

IntelliJ와 GitHub 연동

Git이란?

- 형상 관리 도구 중 하나로 버전 관리 시스템이라고도 불린다.
- Git은 소프트웨어를 개발하는 기업의 핵심 자산인 소스코드를 효과적으로 관리할 수 있게 해주는 오픈 소스이다.
- 기존의 SVN을 쓰던 개발 조직들은 점차 Git으로 형상관리를 진행하고 있다.
- Git과 SVN의 차이점
 - Git은 여러 개발PC와 저장소를 분산해서 저장.
 - SVN은 중앙서버에 소스코드와 히스토리를 저장
 - SVN의 경우 중앙서버에 예기치 못한 에러가 발생되면 업데이트된 소스코드가 날라가거나, 형상관리를 진행하기 위한 커밋, 푸쉬등의 동작을 진행하지 못할 수 있다.
 - Git의 경우 개발PC 즉, 로컬에 저장소를 분산하여 저장함으로 원격저장소에 문제가 발생하여도 로컬에서 최신의 소스코드를 들고 있기 때문에 단지 새롭게 커밋을 진행하면 문제를 중앙서버 방식인 SVN보다 비교적 쉽게 해결가능하다.
- Git을 사용하여 형상관리를 하는 대표적인 무료 소프트웨어로는 GitHub와 GitLab이 존재한다.
- Git에 대한 좀 더 자세한 설명을 알고 싶으면 아래의 링크를 타고 들어가 참조하도록 하자.
Link1 >> <https://blog.naver.com/dldudcks1779/221992321884>
Link2 >> <https://goddaehee.tistory.com/91>

실습1) IntelliJ와 GitHub 연동

※ 이 실습은 해당 실습을 수행하는 분들이 GitHub의 ID를 소유하고 있다는 전제하에 진행됩니다.

1. GitHub에 현재 진행 중인 프로젝트를 공유하기 (원격 저장소 생성하기)

1.1. IntelliJ의 Actions검색창 띄우기

: 윈도우 단축키 >> [Ctrl + Shift + A]

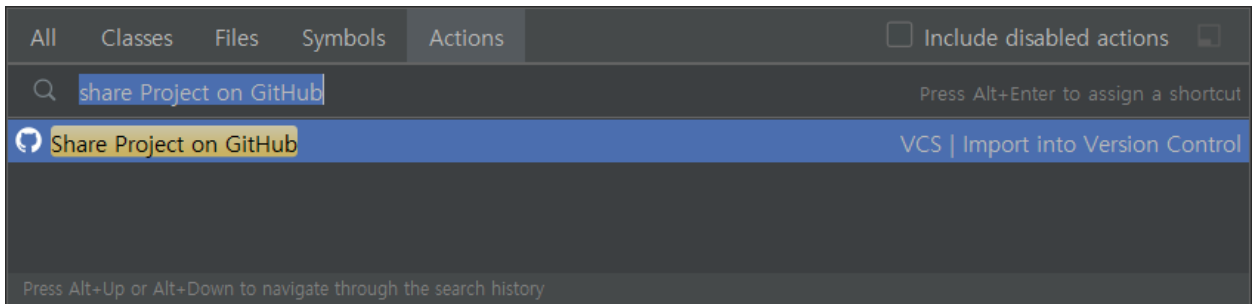
: 맥 단축키 >> [Cmd + Shift + A]



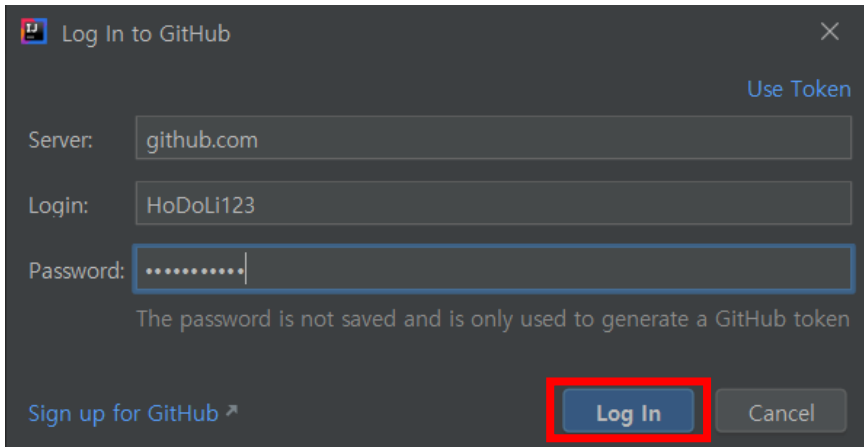
(단축키를 누를 시 IntelliJ에서는 위 사진과 같은 Actions 검색창이 활성화 된다.)

