

## Sonarqube 使用

# 1 Docker 安装、运行 sonarqube 服务

Step1: 安装数据库

```
$docker run --name postgresqllatest -v
/data/postgresql:/var/lib/postgresql/data/ -e POSTGRES_USER=postgres -e
POSTGRES_PASSWORD=postgres -p 54321:5432 -d docker.io/postgres:latest
```

运行 docker run 经常会报错，提示容器名称已经占用，需要移除掉先前的或者使用新的容器名。

```
$docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8ad95254c368	postgres	"docker-entrypoint..."	6 minutes ago	Created		mypostgresql
84ca011a1e21	postgres	"docker-entrypoint..."	About an hour ago	Exited (0) 21 minutes ago		postgresqlb

```
$docker stop ID
```

```
$docker rm -f ID
```

再次运行容器命令：

```
docker run --name postgresqllatest -v
/data/postgresql:/var/lib/postgresql/data/ -e POSTGRES_USER=postgres -e
POSTGRES_PASSWORD=postgres -p 54321:5432 -d docker.io/postgres:latest
```

在 WINDOWS 客户端，连接云端通过 docker 启动的 postgresql：

```
C:\Program Files\PostgreSQL\10\bin>psql -h 39.108.210.27 -p 54321 -U postgres
psql <10.1, 服务器 11.0 (Debian 11.0-1.pgdg90+2)>
WARNING: psql major version 10, server major version 11.
        Some psql features might not work.
输入 "help" 来获取帮助信息.
postgres=#
```

```
CREATE USER sonar WITH PASSWORD 'sonar';
```

Step2:

```
$docker run -d --name mysonarqube --link postgresql:latest -v
/data/sonarqube:/var/lib/sonarqube/data/ -p 9001:9000 -e
SONARQUBE_JDBC_URL=jdbc:postgresql://39.108.210.27:54321/sonar
docker.io/sonarqube:latest
```

清理此容器的网络占用

```
$docker network disconnect --force bridge mysonarqube
```

查看是否还有同名容器占用

```
$docker network inspect mysonarqube
```

查看容器日志:

