

如何用 Sonar 进行静态代码扫描

1.使用目的

Sonar对于从事IT的人员而言，应该并不陌生，我们既可以通过给开发工具例如Eclipse、STS、MyEclipse安装sonarlint插件，监控开发人员所写的每一行代码，也可以通过sonarqube服务的安装体系尽早发现开发源码中的缺陷。通过对源码的分析，使得测试人员可以发现源码中无效引用、注释缺失、模块设计复杂等代码问题。

2.环境搭建

2.1对于测试而言的sonarqube服务搭建

2.1.1JDK的安装

目前针对主流的sonarqube6.5而言支持的JDK版本为1.8，当然安装JDK的时候请先核对自己的电脑系统，Windows32位的电脑系统对应安装32位的jdk1.8，Windows64位的系统安装对应的64位jdk。切记安装的jdk位数必须与自己的电脑系统一致。

2.2.2Mysql的安装

这里有点说明供大家参考：

（1）Windows 10系统的朋友们必须选择安装Mysql56的版本；Windows10以下的系统可选择性安装mysql56或者mysql57的版本，值得说明的是sonarqube对Mysql56的版本兼容性较好，个人建议大家安装Mysql56版本。

（2）安装过程中需要注意的问题是：

第一点：只安装Mysql的server端即可；

第二点：安装完毕之后，要启用本地服务，去启动mysql57（具体服务名称是什么要看最终本地服务的名称是什么）服务。

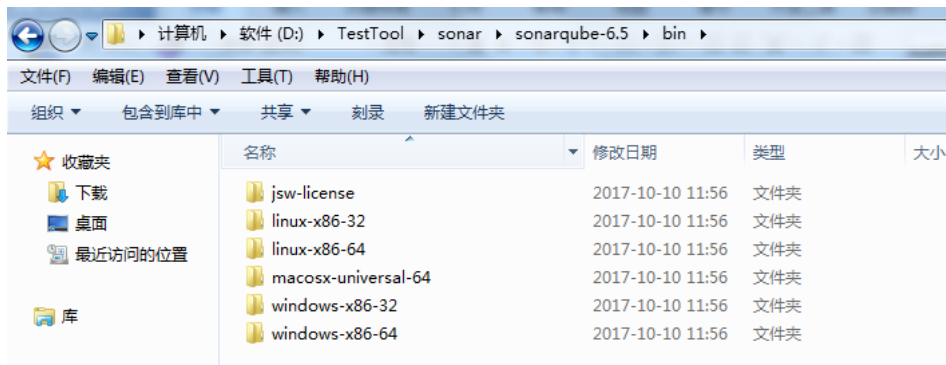
（3）安装完Mysql的服务之后在C:\ProgramData\MySQL\MySQL Server 5.7目录下找到my.ini文件，在文件中找到max_allowed_packet=4M将其修改为：max_allowed_packet=200M；

