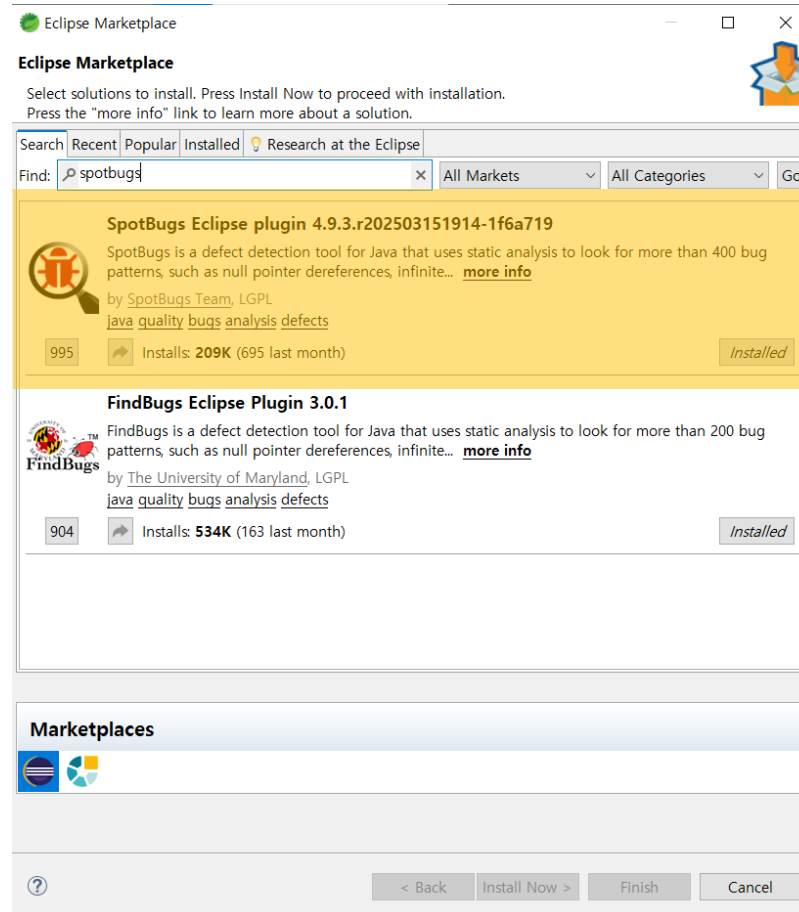
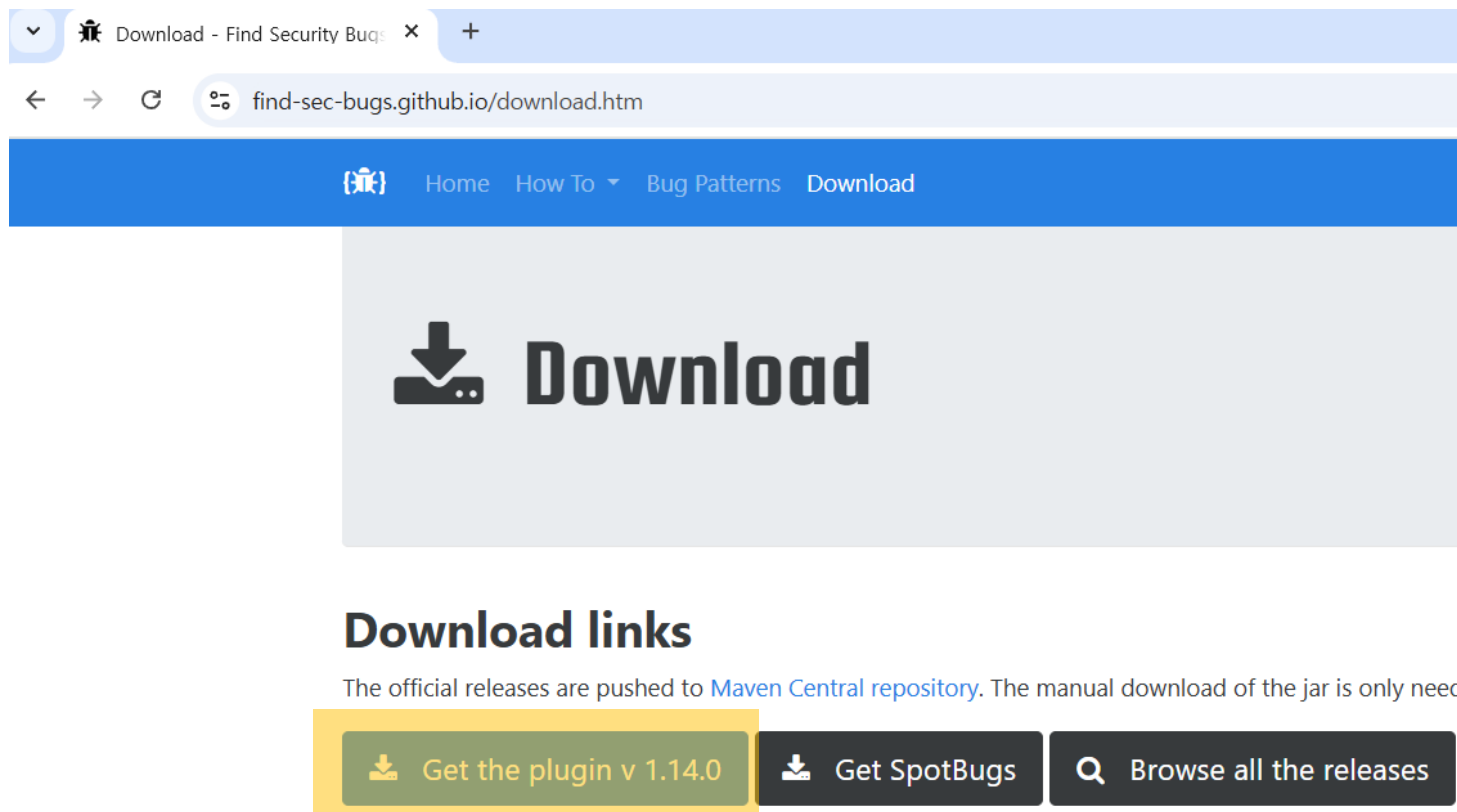


Spotbugs 설치



Spotbugs만 설치할 것

- <https://find-sec-bugs.github.io/download.htm>

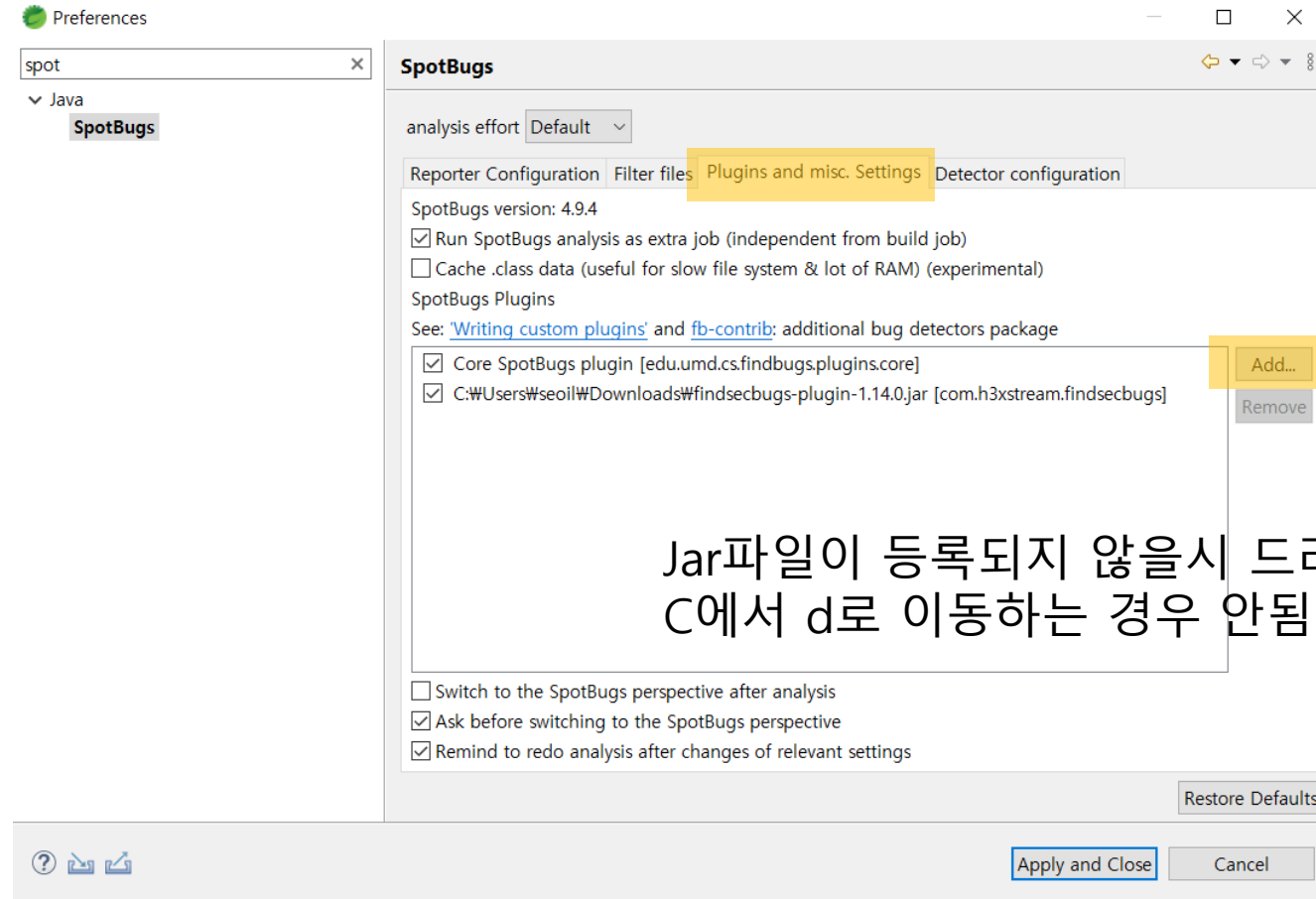


<https://yourusername.tistory.com/19>

Not sure how to integrate the plugin? Check the [Tutorial section](#)

<https://myanjini.tistory.com/entry/FindBugs-%EC%84%A4%EC%B9%98>

플러그인 등록



spot

Java

SpotBugs

SpotBugs

analysis effort Maximal

Reporter Configuration Filter files Plugins and misc. Settings Detector configuration

☐ Merge similar warnings

Minimum rank to report:
(1 is most severe, 20 is least)



20 (Of Concern)