```
$docker logs -f -t --tail 600 mysonarqube
```

```
2018-10-26T05:04:01.139307000Z 2018.10.26 05:04:01 INFO ce[[o.sonar.db.Database] Create JDBC data source for jdbc:postgresql://39.108.210.27:54321/sonar
2018-10-26T05:04:02.658058000Z 2018.10.26 05:04:02 INFO ce[[o.s.s.p.ServerFileSystemImpl] SonarQube home: /opt/sonarqube
2018-10-26T05:04:02.832400000Z 2018.10.26 05:04:02 INFO ce[[o.s.c.c.CePluginRepository] Load plugins
2018-10-26T05:04:03.643697000Z 2018.10.26 05:04:03 INFO ce[[o.s.c.q.PurgeCeActivities] Delete the Compute Engine tasks created before 1524978243642
2018-10-26T05:04:03.657381000Z 2018.10.26 05:04:03 INFO ce[[o.s.c.q.PurgeCeActivities] Delete the Scanner contexts tasks created before 1538111043656
2018-10-26T05:04:03.690125000Z 2018.10.26 05:04:03 INFO ce[[o.s.ce.app.CeServer] Compute Engine is operational
2018-10-26T05:04:04.142011000Z 2018.10.26 05:04:04 INFO app[[o.s.a.SchedulerImpl] Process[ce] is up
2018-10-26T05:04:04.142274000Z 2018.10.26 05:04:04 INFO app[[o.s.a.SchedulerImpl] SonarQube is up
```

## 2 maven 扫描工程

<http://39.108.210.27:9001/projects>

Welcome to SonarQube!

Want to quickly analyze a first project? Follow these 2 easy steps.

### 1 Provide a token

Generate a token



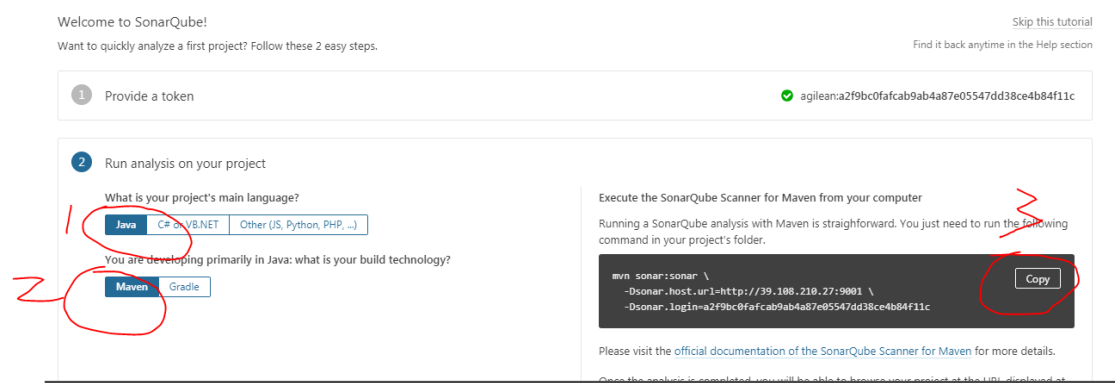
The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your user account.

### 2 Run analysis on your project

生成私有的 Token.

**a2f9bc0fafcab9ab4a87e05547dd38ce4b84f11c**

选择分析的语言和构建:

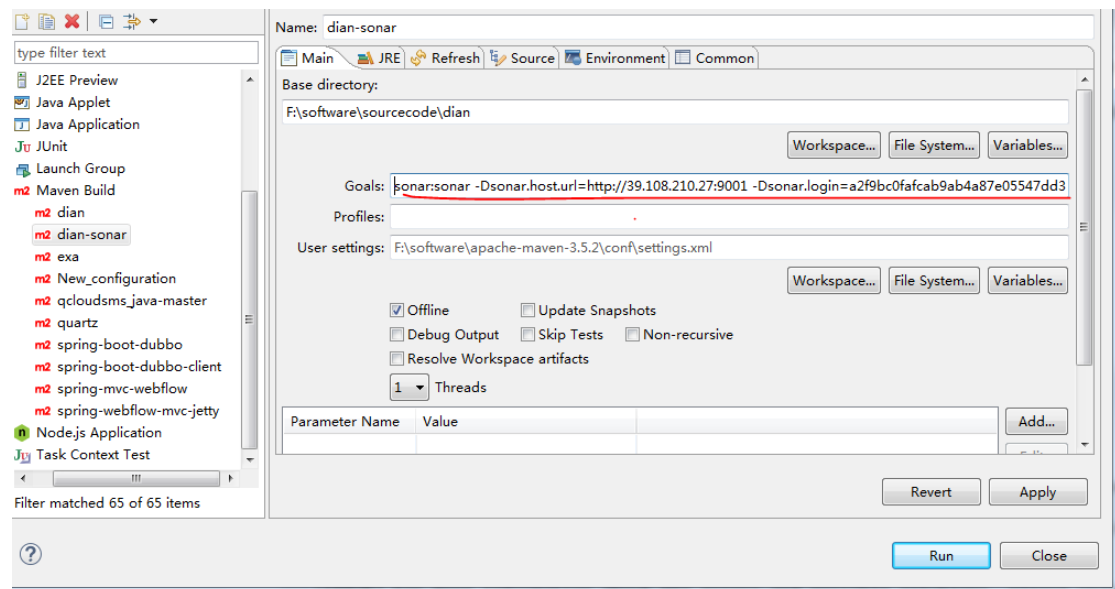


```
mvn sonar:sonar \
```

```
-Dsonar.host.url=http://39.108.210.27:9001 \
```

```
-Dsonar.login=a2f9bc0fafcab9ab4a87e05547dd38ce4b84f11c
```

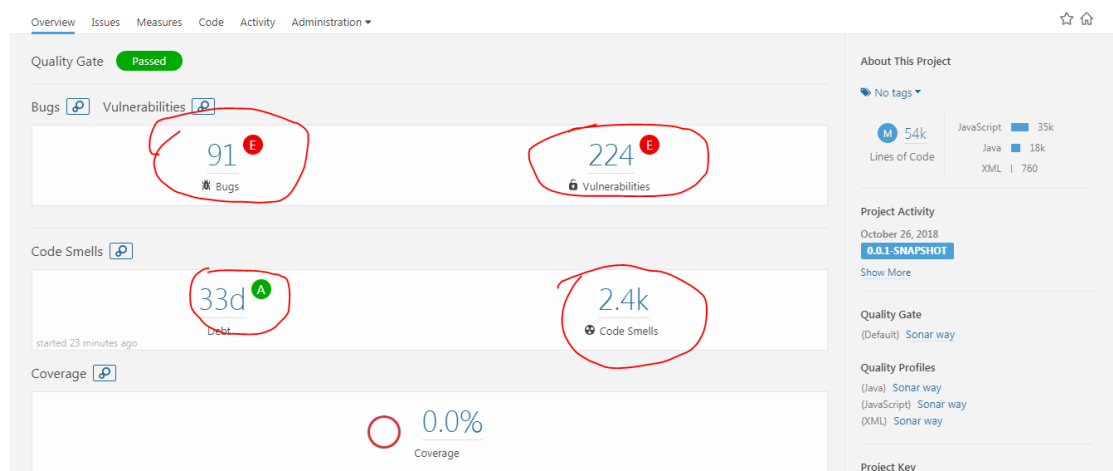
直接在 ECLIPSE 上对我的项目进行质量扫描:



运行完后，直接形成报告。

