

# FOSSLight 실무에서 활용하는 법



LG전자 Open Source TP 김소임



LG Open Source

# CONTENTS

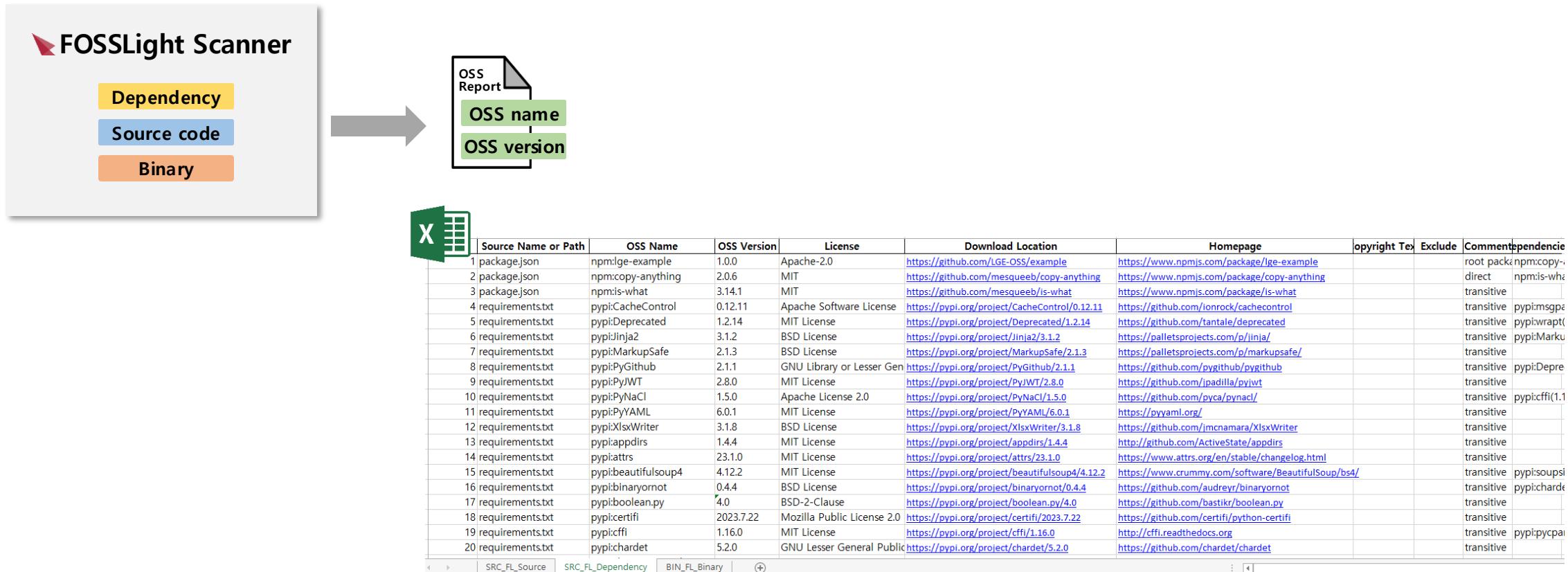
---

- FOSSLight Hub와 Scanner 연동
- FOSSLight Prechecker with Jenkins
- FOSSLight Scanner with Jenkins
- FOSSLight MCP 서버

# FOSSLight Hub와 Scanner 연동

# FOSSLight Hub + FOSSLight Scanner

- FOSSLight Scanner 실행하여 오픈소스 분석 보고서 생성



# FOSSLight Hub + FOSSLight Scanner

- 생성한 Report을 FOSSLight Hub에 업로드하여 OSC Process를 진행

The diagram illustrates the integration between the FOSSLight Hub and FOSSLight Scanner. On the right, the FOSSLight Hub interface shows a list of projects with their details and current status (e.g., Progress, Request). On the left, the FOSSLight Scanner interface displays a detailed report of dependencies and their licenses. A large red arrow points from the FOSSLight Hub towards the FOSSLight Scanner, symbolizing the upload process.

**FOSSLight Hub Project List:**

ID	Project Name	Status	OSC Process	Download	Security	Division	Creator
881	Review Report Demo (1.0.1)	Progress	Identification > Packaging		Need to resolve(10.0)	LG Electro...	foss.lge
872	min_test	Progress	Identification > Packaging	X	Need to resolve(10.0)	LG Electro...	minksk
871	chs_test (1.10)	Request	Identification > Packaging		Discovered(N/A)	LG Electro...	balisa_fist
860	lge-osc-example	Request	Identification > Packaging		Need to resolve(7.1)	LG Electro...	foss.lge
		Progress	Identification > Packaging		Need to resolve(7.5)	LG Electro...	foss.lge
		Progress	Identification > Packaging		Need to resolve(9.8)	LG Electro...	foss.lge
		Progress	Identification > Packaging		Need to resolve(9.8)	LG Electro...	foss.lge