2.2.3Mysql访问工具的安裝

这里安装的软件大家可以自行选择，我自己使用的是dbeaver.exe，也可以使用Navicat for mysql或者是小青蛙toad等等。

安装完毕后，新建一个本地连接，建立一个数据库叫做sonar





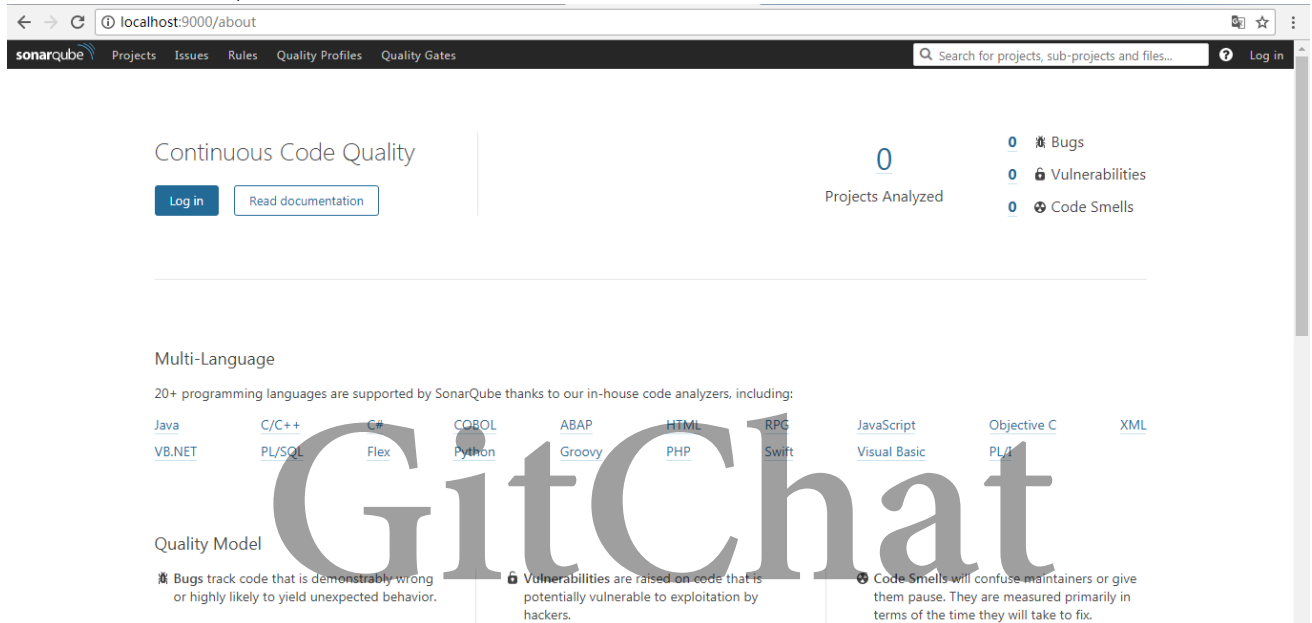
Windows 32位系统启动服务时需要选择：windows-x86-32文件夹下的StartSonar.bat命令双击执行即可启动sonarqube服务。

Windows 64位系统启动服务时需要选择：windows-x86-64文件夹下的StartSonar.bat命令双击执行即可启动sonarqube服务。

(2) 启动sonarqube服务

(3) 启动服务后，打开浏览器输入http://localhost:9000推荐谷歌浏览器；

显示如下图即表示sonarqube成功被安装并且被启用。



(4) 配置sonarqube，安装启动成功后需要对其进行配置：

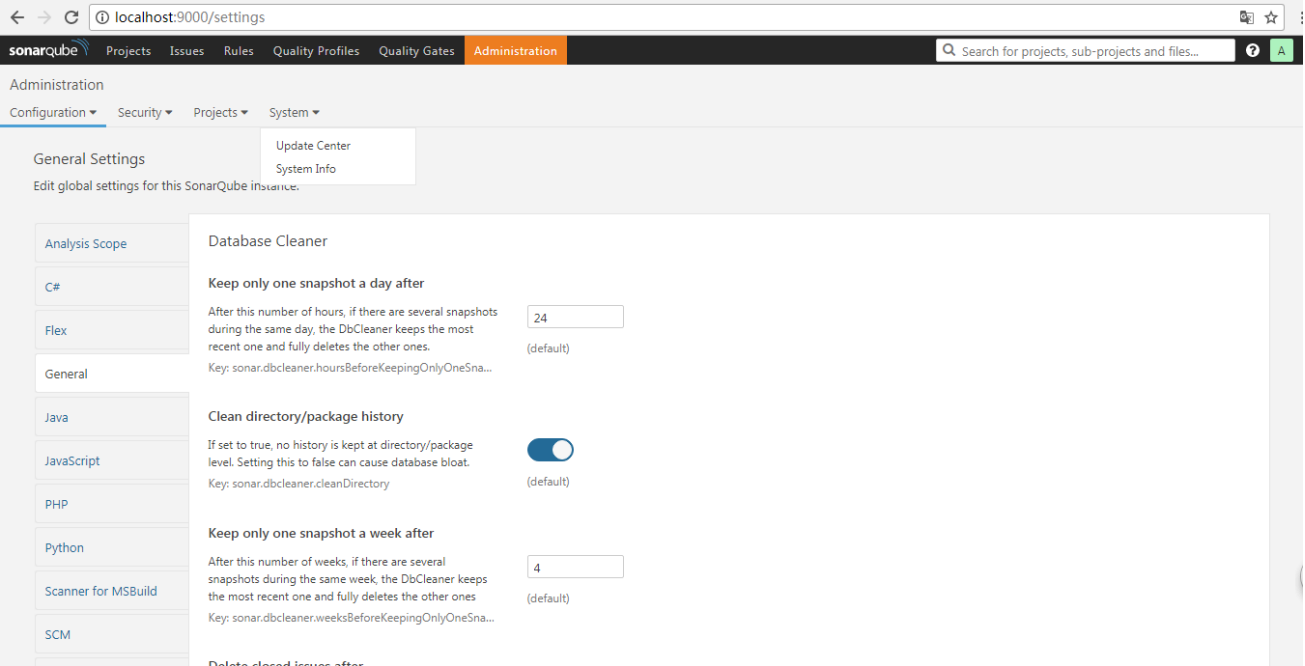
打开Sonarqube安装目录D:\soanr\sonarqube-6.5\conf下的sonar.properties文件，在#--- MySQL 5.6 or greater下输入如下信息：