```
WARNING] Invalid character encountered in file f:/software/sourcecode/dian/src/main/webapp/js/jquery.tancybox.min.js at line :
[INFO] Analysis report generated in 4350ms, dir size=6 MB
[INFO] Analysis reports compressed in 9291ms, zip size=2 MB
[INFO] Analysis report uploaded in 27093ms
[INFO] ANALYSIS SUCCESSFUL, you can browse http://39.108.210.27:9001/dashboard/index.com_hanniu.dian:dian
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at http://39.108.210.27:9001/api/ce/task?id=AWau4EJ_Hqs00bEGziSv
[INFO] Task total time: 5:23.818 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 06:15 min
[INFO] Finished at: 2018-10-26T13:37:27+08:00
[INFO] Final Memory: 18M/247M
[INFO] -----
```

报告如下：

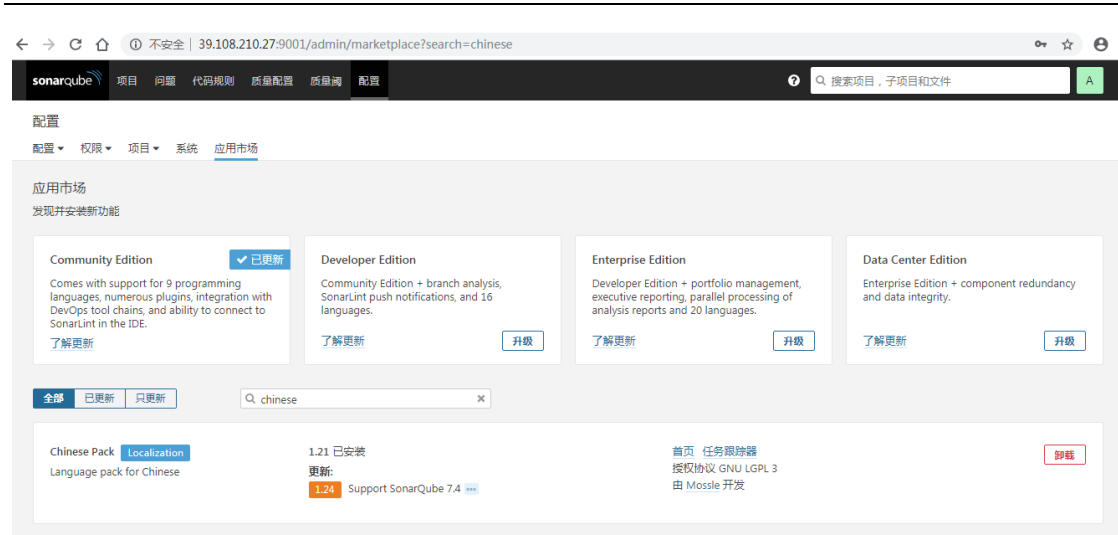


## 3 sonarqube 汉化安装

The screenshot shows the SonarQube Administration interface. The top navigation bar includes 'sonarqube', 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration' (circled in red with a '1'). Below this, the 'Administration' section has sub-tabs: 'Configuration', 'Security', 'Projects', 'System', and 'Marketplace' (circled in red with a '2'). The 'Marketplace' section is titled 'Discover and install new features'. It displays three editions: 'Community Edition' (marked 'Installed'), 'Developer Edition', and 'Enterprise Edition'. Below these, there are tabs for 'All', 'Installed', and 'Updates Only'. A search bar contains the text 'chinese' (circled in red with a '3'). The search results show the 'Chinese Pack' (Localization) for 'SonarQube Chinese Pack' (circled in red with a '4'). A version indicator '1.21' and 'Support SonarQube 7.1' are visible. On the right, there are links for 'Homepage', 'Issue Tracker', and 'Developed by Mossle'.

This screenshot shows the SonarQube Administration interface after a restart. The top navigation bar is the same. The 'Administration' section has sub-tabs: 'Configuration', 'Security', 'Projects', 'System', and 'Marketplace' (circled in red with a '6'). A blue banner at the top of the Marketplace section states 'SonarQube needs to be restarted in order to install 1 plugins' with 'Restart' and 'Revert' buttons. The 'Marketplace' section is titled 'Discover and install new features'. It displays four editions: 'Community Edition' (marked 'Installed'), 'Developer Edition', 'Enterprise Edition', and 'Data Center Edition'. Below these, there are tabs for 'All', 'Installed', and 'Updates Only'. A search bar contains the text 'chinese'. The search results show the 'Chinese Pack' (Localization) for 'SonarQube Chinese Pack'. A version indicator '1.21' and 'Support SonarQube 7.1' are visible. On the right, there are links for 'Homepage', 'Issue Tracker', and 'Developed by Mossle'. The status of the 'Chinese Pack' is now 'Install Pending' (circled in red with a '5').

重启后登陆（）汉化完成。



## 4 自定义规则

<https://docs.sonarqube.org/latest/user-guide/built-in-rule-tags/>

### 4.1 安装 CheckStyle 插件

官网: <http://checkstyle.sourceforge.net/>

### 4.2 配置自定义的 CheckStyle 代码规则

#### 4.2.1 使用 CheckStyle 代码规则配置文件

注意: 这种方法只有新建一个质量配置时才能用, 质量配置创建好后, 就不能利用配置文件来配置代码规则了