Minimum confidence to report: Low

Reported (visible) bug categories

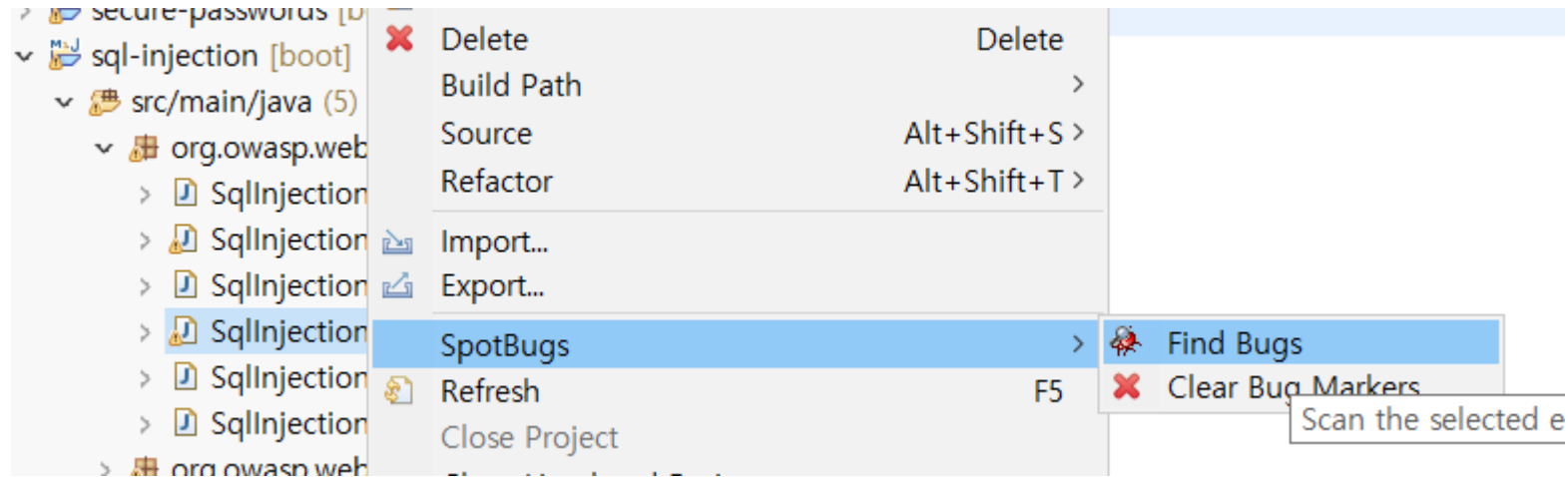
- ☒ Bad practice
- ☐ Malicious code vulnerability
- ☒ Correctness
- ☒ Performance
- ☒ Security
- ☒ Dodgy code
- ☐ Experimental
- ☒ Multithreaded correctness
- ☐ Internationalization

Mark bugs with ... rank as:

- Scariest: Warning
- Scary: Warning
- Troubling: Warning
- Of concern: Warning

Restore Defaults

툴 설치방법



The screenshot shows an IDE interface. On the left, a project tree under 'sql-injection (5)' shows a hierarchy of folders: 'Scary (1)' containing 'Normal confidence (1)' and 'Potential JDBC Injection (1)'. The 'Potential JDBC Injection (1)' folder is expanded, showing 'This use of java/sql/Statement.executeQuery(Ljava/lang/String;)Ljava/sql/ResultSet;'. Below it are 'Troubling (3)', 'High confidence (1)', 'Nonconstant string literals (1)' (containing 'org.owasp.webgoat'), 'Low confidence (2)', and 'Of Concern (1)'. The main code editor displays the following Java code:

```
66         ResultSet.CONCUR_READ_ONLY)) {  
67         ResultSet results = statement.executeQuery(query);  
68  
69         if ((results != null) && (results.first())) {  
70             ResultSetMetaData resultsMetaData = results.getMetaData();  
71             StringBuffer output = new StringBuffer();  
72         }
```

Below the code editor, the 'Bug Reviews' panel is open, showing a bug titled 'Could not create the view: de.toobject.findbugs.view.userannotationsview'. The 'Bug Info' tab is selected, displaying the following information:

- File: SqlInjectionLesson6a.java: 67
- Navigation: This use of java/sql/Statement.executeQuery(Ljava/lang/String;)Ljava/sql/ResultSet; can be vulnerable to SQL injection (with JDBC). Sink method java/sql/Statement.executeQuery(Ljava/lang/String;)Ljava/sql/ResultSet;. Sink parameter 0.
- Vulnerable Code:

```
Connection conn = [...];  
Statement stmt = conn.createStatement();  
ResultSet rs = stmt.executeQuery("update COFFEES set SALES = "+nbSales+" where COF_NAME = '"+coffeeName+"'");
```
- Solution:

```
Connection conn = [...];  
conn.prepareStatement("update COFFEES set SALES = ? where COF_NAME = ?");  
updateSales.setInt(1, nbSales);  
updateSales.setString(2, coffeeName);
```

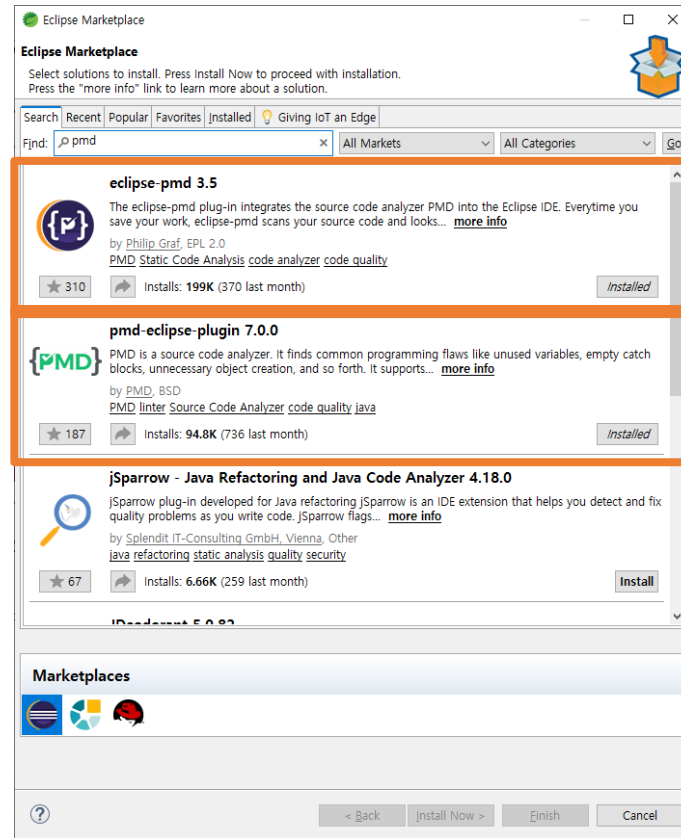
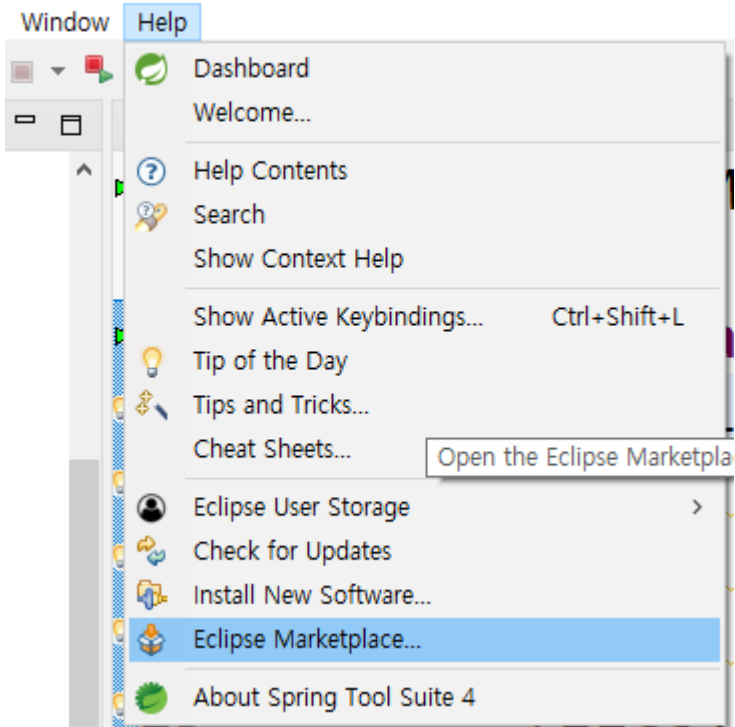
익스플로어에서 최대한 확장을 한 후 마지막 코드 클릭하면 buf info나타 남.
이 정보에 취약한 코드와 해결방법이 나타남

