

Continuous Test & Integration Platform



C

ONTENTS

1. CTIP이란 무엇인가?

2. 요구사항관리_ Github project

3. 이슈관리_ Github issue tracker

4. Jenkins와 연동

5. Question & Answer



1. CTIP이란 무엇인가?

Continuous Testing and Integration Platform

CI + CT

CI: Practice of merging all developer working copies to a shared mainline several times a day.

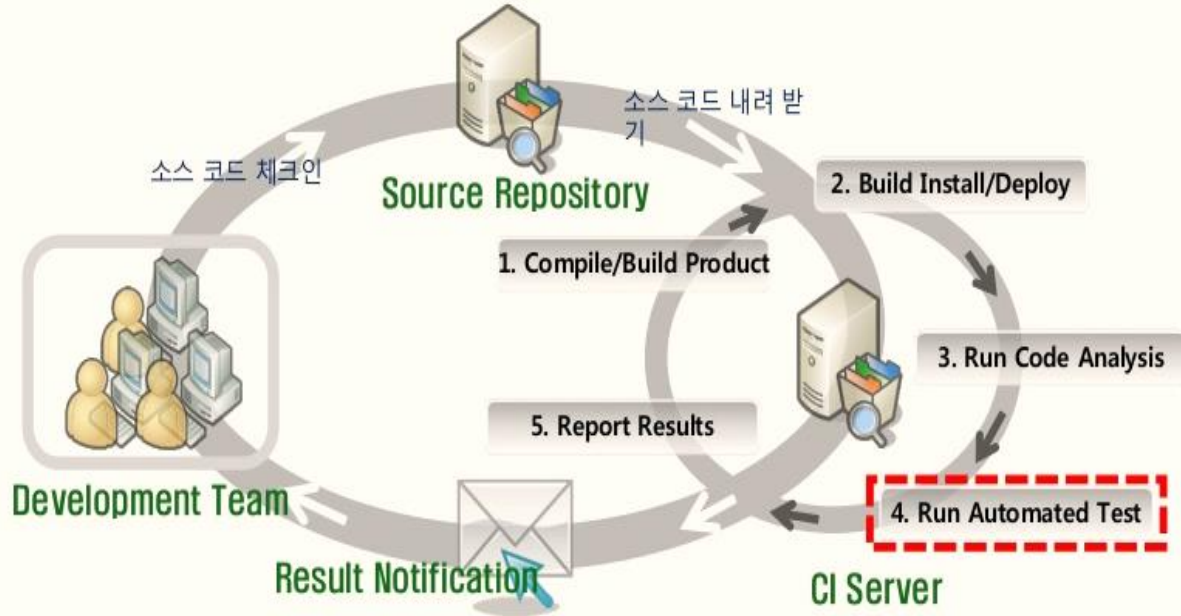
CT: Process of executing **AUTOMATED TEST** as PART of the software delivery pipeline to obtain immediate feedback of the business risks associated with software release candidate.

Def:

CI 개념을 개발 프로젝트에 쉽게 적용하기 위한 Platform

→ 빌드도구, 테스트 도구 및 기타개발 시 유용한 도구들이 유기적으로 동작하도록 묶은 도구들의 집합

∴ 빌드 자동화 + 테스트 자동화



출처:테스터도 알아야할 웹 개발

CTIP을 통해 달성할 수 있는 CI 수행요건

- 단일 Source Repository 관리.
- 빌드 자동화.
- 자체적으로 테스트가능한 빌드.
- 빠른 빌드 수행.
- 운영환경과 유사한 환경 구성.
- 최신 결과물에 대한 쉬운 접근.
- 손쉬운 빌드 상태 모니터링.

제공하는 기능

- CI서버를 통한 지속적 통합 및 빌드
- 품질 도구들을 통한 코드 품질 검토 (테스트 및 정적분석)
- 빌드 결과의 배포 및 관련자에게 통보

_Martin Fowler

2. 요구사항관리

_Github project

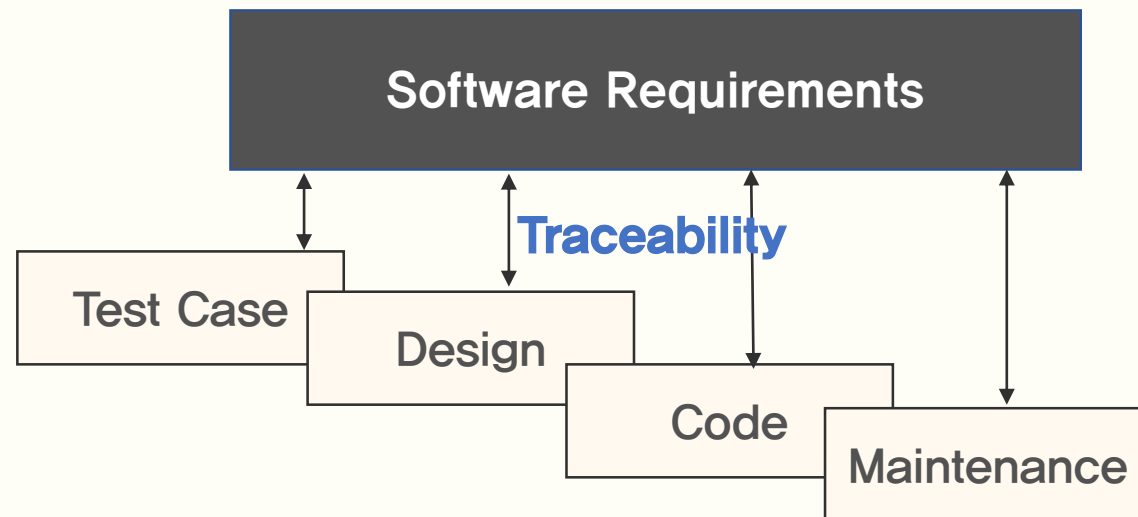
요구사항 관리?

Def:

- 프로젝트와 관련된 이해관계자들로부터 요구사항을 추출, 구성 및 문서화하고 변경에 대한 동의를 설정해 관리하는 시스템적 활동.
- 참여자들 사이에서 효과적인 의사소통 전략을 사용하여 충분한 합의와 협의를 통해 공통의 이해를 구축하고, 개발 생명주기의 전체 기간동안 지속적 관리를 제공함.

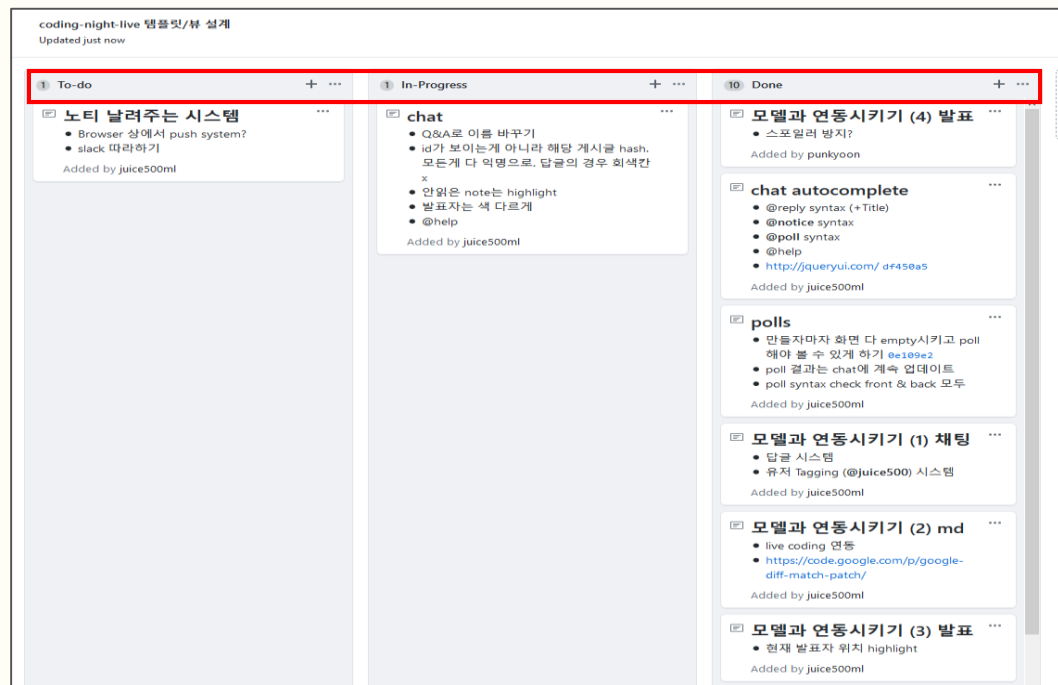
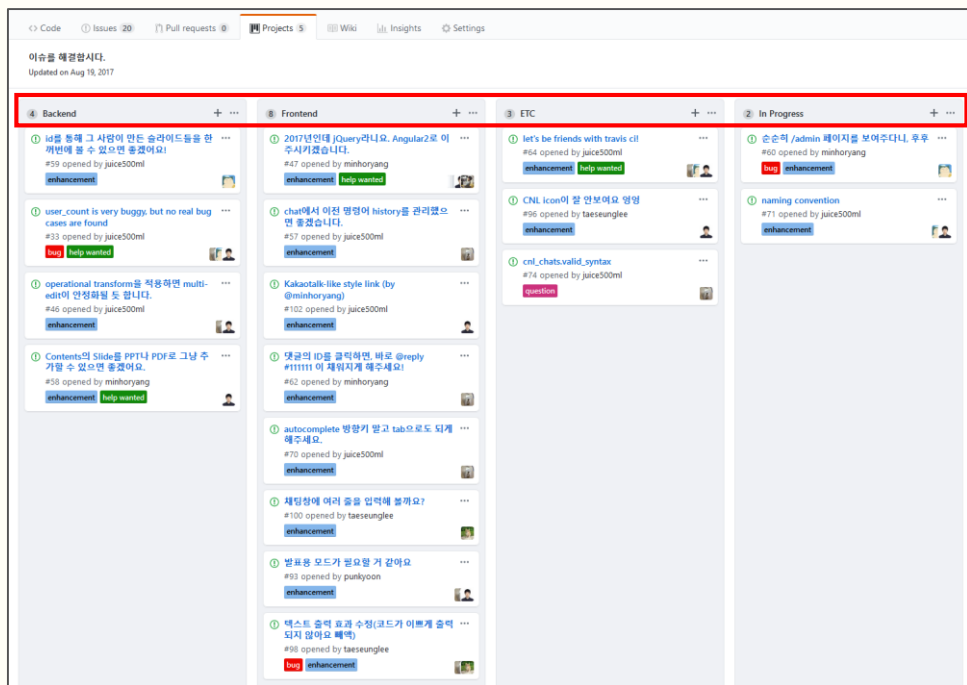
필요성