**FOSSLight Scanner Dependency Report:**

Source Name or Path	OSS Name	OSS Version	License	Download Location	Homepage	Copyright Text	Exclude	Comments	Dependencies
1 package.json	npmLGE-example	1.0.0	Apache-2.0	<a href="https://github.com/LGE-OSS/example">https://github.com/LGE-OSS/example</a>	<a href="https://www.npmjs.com/package/lge-example">https://www.npmjs.com/package/lge-example</a>	root package npm copy-anything			
2 package.json	npmscopy-anything	2.0.6	MIT	<a href="https://github.com/mesqueeb/copy-anything">https://github.com/mesqueeb/copy-anything</a>	<a href="https://www.npmjs.com/package/copy-anything">https://www.npmjs.com/package/copy-anything</a>	direct npms-is-what			
3 package.json	npms-is-what	3.14.1	MIT	<a href="https://github.com/mesqueeb/is-what">https://github.com/mesqueeb/is-what</a>	<a href="https://www.npmjs.com/package/is-what">https://www.npmjs.com/package/is-what</a>	transitive			
4 requirements.txt	pypiCacheControl	0.12.11	Apache Software License	<a href="https://pypi.org/project/cachecontrol/0.12.11">https://pypi.org/project/cachecontrol/0.12.11</a>	<a href="https://github.com/lonnix/cachecontrol">https://github.com/lonnix/cachecontrol</a>	transitive pyplmspaceman			
5 requirements.txt	pypiDeprecated	1.2.14	MIT License	<a href="https://pypi.org/project/Deprecated/1.2.14">https://pypi.org/project/Deprecated/1.2.14</a>	<a href="https://github.com/tantale/deprecated">https://github.com/tantale/deprecated</a>	transitive pyplwrapt			
6 requirements.txt	pypiJinja2	3.1.2	BSD License	<a href="https://pypi.org/project/jinja2/3.1.2">https://pypi.org/project/jinja2/3.1.2</a>	<a href="https://palletsprojects.com/p/jinja/">https://palletsprojects.com/p/jinja/</a>	transitive pyplMarkupsafe			
7 requirements.txt	pypiMarkupSafe	2.1.3	BSD License	<a href="https://pypi.org/project/MarkupSafe/2.1.3">https://pypi.org/project/MarkupSafe/2.1.3</a>	<a href="https://palletsprojects.com/p/markupsafe/">https://palletsprojects.com/p/markupsafe/</a>	transitive			
8 requirements.txt	pypiPyGithub	2.1.1	GNU Library or Lesser General Public License	<a href="https://pypi.org/project/PyGithub/2.1.1">https://pypi.org/project/PyGithub/2.1.1</a>	<a href="https://github.com/PyGithub/PyGithub">https://github.com/PyGithub/PyGithub</a>	transitive pipDependents			
9 requirements.txt	pypiPyJWT	2.8.0	MIT License	<a href="https://pypi.org/project/PyJWT/2.8.0">https://pypi.org/project/PyJWT/2.8.0</a>	<a href="https://github.com/padilla/pyjwt">https://github.com/padilla/pyjwt</a>	transitive			
10 requirements.txt	pypiPyNaCl	1.5.0	Apache License 2.0	<a href="https://pypi.org/project/PyNaCl/1.5.0">https://pypi.org/project/PyNaCl/1.5.0</a>	<a href="https://github.com/PyNaCl/PyNaCl">https://github.com/PyNaCl/PyNaCl</a>	transitive pipicffi(1.1.0)			
11 requirements.txt	pypiPyYAML	6.0.1	MIT License	<a href="https://pypi.org/project/PyYAML/6.0.1">https://pypi.org/project/PyYAML/6.0.1</a>	<a href="https://pyyaml.org/">https://pyyaml.org/</a>	transitive			
12 requirements.txt	pypiXlsxWriter	3.1.8	BSD License	<a href="https://pypi.org/project/xlsxwriter/3.1.8">https://pypi.org/project/xlsxwriter/3.1.8</a>	<a href="https://github.com/mcamara/XlsxWriter">https://github.com/mcamara/XlsxWriter</a>	transitive			
13 requirements.txt	pypiAppdirs	1.4.4	MIT License	<a href="https://pypi.org/project/appdirs/1.4.4">https://pypi.org/project/appdirs/1.4.4</a>	<a href="https://github.com/ActiveState/appdirs">https://github.com/ActiveState/appdirs</a>	transitive			
14 requirements.txt	pypiAtrus	23.1.0	MIT License	<a href="https://pypi.org/project/atrus/23.1.0">https://pypi.org/project/atrus/23.1.0</a>	<a href="https://www.attrs.org/en/stable/changelog.html">https://www.attrs.org/en/stable/changelog.html</a>	transitive			
15 requirements.txt	pypibeautifulsoup4	4.12.2	MIT License	<a href="https://pypi.org/project/beautifulsoup4/4.12.2">https://pypi.org/project/beautifulsoup4/4.12.2</a>	<a href="https://www.crummy.com/software/BeautifulSoup/bs4/doc/">https://www.crummy.com/software/BeautifulSoup/bs4/doc/</a>	transitive pipSoups			
16 requirements.txt	pyplibinarynot	0.44	BSD License	<a href="https://pypi.org/project/binarynot/0.44">https://pypi.org/project/binarynot/0.44</a>	<a href="https://github.com/audreyr/binarynot">https://github.com/audreyr/binarynot</a>	transitive pipichard			
17 requirements.txt	pyplibboolean.py	4.0	BSD-2-Clause	<a href="https://pypi.org/project/boolean.py/4.0">https://pypi.org/project/boolean.py/4.0</a>	<a href="https://github.com/bastirk/boolean.py">https://github.com/bastirk/boolean.py</a>	transitive			
18 requirements.txt	pyplibcertifi	2023.7.22	Mozilla Public License 2.0	<a href="https://pypi.org/project/certifi/2023.7.22">https://pypi.org/project/certifi/2023.7.22</a>	<a href="https://github.com/certifi/python-certifi">https://github.com/certifi/python-certifi</a>	transitive			
19 requirements.txt	pyplibcffi	1.16.0	MIT License	<a href="https://pypi.org/project/cffi/1.16.0">https://pypi.org/project/cffi/1.16.0</a>	<a href="https:// ffi.readthedocs.org">https:// ffi.readthedocs.org</a>	transitive pipCpca			
20 requirements.txt	pyplibchardet	5.2.0	GNU Lesser General Public License	<a href="https://pypi.org/project/chardet/5.2.0">https://pypi.org/project/chardet/5.2.0</a>	<a href="https://github.com/charDET/chardet">https://github.com/charDET/chardet</a>	transitive			