코드인스펙션 pmd

코드점검

프로그램 설치

unittest/Main.java - Spring Tool Suite 4

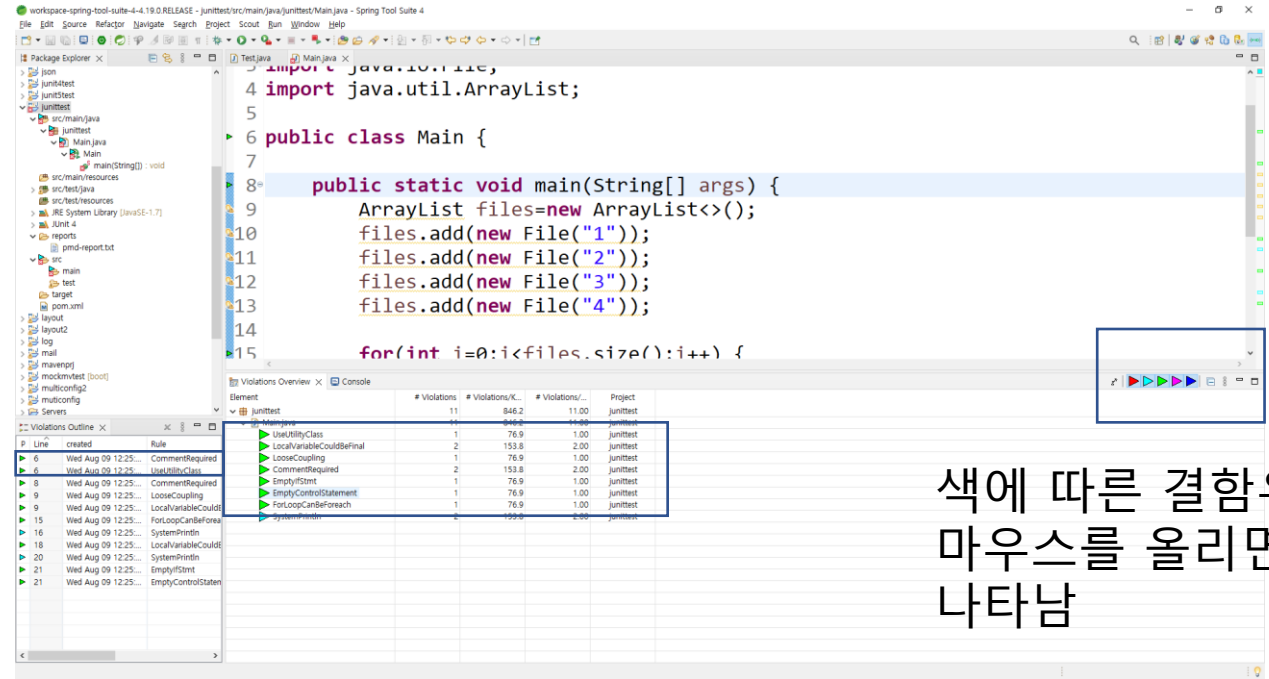
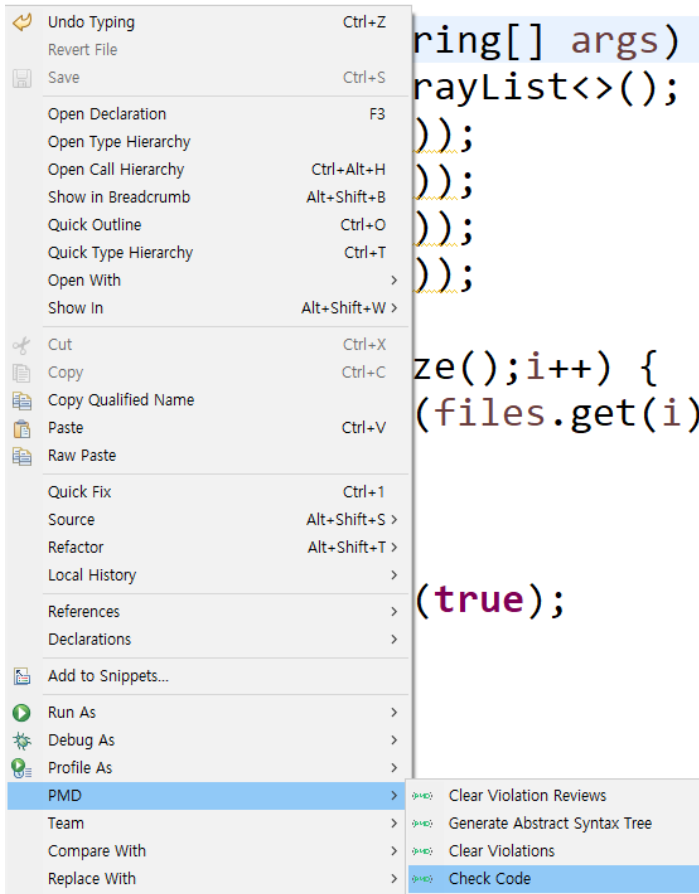


Pmd, plugin 두개 모두 설치한다.

문제 코드 작성

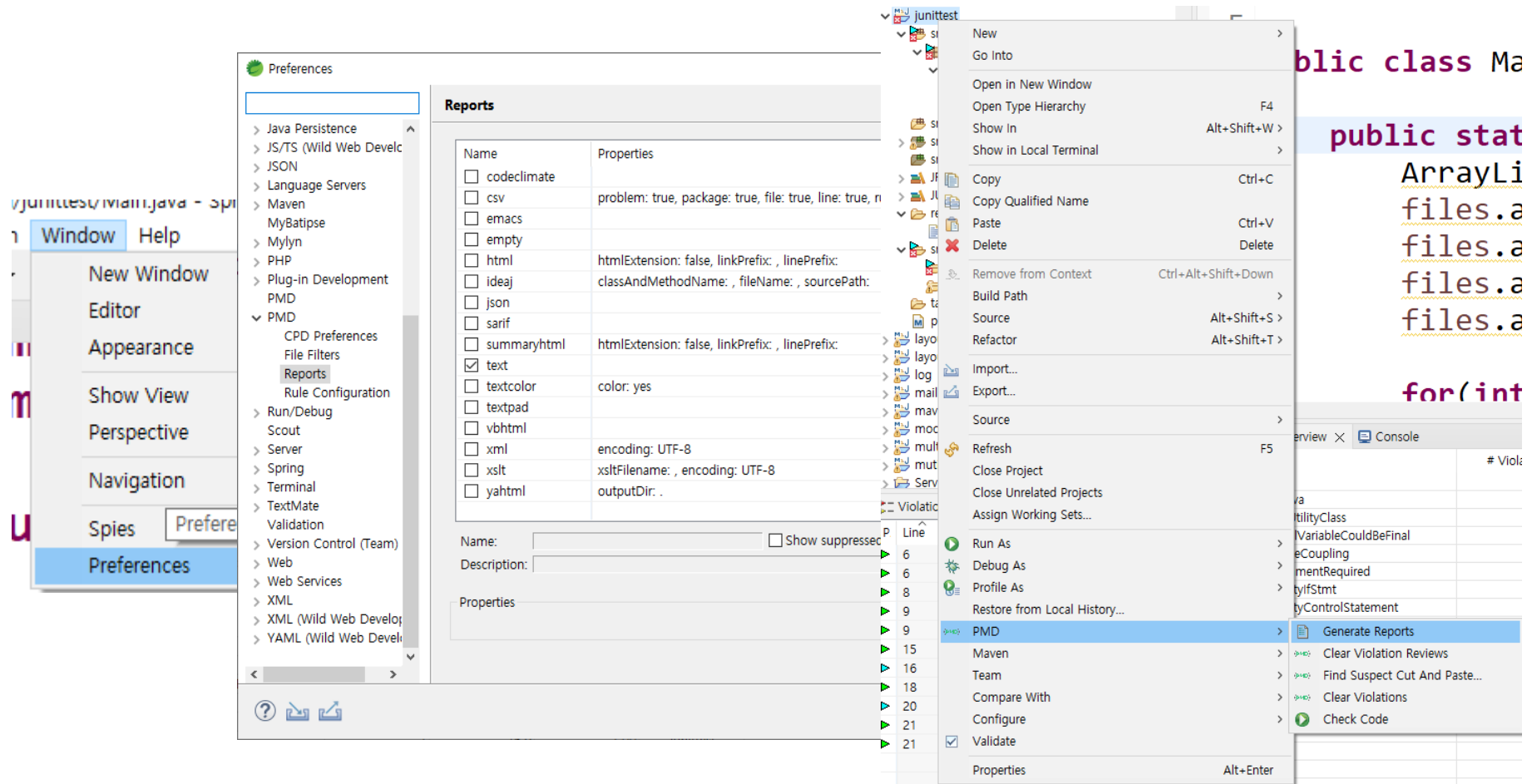
```
public class Main {  
  
    public static void main(String[] args) {  
        ArrayList files=new ArrayList<>();  
        files.add(new File("1"));  
        files.add(new File("2"));  
        files.add(new File("3"));  
        files.add(new File("4"));  
        for(int i=0;i<files.size();i++) {  
            System.out.println(files.get(i));  
        }  
        boolean state=true;  
        if(state) {  
            System.out.println(true);  
        }else { }  
    }  
}
```

코드 체크하기



색에 따른 결함유형확인
마우스를 올리면 영어말
나타남

코드 리포트 만들기



설정을 통해 문서를 어떤 문서로 만들것인지 선택하고
프로젝트에서 우측클릭하여 리포트를 생성한다.

리포트 생성

The screenshot shows the Spring Tool Suite 4 interface. The main editor displays a list of 12 PMD violations in Main.java. The violations are as follows:

Line	Violation	Message
1	src/main/java/junittest/Main.java:6:	CommentRequired: Class c
2	src/main/java/junittest/Main.java:6:	UseUtilityClass: UseUtilityClass: This ut
3	src/main/java/junittest/Main.java:8:	CommentRequired: CommentRequired: Public
4	src/main/java/junittest/Main.java:9:	LocalVariableCouldBeFinal: LocalVariableCou
5	src/main/java/junittest/Main.java:9:	LooseCoupling: LooseCoupling: Avoid using i
6	src/main/java/junittest/Main.java:15:	ForLoopCanBeForeach: ForLoopCanBeForeach:
7	src/main/java/junittest/Main.java:16:	SystemPrintln: SystemPrintln: Usage of Syst
8	src/main/java/junittest/Main.java:18:	LocalVariableCouldBeFinal: LocalVariableCou
9	src/main/java/junittest/Main.java:20:	SystemPrintln: SystemPrintln: Usage of Syst
10	src/main/java/junittest/Main.java:21:	EmptyControlStatement: EmptyControlStatemen
11	src/main/java/junittest/Main.java:21:	EmptyIfStmt: EmptyIfStmt: Avoid empty if
12		

The bottom pane shows a 'Violations Overview' table:

Element	# Violations	# Violations/K	# Violations/...	Project
junitest	11	846.2	11.00	junitest
Main.java	11	846.2	11.00	junitest
UseUtilityClass	1	76.9	1.00	junitest
LocalVariableCouldBeFinal	2	153.8	2.00	junitest
LooseCoupling	1	76.9	1.00	junitest
CommentRequired	2	153.8	2.00	junitest
EmptyIfStmt	1	76.9	1.00	junitest
EmptyControlStatement	1	76.9	1.00	junitest
ForLoopCanBeForeach	1	76.9	1.00	junitest
SystemPrintln	2	153.8	2.00	junitest

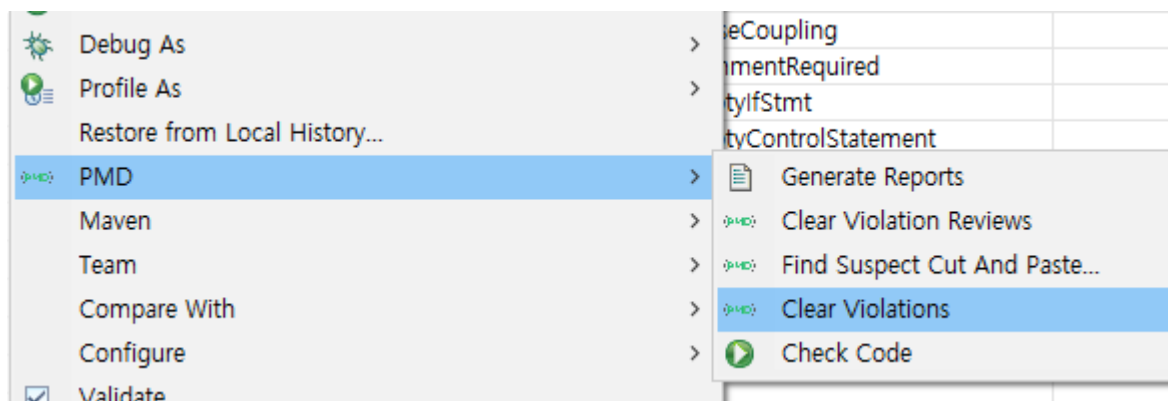
프로젝트에 reports폴더가 생성되고 문서가 안에 포함되어 있다.

코드에 따라 색깔이 다르다.