- 요구수준의 고도화.
- 이해당사자의 다양화.
- 대부분의 에러가 생명주기의 요구사항 단계에서 발생 등.



#Kanban chart: 연속적인 흐름모델

- 데드라인은 없지만, 속도에 대한 압박이 존재하는 시스템.
- Work-In-Process가 동시에 개발이 진행 될 수 있는 아이템의 수를 제한한다.



출처: <https://github.com/punkyoon>



- 계획적이지 않음으로 인해 소프트웨어 품질에 초점을 맞출 수 있게 됨.
- 진행하는 일의 제한을 두어 더 적은 아이템을 더 빨리 만들어 내는 것에 집중 가능.

Issue Tracking System(ITS)?

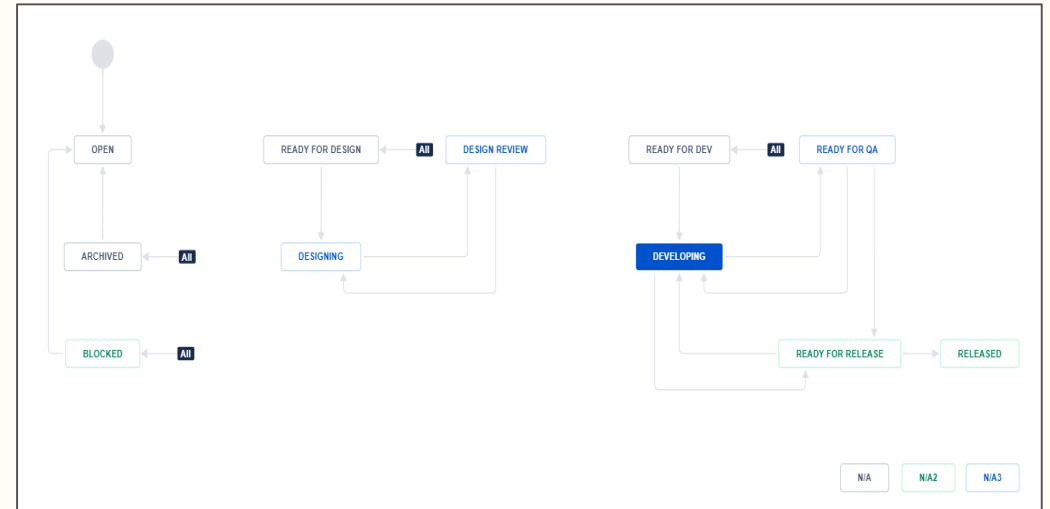
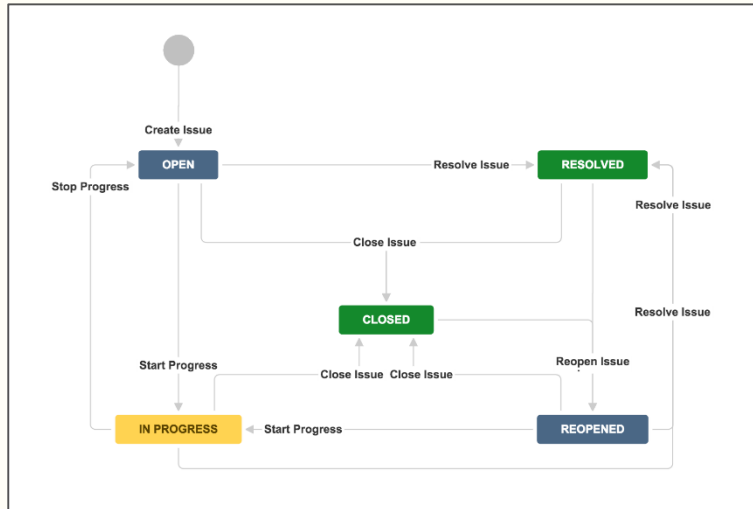
Def:

- 버그, 요구사항, 작업내용 등이 있을 때 해당 시스템에 게시물형태로 올리고 개발자, 테스터들이 작업 진행상황을 기록하는 시스템
- 대부분 버그관리시스템(Bug Tracking System)에서 출발하였으나, 단순히 버그 뿐 아니라 다양한 이슈를 관리할 수 있도록 의미가 확장되고 있다.

Issue vs Bug

- Issue: Does not really indicate that there is a problem in developer's code.
- Bug: Surely indicate that there is problem in developer's code.

Flow chart Example



Issue open 시간

Issue open한 사람

Label

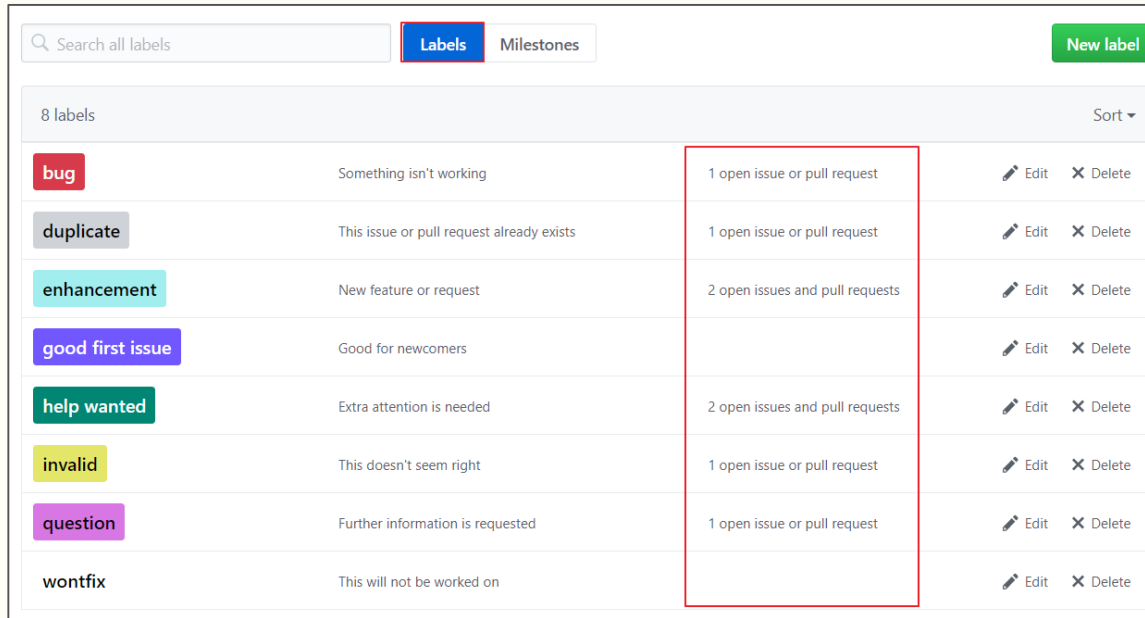
Milestone

Assignee

Github Issue Tracker만의 특징

1. 이슈를 제목과 description만으로 설명할 수 있다.
2. 색이 다른 라벨을 통해 이슈들을 필터링 할 수 있다.
3. Milestone을 프로젝트 내의 프로젝트 처럼 활용할 수 있다.
4. 일을 할당 받은 사람은(Assignee) 대답할 책임 있다.
5. 누구든지 자유롭게 Comments를 남길 수 있다.
6. Issue Dashboard, Pulse를 통해 기여도를 확인이 쉽다.
7. Assignee를 할당할 수 있다.