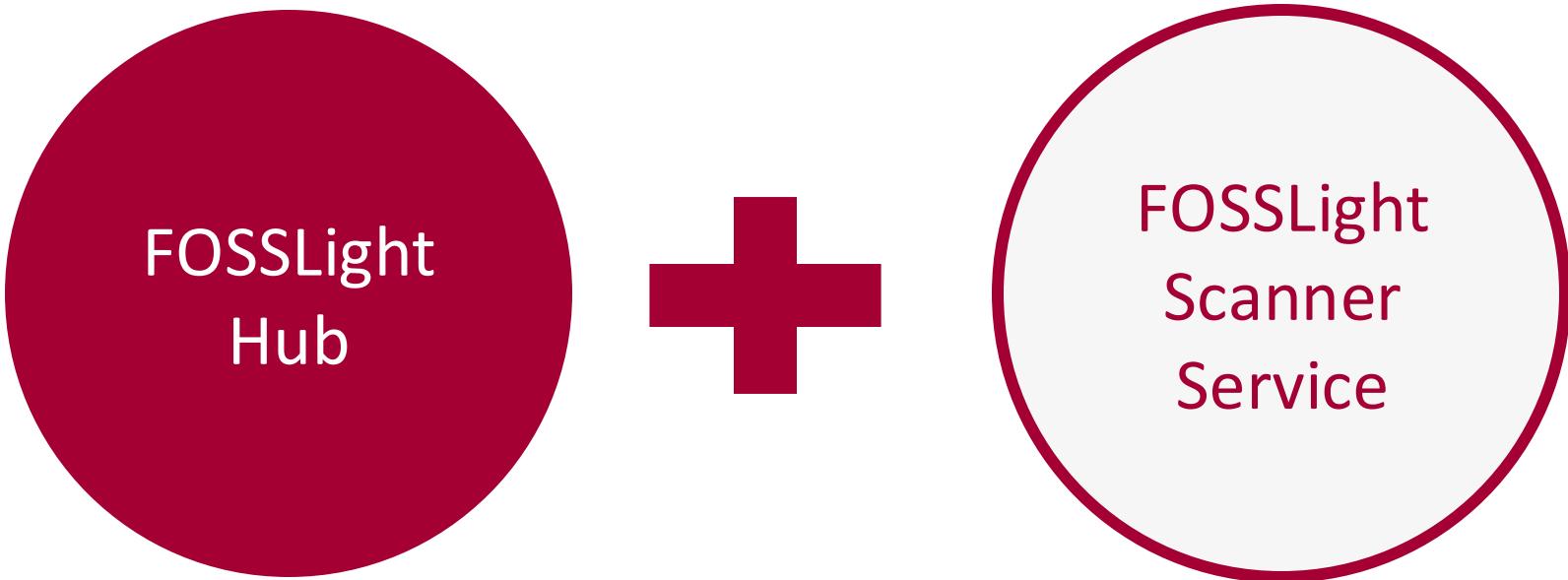
# FOSSLight Hub + FOSSLight Scanner

- FOSSLight Hub에서 한번에 처리할 수 없을까?



# FOSSLight Hub + FOSSLight Scanner

- FOSSLight Hub에서 한번에 처리할 수 없을까?



# FOSSLight Hub에서 한번에 호출

- Self-Check > URL 분석

The screenshot shows the FOSSLight Hub web application. On the left is a dark sidebar with navigation links: Dashboard, License, Open Source, Project, 3rd Party, BAT, Binary DB, Vulnerability, Self-Check (which is selected), Compliance Status, External Link, User Settings, and System. The main content area has a title bar with tabs: Close, Dashboard, License, Open Source, Project, 3rd Party, Self-Check, and 1576\_selfCheck. The 1576\_selfCheck tab is active. Below the title bar, there's a form with fields for Project Name (testtttt), Comment (https://github.com/LGE-OSS/example), and Created (김소임/책임연구원/SW공학(연)Open Source TP CTO SW센터 (2025-02-13)). An 'Edit' button is in the top right of this section. Below this is an 'Edit Permission' section with a '+' button. The main workspace is titled 'Self-Check' and contains a sub-tab 'Notice'. It features three radio buttons: 'Upload Analysis Result' (unchecked), 'URL' (selected, indicated by a checked checkbox), and 'Project Search (fs.lge.com)' (unchecked). Below these is a large input field labeled 'Enter the link of the source to be analyzed' with a dropdown menu set to 'URL'. To the right of the input field is a blue 'send' button. A large red arrow points from the text 'A red arrow points to the URL input field where a GitHub link is being entered.' to the 'URL' input field. At the bottom of the workspace is a 'Pre-Review' section with a table showing rows of data. The first row has columns: ID, Binary Name or Source Pa, OSS Name, OSS Version, License, Download Loc, Homepage, Copyright Text, OSS De, License, Vulner, Notic, Sourc, Restrict, and Ex. The second row shows ID 1, OSS Name pypi:cyclonedx-py..., OSS Version 8.5.1, License Apache-2.0, Download Loc https://pypi.org https://github., Homepage, Copyright Text, OSS De, License, Vulner, Notic, Sourc, Restrict, and Ex. The third row shows ID 4, OSS Name Required, OSS Version Required, License, Download Loc, Homepage, Copyright Text, OSS De, License, Vulner, Notic, Sourc, Restrict, and Ex. The fourth row shows ID 25, OSS Name Required, OSS Version Required, License Apache-2.0, Download Loc, Homepage Copyright (c) 2021 ..., OSS De, License, Vulner, Notic, Sourc, Restrict, and Ex.