- 빨강 (High : Blocker) : 심각한 버그가 발생할 수 있는 코드이기 때문에 반드시 Rule 을 준수해야 한다.
- 하늘 (Medium : High : Critical) : 심각하지는 않지만 버그가 발생할 수 있는 코드이기 때문에 Rule 을 준수해야 한다.
- 초록 (Medium : Urgent) : 복잡한 코딩, Best Practice 및 보안, 성능 등에 관련 된 내용으로 준수 할 것을 권장한다.
- 분홍 (Medium Low : Important) : 버그가 아니며 표준, 코딩 스타일, 불필요한 코드 및 미사용 코드에 관련된 내용이다.
- 파랑 (Low : Warning) : 패키지, 클래스, 필드 등 Naming에 관련된 내용이다.

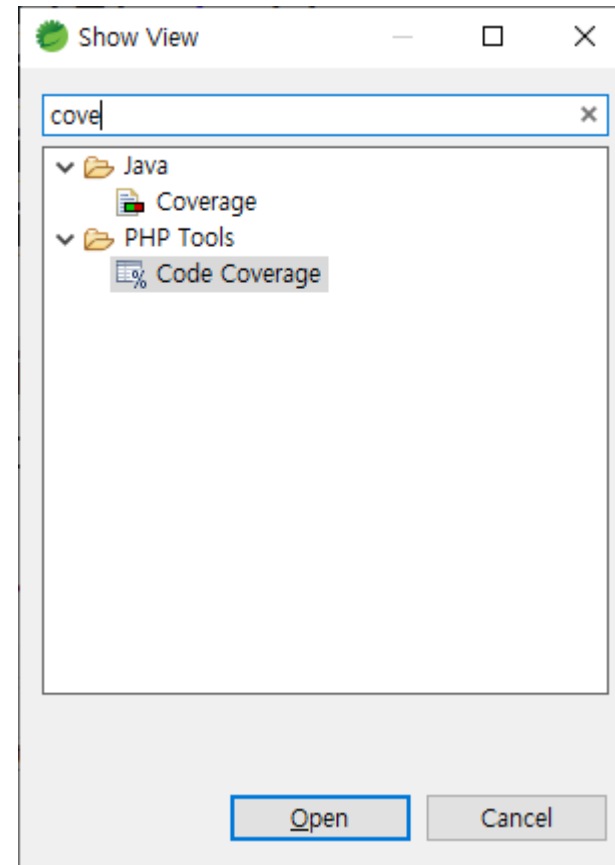
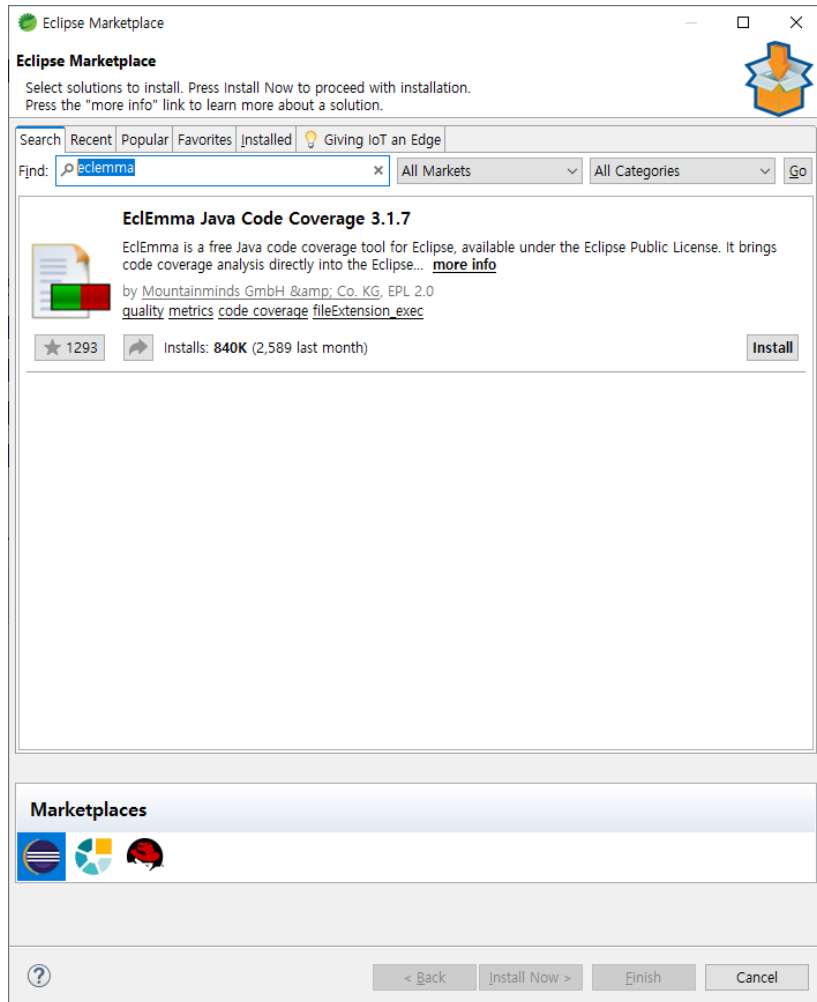


색을 클릭하여 해제하면 해당 코드는 보이지 않는다.

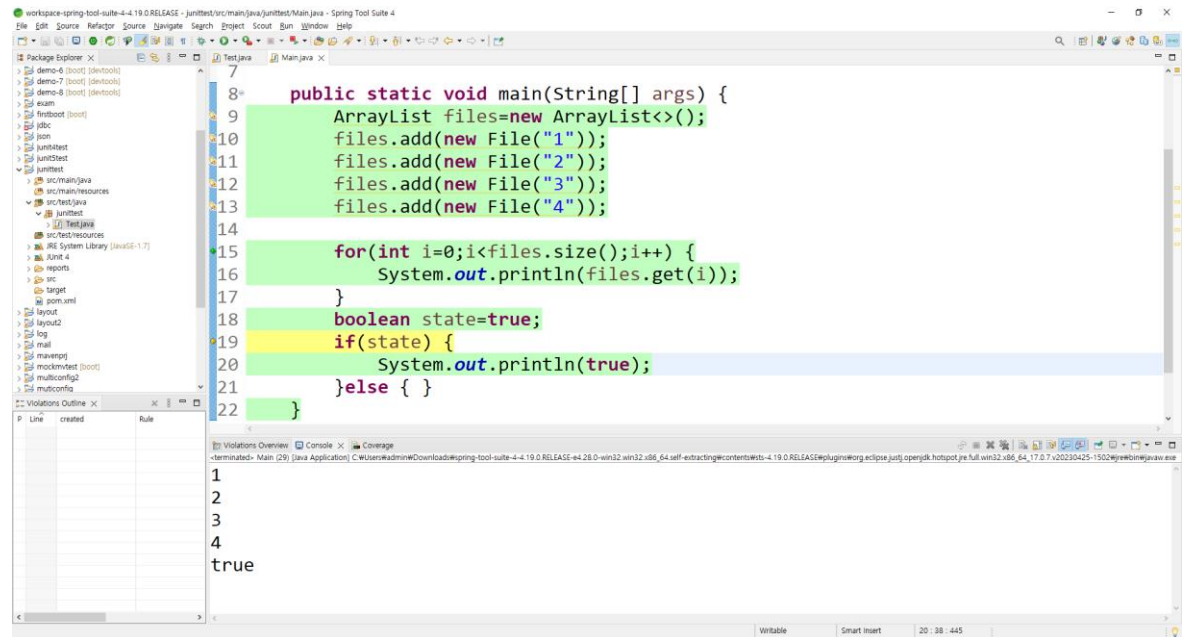
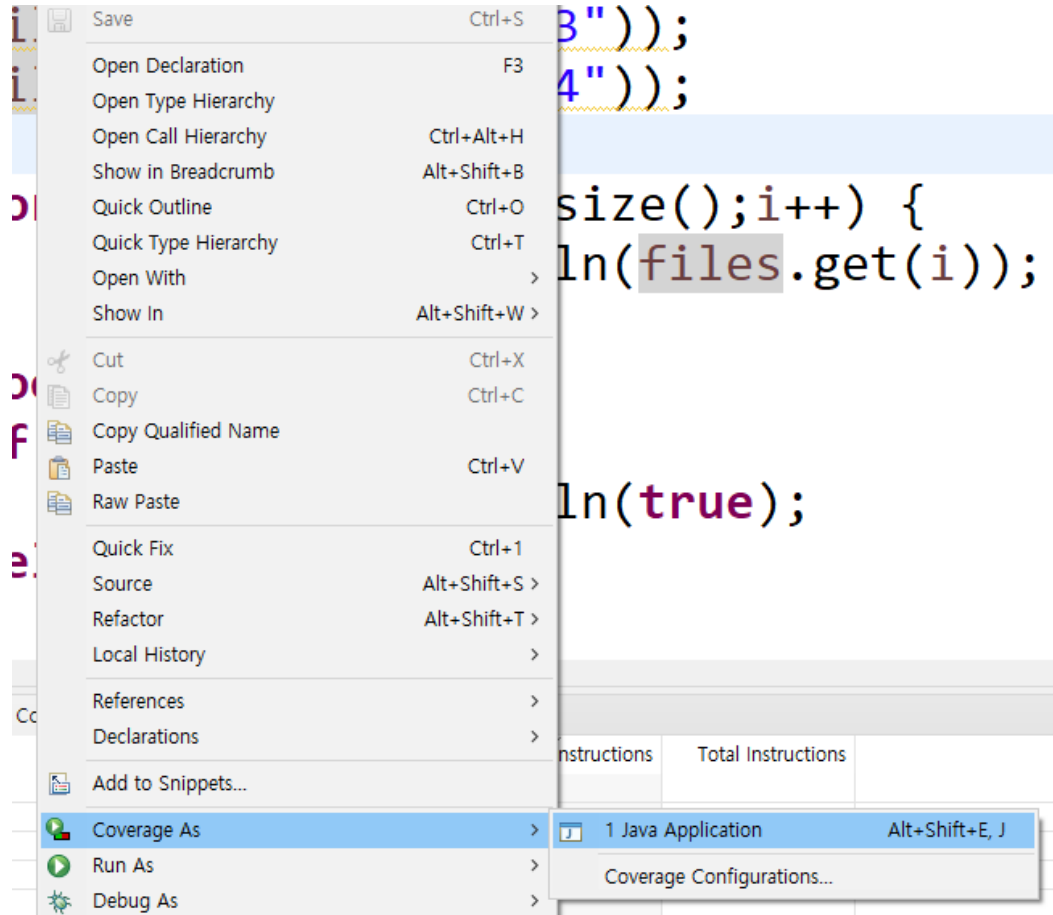


Code coverage 설치

<https://www.eclemma.org/>



Code coverage 실행



결과확인하기

The screenshot shows the Spring Tool Suite 4 IDE with a Java file named `Main.java` open. The code is as follows:

