

## SonarQube Win 安装部署与实践

### 1. SonarQube 简单介绍

SonarQube 能够提供对代码的一整套检查扫描和分析功能，拥有一套服务器端程序，然后再通过客户端或者别的软件的插件的形式完成对各开发环境和软件的支持。

对编程语言的支持非常广泛，包括 C、C++、Java、Objective C、Python、JavaScript、PHP、C#、Swift、Erlang、Groovy 等众多语言  
提供了对 HTML、CSS、JSON、XML、CSV、SQL、JSP/JSF 等类型的文档的支持  
提供了以 FindBugs、PMD、CheckStyle 方式执行代码分析和测试检查的功能  
登录认证方式支持 LDAP、Bitbucket、Azure Active Directory (AAD)、Crowd 等方式  
提供了优美的视图方式下查看代码分析和测试结果报告  
网址：<https://www.sonarqube.org/>

### 2. SonarQube 安装及部署

2.0 在安装 sonarqube 时，必须先安装 jdk，并配置了环境变量

2.1 下载 SonarQube <https://www.sonarqube.org/downloads/>

Download SonarQube below or use it online on [SonarCloud](#) <sup>®</sup>

[System Requirements](#) - [Documentation](#) - [Installation Instructions](#) - [Upgrade Instructions](#) - [License](#)

#### SonarQube 6.4

June 2, 2017

Tag of projects, enhanced "Projects" page with more details/filters and with visualisations, efficient UX for issue multiple locations, more usable global search, core support for private vs. public projects

[Documentation](#) - [Screenshots](#) - [Release notes](#)

DOWNLOAD LATEST

MD5

2.2 解压并放置在任意文件夹下

2.3 进入 bin 目录，然后选择 win\*\*\*\*\*与电脑相应的文件夹，点击 StartSonar.bat 启动启动后，若出现一下界面则表示成功

```
sonar.properties
jvm 1 | 2017.07.17 09:54:03 INFO app[] [o.s.a.SchedulerImpl] Process[ce] is up
jvm 1 | 2017.07.17 09:54:03 INFO app[] [o.s.a.SchedulerImpl] SonarQube is up
```

若没有，则表示启动失败

进入 sonarqube-6.4\sonarqube-6.4\logs\查看 weblog 找到相应的错误并解决。

此时启动成功后，SonarQube 默认使用的是内置的 H2 数据库，当然也可以进行自定义配置，在 sonarqube-6.4\sonarqube-6.4\conf\下，打开 sonar.properties 文件，

```

#-----
# DATABASE
#
# IMPORTANT: the embedded H2 database is used by default. It is recommended for tests but not for
# production use. Supported databases are MySQL, Oracle, PostgreSQL and Microsoft SQLServer.
#
# User credentials.
# Permissions to create tables, indices and triggers must be granted to JDBC user.
# The schema must be created first.
#sonar.jdbc.username=
#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

#----- Oracle 11g/12c
# - Only thin client is supported
# - Only versions 11.2.x and 12.x of Oracle JDBC driver are supported
# - The JDBC driver must be copied into the directory extensions/jdbc-driver/oracle/
# - If you need to set the schema, please refer to http://jira.sonarsource.com/browse/SONAR-5000
#sonar.jdbc.url=jdbc:oracle:thin:@localhost:1521/XE

#----- PostgreSQL 8.x/9.x
# If you don't use the schema named 'public', please refer to http://jira.sonarsource.com/browse/SONAR-5000
#sonar.jdbc.url=jdbc:postgresql://localhost/sonar

#----- Microsoft SQL Server 2012/2014 and SQL Azure
# A database named sonar must exist and its collation must be case-sensitive (CS) and accent-sensitive (AS)
# Use the following connection string if you want to use integrated security with Microsoft Sql Server

