	主要
Videos should have subtitles	坏味道	主要
Attributes deprecated in HTML5 should not be used	坏味道	主要
Sections of code should not be commented out	坏味道	主要
Meta tags should not be used to refresh or redirect	坏味道	主要
Links should not directly target images	坏味道	主要
"<object>" tags should provide an alternative content	坏味道	次要
Track uses of "TODO" tags	坏味道	提示

质量配置	xml:Sonar way Bug:1 漏洞:2 坏味道:3	
规则	类型	违规级别

XML files containing a prolog header should start with "<?xml" characters	Bug	严重
Basic authentication should not be used	漏洞	严重
Restrict access to exported components with appropriate permissions	漏洞	主要
Track uses of "FIXME" tags	坏味道	主要
Sections of code should not be commented out	坏味道	主要
Track uses of "TODO" tags	坏味道	提示