```
<module name="Checker">

  <property name="charset" value="UTF-8" />
  <property name="severity" value="warning" />
  <property name="fileExtensions" value="java, properties, xml" />

  <!-- 检查文件的长度（行） default max=2000 -->
  <module name="FileLength">
    <property name="max" value="2000" />
  </module>

  <module name="TreeWalker">

    <!-- =====命名规范===== -->

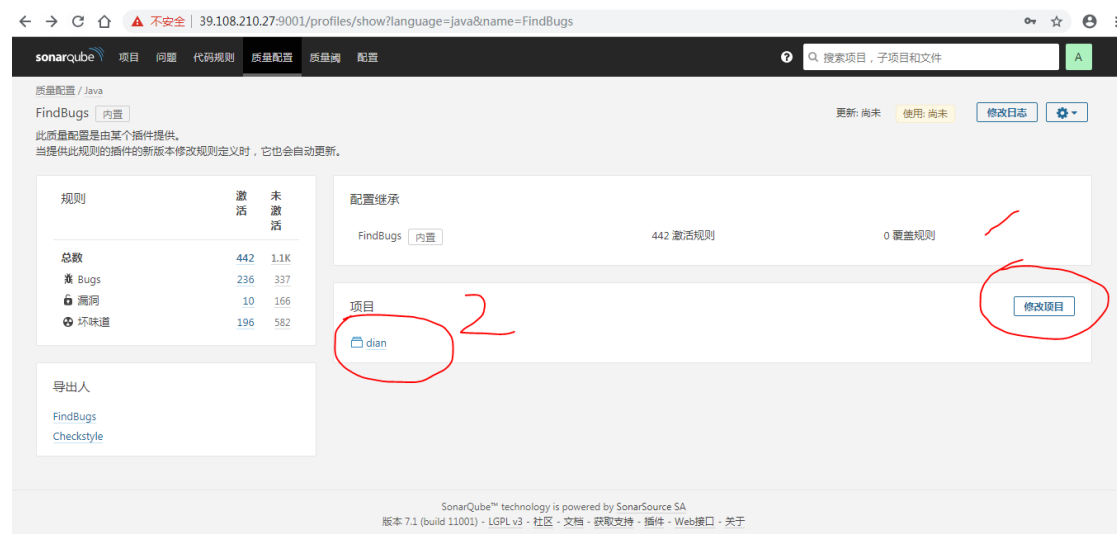
    <!-- 包名的检查（只允许小写字母） -->
    <module name="PackageName">
      <property name="format" value="^[a-z]+(\\.[a-z][a-z0-9]*)*$" />
    </module>

