Sonar Lint

Performing static Code Analysis using

==========

Lint Plugin of the Eclipse IDE

==========

plugin:: It is a patch software that provides the additional funcationalaties to the existing software => SonarLint plugin added to the Eclipse IDE gives local SonarQube to Eclipse IDE to analyze the code

Procedure to install sonarLint Plugin In Eclipse IDE (Now it has become SonarQube Plugin)

==========

=>help menu ---> eclipse market place ----> search for "SonarLint" ---> go ---> select "SonarLint 10.0" ----> next --> accept terms and conditions

---> next --> next---> finish ----> restart the Eclipse IDE

sonarlint

★ 3186

SonarLint 10.0

(in new versions plugin name SonarQube-<ver>)

SonarLint is an IDE extension that helps you detect and fix quality issu Java, JavaScript / TypeScript, PHP, Python, CSS and HTML. more info by SonarSource S.A, LGPL

java PHP javascript Python static analysis

Installs: 1.29M (9,561 last month)

Praticals of Static code Analysis

======

==========

step1) make sure that "sonarLint" plugin is installed

step2) enable the following options/windows in eclipse IDE

window menu ---> show view ---> select sonarLint ---> select the following options

Sonarlint

SonarLint Bindings

SonarLint Issue Locations

SonarLint On-The-Fly

SonarLint Report

note:: From eclipse 2024-06, the SonarLint plugin is changed to Sonar Qube plugin.

Two types of eclise plugins

eclipse's own plugins

eclipse

=>Given by eclipse itself

=> use help menu -->

third party plugins

=> third party companies

have given these plugins

=> use help menu ---> eclipse market place

install new software option option to install the plugins

eg:: GUI Builder plugin,

Lombok plugin and etc..

one

eg:: STS plugin, SonarLint and etc...

note:: Plugins installed in workspace will reflect to all the workspaces of a Eclipse IDE

What is Plugin?

Ans) Plugin is a patch software that provides additional functionalities on the top of existing software

eg1:: amazon fire stick attached to the normal TV makes the TV as Smart TV eg2:: STS Plugin added to the Eclipse IDE gives STS IDE features to Eclipse IDE

step3) Enable SonarLint console from console window

console window icon --> select "SonarLint Console"

note:: once the sonarLint plugin is installed all the existing projects will be linked sonarLint automatically

step4) Develop the Java Project having .java source files

TestApp1.java

```
//TestApp1.java
package com.nt.service;
import java.util.ArrayList;
import java.util.List;
public class TestApp1 {
public void m2() {
}
System.out.println("TestApp1.m2()");
public void m1() {
System.out.println("TestApp1.m1()");
String s1="";
List list=new ArrayList();
list.add(10); list.add(20);
list=null;
list.add(40);
}
public static void main(String[] args) {
System.out.println("TestApp1.main() (start)");
TestApp1 app1=new TestApp1(); app1.m1();
```

```
System.out.println("TestApp1. main() (end)");
}
}
```

step5) Perform static code analysis

right click on TestApp1.java ---> sonarlint ---> analyze

step6) observer the following tabs/windows

sonarLint console

SonarLint processing file /SonarQube Proj01/src/com/nt/service/TestApp1.java... Found 8 issue(s)

SonarLint - On the fly

How to install SonarLint/SonarQube Plugin to the Eclipse IDE with out using Eclipse market Place option?
Ans) use Install new Software option of help menu by specifying sonarlint plugin as shown below Help menu ---> install new software ---> add ---> name:: SonarLint ---> url:: https://eclipse-uc.sonarlint.org

3 items

Resource

Date

Description

6 minutes ago

Provide the parametrized type for this generic.

TestApp1.java

TestApp1.java

9 minutes ago

Replace this use of System.out by a logger.

10 minutes ago

A "NullPointerException" could be thrown; "list" is nullable here. [+2 locations]

TestApp1.java

TestApp1.java

10 minutes ago

Replace this use of System.out by a logger.