如下图位置：

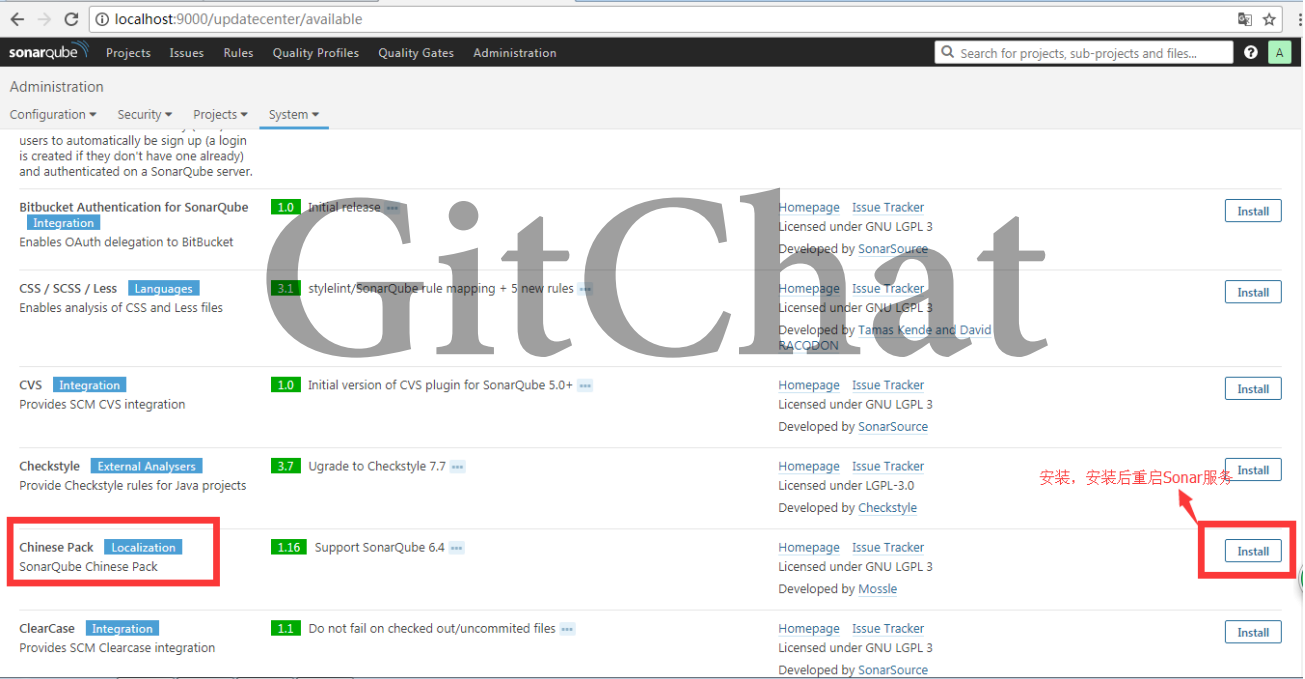
```
#sonar.jdbc.password=

#----- Embedded Database (default)
# H2 embedded database server listening port, defaults to 9092
#sonar.embeddedDatabase.port=9092
#----- MySQL 5.6 or greater
# Only InnoDB storage engine is supported (not myISAM).
# Only the bundled driver is supported. It can not be changed.
sonar.jdbc.url=jdbc:mysql://localhost:3306/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useSSL=false
sonar.jdbc.username=root
sonar.jdbc.password=root
sonar.sourceEncoding=UTF-8
sonar.login=admin
sonar.password=admin
```