  </module>
</module>
```

上述方法不推荐。

## 4.2.2 启用 SonarQube 中 CheckStyle 相关代码规

让项目 dian 使用 findBugs 规则：



CheckStyle: 官网:

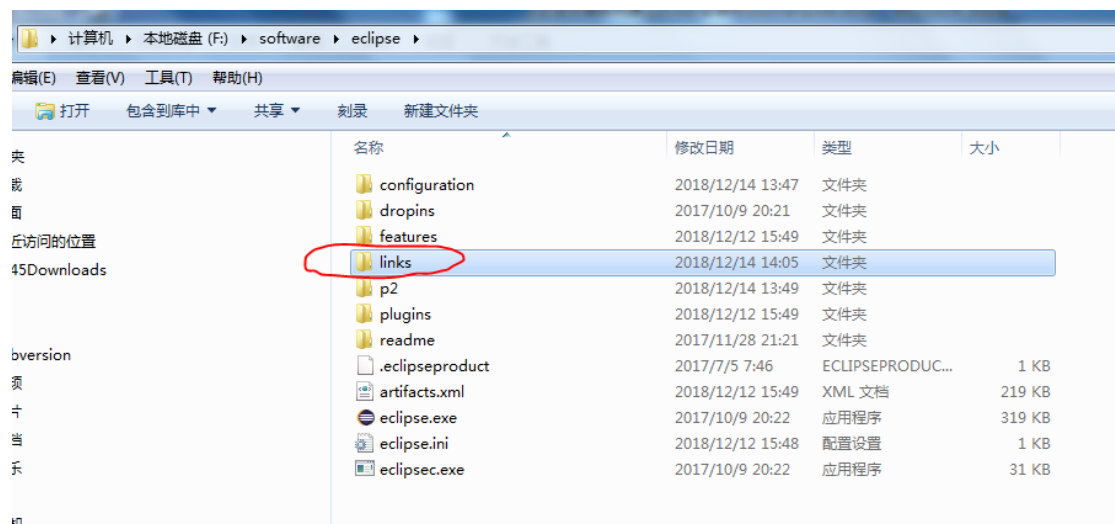
<http://checkstyle.sourceforge.net/>

配置 checkstyle 插件进行代码检查

下载插件:

<https://sourceforge.net/projects/eclipse-cs/files/Eclipse%20Checkstyle%20Plug-in/>

新建 links

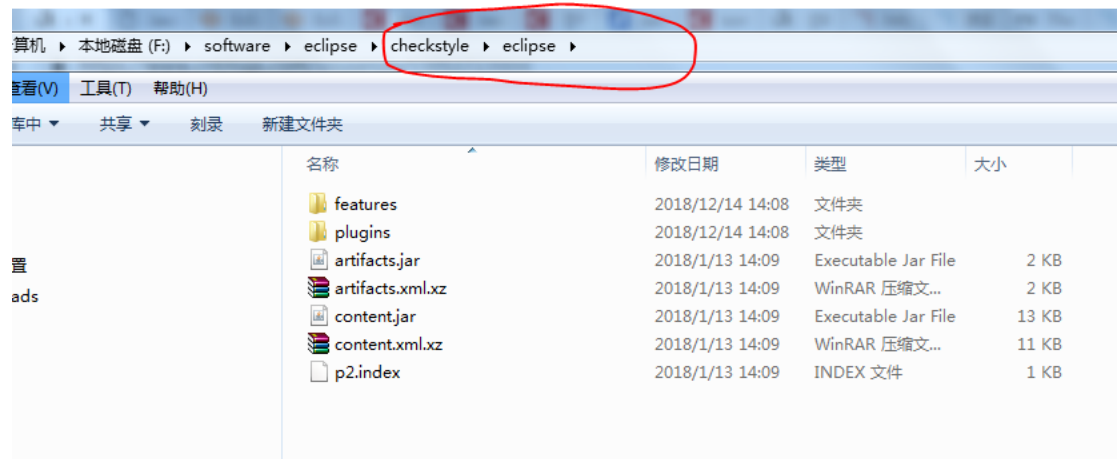


将下载的解压插件（含）拷贝至 F:/softwares/eclipse/checkstyle/

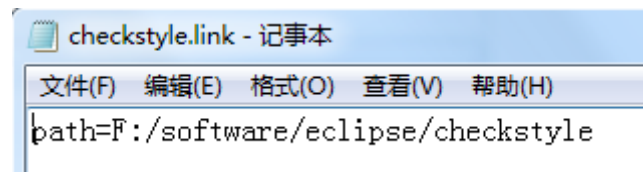
F:/softwares/eclipse/checkstyle/eclipse/plugins