```

**注意：**

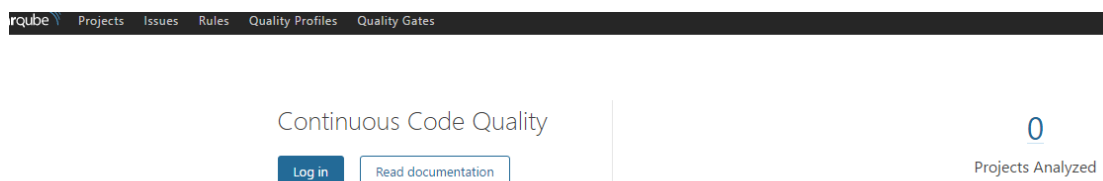
SonarQube server 默认使用的是 9000 端口，如果目前 9000 端口已经占用，进行到 sonarqube-6.4\sonarqube-6.4\conf\下，打开 sonar.properties 文件进行更改端口的设置

```

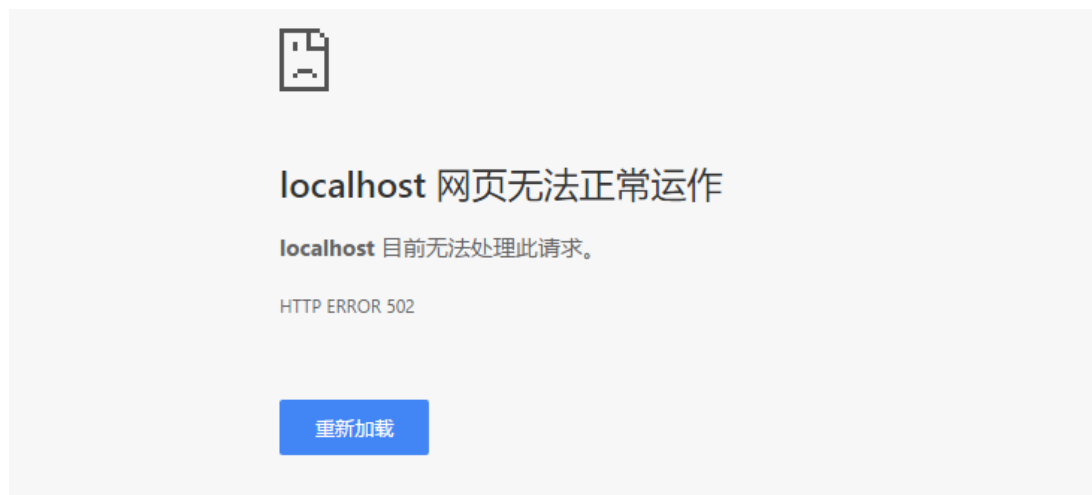
# The context path for the web application. Default value is root context (empty value).
#sonar.web.context=
# TCP port for incoming HTTP connections. Default value is 9000.
#sonar.web.port=9000

```

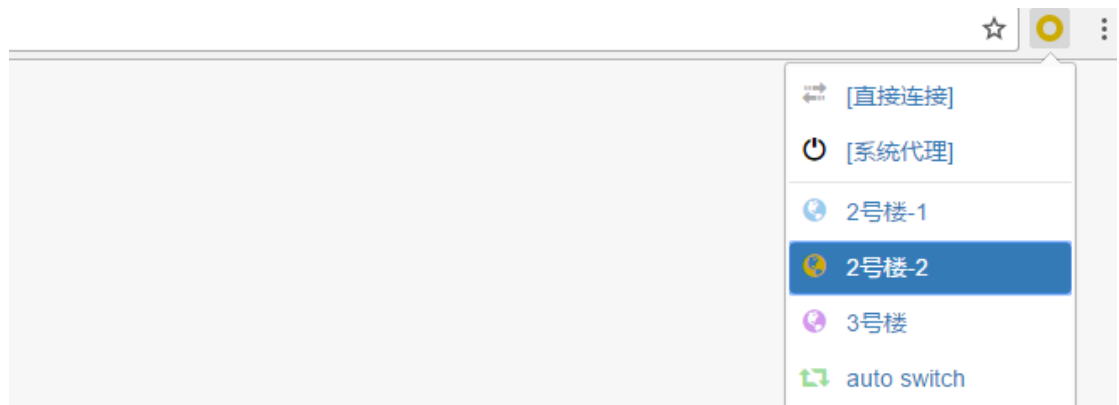
## 2.4 打开浏览器进入 <http://localhost:9000>，将会显示一下界面