1.2. Actions검색창에 GitHub Action검색

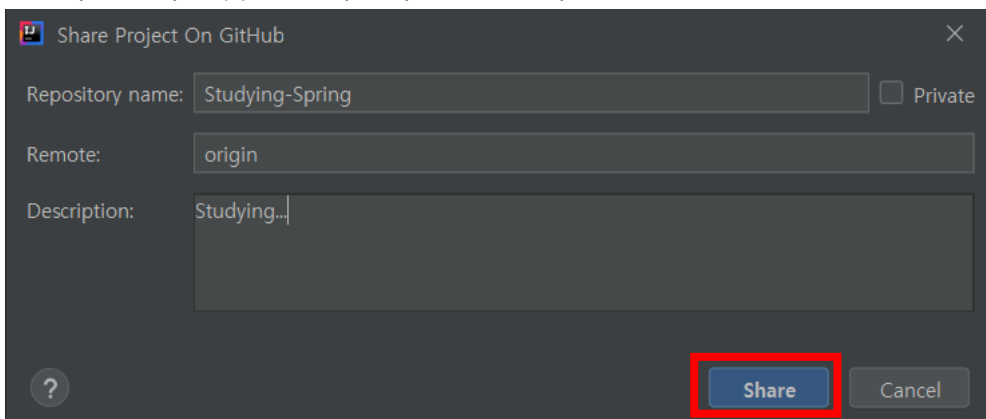
: 입력 >> share project on Github >> Enter



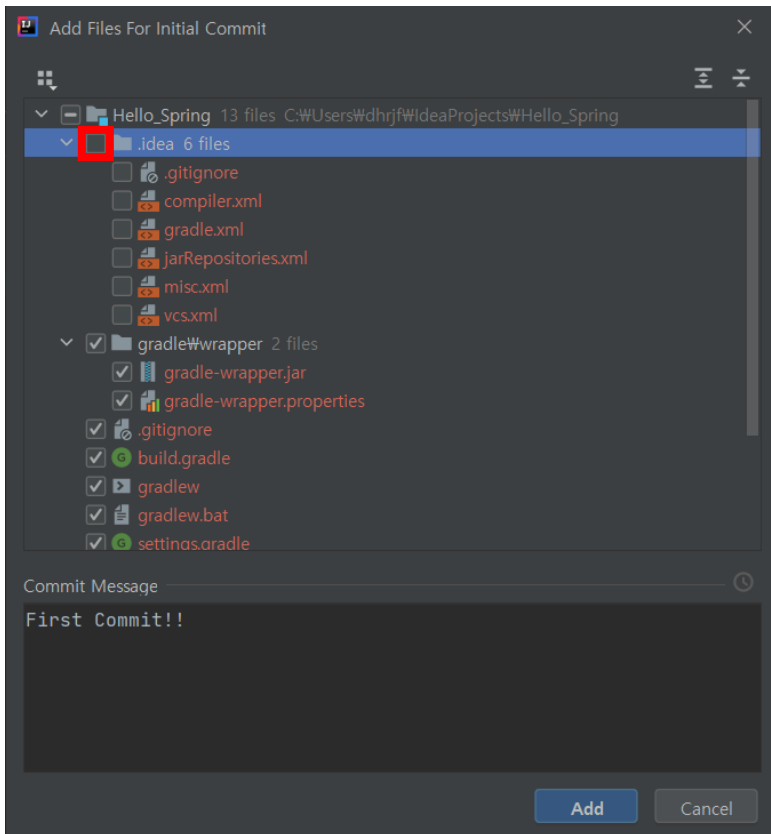
1.3. GitHub의 ID와 Password를 입력 후 Log In 클릭