등의 다양한 기능을 개발자들이 가장 많이 사용하는 Github에서 사용할 수 있다는 것이 가장 큰 장점

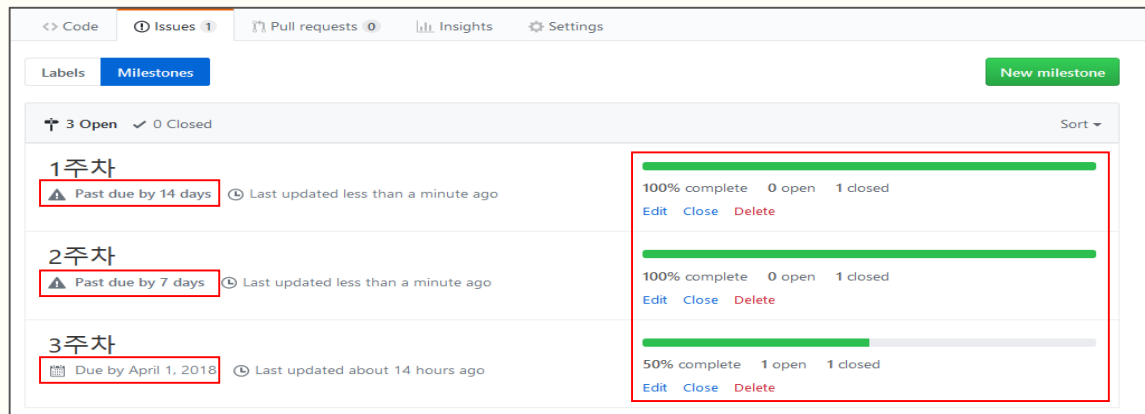


The screenshot shows the Github Labels management interface. At the top, there is a search bar for labels and a 'New label' button. Below, a list of 8 labels is displayed, each with a unique color and a description. A red box highlights the 'Number of open issues or pull requests' column for each label.

Label	Description	Number of open issues or pull requests	Actions
bug	Something isn't working	1 open issue or pull request	Edit Delete
duplicate	This issue or pull request already exists	1 open issue or pull request	Edit Delete
enhancement	New feature or request	2 open issues and pull requests	Edit Delete
good first issue	Good for newcomers		Edit Delete
help wanted	Extra attention is needed	2 open issues and pull requests	Edit Delete
invalid	This doesn't seem right	1 open issue or pull request	Edit Delete
question	Further information is requested	1 open issue or pull request	Edit Delete
wontfix	This will not be worked on		Edit Delete

#Labels

- 다른 종류의 이슈들을 묶는 Label-ing.
- 하나의 이슈는 다수의 Label을 가질 수 있음.
- Label별 sorting으로 이슈 확인 용이.

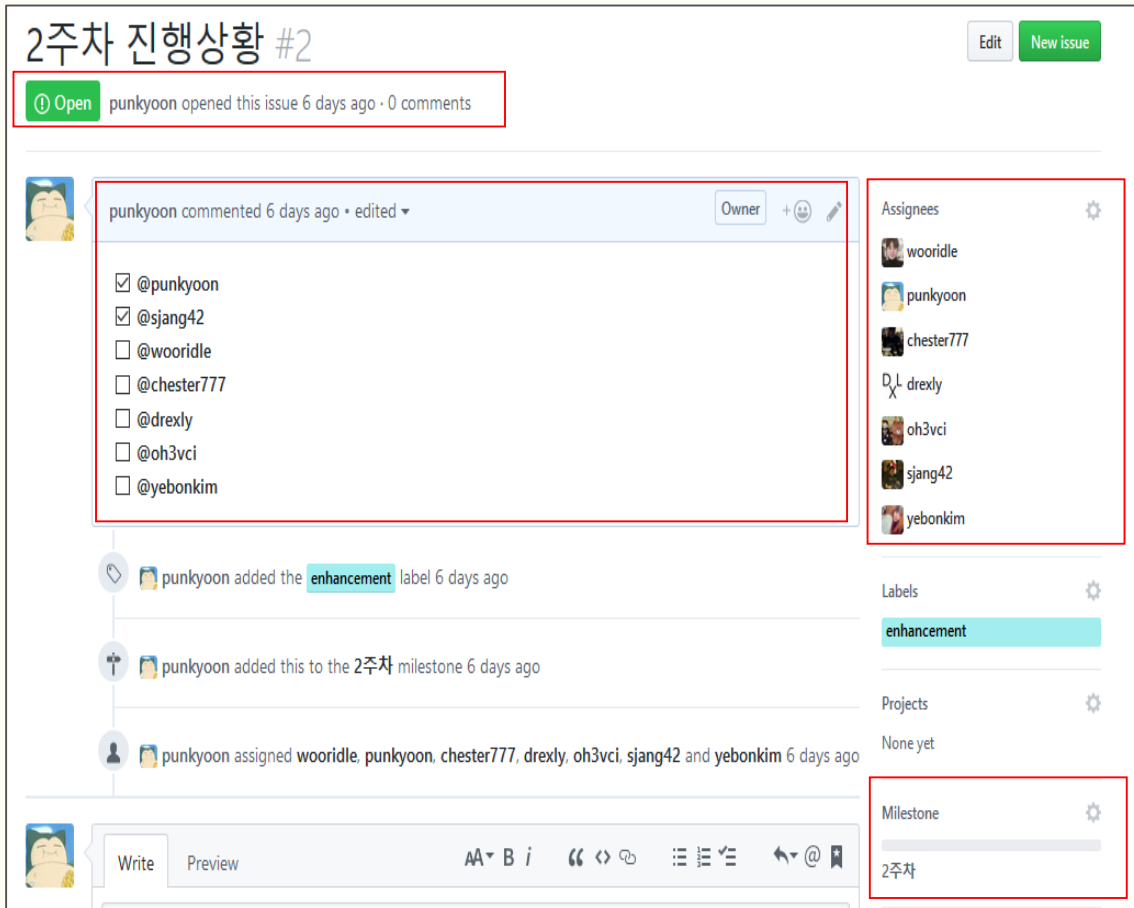


The screenshot shows the Github Milestones management interface. At the top, there are tabs for 'Labels' and 'Milestones', and a 'New milestone' button. Below, a list of 3 milestones is displayed, each with a title, a progress bar, and a completion status. A red box highlights the 'Due by' date for each milestone.

Milestone	Due by	Completion Status	Actions
1주차	Past due by 14 days	100% complete 0 open 1 closed	Edit Close Delete
2주차	Past due by 7 days	100% complete 0 open 1 closed	Edit Close Delete
3주차	Due by April 1, 2018	50% complete 1 open 1 closed	Edit Close Delete

#Milestone

- 이슈들의 그룹 개념: 시간, 특징 등에 따라 Grouping.
- 팀의 성격, 개발 단계에 맞게 다양한 활용 가능.



2주차 진행상황 #2

Open punkyoon opened this issue 6 days ago · 0 comments

punkyoon commented 6 days ago · edited

- @punkyoon
- @sjang42
- @wooridle
- @chester777
- @drexly
- @oh3vci
- @yebonkim

Assignees

- wooridle
- punkyoon
- chester777
- drexly
- oh3vci
- sjang42
- yebonkim

Labels

enhancement

Projects

None yet

Milestone

2주차



Feature extraction (Re-id) #8

Closed punkyoon opened this issue on Oct 29, 2017 · 2 comments

punkyoon commented on Oct 29, 2017 · edited

과연 with에 걸려있는 tensorflow session이 끝나기 전에 return 해서 GPU 메모리가 터지는 것일까? 두둔!

https://github.com/dzzp/argos-back/blob/master/object_detection/main.py#L75-L76

punkyoon added bug enhancement labels on Oct 29, 2017

punkyoon self-assigned this on Oct 29, 2017

punkyoon changed the title from Extract feature (Re-id) to Feature extraction (Re-id) on Oct 29, 2017

punkyoon commented on Oct 29, 2017

세션이 닫힌 후에 코드를 실행한다고 해서, 잘 돌아가는 것이 아니었다...

punkyoon commented on Oct 29, 2017 · edited

멀티프로세싱으로 해결하였다!!