此时如果出现以下界面并且 SonarQube 已经启动

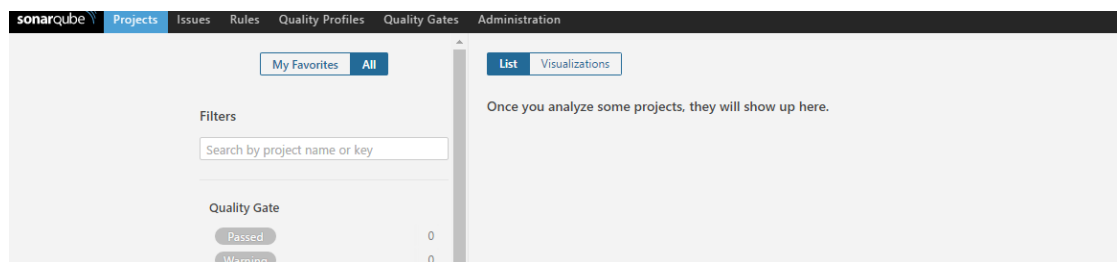


那问题就是出现在了使用了代理服务器访问，此时应该将代理服务器关闭，转为直接链接



再次访问便可以了。

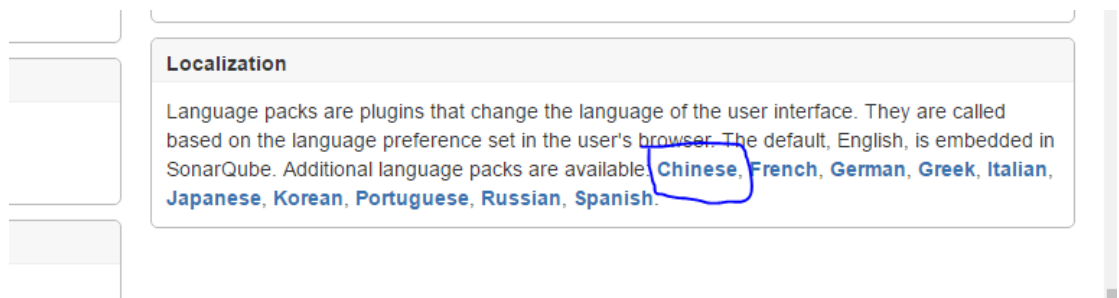
## 2.5 登陆，默认用户名和密码都是 admin



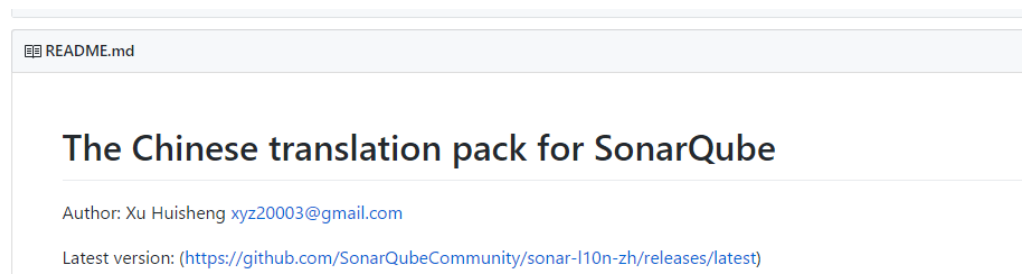
## 2.6 下载中文包插件，汉化网页

链接：<https://docs.sonarqube.org/display/PLUG/Plugin+Library>

在右下角



进入



点击 Lasttest version 对应链接便可以下载，下载成功后，放在 sonarqube-6.4\sonarqube-6.4\extensions\plugins 文件夹下。