# FOSSLight Hub에서 한번에 호출

- **Self-Check > URL 분석**

1. FOSSLight Scanner Service에 project 생성 & 분석
2. 분석 결과를 호출한 FOSSLight Hub 의 self-check project로 업로드

The screenshot shows the FOSSLight Hub web application. On the left is a dark sidebar with navigation links: Dashboard, License, Open Source, Project, 3rd Party, BAT, Binary DB, Vulnerability, Self-Check, Compliance Status, External Link, User Settings, and System. The main content area has a title bar with tabs: Close, Dashboard, License, Open Source, Project, 3rd Party, Self-Check, and 1576\_selfCheck. The 1576\_selfCheck tab is active. Below the title bar, there's a form for a project named 'testtttt' with a comment linking to a GitHub repository. A red arrow points to the bottom right corner of this form area. The next section is titled 'Self-Check' and contains a 'URL' input field with a dropdown menu set to 'URL'. A red arrow points to the bottom right corner of this input field. The final section is titled 'Pre-Review' and displays a table of analysis results. The table has columns: ID, Binary Name or Source Pa, OSS Name, OSS Version, License, Download Loc, Homepage, Copyright Text, OSS De, License, Vulner, Notic, Sourc, Restrict, and Ex. The first row shows 'pypi:cyclonedx-pyt...' with a status of 'New open source'. The second row shows 'src/fosslight\_util/reso...' with a status of 'Required'. The third row shows 'src/fosslight\_util/help...' with a status of 'Required'. A red box highlights the entire 'Pre-Review' table area.

# FOSSLight Hub에서 한번에 호출

- Self-Check > FOSSLight Scanner Service Project Search

The screenshot shows the FOSSLight Hub dashboard with a sidebar containing links like Dashboard, License, Open Source, Project, 3rd Party, BAT, Binary DB, Vulnerability, Self-Check, Compliance Status, External Link, User Settings, and System. The main area shows a project named 'testtttt' created by '김소임/책임연구원/SW공학(연)Open Source TP CTO SW센터 (2025-02-13)'. Below this, there's an 'Edit Permission' button and a 'Self-Check' tab which is currently selected. The 'Project Search (fs.lge.com)' tab is also visible. A red box highlights the 'Project Search (fs.lge.com)' tab, and a red arrow points from it to a callout bubble.

FOSSLight Scanner Service에  
별도로 생성해둔 Project가 있는  
경우 load

# FOSSLight Hub에서 한번에 호출

- 각각 서비스 구동 후 간단하게 연동!

The screenshot shows the FOSSLight Hub application interface. On the left is a dark sidebar with various navigation options. The main area displays a table with two rows of data. A red box highlights the second row, and a red arrow points to it from the bottom right.

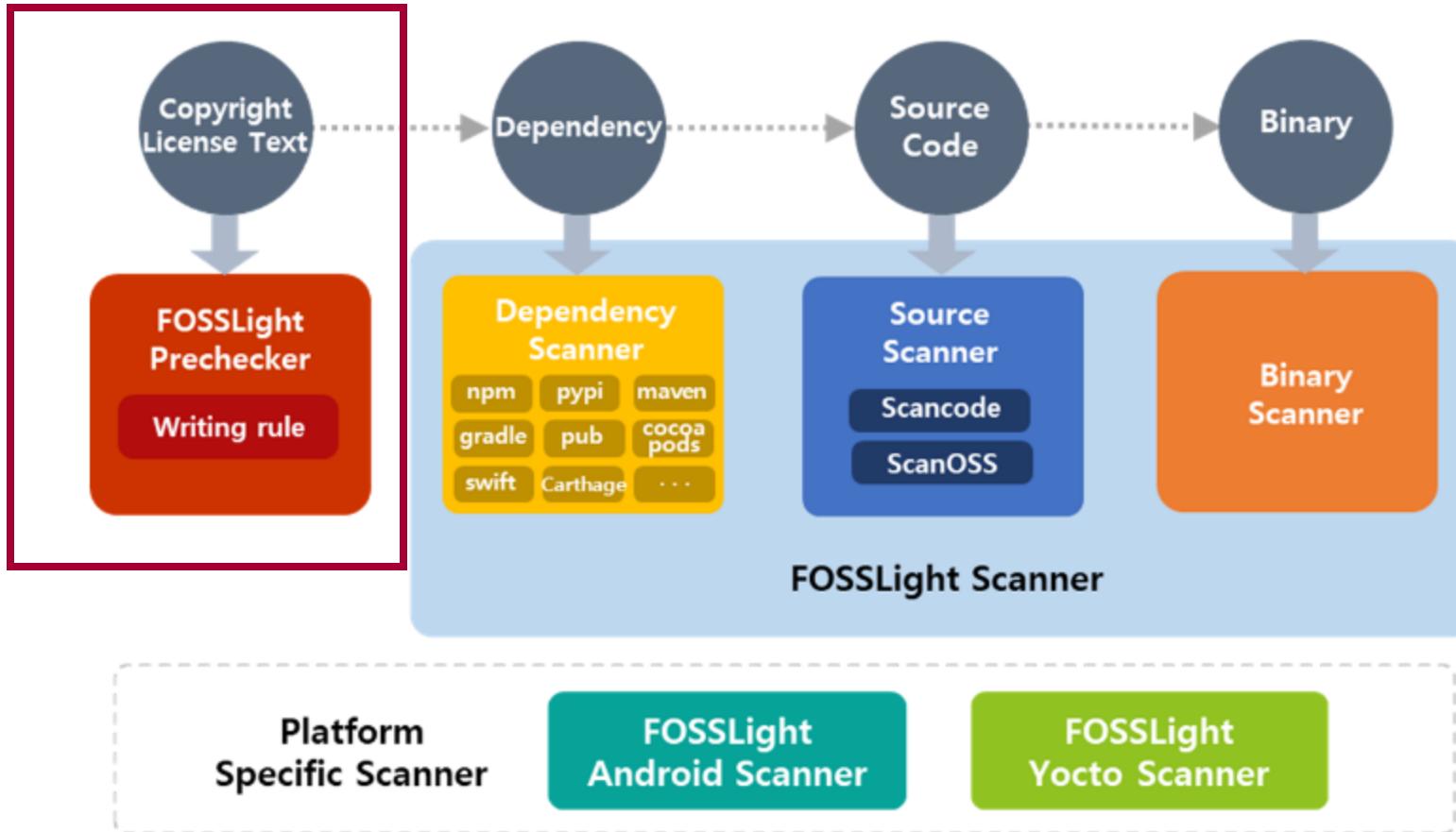
Detail No	Detail Name	Detail Description	Sub Code	Order	Use YN	Action
101	FL Scanner Url	https://fs.lge.com		1	Y	Delete
102	Admin Token	ac024cdef4391e5dceaa70a1a5a6f947e1a56496		2	Y	Delete

