SonarGo

Could not retrieve http://update.sonarsource.org/plugins/go-confluence-include.html - Page not found.

Description

Enables the powerful SonarGo analyzer to scan GoLang 1.0+ files.

Prerequisites

- SonarQube Scanner should run on a x86-64 Windows, macOS or Linux 64bits machine
- You need the Go installation on the scan machine only if you want to import coverage data

First Analysis of a Go Project

1. Install SonarQube Server (see Setup and Upgrade for more details)
2. Install SonarQube Scanner and be sure your can call *sonar-scanner* from the directory where you have your source code
3. Install SonarGo (see Installing a Plugin for more details)
4. Run your analysis with the SonarQube Scanner by executing the following command from the root directory of the project:

   ```
   sonar-scanner -Dsonar.projectKey=xxx -Dsonar.sources=.
   ```

5. Follow the link provided at the end of the analysis to browse your project's quality in SonarQube UI

Further Analyses

Assuming steps 1-3 above have already been completed, you'll want to encapsulate your analysis parameters in a *sonar-project.properties* file at the root of your project (see a sample project on GitHub: https://github.com/SonarSource/sonar-scanning-examples/tree/master/sonarqube-scanner). Then subsequent analyses can simply be run with:

```
sonar-scanner
```

Here is a good first version of a sonar-project.properties, correctly excluding some "vendor" directories and categorizing files as "main" or "test":

```
sonar.projectKey=com.company.projectkey1
sonar.projectName=My Project Name


sonar.sources=.
sonar.exclusions=**/*_test.go,**/vendor/**


sonar.tests=.
sonar.test.inclusions=**/*_test.go
sonar.test.exclusions=**/vendor/**
```

Advanced Usage

With SonarGo, you can also:

- import Coverage Results
- import Unit Tests Results Import
- import GoVet, GoLint and GoMetaLinter issues reports