## 2.7 重新启动 SonarQube

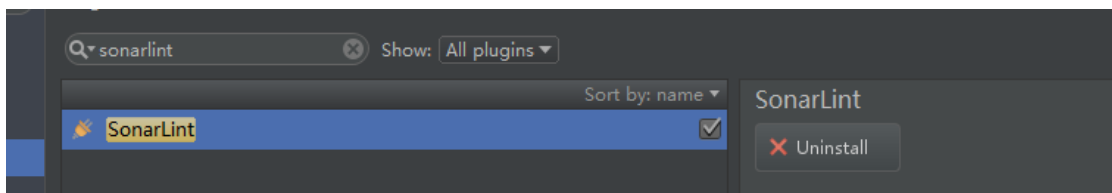
- 1、首先关闭 SonarQube.bat 窗口
- 2、再 Ctrl+Shift+Esc 调出 windows 资源管理器
- 3、在进程中关闭所有 Java.exe 进程
- 4、然后重新进入.\sonarqube-4.5.7\bin\windows-x86-32，运行 StartSonar.bat 文件

2.8 打开 <http://localhost:9000/projects>，已经汉化问中文了

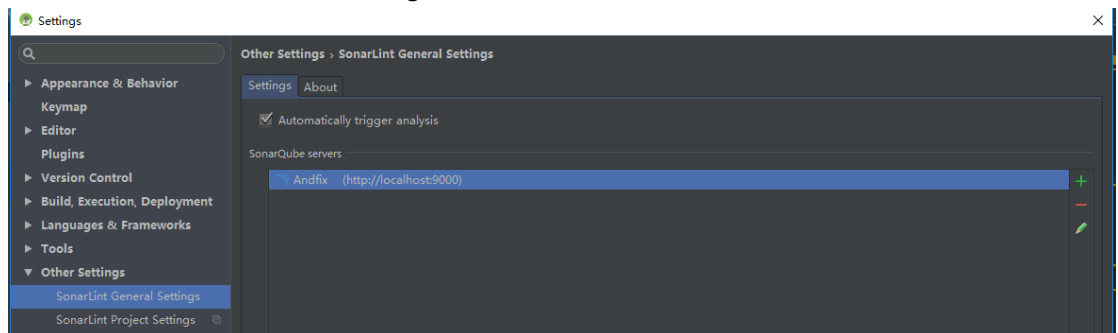


### 3. Android Studio 中配置 SonarLint 与 SonarQube

#### 3.1 安装 SonarLine 插件

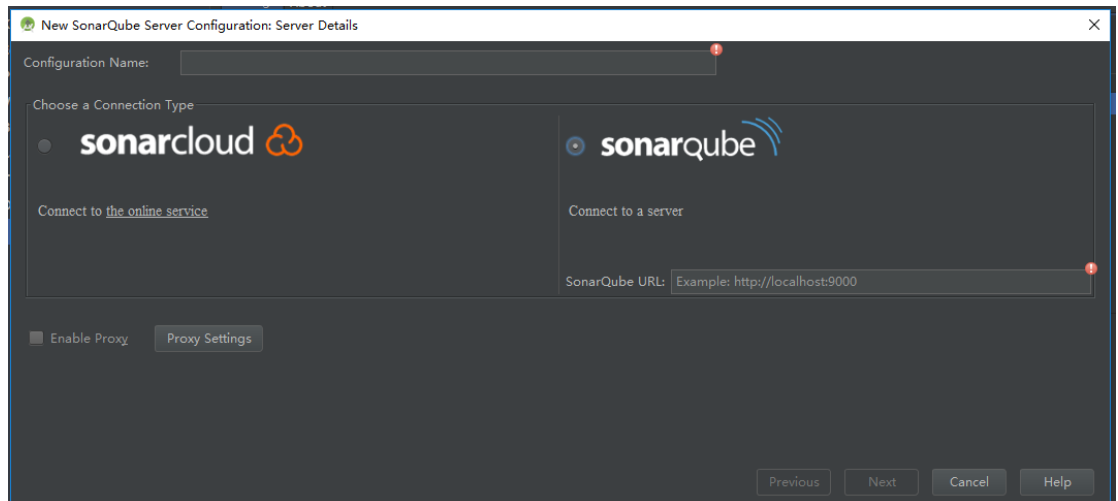