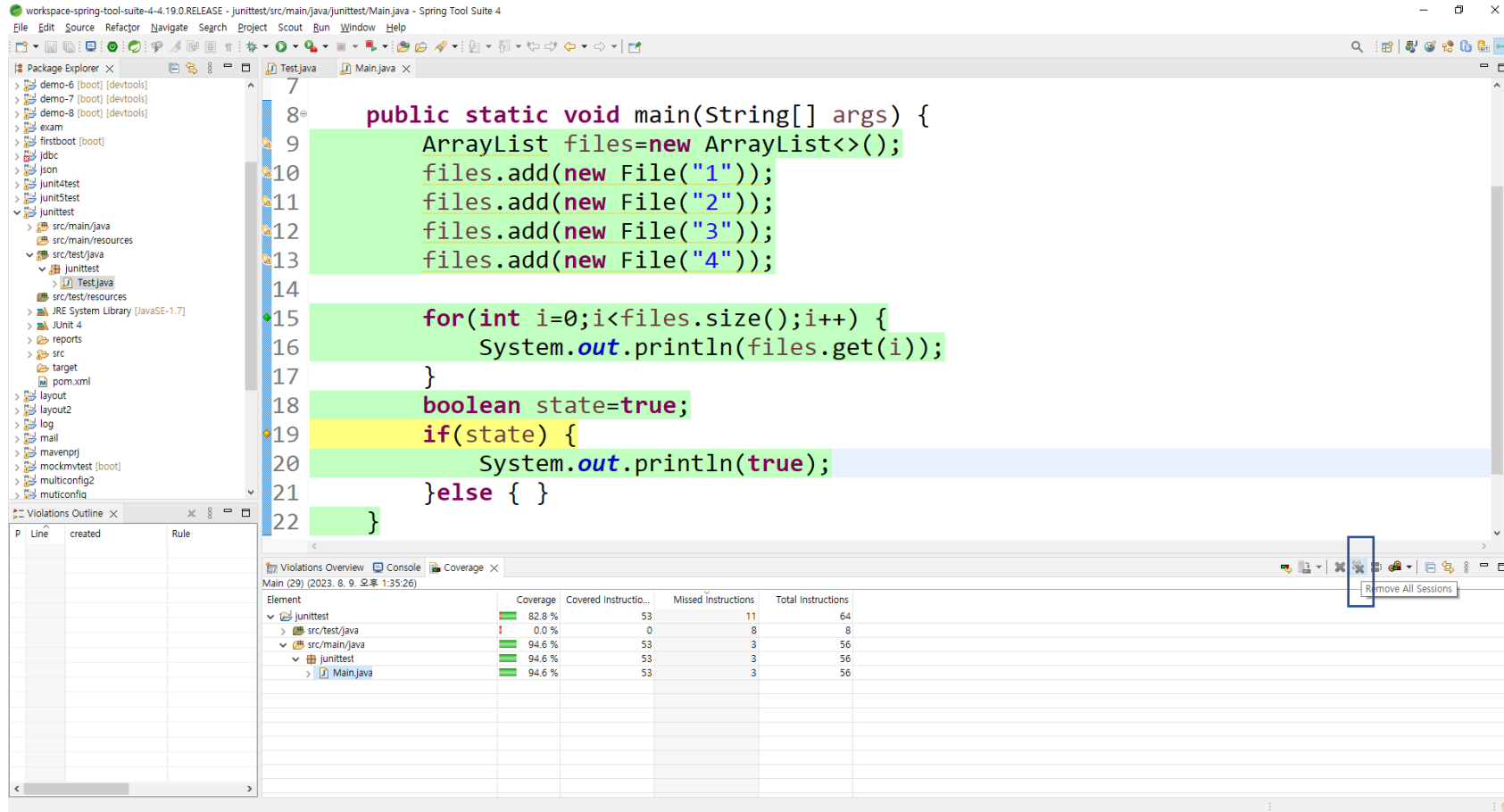
```
7
8 public static void main(String[] args) {
9     ArrayList files=new ArrayList<>();
10    files.add(new File("1"));
11    files.add(new File("2"));
12    files.add(new File("3"));
13    files.add(new File("4"));
14
15    for(int i=0;i<files.size();i++) {
16        System.out.println(files.get(i));
17    }
18    boolean state=true;
19    if(state) {
20        System.out.println(true);
21    }else {
22    }
```

Below the code editor, the **Violations Overview** tab is active, displaying a coverage table for the project.

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
JUnit4	82.8 %	53	11	64
src/test/java	0.0 %	0	8	8
src/main/java	94.6 %	53	3	56
JUnit5	94.6 %	53	3	56
Main.java	94.6 %	53	3	56

The status bar at the bottom indicates the file is **Writable**, **Smart Insert** is active, and the time is **19 : 20 : 415**.

Code coverage 해제하기



The screenshot displays the Spring Tool Suite 4 IDE with a Java project. The main editor shows the `Main.java` file, which contains a `main` method. The code is highlighted in green, indicating 100% coverage. The bottom panel shows a table with coverage data for various elements.

```
7  
8 public static void main(String[] args) {  
9     ArrayList files=new ArrayList<>();  
10    files.add(new File("1"));  
11    files.add(new File("2"));  
12    files.add(new File("3"));  
13    files.add(new File("4"));  
14  
15    for(int i=0;i<files.size();i++) {  
16        System.out.println(files.get(i));  
17    }  
18    boolean state=true;  
19    if(state) {  
20        System.out.println(true);  
21    }else { }  
22 }
```

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
JUnit 4	82.8 %	53	11	64
src/test/java	0.0 %	0	8	8
src/main/java	94.6 %	53	3	56
JUnit 4	94.6 %	53	3	56
Main.java	94.6 %	53	3	56

참고

- pmd 툴킷
- <https://lky1.tistory.com/53>
- spotbugs 설치방법
- <https://lts0606.tistory.com/447>
- <https://chanztudio.tistory.com/33>
- 보안도구
- <https://velog.io/@been/IT%EA%B8%B0%EC%82%AC%EA%B0%9C%EB%B0%9C%EC%9E%90%EA%B0%80-%EA%B6%8C%EC%9E%A5%ED%95%98%EB%8A%94-Java-%EC%BD%94%EB%93%9C-%ED%92%88%EC%A7%88-%EB%8F%84%EA%B5%AC>
- 스프링프레임워크 정부추천도구
- <https://www.egovframe.go.kr/wiki/doku.php?id=egovframework:dev4:findsecuritybugs>
- 젠킨스
- <https://onethejay.tistory.com/147>
- 도커에서 설정하고 도커를 실행할 때 콘솔로 처리해야 웹으로 동작
- 도커를 실행하고 비밀번호입력
- C:\Users\woogi>docker run -d -p 8081:8080 --name jenkins -u root jenkins/jenkins
- C:\Users\woogi>docker logs jenkins

PATH에 대한 이론

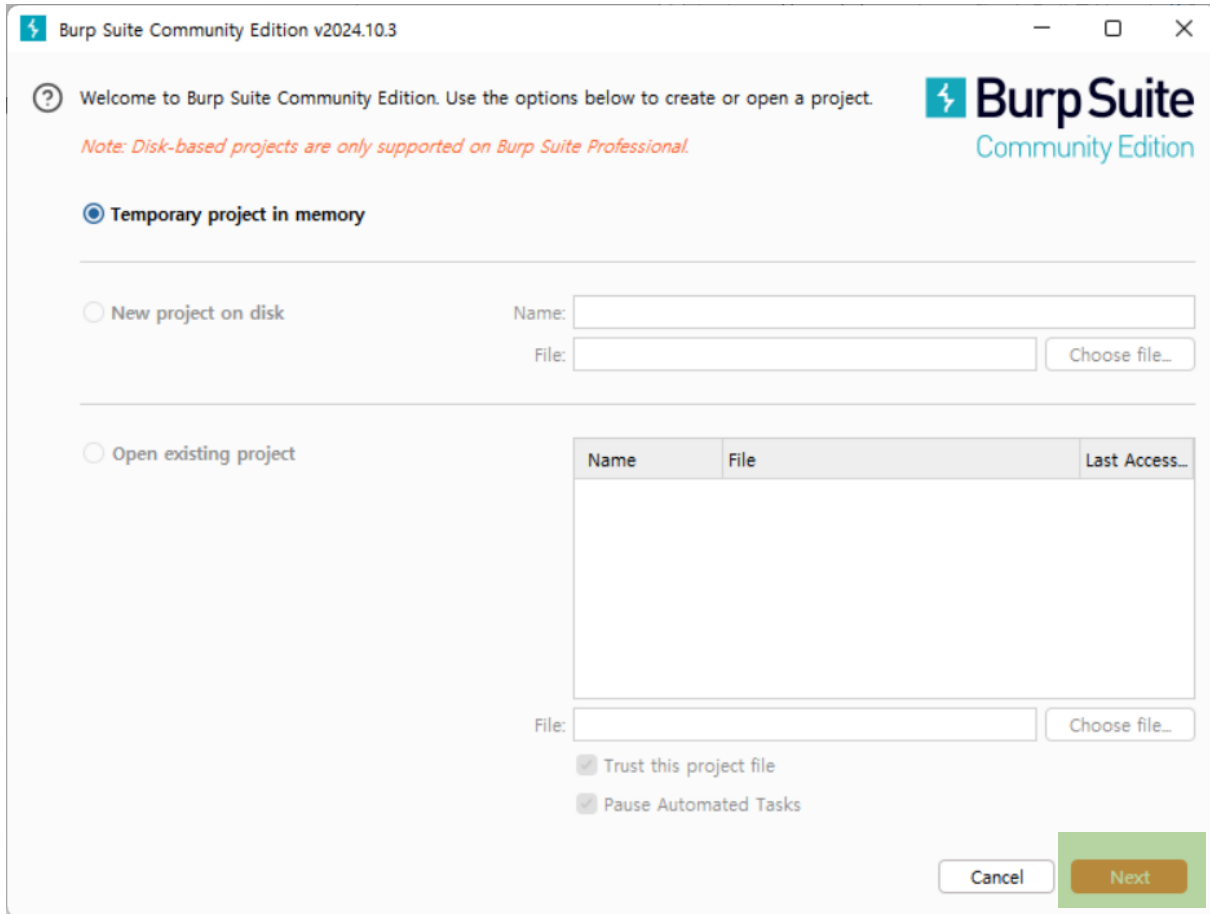
- pwd
 - /home/user
 - /home/user/a.sh 파일이 존재한다고 가정할 경우
 - a.sh : 상대경로
 - /home/user/a.sh : 절대 경로
 - ../ ..은 상위 디렉토리
 - ./ .은 현재 디렉토리
-
- CRLF : 줄바꿈코드
 - CR (Carriage Return): $\backslash r$ (ASCII 13)
 - LF (Line Feed): $\backslash n$ (ASCII 10)
 - CRLF: $\backslash r\backslash n$ → Windows 시스템에서 주로 줄 바꿈으로 사용됨
-
- CSRF(Cross-Site Request Forgery) : 페이지 인증관련 코드

Burp suite 설치

Burp Suite설치

[Burp Suite - Application Security Testing Software - PortSwigger](https://portswigger.net/burp)

<https://portswigger.net/burp>



Burp Suite Community Edition v2024.10.3

Welcome to Burp Suite Community Edition. Use the options below to create or open a project.