F:/softwares/eclipse/checkstyle/eclipse/features





新建 checkstyle.link 文件，内容如下：




## 5 如何调整 sonar 规则

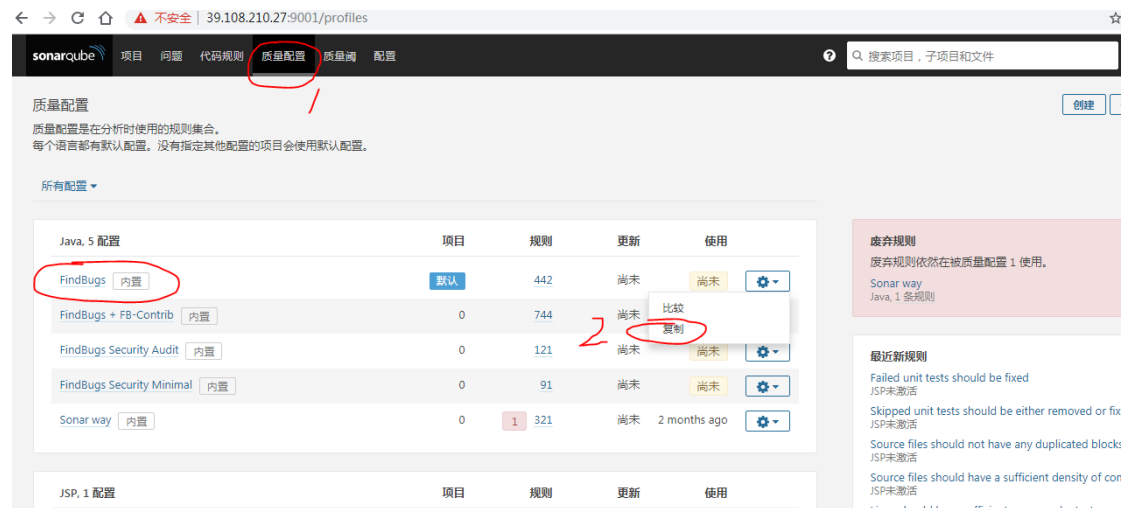
### 5.1 自定义

需要按照 sonarqube API 代码实现。（知识量大，下一版本介绍）

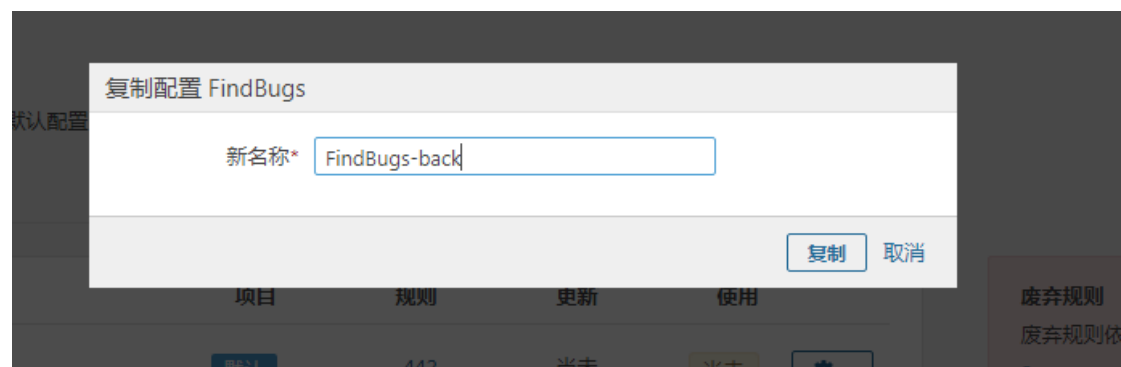
### 5.2 服务器端修改模板

➤ 第一步：导出模板 xml 文件

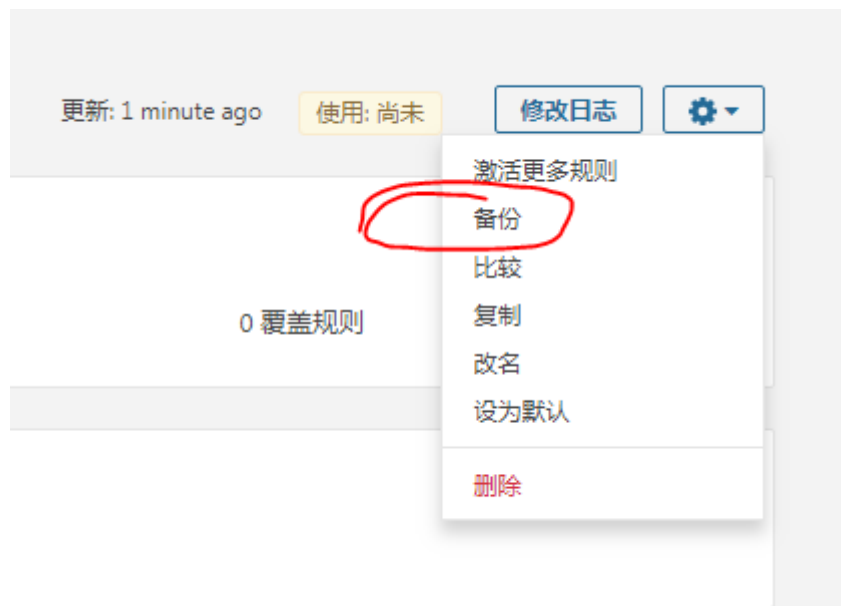
管理员账号登陆 sonarqube, 点击“质量配置”菜单, 显示所有插件配置, 如下图所示: 我们觉得 findBugs 插件规则太多, 可以点击最右边的  键, 选择“复制”。



取名为: findBugs-back



点击“备份”, 如下图所示。



点击“备份”后，可以下载 findBugs 的 xml 格式的模板文件。文件格式如下：

---