실패 기록

- 스레드
- 프로세스 풀(pool)
- `nvidia-smi --gpu-reset -i 0`
- `tf.Session().close()`
- `tf.reset_default_graph`

punkyoon added a commit that referenced this issue on Oct 31, 2017

fix issue #8

punkyoon closed this on Oct 31, 2017

Milestone 내의 이슈 예시

이슈 예시

4. Jenkins 연동

Jenkins 사용 순서

1. 환경설정(Jenkins 관리)

- 플러그인 관리
- Global Tool Configuration
- 시스템 설정

2. 보안설정(Jenkins 관리)

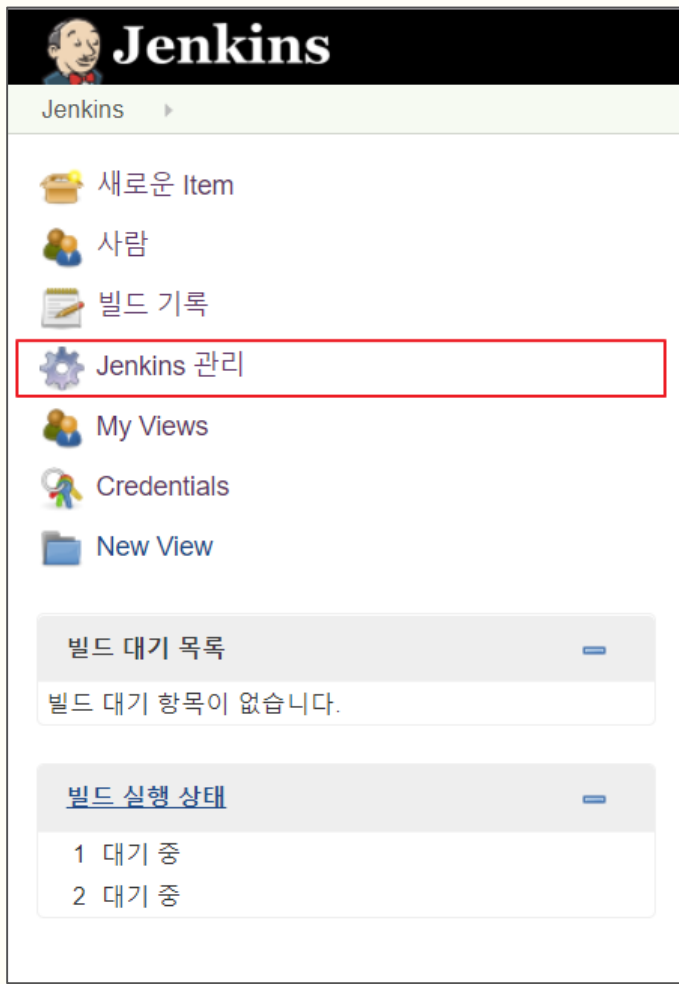
: Configure Global Security

3. 새로운 프로젝트 생성(새로운 Item)



: Freestyle project

4. 빌드 기록 확인

- 프로젝트 별
- 전체 빌드 기록



Jenkins 관리

-  **플러그인 관리**
Jenkins의 기능을 확장하기 위한 플러그인을 추가, 제거, 사용, 미사용으로 설정할 수 있습니다.
🔔 업데이트 가능함
-  **Global Tool Configuration**
Configure tools, their locations and automatic installers.
-  **시스템 설정**
환경변수 및 경로 정보등을 설정합니다.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials
Configure the credential providers and types

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

시스템 정보
문제 해결을 돕기위한 다양한 환경 정보를 보여줍니다.

		Filter: <input type="text"/>		
업데이트된 플러그인 목록		설치 가능	설치된 플러그인 목록	고급
사용가능	이름 ↓	버전	이전 설치 버전	설치 제거
<input checked="" type="checkbox"/>	bouncycastle API Plugin This plugin provides an stable API to Bouncy Castle related tasks.	2.16.2		설치 제거
<input checked="" type="checkbox"/>	Command Agent Launcher Plugin Allows agents to be launched using a specified command.	1.2		설치 제거
<input checked="" type="checkbox"/>	Gradle Plugin This plugin allows Jenkins to invoke Gradle build scripts directly.	1.28		설치 제거
<input checked="" type="checkbox"/>	Structs Plugin Library plugin for DSL plugins that need names for Jenkins objects.	1.14		설치 제거
<input checked="" type="checkbox"/>	GitHub Branch Source Plugin Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.	2.3.3		설치 제거
<input checked="" type="checkbox"/>	GitHub Issues Plugin This plugin creates GitHub issues when builds fail, and automatically closes the issue when the build starts passing again.	1.2.4		설치 제거

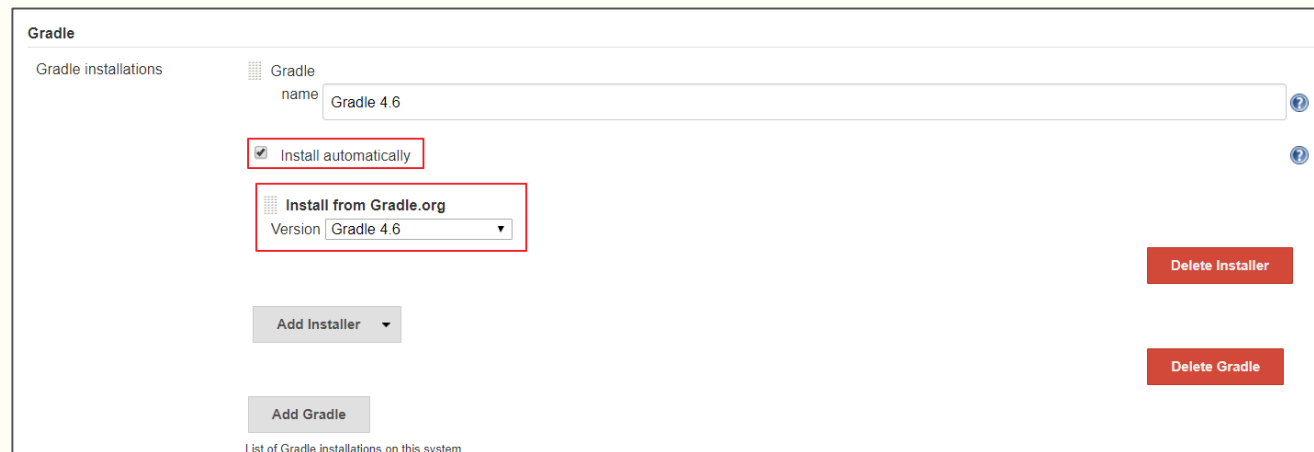
프로젝트에 필요한 플러그인을 검색한 후 설치한다.

→ Build 도구로 Gradle을, 형상관리 도구로 Github, 이슈관리 도구로 Github Issue tracker를 사용.



JDK 설정

- 자동 or 수동으로 JDK를 설정할 수 있다.
- 자동을 선택하면 Oracle 계정을 입력해야 한다.
- 복수의 JDK version 등록 가능.



Gradle 설정

- 자동 or 수동으로 Gradle을 설정할 수 있다.
- 복수의 Gradle version 등록 가능.

Jenkins Location

Jenkins URL

System Admin e-mail address

Jenkins Location 설정

GitHub

GitHub Servers

Override Hook URL Specify another hook URL for GitHub configuration

Shared secret

Additional actions

Convert login and password to token

GitHub API URL

From credentials

From login and password

Login

Password

GitHub Server

Name

API URL

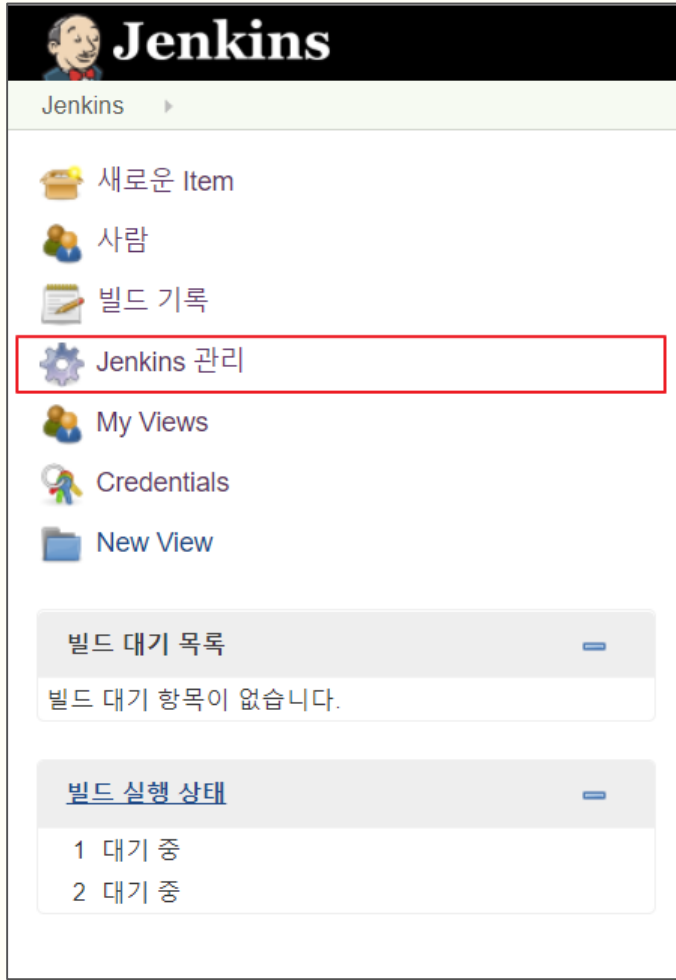
Credentials

Credentials verified for user 3ssarah, rate limit: 4998

Manage hooks

Github 서버 설정

- Convert login and password 선택 후 토큰자격 증명 만들기.
- [Test Connection] 클릭하여 인증이 작동하는지 확인.