第一步：用第（4）配置的sonar.password和sonar.login登录sonarqube服务的web界面
操作System下的Update Center（如图）



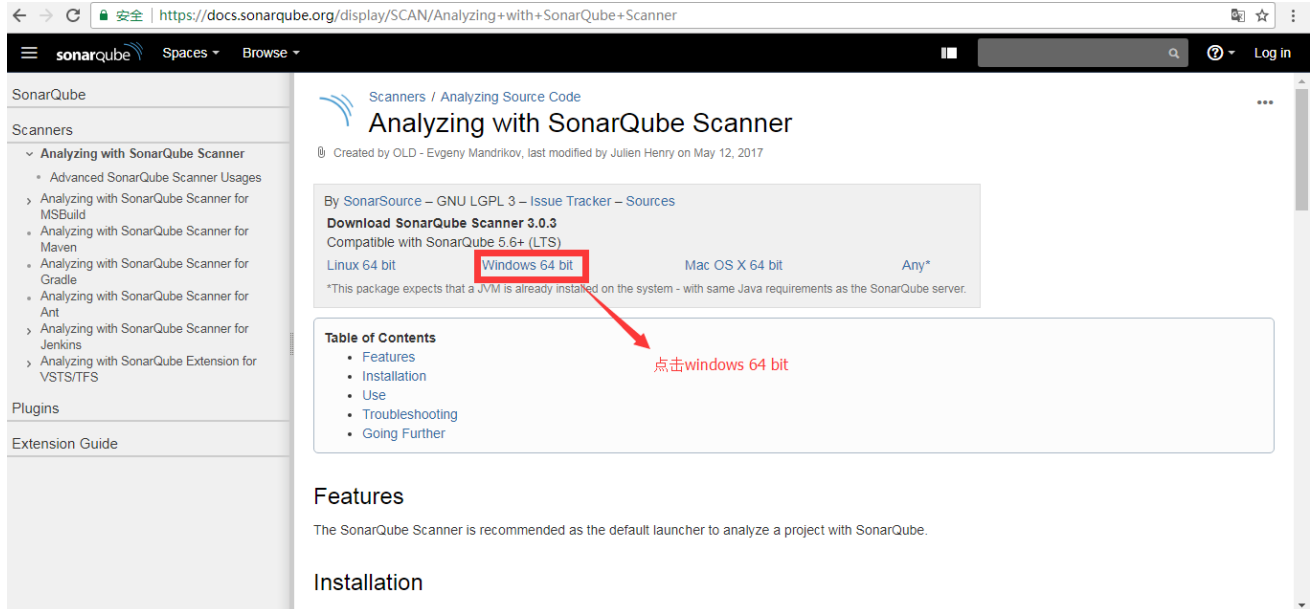
第二步：操作【Available】，在该页面选择Chinese Pack（如图）



请注意这里可能会安装插件失败，原因是因为你的浏览器问题，换个浏览器重新安装即可，建议使用谷歌浏览器进行安装，效果最佳；
第三步：安装后重启SonarQube服务，重新登录页面显示如下图。

<https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner>

文件名：sonar-scanner-cli-3.0.3.778-windows (如图)



(2) Sonar-scanner安装配置

解压sonar-scanner-cli-3.0.3.778-windows。

在D:\soanr\sonar-scanner-3.0.3.778-windows\conf文件夹下打开

sonar-scanner.properties文件，配置信息如下：

Configure here general information about the environment, such as SonarQube DB details for example