```

▼<profile>
  <name>FindBugs-new</name>
  <language>java</language>
  ▼<rules>
    ▼<rule>
      <repositoryKey>findbugs</repositoryKey>
      <key>AM_CREATES_EMPTY_JAR_FILE_ENTRY</key>
      <priority>MAJOR</priority>
      <parameters/>
    </rule>
    ▼<rule>
      <repositoryKey>findbugs</repositoryKey>
      <key>AM_CREATES_EMPTY_ZIP_FILE_ENTRY</key>
      <priority>MAJOR</priority>
      <parameters/>
    </rule>
    ▼<rule>
      <repositoryKey>findbugs</repositoryKey>
      <key>AT_OPERATION_SEQUENCE_ON_CONCURRENT_ABSTRACTION</key>
      <priority>MAJOR</priority>
      <parameters/>
    </rule>
    ▼<rule>
      <repositoryKey>findbugs</repositoryKey>
      <key>BAC_BAD_APPLET_CONSTRUCTOR</key>
      <priority>MAJOR</priority>
      <parameters/>
    </rule>
    ▼<rule>
      <repositoryKey>findbugs</repositoryKey>
      <key>BC_BAD_CAST_TO_ABSTRACT_COLLECTION</key>
      <priority>INFO</priority>
      <parameters/>
    </rule>
  </rules>
</profile>

```

➤ 第二步：修改模板文件：

findBugs 自带 400 多种规则，实际用到的不多，可以根据实际情况删减部分规则，直接修改 XML 模板文件并保存。

➤ 第三步：新建新的模板文件

点击“新建”，取名为：findBug-simple



点击“选择文件”，将前面修改的模板文件上传：

如下图所示：新的规则已经生效。



## 6 sonar rules 整理

参考文献:

<https://rules.sonarsource.com/java/RSPEC-4434>

### 5.1 LDAP 初始化不允许反序列化

JNDI supports the deserialization of objects from LDAP directories, which is fundamentally insecure and can lead to remote code execution.

This rule raises an issue when an LDAP search query is executed with SearchControls configured to allow deserialization.

#### Noncompliant Code Example

```
DirContext ctx = new InitialDirContext();
// ...
ctx.search(query, filter,
    new SearchControls(scope, countLimit, timeLimit, attributes,
        true, // Noncompliant; allows deserialization
        deref));
```

#### Compliant Solution

```
DirContext ctx = new InitialDirContext();
// ...
ctx.search(query, filter,
    new SearchControls(scope, countLimit, timeLimit, attributes,
        false,
        deref));
```

## 5.2 Cryptographic keys should not be too short

<https://rules.sonarsource.com/java/RSPEC-4426>

When generating cryptographic keys (or key pairs), it is important to use a key length that provides enough entropy against brute-force attacks. For the Blowfish algorithm the key should be at least 128 bits long, while for the RSA algorithm it should be at least 2048 bits long.

This rule raises an issue when a Blowfish key generator or RSA key-pair generator is initialized with too small a length parameter.

### Noncompliant Code Example

```
KeyGenerator keyGen = KeyGenerator.getInstance("Blowfish");
keyGen.init(64); // Noncompliant


KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("RSA");
keyPairGen.initialize(512); // Noncompliant
```

### Compliant Solution

```
KeyGenerator keyGen = KeyGenerator.getInstance("Blowfish");
keyGen.init(128);

KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("RSA");
keyPairGen.initialize(2048);
```

## 5.3 SQL 注入



```

public boolean authenticate(javax.servlet.http.HttpServletRequest request, java.sql.Connection connection) throws SQLException {
    String user = request.getParameter("user");
    String pass = request.getParameter("pass");

    String query = "SELECT * FROM users WHERE user = '" + user + "' AND pass = '" + pass + "'"; // Unsafe

    // If the special value "foo' OR 1=1 --" is passed as either the user or pass, authentication is bypassed
    // Indeed, if it is passed as a user, the query becomes:
    // SELECT * FROM users WHERE user = 'foo' OR 1=1 --' AND pass = '...'
    // As '--' is the comment till end of line syntax in SQL, this is equivalent to:
    // SELECT * FROM users WHERE user = 'foo' OR 1=1
    // which is equivalent to:
    // SELECT * FROM users WHERE 1=1
    // which is equivalent to:
    // SELECT * FROM users

    java.sql.Statement statement = connection.createStatement();
    java.sql.ResultSet resultSet = statement.executeQuery(query); // Noncompliant
    return resultSet.next();
}

```

正确写法:

```

public boolean authenticate(javax.servlet.http.HttpServletRequest request, java.sql.Connection connection) throws SQLException {
    String user = request.getParameter("user");
    String pass = request.getParameter("pass");

    String query = "SELECT * FROM users WHERE user = ? AND pass = ?"; // Safe

    java.sql.PreparedStatement statement = connection.prepareStatement(query);
    statement.setString(1, user); // Will be properly escaped
    statement.setString(2, pass);
    java.sql.ResultSet resultSet = statement.executeQuery();
    return resultSet.next();
}

```

## 5.4 不再安全的加密方式

According to the US National Institute of Standards and Technology (NIST), the Data Encryption Standard (DES) is no longer considered secure:

### Noncompliant Code Example

```
Cipher c = Cipher.getInstance("DESede/ECB/PKCS5Padding");
```

### Compliant Solution

```
Cipher c = Cipher.getInstance("AES/GCM/NoPadding");
```



## 5.5 操作系统命令需要校验

### Noncompliant Code Example

```
public void run(javax.servlet.http.HttpServletRequest request) throws IOException {
    String binary = request.getParameter("binary");

    // If the value "/sbin/shutdown" is passed as binary and the web server is running as
    // then the machine running the web server will be shut down and become unavailable f

    Runtime.getRuntime().exec(binary); // Noncompliant
}
```

### Compliant Solution

```
public void run(javax.servlet.http.HttpServletRequest request) throws IOException {
    String binary = request.getParameter("binary");

    // Restrict to binaries within the current working directory whose name only contains
    if (!binary.matches("[a-zA-Z]++")) {
        throw new IllegalArgumentException();
    }

    Runtime.getRuntime().exec(binary);
}
```

### Noncompliant Code Example

```
public void run(javax.servlet.http.HttpServletRequest request) throws IOException {
    String binary = request.getParameter("binary");

    // If the value "/sbin/shutdown" is passed as binary and the web server is running as
    // then the machine running the web server will be shut down and become unavailable f

    Runtime.getRuntime().exec(binary); // Noncompliant
}
```

### Compliant Solution

```
public void run(javax.servlet.http.HttpServletRequest request) throws IOException {
    String binary = request.getParameter("binary");

    // Restrict to binaries within the current working directory whose name only contains
    if (!binary.matches("[a-zA-Z]++")) {
        throw new IllegalArgumentException();
    }

    Runtime.getRuntime().exec(binary);
}
```

## 5.6 不要使用 ThreadGroup

Even though thread groups are useful for keeping threads organized, programmers seldom benefit from their use because many of the methods of the `ThreadGroup` class (for example, `allowThreadSuspension()`, `resume()`, `stop()`, and `suspend()`) are deprecated. Furthermore, many nondeprecated methods are obsolete in that they offer little desirable functionality. Ironically, a few `ThreadGroup` methods are not even [thread-safe](#) [Bloch 2001].

使用线程池：

```
ThreadFactory threadFactory = Executors.defaultThreadFactory();
ThreadPoolExecutor executorPool = new ThreadPoolExecutor(3, 10, 5, TimeUnit.SECONDS,
    new ArrayBlockingQueue<Runnable>(2), threadFactory);

for (int i = 0; i < 10; i++) {
    executorPool.execute(new JobThread("Job: " + i));
}

System.out.println(executorPool.getActiveCount()); // Compliant
executorPool.shutdown();
```