Jenkins 관리



시스템 설정

환경변수 및 경로 정보등을 설정합니다.



Configure Global Security

Secure Jenkins; define who is allowed to access/use the system.



Configure Credentials

Configure the credential providers and types



Global Tool Configuration

Configure tools, their locations and automatic installers.



Reload Configuration from Disk

Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.



플러그인 관리

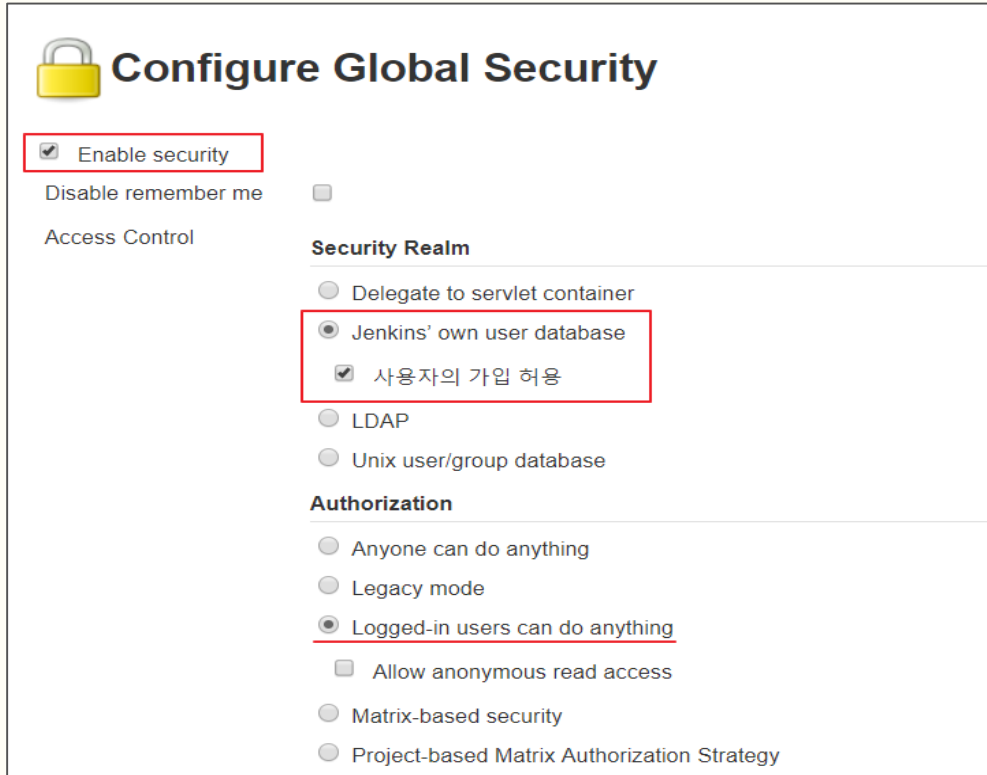
Jenkins의 기능을 확장하기 위한 플러그인을 추가, 제거, 사용, 미사용으로 설정할 수 있습니다.

🔔 업데이트 가능함



시스템 정보

문제 해결을 돕기위한 다양한 환경 정보를 보여줍니다.



Configure Global Security

Enable security

Disable remember me

Access Control

Security Realm

- Delegate to servlet container
- Jenkins' own user database
 - 사용자의 가입 허용
- LDAP
- Unix user/group database

Authorization

- Anyone can do anything
- Legacy mode
- Logged-in users can do anything
 - Allow anonymous read access
- Matrix-based security
- Project-based Matrix Authorization Strategy

모든 사용자가 기본적으로 로그인 없이 접근하여 관리자 권한 가짐

→ 계정에 따른 권한 제어로 수정하는 것이 필요함.

Root: Jenkins 관리 > Configure Global Security

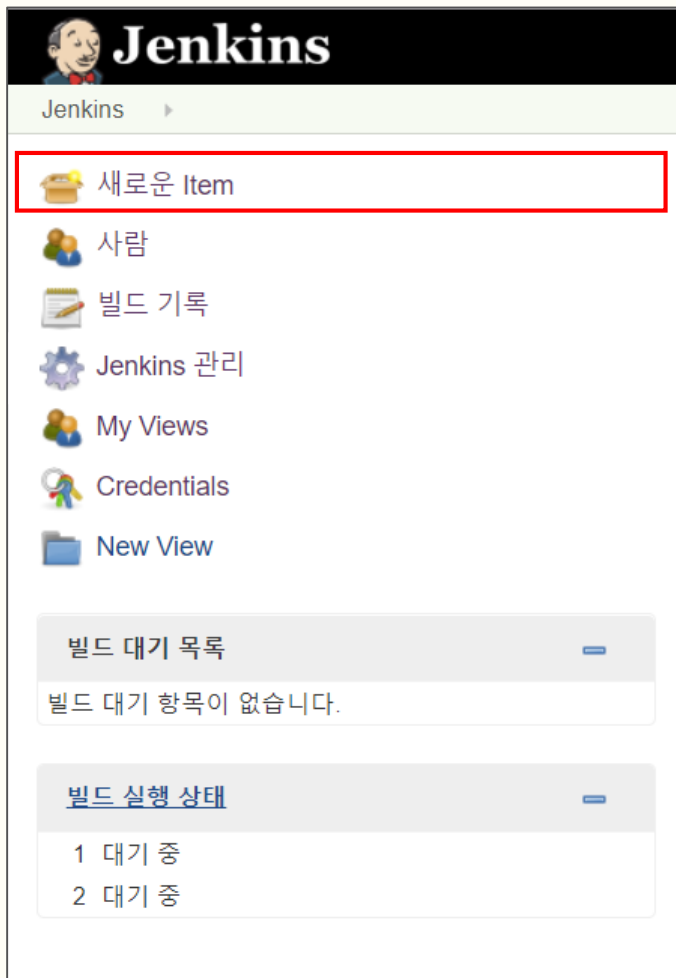
1. Enable Security 체크
2. Security realm 설정
3. Authorization 설정

Matrix-based security

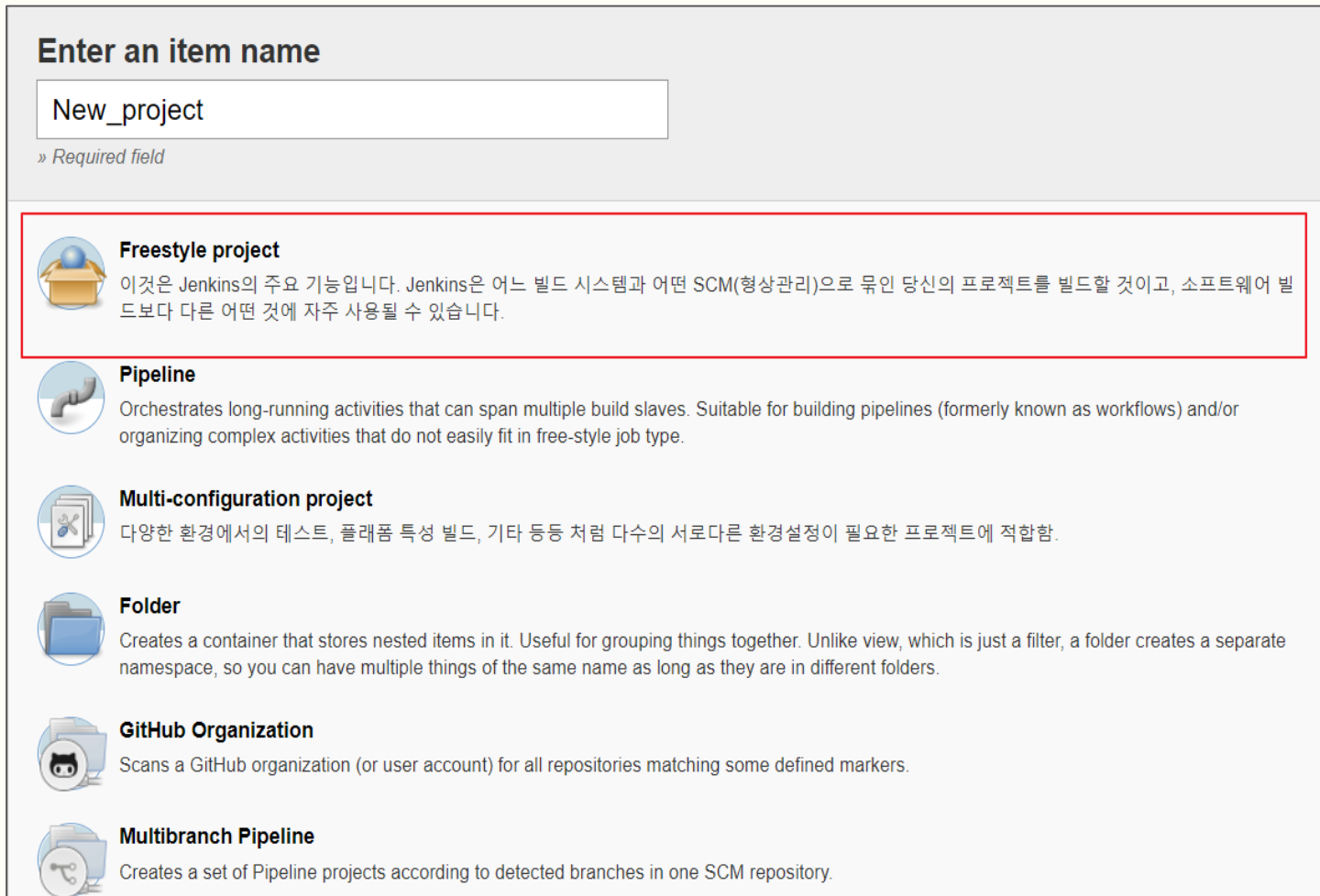
User/group	Overall	Credentials				Agent				Job				Run				View				SCM									
	Administer	Read	Create	Delete	Manage	Domains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Build	Cancel	Configure	Create	Delete	Discover	Move	Read	Workspace	Delete	Replay	Update	Configure	Create	Delete	Read	Tag
Anonymous Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authenticated Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
관리자	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