No information about specific project should appear here

—— Default SonarQube server

sonar.host.url=<http://192.168.18.95:9000/>

sonar.host.url=<http://localhost:9000/>

—— Default source code encoding

sonar.jdbc.password=sonar

— PostgreSQL

sonar.jdbc.url=jdbc:postgresql://localhost/sonar

— MySQL

sonar.jdbc.url=jdbc:mysql://192.168.18.95:3306/sonar?
useUnicode=true&characterEncoding=utf8

sonar.jdbc.url=jdbc:mysql://localhost:3306/sonar?
useUnicode=true&characterEncoding=utf8

sonar.jdbc.username=sonar

sonar.jdbc.password=sonar

— Oracle

sonar.jdbc.url=jdbc:oracle:thin:@localhost/XE

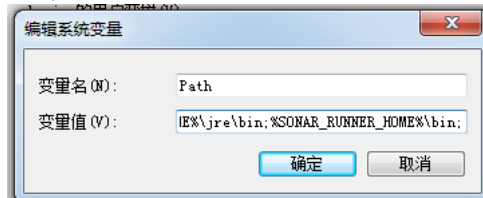
配置环境变量

变量名Name=SONAR_RUNNER_HOME

变量值Value=D:\soanr\sonar-scanner-3.0.3.778-windows (如图)



打开path, 输入%SONAR_RUNNER_HOME%\bin;



在命令行窗口输入sonar-scanner -version出现以下信息, 则表示环境变量设置成功 (如图)。



(3) sonar-scanner执行

打开要进行代码分析的项目根目录, 新建sonar-project.properties文件。

输入以下信息:

GitChat

must be unique in a given SonarQube instance

this is the name displayed in the SonarQube UI

Path is relative to the sonar-project.properties file.

the sonar-project.properties file.

must be unique in a given SonarQube instance

```
sonar.projectKey=Juzai
```

Encoding of the source code. Default is
defasonar.ce.javaOpts=-Xmx2560m -Xms853m -
XX:+HeapDumpOnOutOfMemoryError

```
sonar.projectName=Helloworld  
sonar.projectVersion=1.1
```

```
sonar.ce.workerCount=1  
sonar.language=java  
sonar.sources=comm-test  
sonar.java.binaries=Webdriver/bin  
sonar.ce.javaOpts=-Xmx2560m -Xms853m -XX:+HeapDumpOnOutOfMemoryError  
sonar.ce.workerCount=1  
sonar.language=java
```

