資訊人才軟體培育推廣計畫

Java安全程式設計

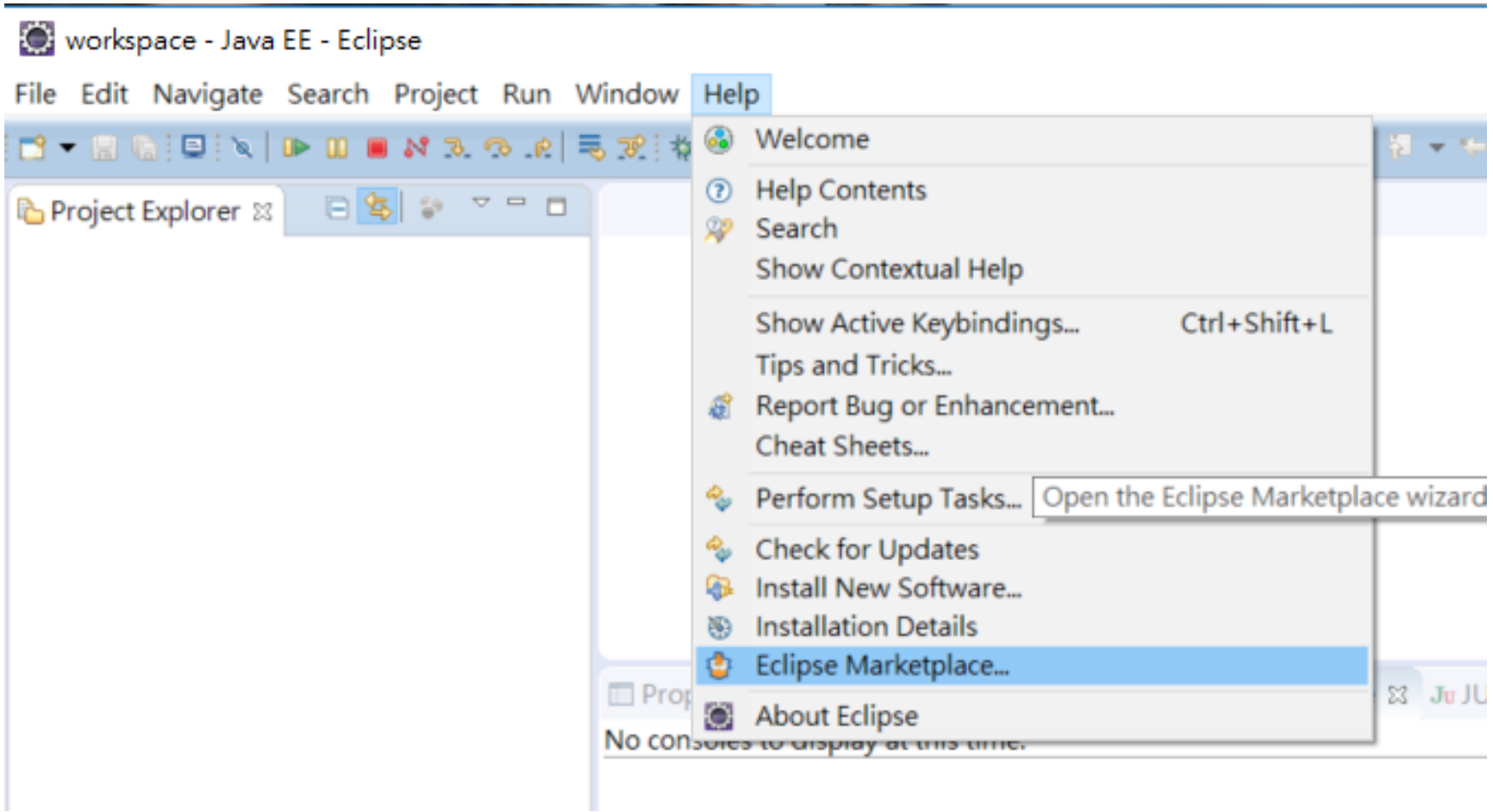
逢甲大學 資訊工程系
陳錫民 助理教授

2016年8月

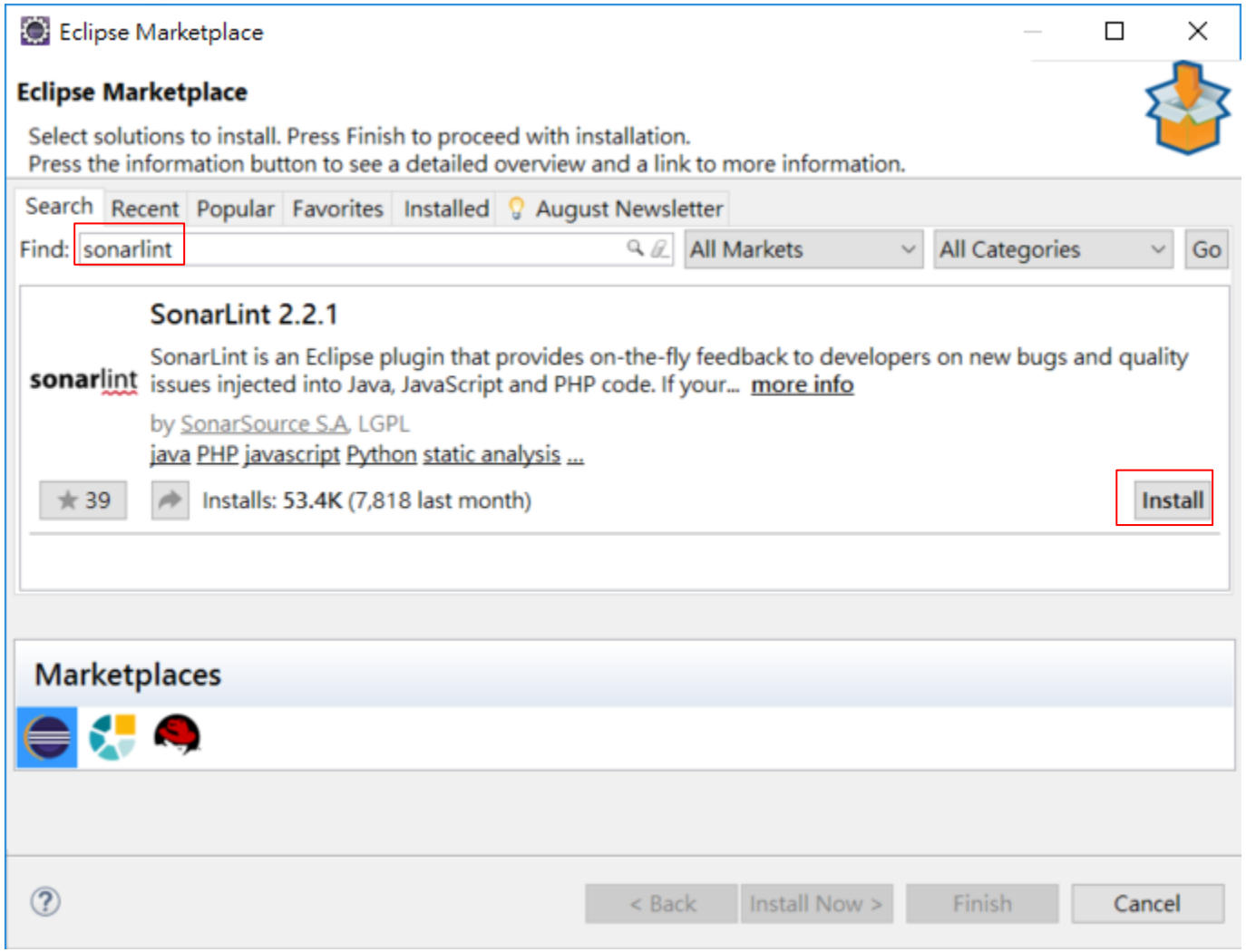
SonarLint for Eclipse

- 靜態程式碼分析工具
  - 偵測潛在的Bug、安全缺陷與不良的程式撰寫方式
- 支援Java、JavaScript與PHP
- Open-source Eclipse plugin
- 系統需求: Java 8 ↑ 、 Eclipse