# **FOSSLight Prechecker with Jenkins**

---

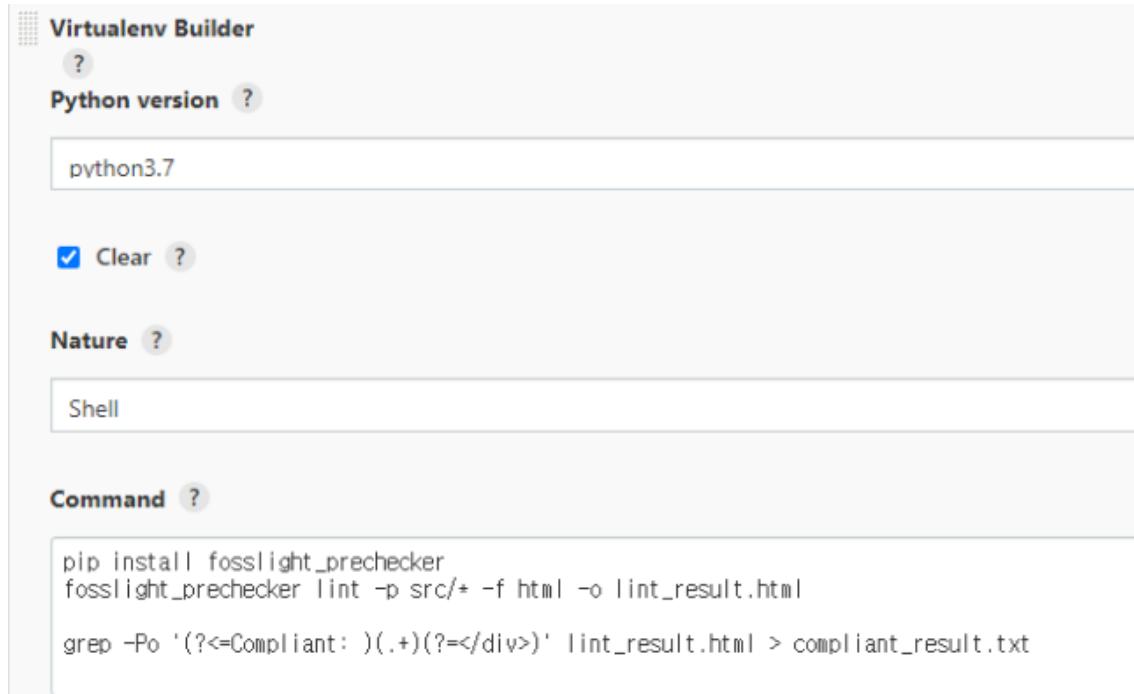
# FOSSLight Prechecker

- 소스 코드 내 저작권 및 라이선스 규칙 준수 여부 체크



# Jenkins 프로젝트에서 FOSSLight Prechecker 연동하기

1. Project 생성
2. 실행 주기 설정 -> 빌드 유발
3. FOSSLight Prechecker 실행 환경 설정 (lint 모드 실행)



4. FOSSLight Prechecker 결과 파일 메일링

# FOSSLight Prechecker lint 모드 결과 메일링 예시

File ( Total 1 )   lint\_result.html (2.9KB)

Internet Header ▾

## FOSSLight Reuse Lint

**Compliant: OK**

- Open Source Package file: N/A
- Detected licenses:
  - LicenseRef-LGE-Proprietary
- Files without copyright / total: 0 / 2
- Files without license / total: 0 / 2

