

Project: SpringMvc

The screenshot shows the SonarQube web interface at `localhost:9000/projects`. The 'Projects' tab is active. The 'SpringMvc' project is listed with an overall status of 'Passed'. The project details show 6 Bugs (E), 4 Vulnerabilities (B), 16 Code Smells (A), 0.0% Coverage, and 0.0% Duplications. The last analysis was on January 24, 2019, at 11:24 AM. The project is associated with 344 lines of code (Java, XML). A warning message states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

Quality Gate: Passed

Reliability (Bugs): 0 | 0 | 0 | 0 | 1

Security (Vulnerabilities): 0 | 1 | 0 | 0 | 0

1 of 1 shown

Community Edition - Version 7.5 (build 20543) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About

The screenshot shows the SonarQube project dashboard for 'SpringMvc' at `localhost:9000/dashboard?id=SpringMvc`. The 'Overview' tab is active. The project is in the 'master' branch. The last analysis was on January 24, 2019, at 11:24 AM, with a version of 1.0. The project has 2 warnings. The dashboard shows the following metrics:

- Quality Gate: Passed
- Bugs: 6 (E)
- Vulnerabilities: 4 (B)
- Code Smells: 16 (A)
- Coverage: 0.0%

Project Activity: January 24, 2019, 1.0

Quality Gate: (Default) Sonar way

Quality Profiles: (Java) Sonar way, (XML) Sonar way

Project Key:

Bugs:

The screenshot shows the SonarQube web interface for the 'SpringMvc' project. The 'Issues' tab is selected, displaying a list of bugs. The left sidebar shows filters for Type (Bug, Vulnerability, Code Smell, Security Hotspot) and Severity (Blocker, Critical, Major, Minor, Info). The main panel lists six bugs, all with a severity of 'Blocker' and a status of 'Open'. Each bug has a description, a severity level (L55 to L107), and a list of tags (cert, cwe, denial-of-service, leak). The top of the interface shows the project name 'SpringMvc' and the master branch. The bottom of the interface shows the Windows taskbar with various application icons and the system clock.

localhost:9000/project/issues?id=SpringMvc&resolved=false&types=BUG

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

SpringMvc master

Last analysis had 2 warnings January 24, 2019, 11:24 AM Version 1.0

Overview Issues Security Reports Measures Code Activity Administration

My Issues All

Filters Clear All Filters

Type Bug Reset

Severity Blocker 6 Minor 0 Critical 0 Info 0 Major 0

Resolution Status

Bulk Change

src/main/java/com/accolite/springmvc/dao/UserDao.java

Use try-with-resources or close this "PreparedStatement" in a "finally" clause. *** 2 hours ago L55 cert, cwe, denial-of-service, leak

Use try-with-resources or close this "PreparedStatement" in a "finally" clause. *** 2 hours ago L70 cert, cwe, denial-of-service, leak

Use try-with-resources or close this "ResultSet" in a "finally" clause. *** 2 hours ago L73 cert, cwe, denial-of-service, leak

Use try-with-resources or close this "Statement" in a "finally" clause. *** 2 hours ago L90 cert, cwe, denial-of-service, leak

Use try-with-resources or close this "ResultSet" in a "finally" clause. *** 2 hours ago L91 cert, cwe, denial-of-service, leak

Use try-with-resources or close this "PreparedStatement" in a "finally" clause. *** 2 hours ago L107 cert, cwe, denial-of-service, leak

Maintainability:

The screenshot shows the SonarQube web interface for the 'SpringMvc' project. The 'Measures' tab is selected, displaying the Maintainability Rating. The left sidebar shows the Project Overview with sections for Reliability, Security, Maintainability, Coverage, Duplications, Size, Complexity, and Issues. The main panel shows the Maintainability Rating for the SpringMvc project, with a green 'A' rating. Below the rating, a list of files is shown, each with a green 'A' rating. The top of the interface shows the project name 'SpringMvc' and the master branch. The bottom of the interface shows the Windows taskbar with various application icons and the system clock.

localhost:9000/component_measures?id=SpringMvc&metric=sqale_rating

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

SpringMvc master

Last analysis had 2 warnings January 24, 2019, 11:24 AM Version 1.0

Overview Issues Security Reports Measures Code Activity Administration

Project Overview

Reliability

Security

Maintainability Maintainability R...