User/group to add: Add

Project-based Matrix Authorization Strategy

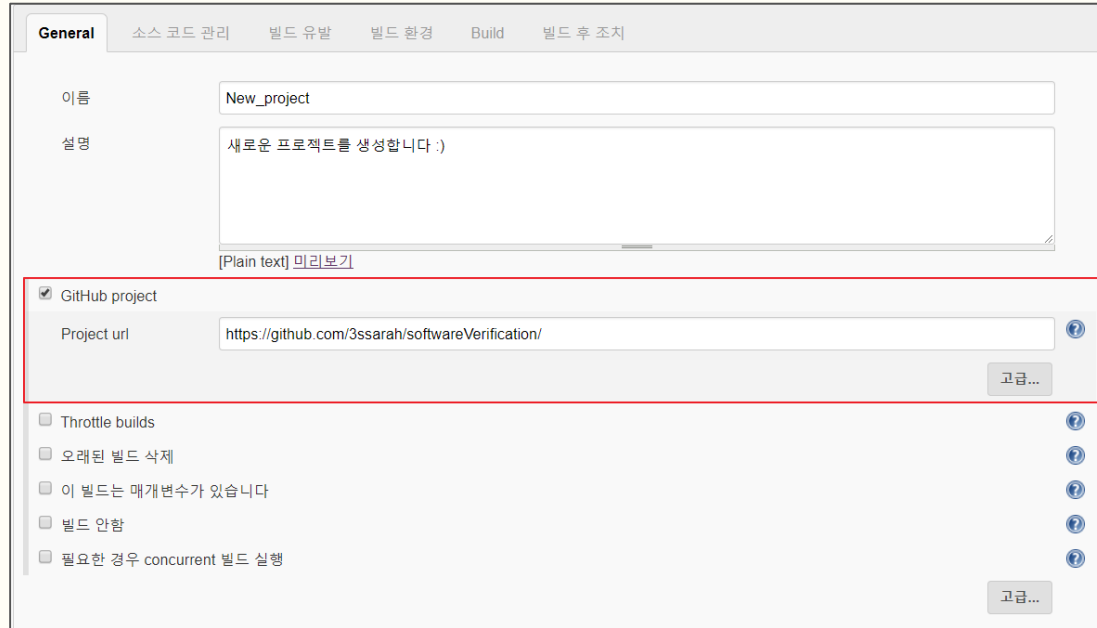


The screenshot shows the Jenkins dashboard. At the top, there is a header with the Jenkins logo and name. Below it, a navigation menu lists several options: '새로운 Item' (New Item), '사람' (People), '빌드 기록' (Build History), 'Jenkins 관리' (Jenkins Management), 'My Views', 'Credentials', and 'New View'. The '새로운 Item' option is highlighted with a red box. Below the menu, there are two sections: '빌드 대기 목록' (Build Queue) and '빌드 실행 상태' (Build Execution Status). The '빌드 대기 목록' section shows '빌드 대기 항목이 없습니다.' (No build items in queue). The '빌드 실행 상태' section shows '1 대기 중' (1 in progress) and '2 대기 중' (2 in progress).



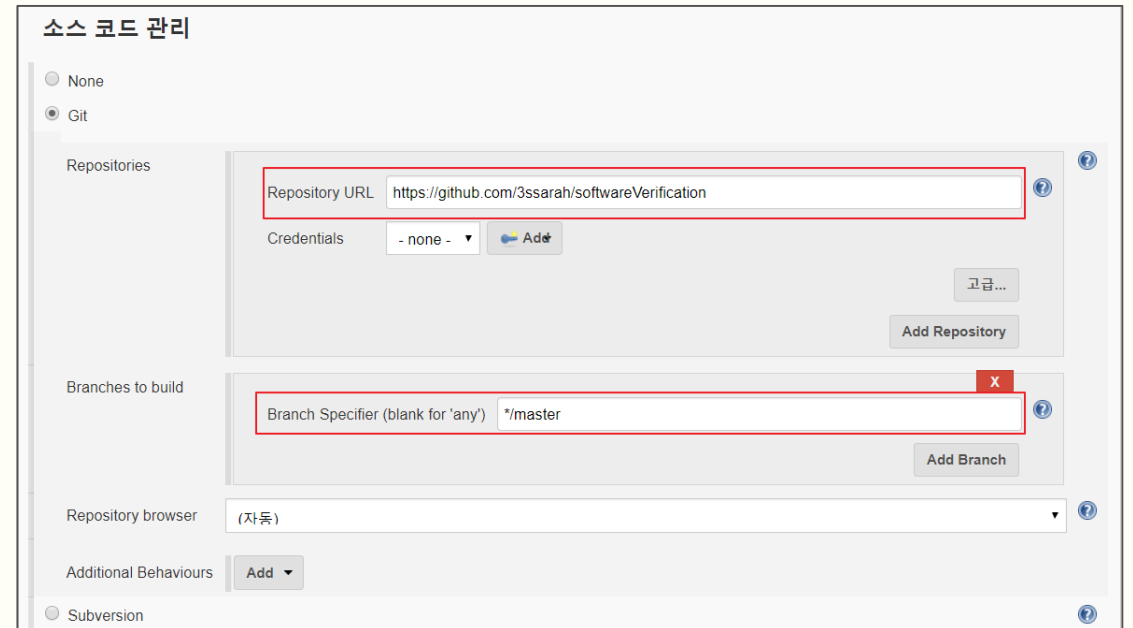
The screenshot shows the 'Enter an item name' dialog in Jenkins. The input field contains 'New_project'. Below the input field, there is a list of project types, with 'Freestyle project' highlighted by a red box. The list includes:

- Freestyle project**: 이것은 Jenkins의 주요 기능입니다. Jenkins은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝트를 빌드할 것이고, 소프트웨어 빌드보다 다른 어떤 것에 자주 사용될 수 있습니다.
- Pipeline**: Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: 다양한 환경에서의 테스트, 플래폼 특성 빌드, 기타 등등 처럼 다수의 서로다른 환경설정이 필요한 프로젝트에 적합함.
- Folder**: Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- GitHub Organization**: Scans a GitHub organization (or user account) for all repositories matching some defined markers.
- Multibranch Pipeline**: Creates a set of Pipeline projects according to detected branches in one SCM repository.



The screenshot shows the 'General' tab of a new Jenkins project configuration. The '이름' (Name) field is 'New_project' and the '설명' (Description) is '새로운 프로젝트를 생성합니다 :)'. The 'GitHub project' checkbox is checked, and the 'Project url' is 'https://github.com/3ssarah/softwareVerification/'. Below this, there are several unchecked options: 'Throttle builds', '오래된 빌드 삭제', '이 빌드는 매개변수가 있습니다', '빌드 안함', and '필요한 경우 concurrent 빌드 실행'. A red box highlights the 'GitHub project' section.