TestApp1.java

11 minutes ago

Remove this unused "$1" local variable.

11 minutes ago

Provide the parametrized type for this generic.

TestApp1.java

TestApp1.java

11 minutes ago

11 minutes ago

^ Replace this use of System.out by a logger.

Replace this use of System.out by a logger.

TestApp1.java

SonarLint Report

items

Resource

Date

TestApp1.java

4 minutes ago

TestApp1.java

7 minutes ago

TestApp1.java

8 minutes ago

TestApp1.java

8 minutes ago

TestApp1.java

9 minutes ago

TestApp1.java

9 minutes ago

TestApp1.java

9 minutes ago

TestApp1.java

9 minutes ago

SonarLint Rule Desscription

Why is this an issue?

Description

00000000

Provide the parametrized type for this generic.

Replace this use of System.out by a logger.

A "NullPointerException" could be thrown; "list" is nullable here. [+2 locations]

Replace this use of System.out by a logger.

Remove this unused "s1" local variable.

Provide the parametrized type for this generic. Replace this use of System.out by a logger.

Replace this use of System.out by a logger.

Naresh i Technologie

If a local variable 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 is used for.

Noncompliant code example

public int numberOfMinutes(int hours) {

int seconds = 0; // seconds is never used

return hours* 60;

Compliant solution

public int numberOfMinutes(int hours) {

return hours * 60;

note:: we can analyze the total project or separate or each .java file

step7) Modify the code according to sonarLint tool report suggestions and fix them

//TestApplication.java

```
package com.nt.client;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
public class TestApplication {
private static final Logger logger=Logger.getLogger("TestApplication");
public void m2() {
logger.fine("from m2()");
```

Z Meeting Chat

Pavan Wadbudhe to You (Direct M

noto Windows

-->select all checkboxes ---> next --> next ---> accept terms and conditions --> -->next ---> ... ---> restart the IDE

activate

How to deactivate one rule of sonarLint?

ans1) Go to SonarLint report tab --> right click

on the any report statement ---> deactivate the rule

Ans2) window menu ---->preferences ---> sonarLint ---> rules configuration ---> search for the rule(s) and select or deselect it [activate] [deactivate]

}

```
public void m1() {
logger.fine("from m1()");
m2();
ArrayList<Integer> list=new ArrayList<>();
list.add(10);list.add(20);list.add(30);
list.add(40);
logger.log(Level.FINE," data is :: {0}",list);
```

=>SonarQube does not allow commented code in our java source file but

the same SonarQube allows comments to explain the code

// logger.trace("Enduser experienced the generated output"); SelectTest.java few seconds ago

//establish the connection

---'F"ᒧ

This block of commented-out lines of code should be removed. [+2 locations]

Connection con=DriverManager.getConnection(props.getProperty("jdbc.url"),

note:: This comment is not a SonarQube Error

}

public static void main(String[] args) {

TestApplication app=new TestApplication(); app.m1();

}

}

The above is SonarLint error free code i.e There are no static Code Testing related errors

props.getProperty("jdbc.username"), props.getProperty("jdbc.password"));

JDBC App with SLF4J

====

(With out performing static code analysis)

====

//SelectTest.java

package com.nt.main;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class SelectTest {

private static final String GET_EMPS_QUERY="SELECT EMPNO1,ENAME,JOB,SAL, DEPTNO FROM EMP";

private static Logger logger = LoggerFactory.getLogger(SelectTest.class);

public static void main(String[] args) {

logger.debug("start of the main(-) method");

try(//establish the connection

Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "tiger");
//create PreparedStatement

//execute the Query

PreparedStatement ps=con.prepareStatement (GET_EMPS_QUERY);