Overview

Overall

Code Smells 16

Debt 1h 53min

Debt Ratio 1.1%

Rating A

Effort to Reach A 0

Coverage

Duplications

Size

Complexity

Issues

SpringMvc

Maintainability Rating A

src/main/java/com/accolite/springmvc/config/DaoBeanConfig.java A

src/main/java/com/accolite/springmvc/indto/LoginUserDto.java A

src/main/java/com/accolite/springmvc/config/MainController.java A

src/main/java/com/accolite/springmvc/controller/SampleController.java A

src/main/java/com/accolite/springmvc/config/ServiceBeanConfig.java A

src/main/java/com/accolite/springmvc/model/User.java A

src/main/java/com/accolite/springmvc/controller/UserController.java A

src/main/java/com/accolite/springmvc/dao/UserDao.java A

src/main/java/com/accolite/springmvc/service/UserService.java A

src/main/webapp/WEB-INF/web.xml A

Code Smells:

The screenshot shows the SonarQube web interface for a project named 'SpringMVC'. The 'Issues' tab is selected, displaying a list of 1/26 issues on the left. The main area shows the source code for 'src/_/com/accolite/springmvc/config/WebServletConfig.java'. A red box highlights a code smell: 'This block of commented-out lines of code should be removed.' The code includes imports for Spring Framework classes and a class definition for 'WebServletConfig'.

Issues list:

- Define a constant instead of duplicating this literal "userName" 3 times. (Code Smell, Major)
- Define a constant instead of duplicating this literal "index.jsp" 3 times. (Code Smell, Critical)
- Define a constant instead of duplicating this literal "message" 5 times. (Code Smell, Critical)

Code snippet (WebServletConfig.java):

```
4 import org.springframework.context.annotation.ComponentScan;
5 import org.springframework.context.annotation.Configuration;
6 import org.springframework.web.servlet.ViewResolver;
7 import org.springframework.web.servlet.config.annotation.EnableWebMvc;
8 import org.springframework.web.servlet.config.annotation.WebMvcConfigurationSupport;
9 import org.springframework.web.servlet.view.InternalResourceViewResolver;
10
11 @Configuration
12 @EnableWebMvc
13 @ComponentScan({"com.accolite.springmvc"})
14 public class.WebServletConfig extends WebMvcConfigurationSupport {
15     // @Bean
16     // public ViewResolver getViewResolver() {
17         // InternalResourceViewResolver resolver = new InternalResourceViewResolver();
18         // resolver.setPrefix("/WEB-INF/views/");
19         // resolver.setSuffix(".jsp");
20         // return resolver;
21     }
22 }
```

The screenshot shows the SonarQube web interface for the 'SpringMVC' project, specifically the 'Measures' tab. The 'Project Overview' sidebar on the left shows various metrics, including 'Code Smells' (16). The main area displays the source code for 'src/_/springmvc/controller/UserController.java'. A red box highlights a code smell: 'Define a constant instead of duplicating this literal "userName" 3 times.' The code includes imports for Spring Framework classes and a class definition for 'UserController'.

Project Overview Metrics:

- Reliability: 0
- Security: 0
- Maintainability: 0
- Code Smells: 16
- Debt: 1h 53min
- Debt Ratio: 1.1%
- Rating: A
- Effort to Reach A: 0
- Coverage: 0
- Duplications: 0
- Size: 0
- Complexity: 0
- Issues: 0

Code snippet (UserController.java):

```
13 import org.springframework.web.bind.annotation.RequestMethod;
14 import org.springframework.web.bind.annotation.RequestParam;
15 import org.springframework.web.servlet.ModelAndView;
16
17 import com.accolite.springmvc.indto.LoginUserDto;
18 import com.accolite.springmvc.model.User;
19 import com.accolite.springmvc.service.UserService;
20
21 @Controller
22 public class UserController {
23     @Autowired
24     UserService userService;
25
26     @RequestMapping(value="/register",method=RequestMethod.POST)
27     ModelAndView registerUser(HttpServletRequest request,HttpServletResponse response) {
28         User user=new User();
29         user.setUsername(request.getParameter("userName"));
30
31         user.setPassword(request.getParameter("password"));
32         user.setAge(Integer.parseInt(request.getParameter("age")));
33     }
34 }
```