#### 3.2 安装成功后，进入到 Other Setting 界面

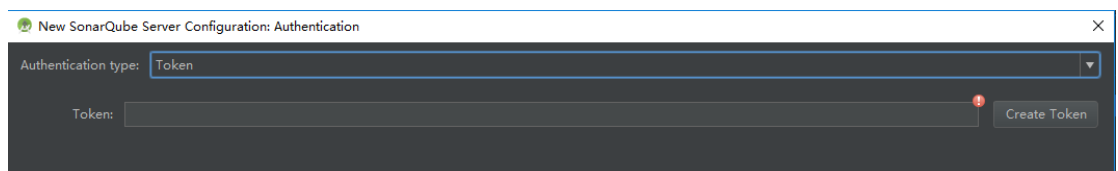


点击+添加 SonarQube 服务器

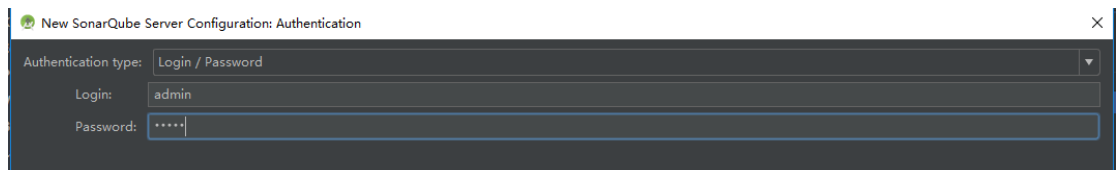
#### 3.3 Configuration Name 就是标志这个服务器的名字



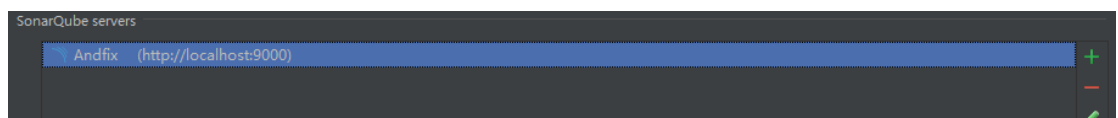
### 3.4 下拉选择 Token 登陆还是 login/password 登陆



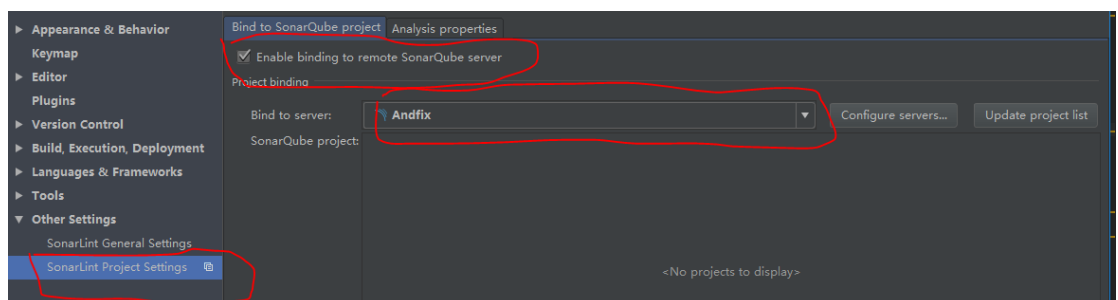
建议使用 login\password 登陆，都填写 admin，这样即使你没有在网页上登陆，依然可以进行分析。