### « Files without License or Copyright »

File	License	Copyright
------	---------	-----------

신규 수정 사항을 반영하면서  
저작권이나 License 표기가 없는  
파일 알림하여 보완 요청

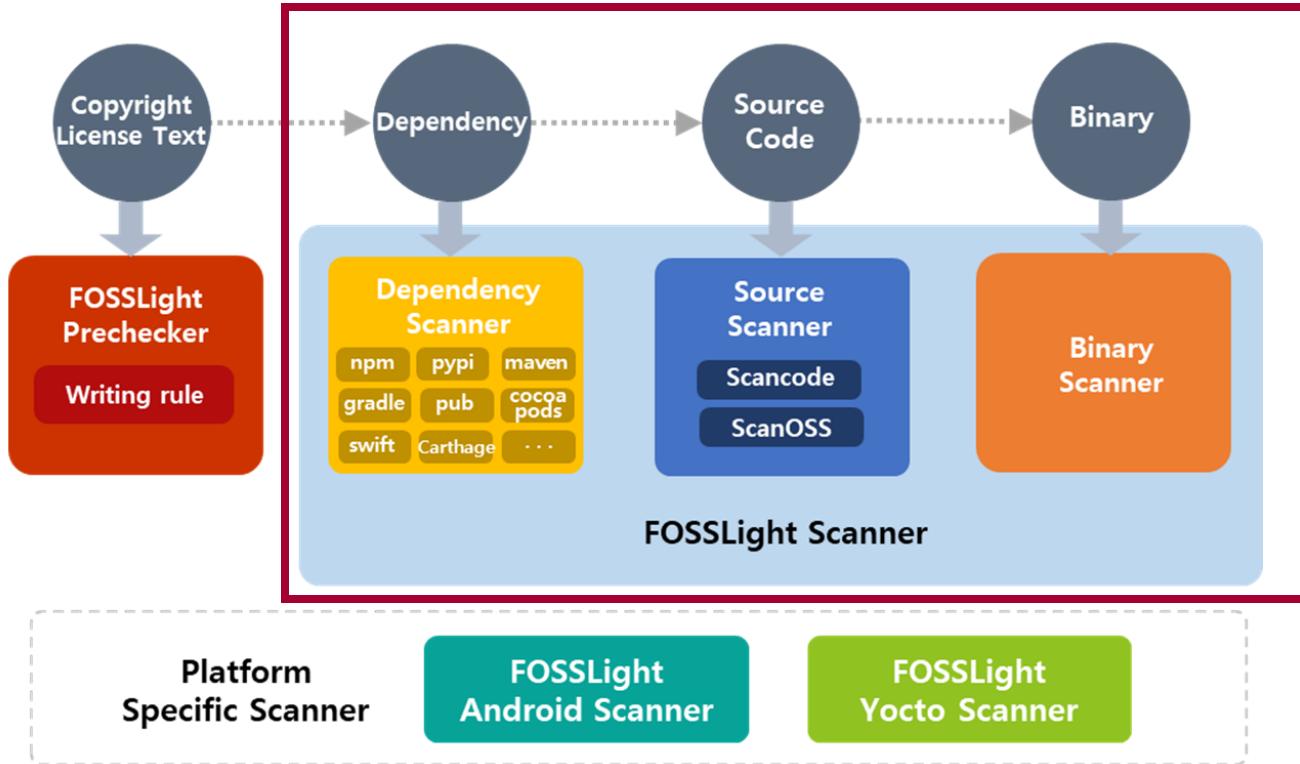
Compliant가 NOT-OK인 경우, [Source Code 내 저작권 및 License 표기 규칙](#)을 참조하시기 바랍니다.

# **FOSSLight Scanner with Jenkins**

---

# FOSSLight Scanner

- 하기 분석을 수행하는 Scanner
  - FOSSLight Dependency Scanner
  - FOSSLight Source Scanner
  - FOSSLight Binary Scanner



- 특정 Path 또는 다운로드 받을 수 있는 링크에 대해  
Open Source 분석을 수행하고, FOSSLight Report를 출력하는 도구

# Jenkins 프로젝트에서 FOSSLight Scanner 연동하기

1. Project 생성
2. 실행 주기 설정 -> 빌드 유발
3. FOSSLight Scanner 실행 환경 설정



The screenshot shows the Jenkins build configuration for a project named 'Virtualenv Builder'. It includes sections for 'Virtualenv Builder', 'Python version' (set to python3.7), 'Nature' (set to Shell), and 'Command'. The 'Command' section contains the following Jenkinsfile code:

```
pip install fosslight_scanner  
fosslight source -p src/* -f yaml -o fl_source_${BUILD_NUMBER}.yaml  
if [ -f fl_source_latest_build.yaml ]; then  
    fosslight compare -p fl_source_latest_build.yaml fl_source_${BUILD_NUMBER}.yaml -f html -o fl_compare_result.html  
else  
    echo "<b>This is a first build job for ${JOB_NAME}, so it cannot display the BOM comparison result.</b><br/>" > fl_compare_result.html  
fi  
mv fl_source_${BUILD_NUMBER}.yaml fl_source_latest_build.yaml
```

분석 후 추출된 SBOM을 이전 버전에서 분석된 SBOM파일과 Compare

4. 결과 파일 메일링

# FOSSLight Scanner Compare 결과 예시

File ( Total 3 ) ▾ fl\_compare\_result.xlsx (5.2KB)

Internet Header ▾

## FOSSLight Scanner Compare Result

### BOM Compare Result

- Before FOSSLight Report file: /var/lib/jenkins/workspace/FL Scanner Example project/fl\_source\_latest\_build.yaml
- After FOSSLight Report file: /var/lib/jenkins/workspace/FL Scanner Example project/fl\_source\_2.yaml