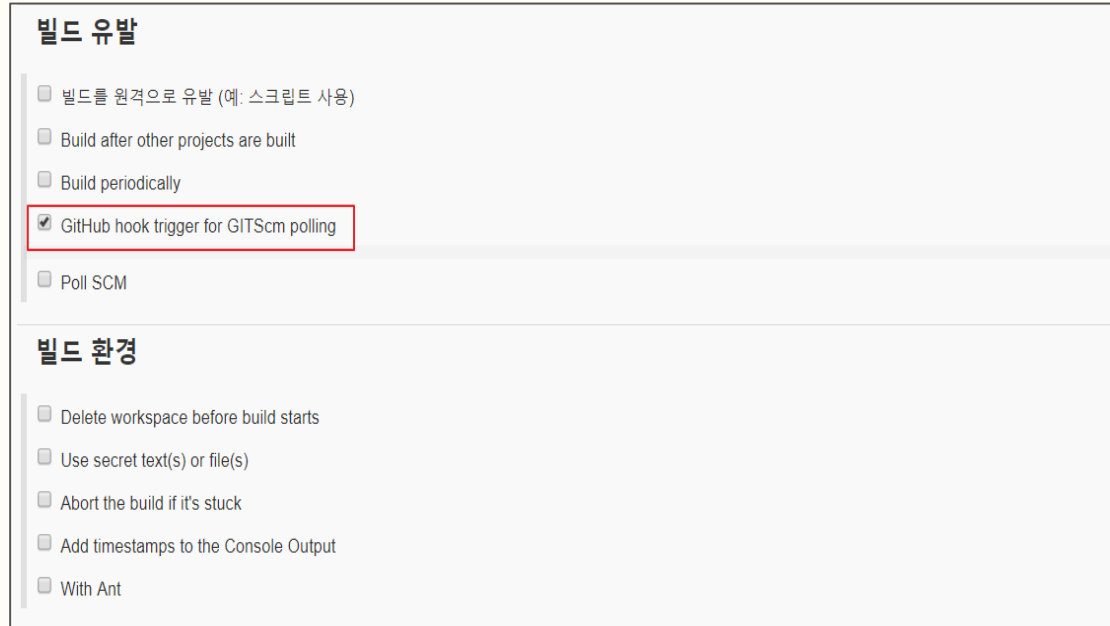
Root: 새로운 Item > Freestyle project > General



The screenshot shows the '소스 코드 관리' (Source Code Management) tab. The 'Git' radio button is selected. Under 'Repositories', the 'Repository URL' is 'https://github.com/3ssarah/softwareVerification/' and 'Credentials' is '- none -'. Under 'Branches to build', the 'Branch Specifier (blank for 'any')' is '*/master'. The 'Repository browser' is set to '(자동)'. A red box highlights the 'Repository URL' and 'Branches to build' fields.

Root: Freestyle project > 소스코드 관리

- 사용하는 형상관리 URL을 입력한다.
- 특정 브랜치만 관리 가능.



빌드 유발

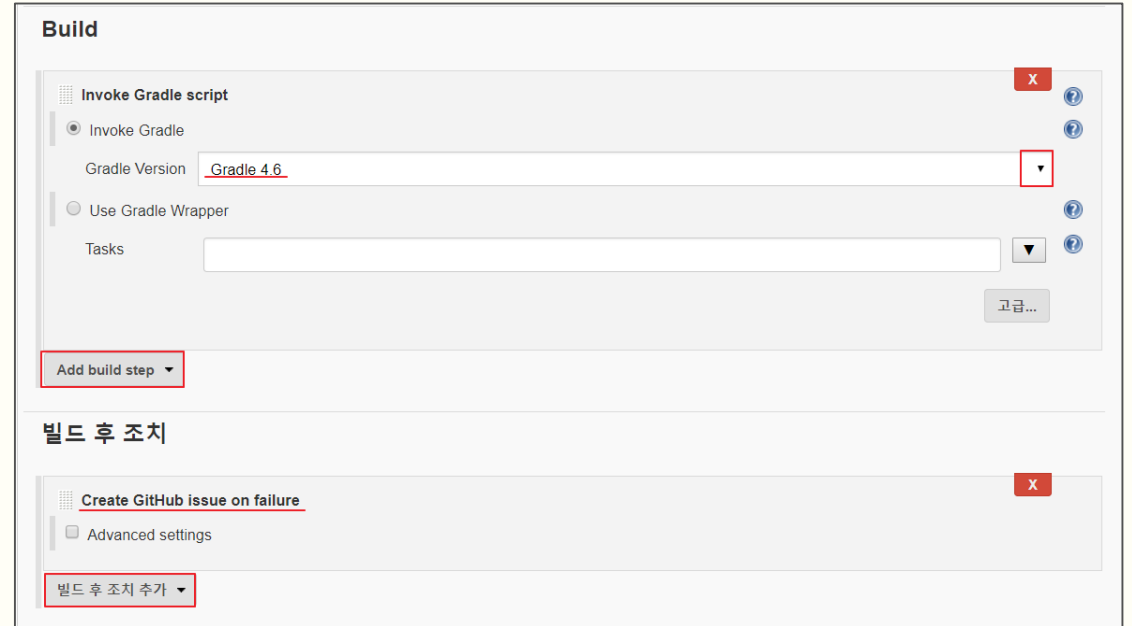
- 빌드를 원격으로 유발 (예: 스크립트 사용)
- Build after other projects are built
- Build periodically
- GitHub hook trigger for GITScm polling
- Poll SCM

빌드 환경

- Delete workspace before build starts
- Use secret text(s) or file(s)
- Abort the build if it's stuck
- Add timestamps to the Console Output
- With Ant

- # Root: Freestyle project > 빌드 유발
- 어떤 때 빌드를 할 지 Trigger를 설정.
 - 여기서는 Github에 push할 때

Root: Freestyle project > 빌드 환경



Build

Invoke Gradle script

- Invoke Gradle
- Gradle Version: Gradle 4.6
- Use Gradle Wrapper
- Tasks:
- 고급...

Add build step

빌드 후 조치

Create GitHub issue on failure

- Advanced settings

빌드 후 조치 추가

- # Root: Freestyle project > Build
- 각 프로젝트에 맞는 Build 도구를 설정.
 - Gradle을 사용할 경우, 고급설정으로 Task, 빌드 파일 등을 지정 가능.

- # Root: Freestyle project > 빌드 후 조치
- 여기서는 Github Issue tracker를 사용하기 위해 선택함.

빌드 유발

- 빌드를 원격으로 유발 (예: 스크립트 사용)
- Build after other projects are built
- Build periodically
- GitHub hook trigger for GITScm polling
- Poll SCM

Build

- Invoke Gradle script**
 - Invoke Gradle
 - Gradle Version:
 - Use Gradle Wrapper
 - Tasks:
 -
-

빌드 후 조치

- Publish HTML reports**
 - HTML directory to archive:
 - Index page[s]:
 - Index page title[s] (Optional):
 - Report title:
 -
 -
- Create GitHub issue on failure**
- Advanced settings

Root: Freestyle project > 빌드 유발

- 어떤 때 빌드를 할 지 Trigger를 설정.
- 여기서는 Github에 push할 때

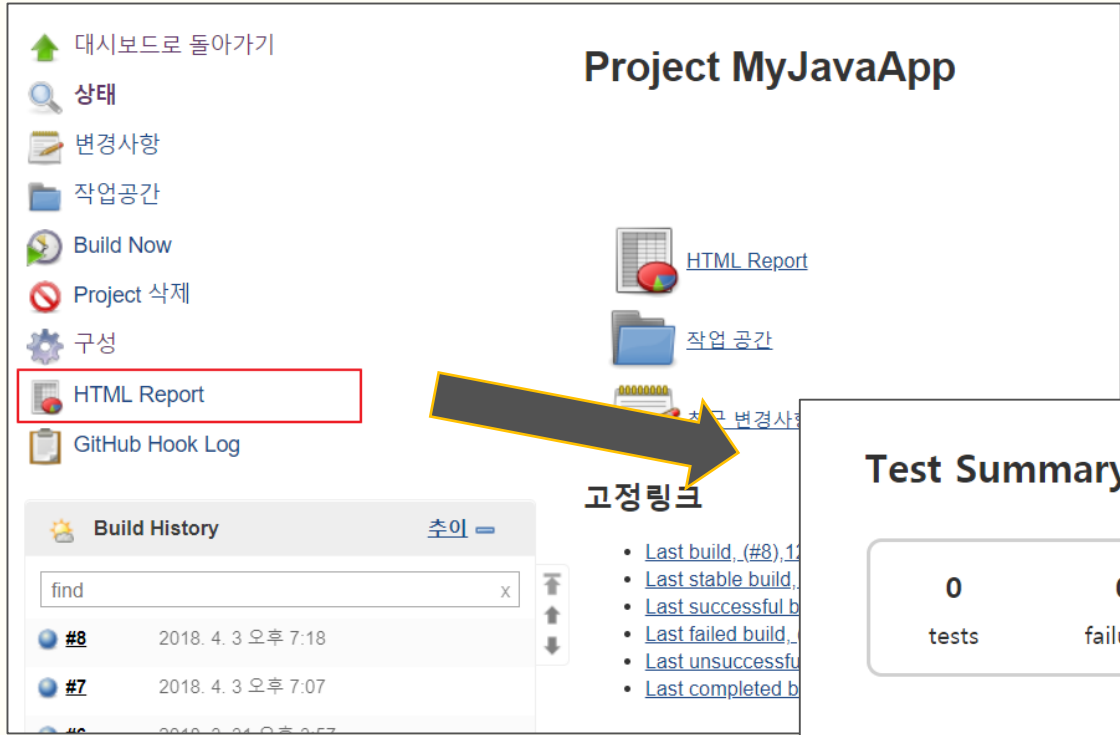
Root: Freestyle project > Build

- 각 프로젝트에 맞는 Build 도구를 설정.
- Gradle을 사용할 경우, 고급설정으로 Task, 빌드 파일 등을 지정 가능.