Note: Disk-based projects are only supported on Burp Suite Professional.

☒ Temporary project in memory

☐ New project on disk

Name:

File:

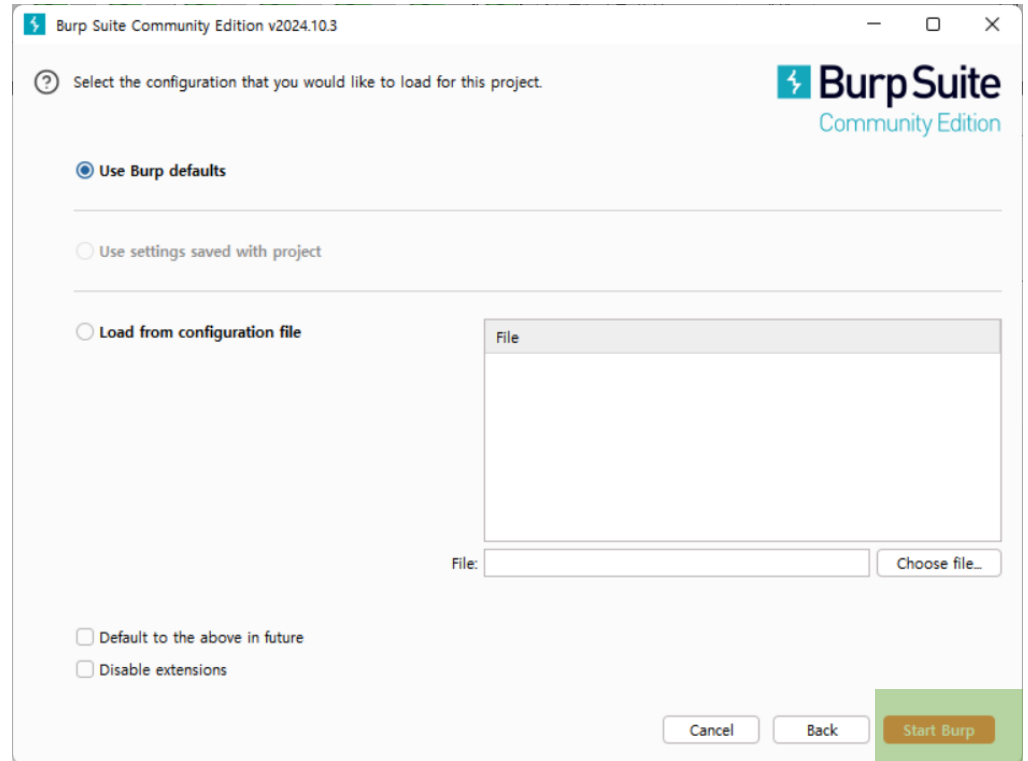
☐ Open existing project

Name	File	Last Access...
------	------	----------------

File:

☒ Trust this project file

☒ Pause Automated Tasks



Burp Suite Community Edition v2024.10.3

Select the configuration that you would like to load for this project.

☒ Use Burp defaults

☐ Use settings saved with project

☐ Load from configuration file

File

File:

☐ Default to the above in future

☐ Disable extensions

프록시 설정(사용하지 않는 포트 설정)

웹서버 주소가 아니며 별도로 연결할 주소이므로 반드시 사용하지 않는 주소를 선택할 것

The screenshot displays the Burp Suite application interface. The top menu bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'View', and 'Help'. Below it, a toolbar contains tabs for 'Dashboard', 'Target', 'Proxy', 'Intruder', 'Repeater', 'Collaborator', 'Sequencer', and 'Decompile'. The 'Proxy' tab is selected, and the 'Proxy settings' button is highlighted. The left sidebar shows the 'Settings' section with a search bar and tabs for 'All', 'User', and 'Project'. Under 'Tools', 'Proxy' is selected. The main panel shows 'Tools > Proxy' with a section for 'Proxy listeners'. A table lists the current listener: 'Running', 'Interface', 'Invisible', and '127.0.0.1:8082'. Below the table, a note states: 'Each installation of Burp generates its own CA certificate that Proxy listens to'. A dialog box titled 'Add a new proxy listener' is open, showing the 'Binding' tab. It contains the text: 'These settings control how Burp binds the proxy listener.' The 'Bind to port' field is set to '8082'. The 'Bind to address' section has 'Loopback only' selected, with 'All interfaces' and 'Specific address: 127.0.0.1' as options. The 'OK' button is highlighted.

Tools > Proxy

Proxy listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser.

Running	Interface	Invisible
<input checked="" type="checkbox"/>	127.0.0.1:8082	

Each installation of Burp generates its own CA certificate that Proxy listens to.

Add a new proxy listener

Binding Request handling Certificate TLS Protocols HTTP

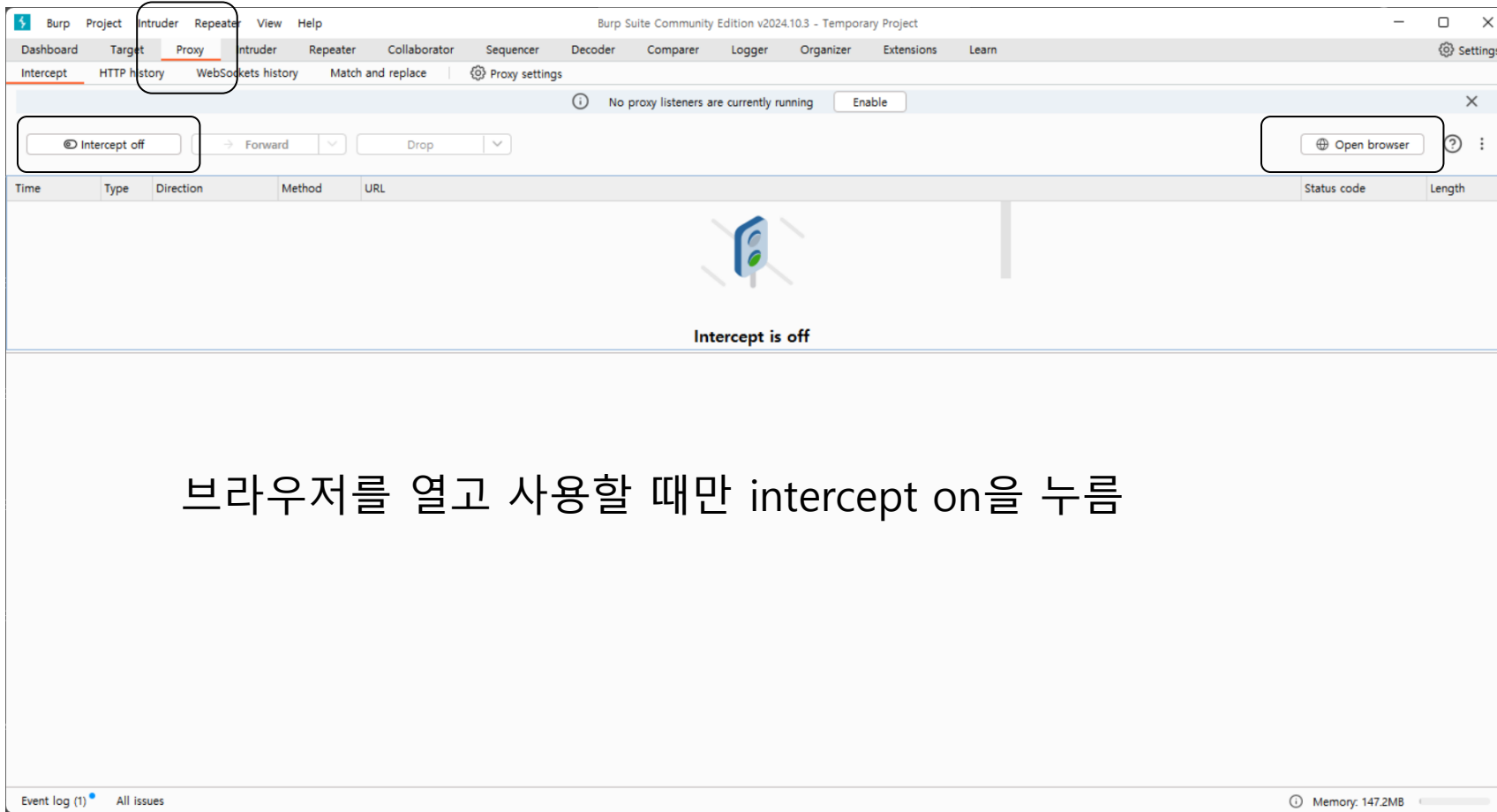
These settings control how Burp binds the proxy listener.