### 3.5 最后点击 next，finish，便看到添加成功



### 3.6 在 Other Setting 中 SonarLint Project Setting 中设置绑定服务器



### 3.7 点击 ok，完成 Android Studio 与 SonarQube 服务器交互配置


#### 4. Android Studio 中 Gradle 配置需要分析的项目

##### 4.1 在需要配置的项目中 build.gradle 文件将以下内容放在最前面

参考链接: <https://plugins.gradle.org/plugin/org.sonarqube>

```
buildscript {  
    repositories {  
        maven {  
            url "https://plugins.gradle.org/m2/"  
        }  
    }  
    dependencies {  
        classpath "org.sonarsource.scanner.gradle:sonarqube-gradle-plugin:2.5"  
    }  
}
```

apply plugin: "org.sonarqube"



```
1 buildscript {  
2     repositories {  
3         maven {  
4             url "https://plugins.gradle.org/m2/"  
5         }  
6     }  
7     dependencies {  
8         classpath "org.sonarsource.scanner.gradle:sonarqube-gradle-plugin:2.5"  
9     }  
10 }  
11  
12  
13 apply plugin: "org.sonarqube"
```

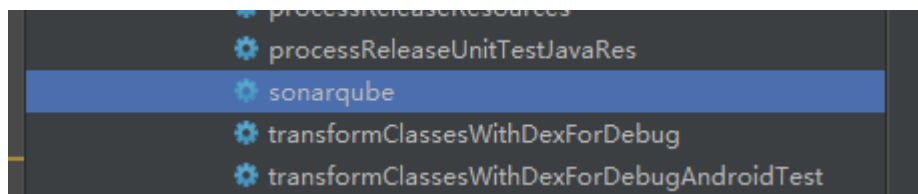
这里有一种更加便捷的方式

Build script snippet for new, incubating, plugin mechanism introduced in Gradle 2.1:

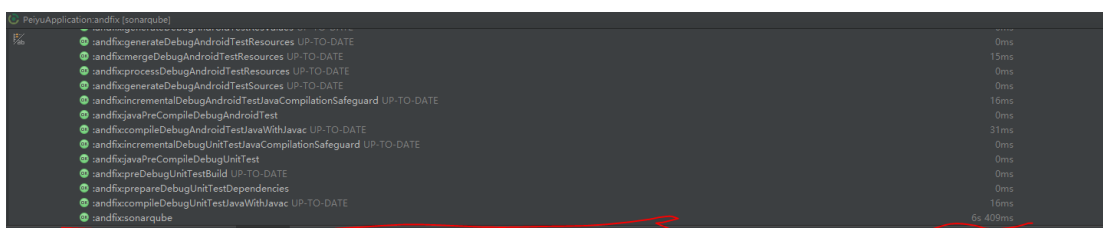
```
plugins {  
    id "org.sonarqube" version "2.5"  
}
```