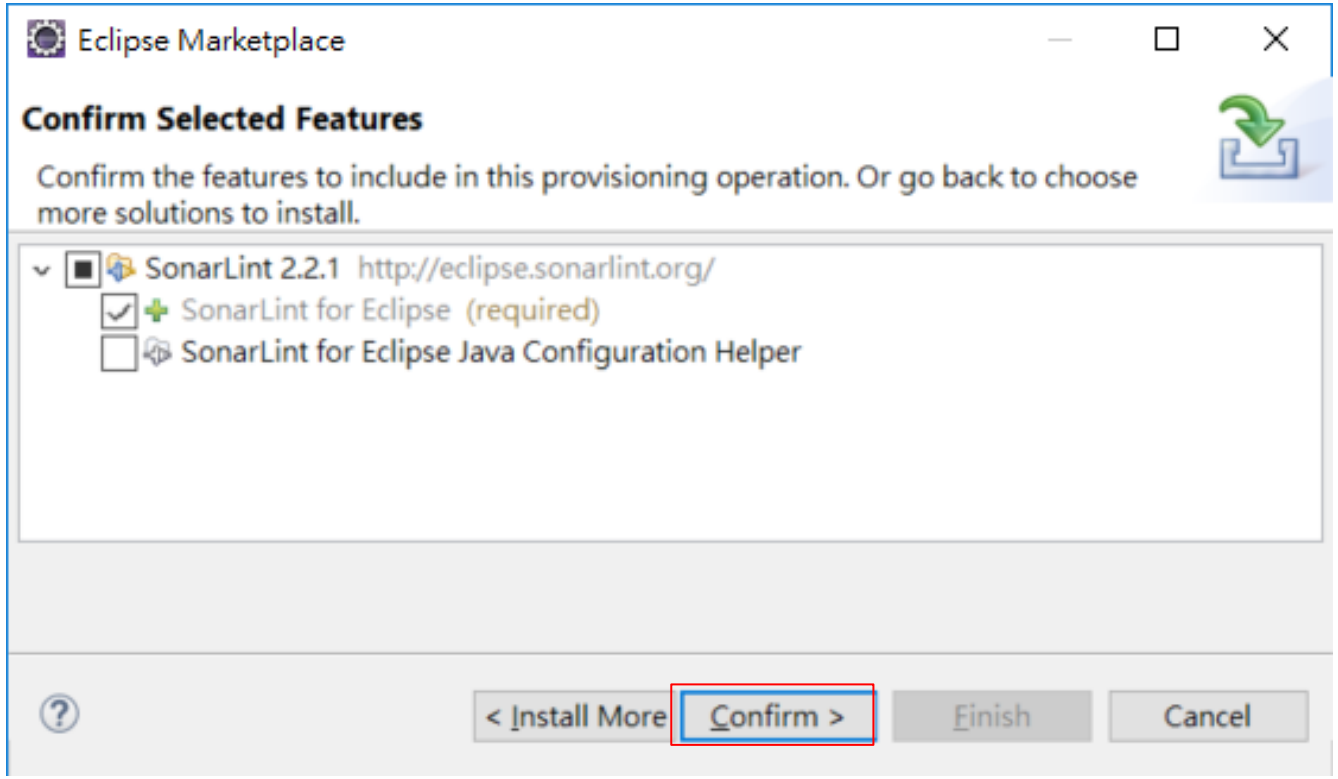
安裝SnoarLint for Eclipse$_1$



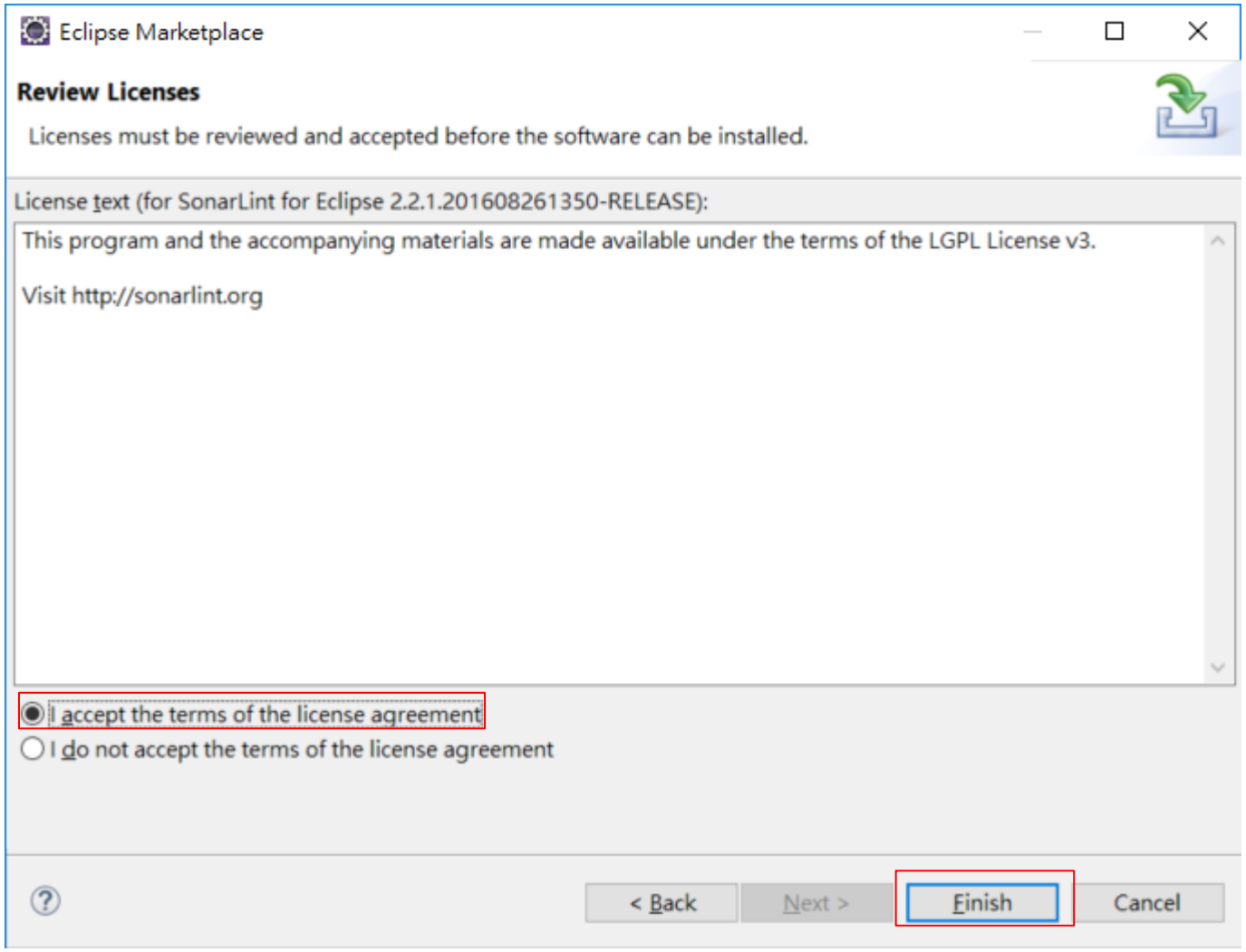
安裝SnoarLint for Eclipse$_2$



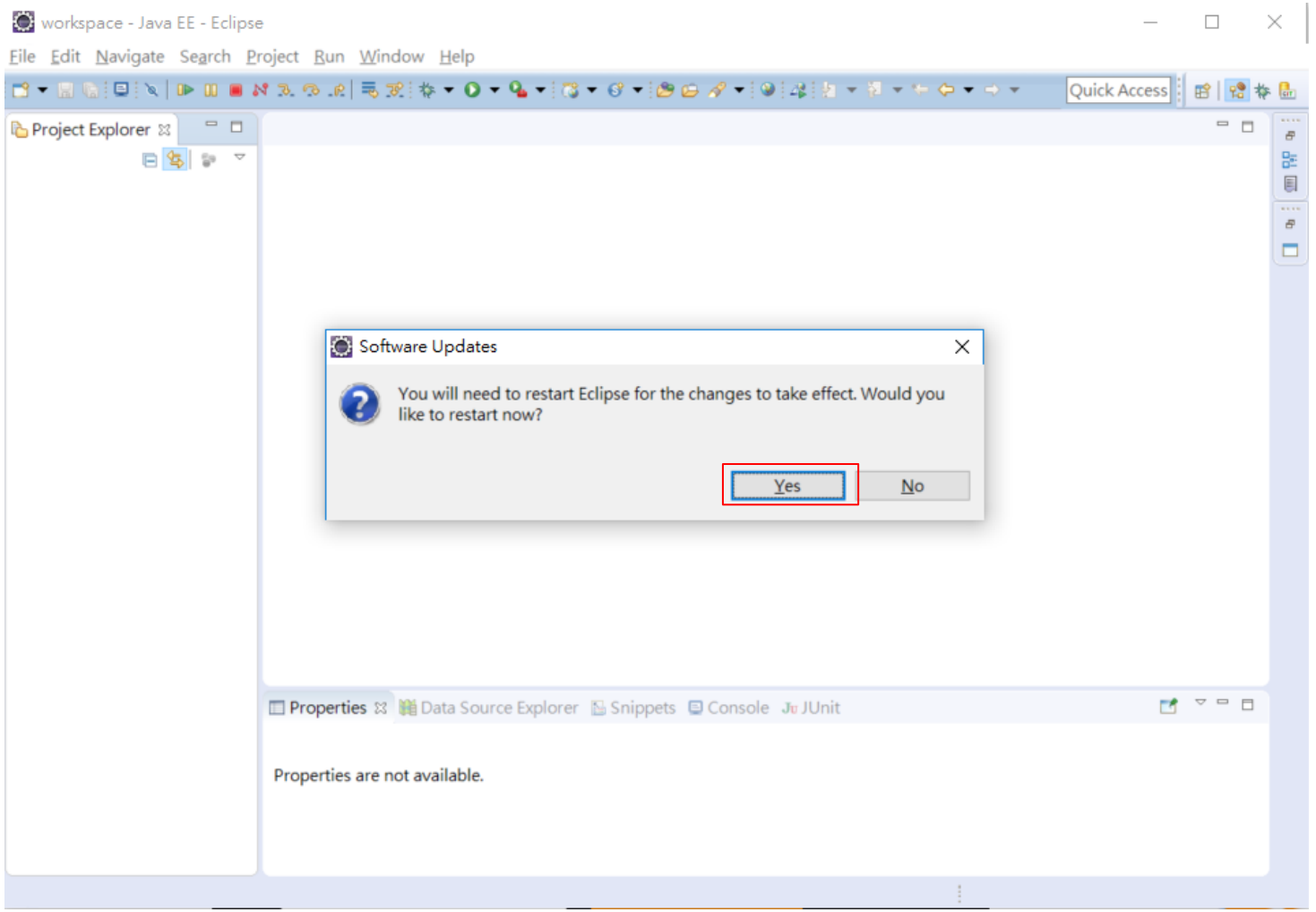
安裝SnoarLint for Eclipse3



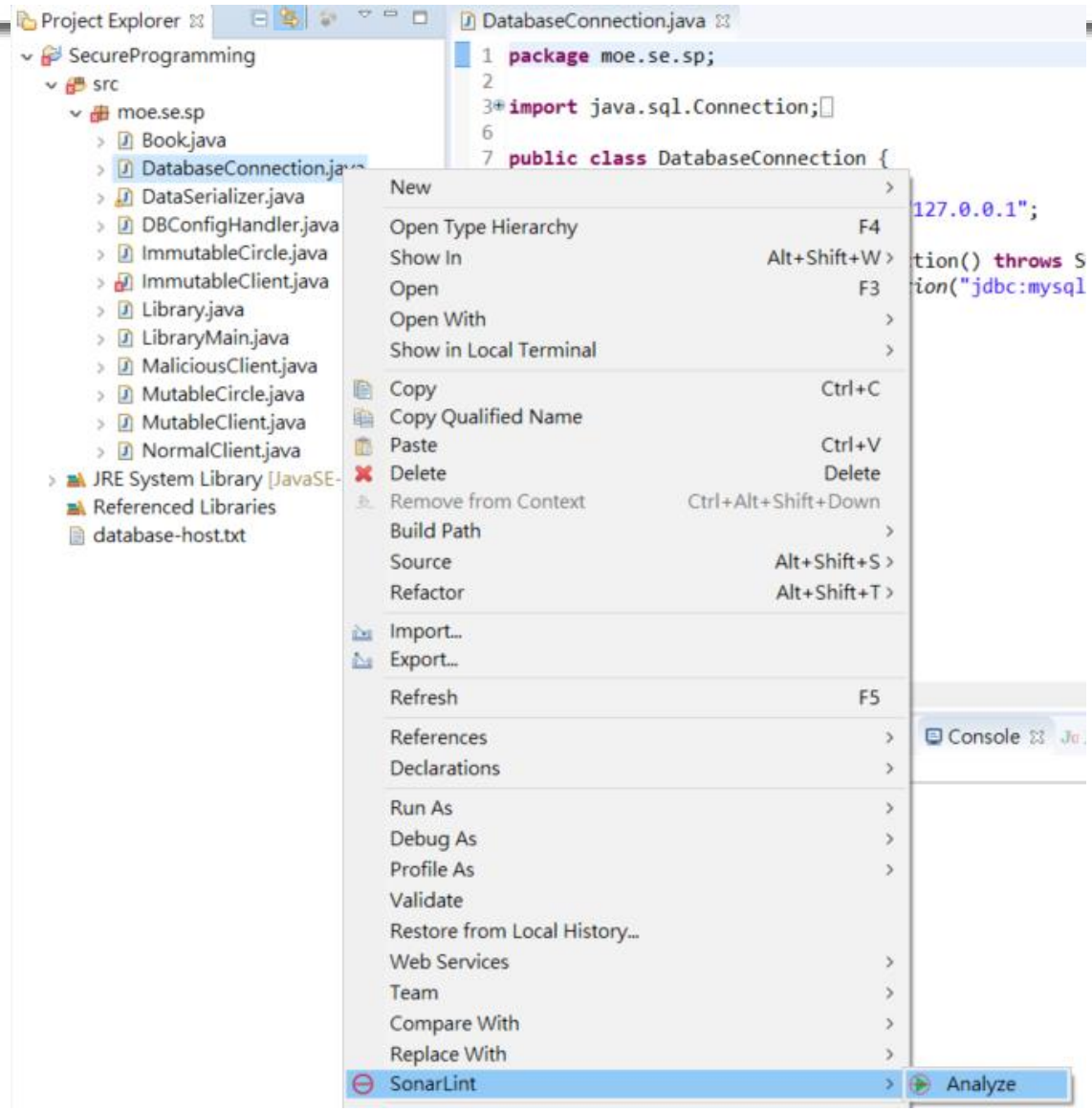
安裝SnoarLint for Eclipse4



安裝SnoarLint for Eclipse$_5$



執行SonarLint分析程式碼