Bind to port: 8082

Bind to address: ☒ Loopback only
☐ All interfaces
☐ Specific address: 127.0.0.1

OK Cancel

Proxy켜기



브라우저를 열고 사용할 때만 intercept on을 누름

Proxy 탭 선택

Open browser 클릭

Proxy를 사용할 때만
Intercept off를 클릭
해당 버튼을 클릭하면
Packcet Capture상태가 됨

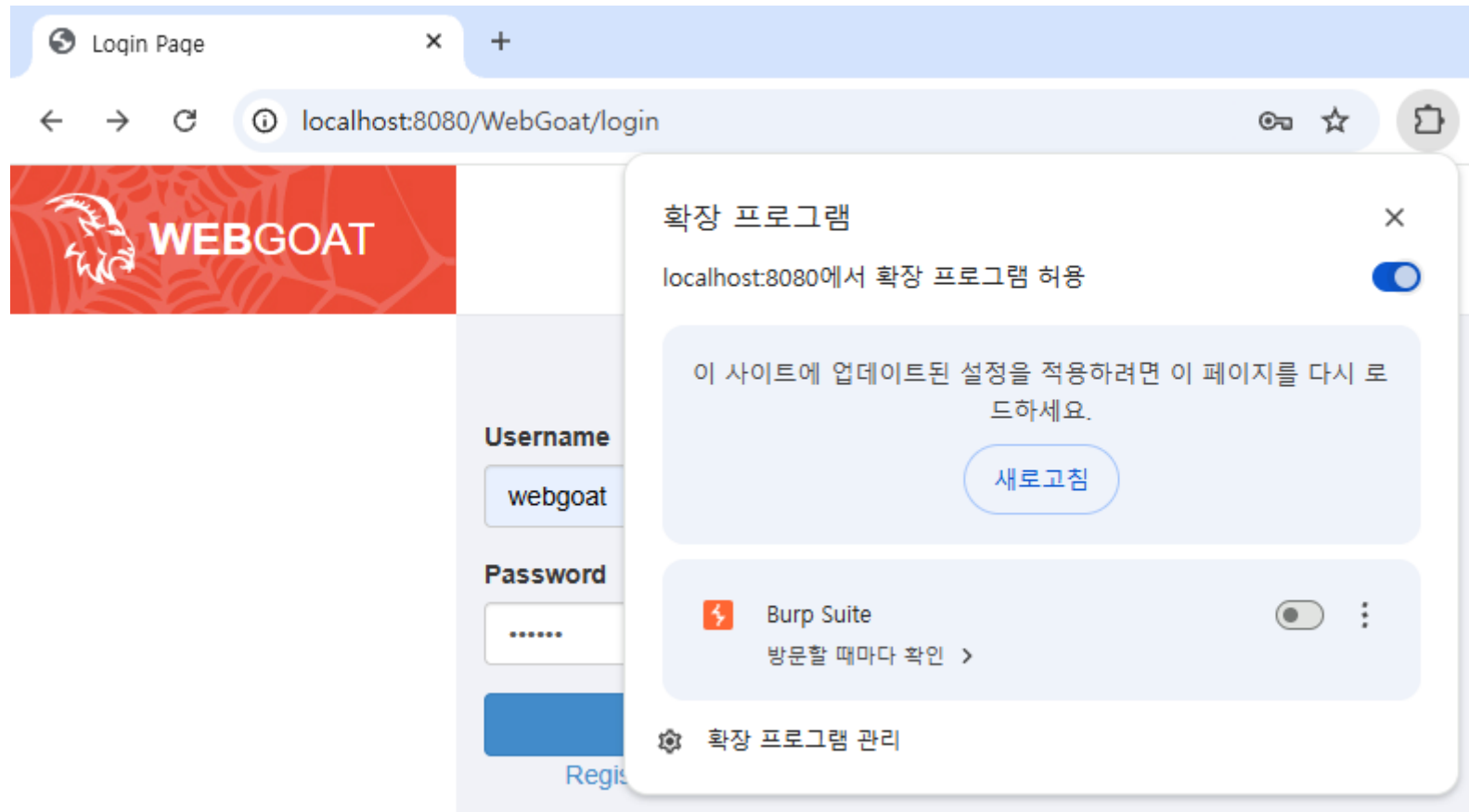
주소입력 후 forward를 눌러야 페이지이동(여러 번)

The image shows a screenshot of a web browser window displaying the WebGoat login page. The browser's address bar shows the URL `localhost:8888/WebGoat/login`. The login page has a red header with the WebGoat logo and the text "WEBGOAT". Below the header, there are two input fields: "Username" with the value "testuser" and "Password" with masked characters ".....". There is a blue "Sign in" button and a link "Register new user" below it.

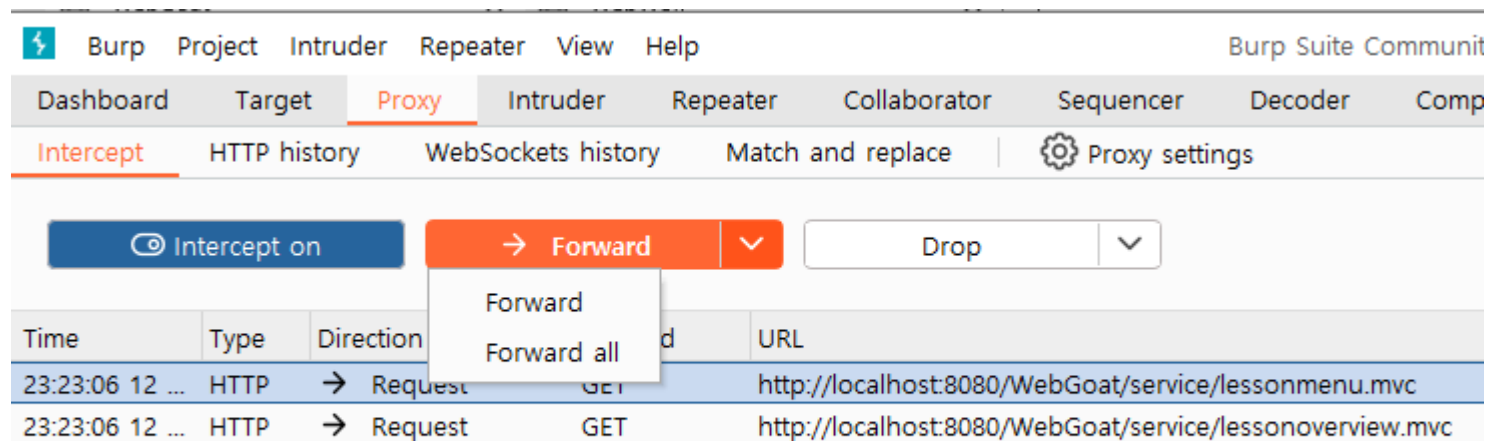
In the background, the Burp Suite interface is visible. The "Proxy" tab is selected, and the "Intercept" sub-tab is active. The "Request to http://localhost:8888 [127.0.0.1]" is shown. The "Forward" button is highlighted. The "Raw" view of the request is displayed, showing the following details:

```
1 POST /WebGoat/login HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 33
4 Cache-Control: max-age=0
5 sec-ch-ua: "Not:A-Brand";v="99", "Chromium";v="112"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Windows"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://localhost:8888
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost:8888/WebGoat/login
18 Accept-Encoding: gzip, deflate
19 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
20 Cookie: JSESSIONID=niolomdn7gSDqAFbx-wWC_3xEzuhl2UmRD:
21 Connection: close
22
23 username=testuser&password=123456
```

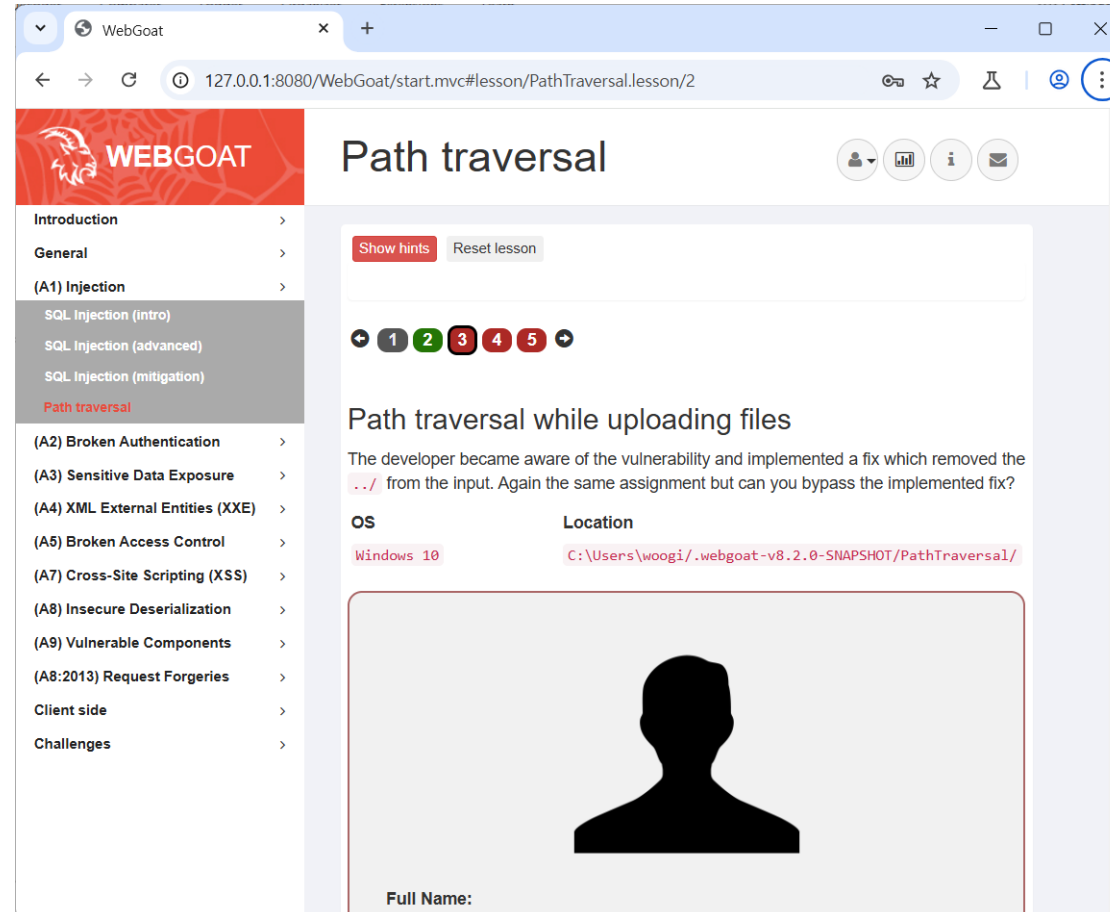
접속이 안될 경우 확장프로그램 허용중지



최신 프로그램으로 변경 forward all가능



Path traversal



The screenshot shows the WebGoat web application interface. The browser address bar displays the URL `127.0.0.1:8080/WebGoat/start.mvc#lesson/PathTraversal.lesson/2`. The page title is "Path traversal". On the left, a sidebar menu lists various security topics, with "Path traversal" highlighted in red. The main content area features a "Show hints" button and a "Reset lesson" button. Below these, a progress indicator shows five steps, with the third step (3) highlighted in red. The lesson title is "Path traversal while uploading files". The text explains that a developer removed the `../` from the input to fix a vulnerability, but asks if the user can bypass the fix. Below the text, there are two input fields: "OS" with the value "Windows 10" and "Location" with the value `C:\Users\woogi\.webgoat-v8.2.0-SNAPSHOT/PathTraversal/`. At the bottom, there is a silhouette of a person and a label "Full Name:".