Status	OSS_Before	License_Before	OSS_After	License_After
Same				

### BUILD SUCCESS

URL: <http://10.178.94.16:8088/job/FL%20Scanner%20Example%20project/2/>

Project: FL Scanner Example project

Date: Tue, 16 Aug 2022 19:37:54 +0900

Duration: 1 min 43 sec

Cause: Started by user admin

이전 버전 대비  
추가/수정/삭제된 OSS 정보를  
리포팅

-> 신규 추가된 OSS에 대한  
보안취약점 확인하고,  
License를 확인하여 Compliance  
준수하도록 가이드

# FOSSLight MCP 서버

---

# FOSSLight MCP 서버

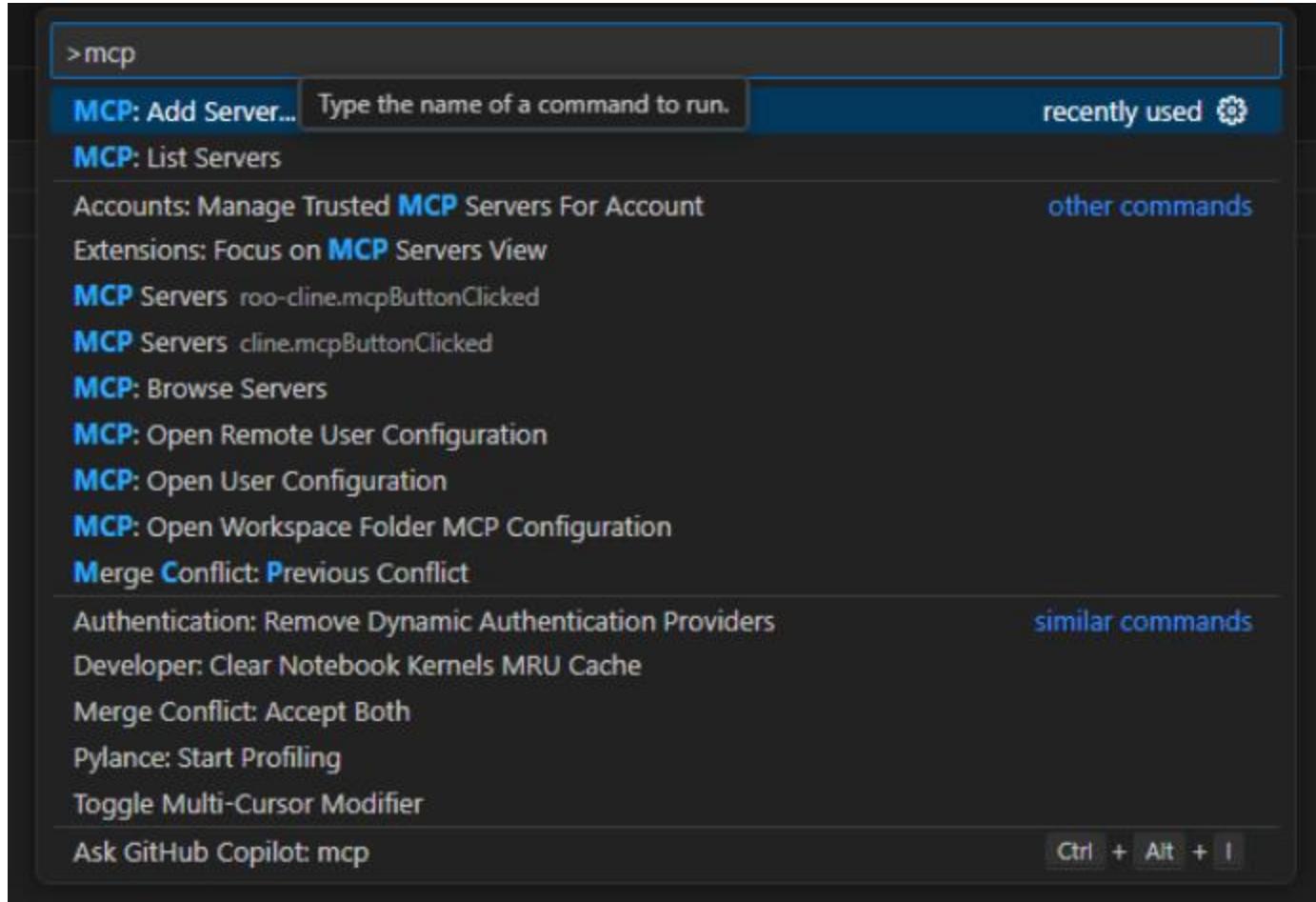
- FOSSLight Scanner Service에 프로젝트 생성 / 조회 / 분석 / 요약 등의 기능 제공

The screenshot displays the FOSSLight MCP Server interface. At the top, there is a navigation bar with the FOSSLight logo, 'Projects', 'Guide', 'Support', 'Notice', and a user profile for 'soim.kim'. Below the navigation bar is a search bar labeled 'Search projects' and filter buttons for 'ViewPermission', 'Pipeline', 'Status', and 'Sort'. The main area shows a table of projects with columns: Name, Dependencies, Detected Open Source, Pipelines, Status, and ViewPermission. There are four large red circles overlaid on the table, each containing a Korean word: '생성' (top-left), '분석' (top-right), '조회' (bottom-left), and '요약' (bottom-right). The table data includes:

Name	Dependencies	Detected Open Source	Pipelines	Status	ViewPermission
testtttttt Created 3 days ago	8	17	scan_all	Success	PRIVATE
test Created 2 weeks ago	8	17	scan_all	Success	PRIVATE
test Created 2 weeks ago	8	17	scan_all	Success	PRIVATE
test Created 2 weeks ago	8	17	scan_all	Success	PRIVATE

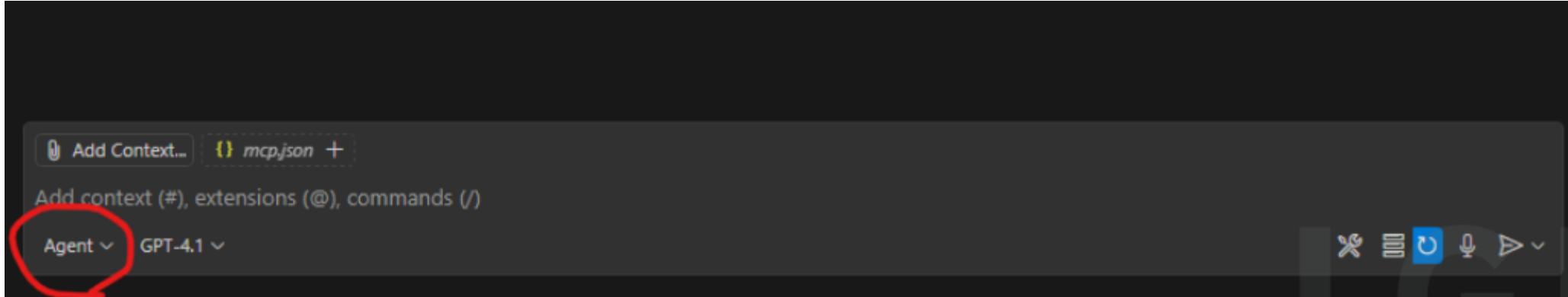
# VSCODE와 Github Copilot 연동 예시

- MCP : Add Server 로 FOSSLight MCP 서버 연동



# vSCode내 Github Copilot 연동 예시

- Copilot Chat에서 Agent 모드로 설정



# VSCODE내 Github Copilot 연동 예시

- FOSSLight Scanner Service Project 생성 및 분석

The screenshot shows a terminal window in VS Code with the following content:

fosslight에 minss\_test 프로젝트를 만들고 분석해줘. repository는 [https://github.com/fosslight/fosslight\\_source\\_scanner.git](https://github.com/fosslight/fosslight_source_scanner.git) 약