查看分析結果

DatabaseConnection.java

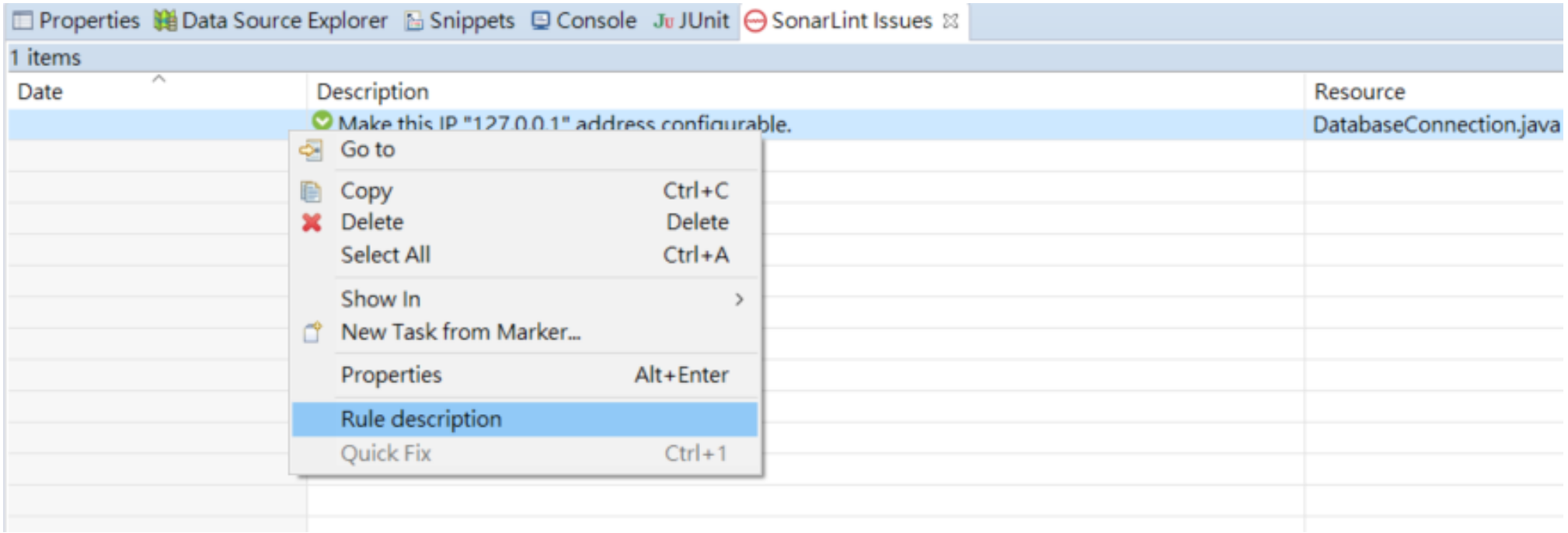
```
1  package moe.se.sp;
2
3  import java.sql.Connection;
6
7  public class DatabaseConnection {
8
9    private String databaseServer = "127.0.0.1";
10
```

Properties | Data Source Explorer | Snippets | Console | JUnit | SonarLint Issues

1 items

Date	Description	Resource
	Make this IP "127.0.0.1" address configurable.	DatabaseConnection.java

查看缺陷詳細資訊1



查看缺陷詳細資訊2

Properties | Data Source Explorer | Snippets | Console | JUnit | SonarLint Issues | SonarLint Rule Description

IP addresses should not be hardcoded (squid:S1313)

Hardcoding an IP address into source code is a bad idea for several reasons:

- a recompile is required if the address changes
- it forces the same address to be used in every environment (dev, sys, qa, prod)
- it places the responsibility of setting the value to use in production on the shoulders of the developer
- it allows attackers to decompile the code and thereby discover a potentially sensitive address

Noncompliant Code Example

```
String ip = "127.0.0.1";
Socket socket = new Socket(ip, 6667);
```

Compliant Solution

```
String ip = System.getProperty("myapplication.ip");
Socket socket = new Socket(ip, 6667);
```

See

- CERT, MSC03-J. - Never hard code sensitive information