WebGoat

Path traversal

Introduction >

General >

(A1) Injection >

SQL Injection (intro)

SQL Injection (advanced)

SQL Injection (mitigation)

Path traversal

(A2) Broken Authentication >

(A3) Sensitive Data Exposure >

(A4) XML External Entities (XXE) >

(A5) Broken Access Control >

(A7) Cross-Site Scripting (XSS) >

(A8) Insecure Deserialization >

(A9) Vulnerable Components >

(A8:2013) Request Forgeries >

Client side >

Challenges >

Show hints Reset lesson

1 2 3 4 5

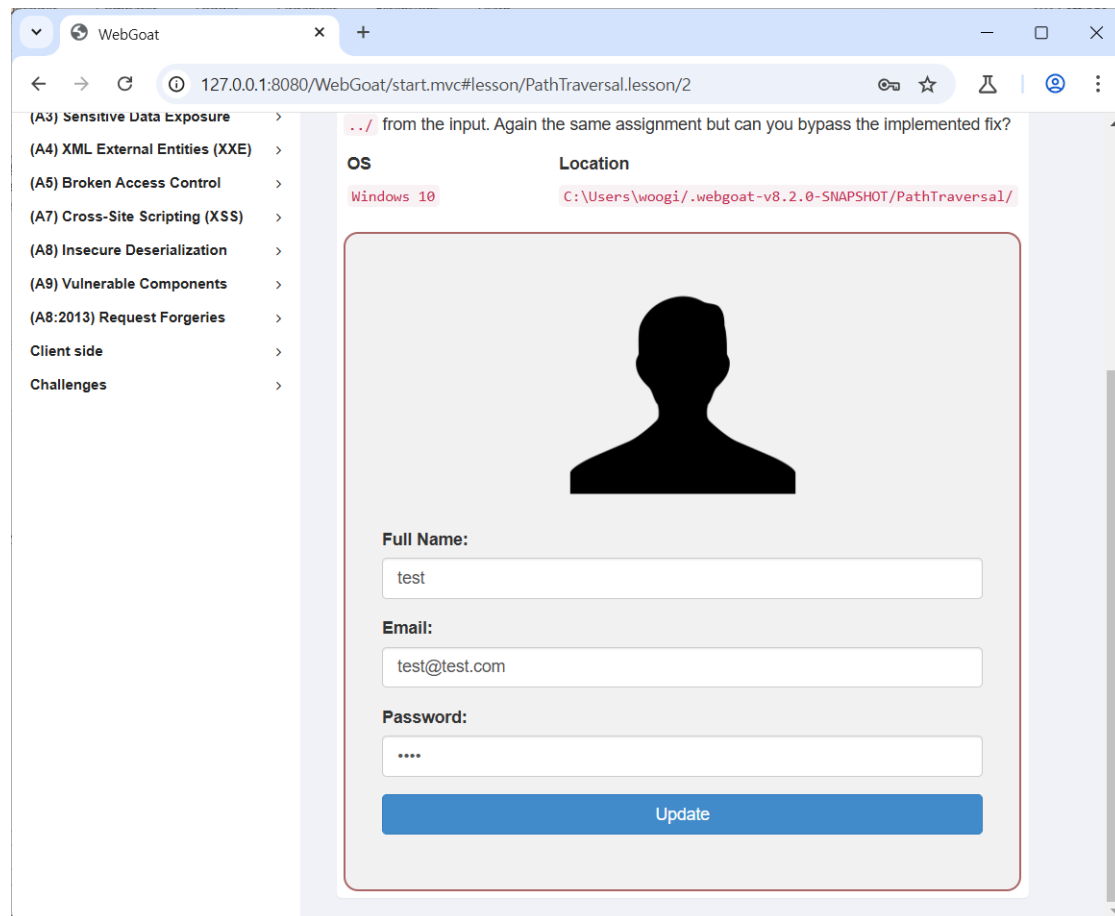
Path traversal while uploading files

The developer became aware of the vulnerability and implemented a fix which removed the `../` from the input. Again the same assignment but can you bypass the implemented fix?

OS Location

Windows 10 `C:\Users\woogi\.webgoat-v8.2.0-SNAPSHOT/PathTraversal/`

Full Name:



⚡ Burp Project Intruder Repeater View Help

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

Intercept HTTP history WebSockets history Match and replace Proxy settings

Intercept on

Forward

Drop

Request to http://127.0.0.1:8080 [Open browser](#)

Time	Type	Direction	Method	URL	Status code	Length
12:44:41	18 De...	HTTP	→ Request	GET http://127.0.0.1:8080/WebGoat/service/lessonmenu.mvc		
12:44:41	18 De...	HTTP	→ Request	GET http://127.0.0.1:8080/WebGoat/service/lessonoverview.mvc		
12:44:52	18 De...	HTTP	→ Request	POST http://127.0.0.1:8080/WebGoat/PathTraversal/profile-upload-fix		
12:45:06	18 De...	HTTP	→ Request	GET http://127.0.0.1:8080/WebGoat/service/lessonoverview.mvc		
12:45:06	18 De...	HTTP	→ Request	GET http://127.0.0.1:8080/WebGoat/service/lessonmenu.mvc		
12:45:11	18 De...	HTTP	→ Request	GET http://127.0.0.1:8080/WebGoat/service/lessonmenu.mvc		

Request

Pretty Raw Hex

1 POST /WebGoat/PathTraversal/profile-upload-fix HTTP/1.1

2 Host: 127.0.0.1:8080

3 Content-Length: 138688

4 sec-ch-ua-platform: "Windows"

5 Accept-Language: ko-KR,ko;q=0.9

6 sec-ch-ua: "Chromium";v="131", "Not_A Brand";v="24"

7 sec-ch-ua-mobile: ?0

8 X-Requested-With: XMLHttpRequest

9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36

10 Accept: */*

11 Content-Type: multipart/form-data; boundary=----WebKitFormBoundarygRboo2FAFZevKC6I

12 Origin: http://127.0.0.1:8080

13 Sec-Fetch-Site: same-origin

14 Sec-Fetch-Mode: cors

15 Sec-Fetch-Dest: empty

16 Referer: http://127.0.0.1:8080/WebGoat/start.mvc

17 Accept-Encoding: gzip, deflate, br

18 Cookie: JSESSIONID=Wyzd6ej3JrLeoB59cjJsEB5HPHVrxSpA_P3uw20

19 Connection: keep-alive

20

21 -----WebKitFormBoundarygRboo2FAFZevKC6I

22 Content-Disposition: form-data; name="uploadedFileFix"; filename="iDm1D-@:iD 2024-09-20 111637.png"

23 Content-Type: image/png

24

25 PNG

Inspector

Request attributes2

Request query parameters0

Request body parameters4

Request cookies1

Request headers18

Notes

0 highlights

Event log (1)

All issues

Memory: 135.6MB

아래쪽으로 이동
변경할 사항을 클릭하면
우측에 변경사항 변경
Apply후 forward

Intercept on Forward Drop Request to http://127.0.0.1:8080 Open browser ?

Time	Type	Direction	Method	URL	Status code	Length
12:44:41.18	De	Request	GET	http://127.0.0.1:8080/WebGoat/service/lessonmenu.mvc		
12:44:41.18	De	Request	GET	http://127.0.0.1:8080/WebGoat/service/lessonoverview.mvc		
12:44:52.18	De	Request	POST	http://127.0.0.1:8080/WebGoat/PathTraversal/profile-upload-fix		
12:45:06.18	De	Request	GET	http://127.0.0.1:8080/WebGoat/service/lessonoverview.mvc		
12:45:06.18	De	Request	GET	http://127.0.0.1:8080/WebGoat/service/lessonmenu.mvc		
12:45:11.18	De	Request	GET	http://127.0.0.1:8080/WebGoat/service/lessonmenu.mvc		

Request

Pretty Raw Hex

```
482 as=VamS*ny:U.<al9A;aaaB_Z+eTuer0*5>P8ou|aauxK|UJ,>5% e"s|»ipUUUQUY.AK_u2UUBU#oyeLj
483 Di~A*DuIDSSDDè,iKAS,'Drô,Râ_YT|)wôaAd>DuB
ôÂoA000bdx9*+de-up+QVUWU4D1eSH0)e,0+|SDAmçimEz,7'è+II@ (GIU?SôYU'D8U'~x007D*ôô'ui*mp~*E#SôT?S*~çD+IÂÏ000H80âôêS~PVe2C±8~M«Tô*y00 (:0?0
J0Tr (<0Y<ôqâ,3;Â|è2[ç~*u1i10U*~ôY1l1b0DE6*00 0~JNDVizM0fôqxÂ-ZByuôIw I 8D0I'my4I*00è
yD*ouDU;0% (#00èH0 (x'=SDHD10a1P*0Y*H0$*Du-Y'SDx2eID0e000)00t>vD~I10*Â~0~ôLaxh8000006EA800 çBô$EÜ00H0âgwi fô
e>MYÜD eo/7~bP% u~*0RD0$K000uy0' DL~*9i|PVGs*0)x^ c$5*0aDuEH0Y;P8|AAU'!)çÂnu80±+y;>xD @ui,EE800 (ô808fuu00? ly?
484 %; 808xi'Â~AD*qi I' ZYÂi+<c0>V;ÂB00$%xB5I (:K0'ôPO<X' Z~u0U (x~0000nV10'Tw*giô;g0M00éV~DX*o'0~AA$90800xAD;0io>ô%t0c'x0000EE0,,fôg0U,S»çH0c.âmAÂ?1%Â',S±
ô'00MVOH40e4YJ0;,'xWh0_IDAD0 R|0 t1eeiuwçkxnp;AÂKD00$;8auI JÂ«(08Q[WAD;xt000*0yu+00V~Xa|0z/|)0% ("0%80<--m'ü800;Â'3;âé'â
485 )"Si1eDa0èetceñ4u1ÄV,7E0è;ñeuhz(-1APO ujuÂTAU~--0'0+71B+ç0m0YçID«<2M0-,5y'OH0fDu010'çD"D080#|0;0u~rè=,(0 AT;1EiW1NDW; (Iç0%*èRu*âôjM0C'--#9%100P0D%0s)»
486 0'FV«~IV~i40BD?Â+ç40D_>*AD0D1a)b (Du000Dx0'0DÂ0'Da0I$kyD00RÂ.ÜNqx~*qè*00rZb*0ySD0u8DÂ*K000y0D~*Vu000ç0Uia:èZ~E407E84è.0DueeipèçôânuéZQ0,400rç*0qeyu0P) iÂç000
487 t~*tiè0iWD0e49 N008ilyfè|008I#8SY0u00lyç:4'0'âDés00+ç0I<ç|0=
488 :D0P))J8Â0(ôP)s0)edèÂiD00x>0>Du8âA00?xi11ED>|es00t>|e8ââ>Du8âA00?py0|b0'I80I8H08 0
489 -----WebKitFormBoundarygRboozFAFZevKC6I
490 Content-Disposition: form-data; name="fullNameFix"
491
492 test
493 -----WebKitFormBoundarygRboozFAFZevKC6I
494 Content-Disposition: form-data; name="emailFix"
495
496 test@test.com
497 -----WebKitFormBoundarygRboozFAFZevKC6I
498 Content-Disposition: form-data; name="passwordFix"
499
500 test
501 -----WebKitFormBoundarygRboozFAFZevKC6I--
```

Inspector

Selection 4 (0x4)

Selected text

.../test

Decoded from: Select

Cancel Apply changes

Request attributes 2

Request query parameters 0

Request body parameters 4

Request cookies 1

Request headers 18

Event log (1) All issues

Memory: 135.6MB