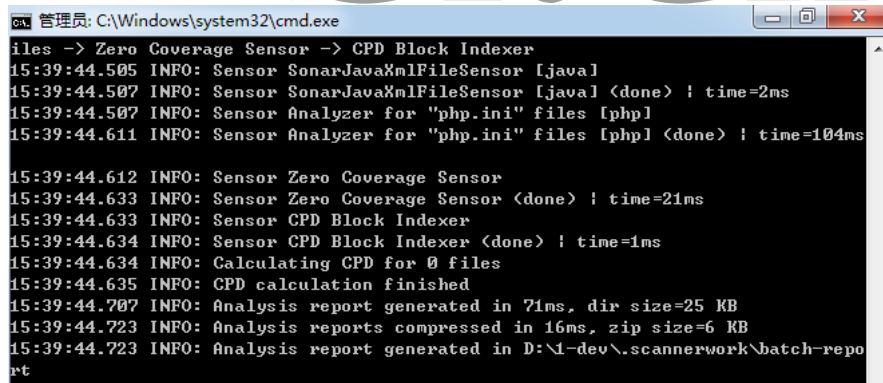
其中sonar.projectName为项目名称

sonar.java.binaries为二进制文件即.class文件所在的路径

sonar.sources为对应项目的根目录。

设置成功后，启动sonarqube服务，并启动cmd。

在cmd进入项目所在的根目录，输入命令：sonar-scanner，分析成功后会出现下图

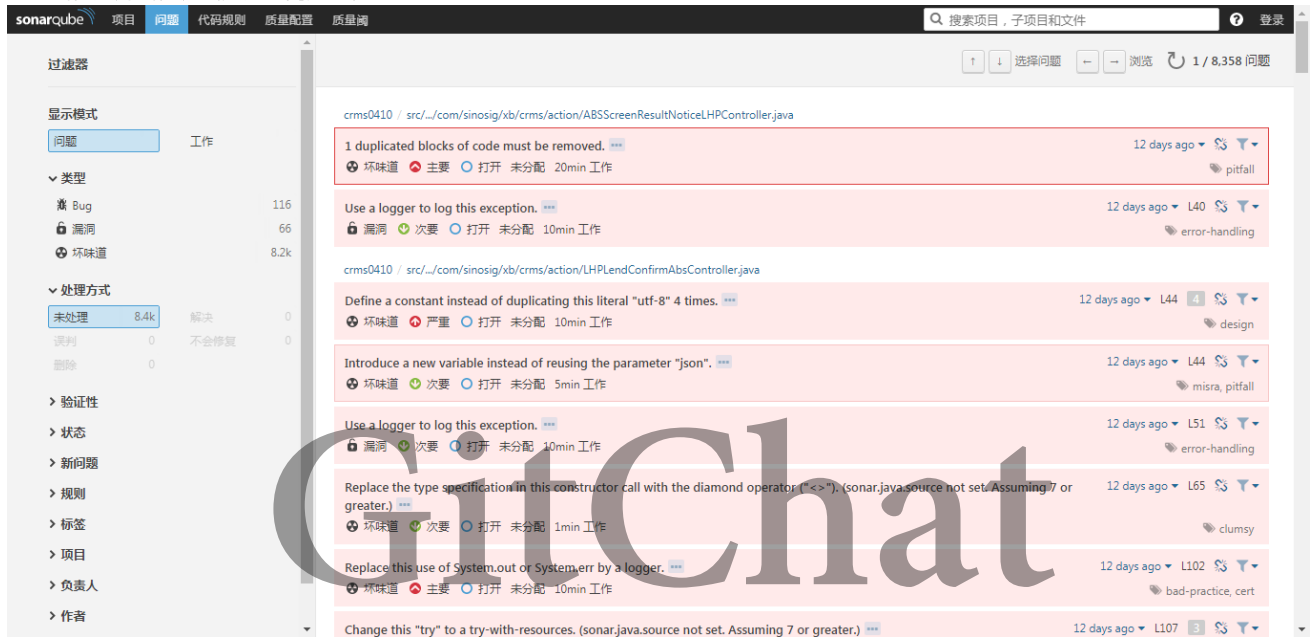


```
管理员: C:\Windows\system32\cmd.exe  
iles -> Zero Coverage Sensor -> CPD Block Indexer  
15:39:44.505 INFO: Sensor SonarJavaXmlFileSensor [java]  
15:39:44.507 INFO: Sensor SonarJavaXmlFileSensor [java] <done> ! time=2ms  
15:39:44.507 INFO: Sensor Analyzer for "php.ini" files [php]  
15:39:44.611 INFO: Sensor Analyzer for "php.ini" files [php] <done> ! time=104ms  
  
15:39:44.612 INFO: Sensor Zero Coverage Sensor  
15:39:44.633 INFO: Sensor Zero Coverage Sensor <done> ! time=21ms  
15:39:44.633 INFO: Sensor CPD Block Indexer  
15:39:44.634 INFO: Sensor CPD Block Indexer <done> ! time=1ms  
15:39:44.634 INFO: Calculating CPD for 0 files  
15:39:44.635 INFO: CPD calculation finished  
15:39:44.707 INFO: Analysis report generated in 71ms, dir size=25 KB  
15:39:44.723 INFO: Analysis reports compressed in 16ms, zip size=6 KB  
15:39:44.723 INFO: Analysis report generated in D:\1-dev\scannerwork\batch-repo  
rt
```

此时打开sonarqube服务器会出现项目的相关信息，如下图所示：



此时我们点开我们测试关注的问题一栏：



所有的BUG就都会在这里。当然sonar还可以从开发角度进行使用，通过在开发工具中安装sonarlint插件，实现代码的即时监控，并将开发项目与sonarqube服务进行绑定，即可以使用sonarqube服务的配置和规则，并且sonar还可以与maven等工具进行集成。大家有兴趣的话可以下去自行研究下。