Root: Freestyle project > 빌드 후 조치

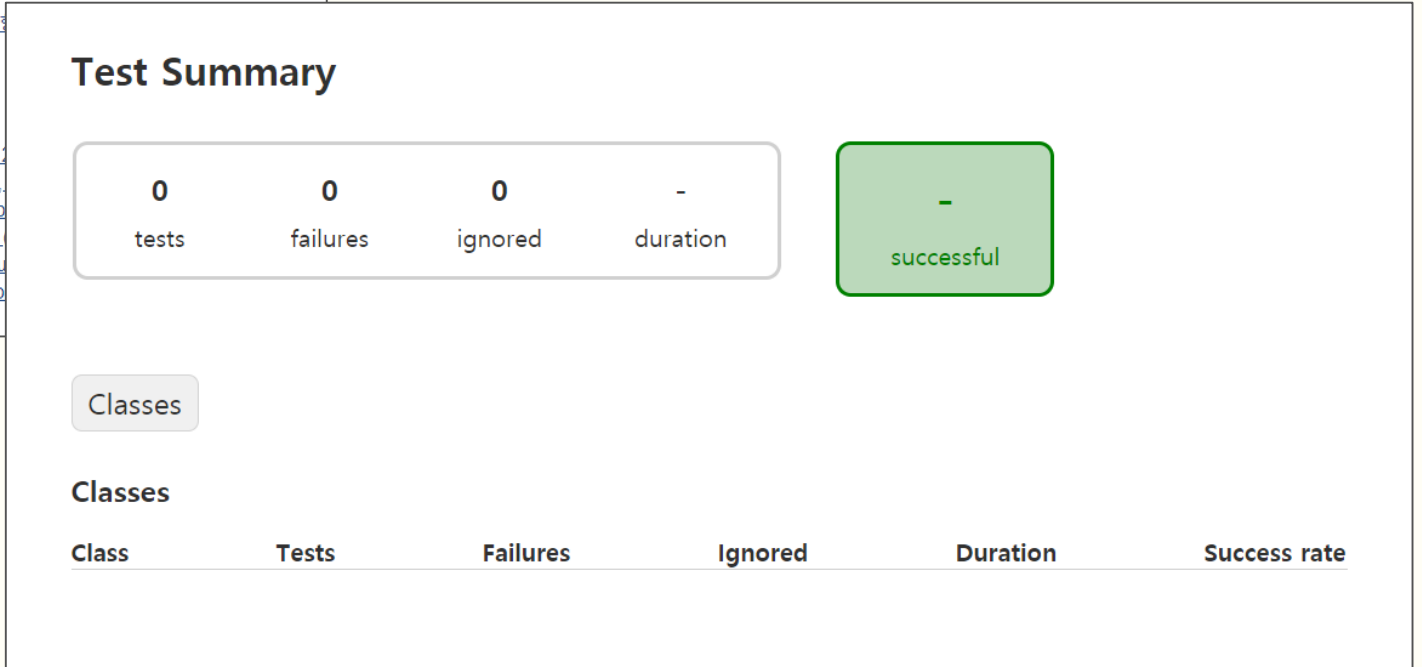
- 빌드 fail시 Github Issue를 등록하는 기능.
- Junit을 통해 단위 테스트한 결과를 보기 위해 설정.

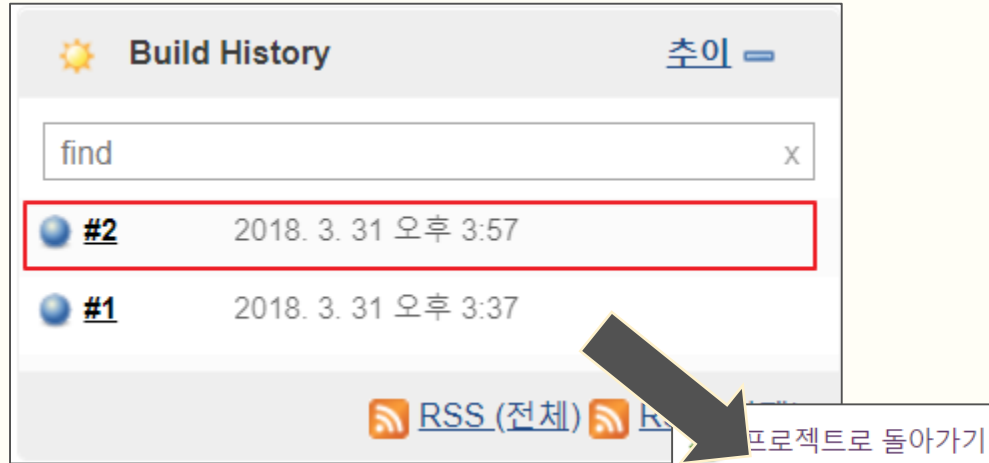
3. 새로운 프로젝트 생성_Freestyle project



#Root: 프로젝트 > HTML Reporter

- 단위테스트(Junit) 결과를 한 눈에 파악 가능.





#Root: 메인화면 > 대시보드 > 프로젝트 선택

- 각 프로젝트의 빌드 히스토리 확인 가능.
- 각 빌드 히스토리 클릭 시 빌드 시작, 변경 내용 확인 가능.
- [Console Output] 클릭 시 상세 정보 확인 가능(실패이유).

프로젝트로 돌아가기

-  상태
-  바뀐점
-  Console Output
-  빌드 정보 수정
-  이 빌드를 삭제
-  Polling Log
-  Git Build Data
-  No Tags
-  이전 빌드

빌드 #2 (2018. 3. 31 오후 3:57:09)


 Changes

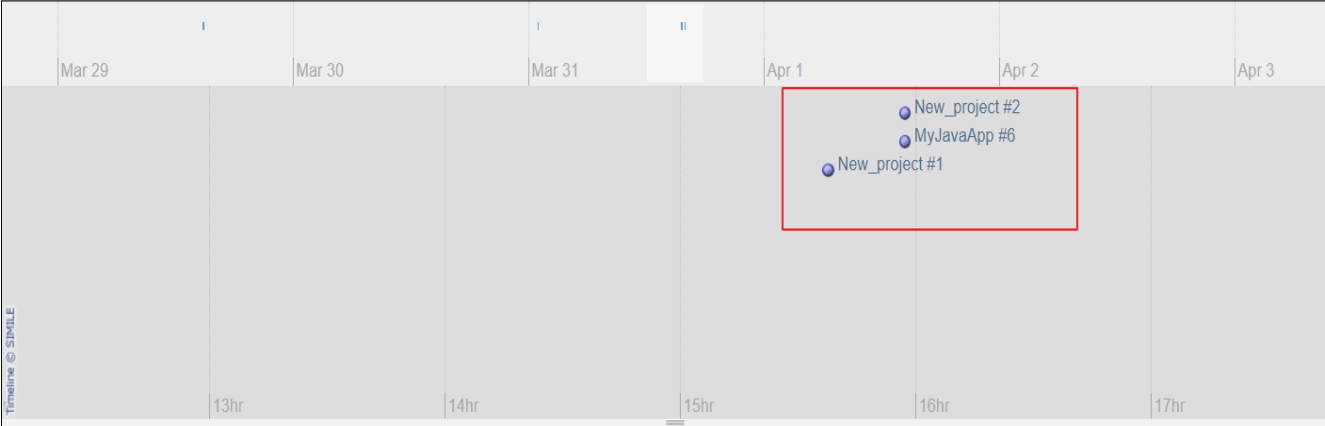
1. done ([commit: faaec6e](#)) ([detail / githubweb](#))
2. Korean to number ([commit: f6da749](#)) ([detail / githubweb](#))

 Started by [GitHub push by 3ssarah](#)











 Revision: f6da749a63eb39ae76361df2232539266ccfcde5

- refs/remotes/origin/master

 **Jenkins의 빌드 기록**



XML 문서로 변환하기 (Recurse in subfolders)

빌드	경과시간 ↑	상태
 New_project #2	1 min 34 sec	stable 
 MyJavaApp #6	1 min 34 sec	stable 
 New_project #1	21 min	stable 
 MyJavaApp #5	14 hr	back to normal 
 MyJavaApp #4	15 hr	broken for a long time 

Root: 메인화면 > 빌드 기록

- 시간 순으로 전체적인 빌드 기록 확인.
- 목록의 [빌드]를 클릭하면 해당 프로젝트로 이동.

5. Question & Answer



Thank You