但是这并不是所有的 gradle 版本都适用, 因此推荐使用最上面的版本。

##### 4.2 同步之后, 在 As 的右边, 打开 Gradle 栏, 在自己配置的项目或者 module 中打开, 选择 other 打开, 下拉便可以看到 sonarqube



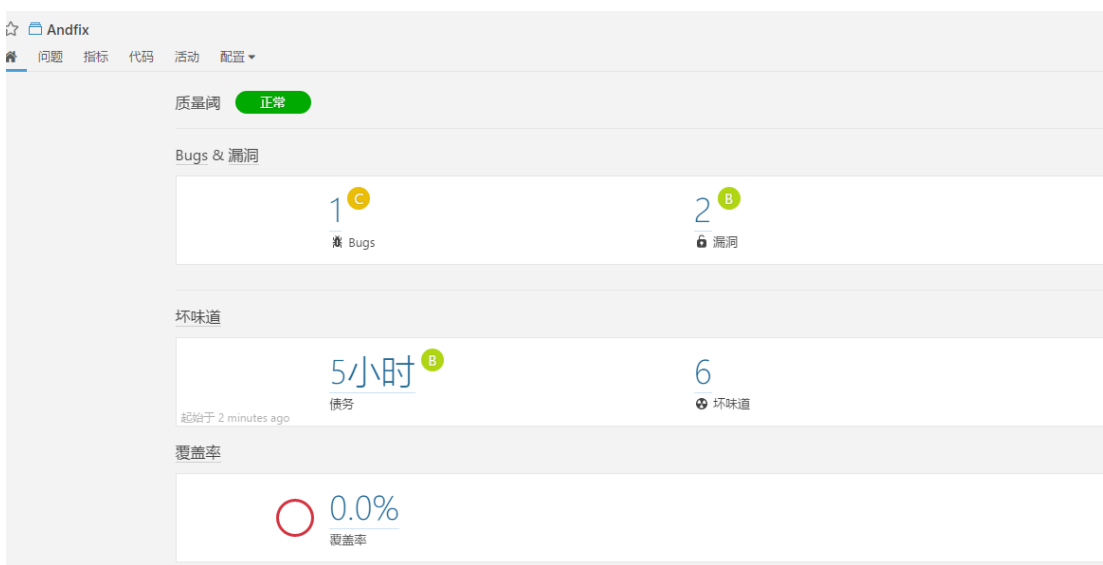
#### 4.3 点击，便会分析



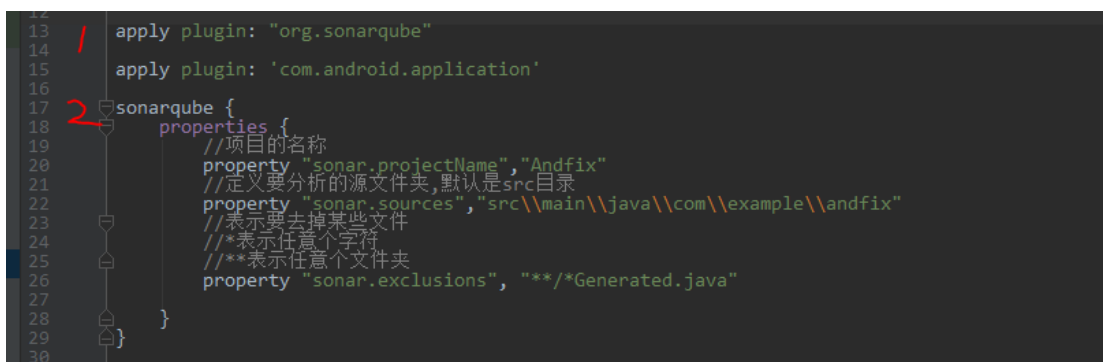
#### 4.4 分析完成后，刷新 <http://localhost:9000/projects>



便可以看到刚才分析的项目，分析结果可视化的展示，非常直观。



4.5 在实际分析项目中，可能需要指定，哪些文件或者文件夹不需要分析，哪些需要，这些的配置是在刚才添加插件的 build.gradle 文件完成



注意：

必须要先应用插件，才可以配置 sonarqube

4.6 在上面配置 sonarqube 属性中，只是使用了  
sonar.projectName 表示在 sonarqube 中显示的项目名字  
sonar.sources 要分析的文件路径  
sonar.exclusions 不分析某些文件  
更多配置请详细看

<https://docs.sonarqube.org/display/SONAR/Analysis+Parameters>

注意：

1. 文件的路径可以不使用绝对路径，src 开头即可
2. 在没有配置相应属性时，默认分析的文件夹时 src，详情可以看上面的链接。
3. sonar.exclusions 和 sonar.inclusions 等属性配置不分析的文件中，或者要分析的文件中，使用了一个匹配模式

## Patterns

Paths are relative to the project base directory.