Ran `create_and_analyze_project` - fosslight (MCP Server) ✓

**Input**

```
{  
  "name": "minss_test",  
  "url": "https://github.com/fosslight/fosslight_source_scanner.git"  
}
```

**Output**

```
{"success":true,"reply":": **프로젝트 생성 및 분석 시작 완료!**\n\n💡 **프로젝트 URL**: https://fs.lge.com/project/minss\_test-202509092357-e384876b\n📊 **분석 데이터**: {'project_uuid': 'e384876b-d8b5-4b54-b098-bad5d5e60f50'}\n\n⌚ 분석이 진행 중입니다. 완료되면 결과를 다운로드할 수 있습니다."}
```

```
{  
  "success": true,  
  "reply": ": **프로젝트 생성 및 분석 시작 완료!**\n\n💡 **프로젝트 URL**: https://fs.lge.com/project/minss\_test-202509092357-e384876b\n📊 **분석 데이터**: {'project_uuid': 'e384876b-d8b5-4b54-b098-bad5d5e60f50'}\n\n⌚ 분석이 진행 중입니다. 완료되면 결과를 다운로드할 수 있습니다."  
}
```

minss\_test 프로젝트가 FOSSLight에 생성되었고 분석이 시작되었습니다.

- 프로젝트 URL: [https://fs.lge.com/project/minss\\_test-202509092357-e384876b](https://fs.lge.com/project/minss_test-202509092357-e384876b)
- 분석 진행 중이며, 완료 후 결과를 다운로드할 수 있습니다.

분석이 완료되면 결과를 바로 안내해드릴 수 있습니다.

# VSCODE내 Github Copilot 연동 예시

- FOSSLight Scanner Service Project 조회

The screenshot shows a terminal window with the following text:

```
fosslight에서 나의 프로젝트 리스트 보여줘
> Ran list_projects - fosslight (MCP Server) ✓
OSSLight에서 조회된 나의 프로젝트 리스트(총 164개)입니다. 최근 생성된 주요 프로젝트 일부는 아래와 같습니다:
1. fosslight_source_scanner_analysis_retry-202509090701 (분석결과 있음)
2. ethan_mcp_test-202509090658 (분석결과 없음)
3. fosslight_mcp_test-202509031359 (분석결과 있음)
4. file_upload_mcp-202509011532 (분석결과 없음)
5. AISE_SAMPLE_project-202509010520 (분석결과 없음)
더 많은 프로젝트가 있으며, 특정 프로젝트의 분석결과가 궁금하면 [번호]번 프로젝트 결과 분석해줘라고 요청하시면 됩니다.
```

At the bottom left of the terminal window, there are small icons: a circular arrow, a double arrow, a square, and a triangle.

# VSCODE내 Github Copilot 연동 예시

- FOSSLight Scanner Service Project 결과 요약

The screenshot shows a dark-themed web interface for analyzing open-source projects. At the top right, there's a button labeled "1번 프로젝트 결과 분석해줘". Below it, a message says "1번 프로젝트(fosslight\_source\_scanner\_analysis\_retry-202509090701) 결과 분석입니다." A section titled "주요 위험 요소" lists vulnerabilities categorized by license: GPL (high), LGPL (medium), and MPL (medium). Another section, "시트별 현황", provides a breakdown of files and packages across source code, third-party packages, and binary files, showing the count and risk level for each. A "대응 방안" section offers three steps for handling GPL/MPL dependencies. A "참고자료" section links to the FOSSLight Hub guide and other resources. At the bottom, a note mentions additional information like per-package license details.

1번 프로젝트(fosslight\_source\_scanner\_analysis\_retry-202509090701) 결과 분석입니다.

주요 위험 요소

- GPL 라이선스(매우 높음): pypi:rfc3987, pypi:gemfileparser2, pypi:text-unidecode (총 6개 파일)
- LGPL 라이선스(높음): pypi:chardet, pypi:crc32c, pypi:intbitset, pypi:psycopg2-binary (총 8개 파일)
- MPL 라이선스(중간): pypi:certifi, pypi:pathspec, pypi:fqdn (총 6개 파일)

시트별 현황

구분	파일/패키지 수	주요 위험 라이선스	위험도
소스코드	36개	없음	낮음
의존성 패키지	297개	GPL, LGPL, MPL	높음/중간
바이너리 파일	데이터 없음	없음	낮음

대응 방안

- GPL/AGPL 패키지 제거 또는 대체: 위험 패키지(GPL 등)는 MIT, Apache 2.0, BSD 등으로 대체 권장
- LGPL/MPL 패키지 검토: 수정/링크 여부 및 공개 요구사항 확인
- 라이선스 클리어링 문서화 및 컴플라이언스 프로세스 구축

참고자료

- FOSSLight Hub 가이드
- 라이선스 가이드
- 대체 OSS 검색

추가로 상세 시트별 위험도, 패키지별 라이선스 정보도 제공 가능합니다. 필요한 정보가 있으면 말씀해 주세요!



# Q&A