ResultSet rs=ps.executeQuery();

```
X{
logger.info("Connection is established");
logger.debug("Prepared Statement obj is ready");
logger.debug("Select Query is executed and the ResultSet obj is generated");
}
//process the ResultSet object
while(rs.next()) {
}
System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4)+" "+rs.getInt(5));
logger.warn("for Salary rs.getFloat(-) is used rahter rs.getDouble(-)");
logger.debug("ResutSet of is Processed");
logger.info("Connection, PreparedStatement, ResultSet objs are closed automatically at end of try with Resource");
catch(SQLException se) {
logger.error("Known DB problem ::" +se.getMessage());
se.printStackTrace();
}
catch (Exception e) {
logger.error("Unknow Problem ::" + e.getMessage());
e.printStackTrace();
}
logger.debug("end of the main(-) method");
}//main
}//class
```

Right click on SelectTest.java file ----> sonarlint---> analyze

Resource

Date

SelectTest.java 1 minute ago

SelectTest.java 1 minute ago

SelectTest.java 1 minute ago

Solution code

==========

```
//SelectTest.java
package com.nt.main;
import java.io.FileInputStream;
import java.sql.Connection;
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Properties;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
public class SelectTest {
```

Description

Format specifiers should be used instead of string concatenation. Format specifiers should be used instead of string concatenation. Revoke and change this password, as it is compromised.

```
private static final String GET_EMPS_QUERY="SELECT EMPNO1,ENAME, JOB,SAL, DEPTNO FROM EMP";
private static Logger logger = LoggerFactory.getLogger(SelectTest.class);
private static Properties props;
static{
try(//load properties file
FileInputStream fis=new FileInputStream("log4j.properties");) {_ logger.debug("Properties file is loaded");
//write properties file content to java.util.Properties class obj
props=new Properties();
props.load(fis);
logger.info("Properties file content is loaded to java.util.Properties class obj");
}
catch(Exception e) {
logger.error("Problem in loading the properties file {}", e.getMessage());
// Here the DB url, db username and password details are gathered and stored in java.util.Properties class object to avoid their hard coding
}//static
public static void main(String[] args) {
logger.debug("start of the main(-) method");
try( //establish the connection
Connection con= DriverManager.getConnection(props.getProperty("db.url"),
props.getProperty("db.username"),
props.getProperty("db.password"));
//create PreparedStatement
PreparedStatement ps=con.prepareStatement (GET_EMPS_QUERY);
//execute the Query
ResultSet rs=ps.executeQuery();
X{
logger.info("Connection is established");
```

```
logger.debug("Prepared Statement obj is ready");
logger.debug("Select Query is executed and the ResultSet obj is generated");
//process the ResultSet object
while(rs.next()) {
System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4)+" "+rs.getInt(5));
}
logger.warn("for Salary rs.getFloat(-) is used rahter rs.getDouble(-)");
logger.debug("ResutSet of is Processed");
logger.info("Connection, PreparedStatement, ResultSet objs are closed automatically at end of try with Resource");
}
catch (SQLException se) {
logger.error("Known DB problem :: {}",se.getMessage());
se.printStackTrace();
}
catch(Exception e) {
logger.error("Unknow Problem :: {}",e.getMessage());
e.printStackTrace();
}
logger.debug("end of the main(-) method");
}//main
}//class
```

FAQS

====

note: We can deactivate the SonarQube rule from the SonarQube report window

by right click on the rule --> Deactivate Rule

b/w

Q) What is the difference static testing and dynamic testing?

Q) List out various tools for static testing and dynamic testing?

Q) What is the difference b/w functional testing and non-functional testing?

Q) what is the difference b/w Black box testing and White box testing?

Q) List out various Code Review tools or Static Code analysis tools?

Q) What are potential bugs that SonarQube can find out?

Q) What is security vulnerability Problem as part of Code Review process?

Q) How to work with SonarQube in Eclipse IDE?

rule

Q) How to deactivate certain using SonarQube?

Q) What is difference b/w SonarQube(Code review) and Jacoco (code Coverage tool) Tool?

Q) Explain the features of SonarQube ?

Q) How to install SonarQube plugin in the Eclipse IDE?

=> SonarQube performs Code Review / static Code Analysis activity where as Jacoco performs Test Code Coverage that means it checks whether the testcases that we have written are covering entire code for execution or not?