The following wildcards can be used:

Wildcard	Matches
*	zero or more characters
**	zero or more directories
?	a single character

举例：

### Examples

```
# Exclude all classes ending by "Bean"
# Matches org/sonar/api/MyBean.java, org/sonar/util/MyOtherBean.java, org/sonar/util/MyDTO.java, etc.
sonar.exclusions=**/*Bean.java,**/*DTO.java

# Exclude all classes in the "src/main/java/org/sonar" directory
# Matches src/main/java/org/sonar/MyClass.java, src/main/java/org/sonar/MyOtherClass.java
# But does not match src/main/java/org/sonar/util/MyClassUtil.java
sonar.exclusions=src/main/java/org/sonar/*

# Exclude all COBOL programs in the "bank" directory and its sub-directories
# Matches bank/ZTR00021.cbl, bank/data/CBR00354.cbl, bank/data/REM012345.cob
sonar.exclusions=bank/**/*

# Exclude all COBOL programs in the "bank" directory and its sub-directories whose extension is .cbl
# Matches bank/ZTR00021.cbl, bank/data/CBR00354.cbl
sonar.exclusions=bank/**/*.*cbl
```

详情可看：<https://docs.sonarqube.org/display/SONAR/Narrowing+the+Focus>

4.7 可能还有一种情况，sonarqube 插件配置的是整个项目，而项目中有很多个 module，此时可以使用



## Analyzing Multi-Project Builds

To analyze a project hierarchy, apply the SonarQube plugin to the root project of the hierarchy. Typically (but not necessarily) this will be the root project of the Gradle build. Information pertaining to the analysis as a whole has to be configured in the `sonarqube` block of this project. Any properties set on the command line also apply to this project.

### Global configuration settings

```
build.gradle

sonarqube {
    properties {
        property "sonar.sourceEncoding", "UTF-8"
    }
}
```

### Shared configuration settings

Configuration shared between subprojects can be configured in a `subprojects` block.

```
build.gradle

subprojects {
    sonarqube {
        properties {
            property "sonar.sources", "src"
        }
    }
}
```

### Individual configuration settings

Project-specific information is configured in the `sonarqube` block of the corresponding project.

```
build.gradle

project(":project1") {
    sonarqube {
        properties {
            property "sonar.branch", "Foo"
        }
    }
}
```

### Skipping analysis of a project

To skip SonarQube analysis for a particular subproject, set `sonarqube.skipProject` to `true`.

```
build.gradle

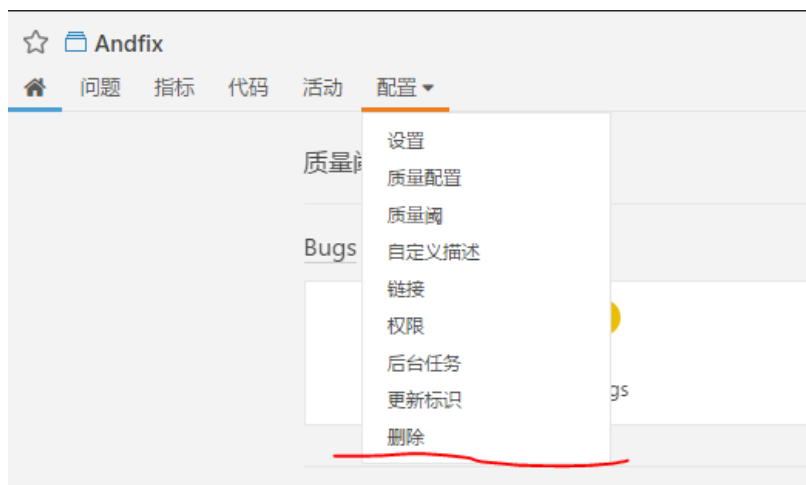
project(":project2") {
    sonarqube {
        skipProject = true
    }
}
```

详情请看：

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

## 5. 注意

1. 如果 Android studio 已经分析了一个项目，在 sonarqube server 中已经保存，当要再进行分析时，需要先删除 sonarqube server 中的项目，然后再进行分析



2. 在 bin 下点击 StartSonar.bat 时，如果启动失败，查看 log，找到相应的 exception 并解决
3. Sonarqube 默认使用的是 H2 内置数据库，推荐使用 MySQL，或者 Oracle 等数据库。

## 6. 结束