1.4. 저장소 이름 및 설명 작성 후 Share 클릭

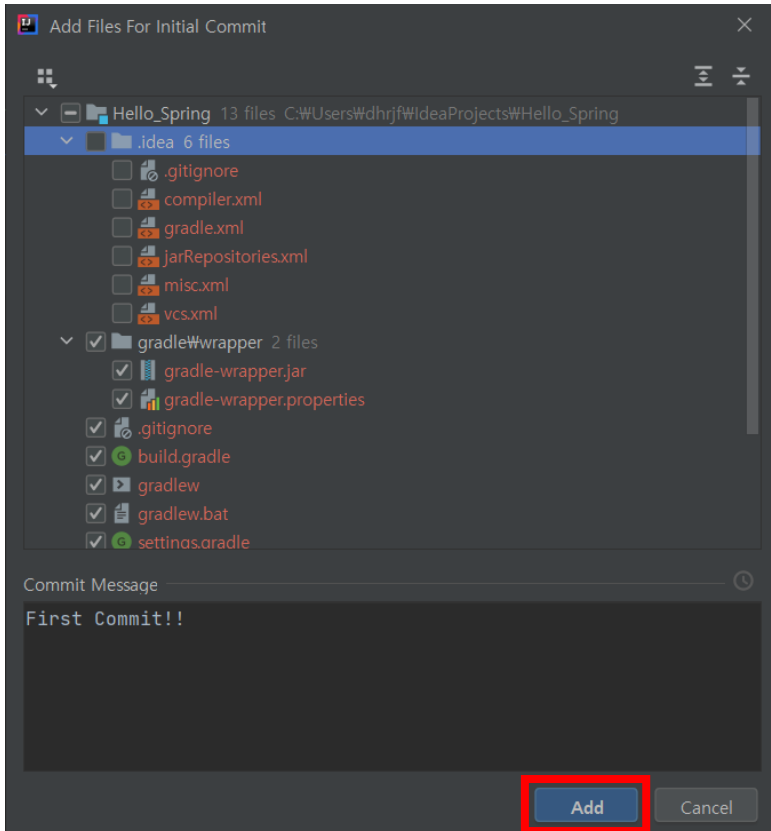


1.5. 커밋할 대상에서 .idea 디렉터리(폴더)를 커밋 대상에서 제외하기 : .idea에 체크되어 있는 체크박스를 해제




(.idea 디렉터리는 IntelliJ에서 프로젝트를 생성 시 자동으로 생성되는 폴더임으로 GitHub에 올리기에는 불필요 하다.)

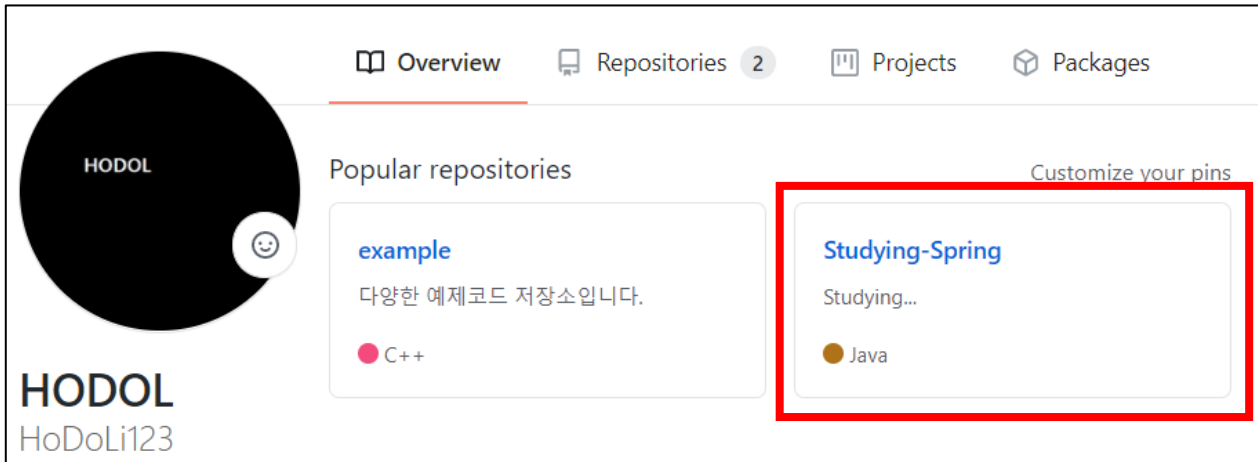
1.6. .idea를 커밋 대상에서 제외하였다면 Add를 클릭하여 GitHub의 저장소와 연동



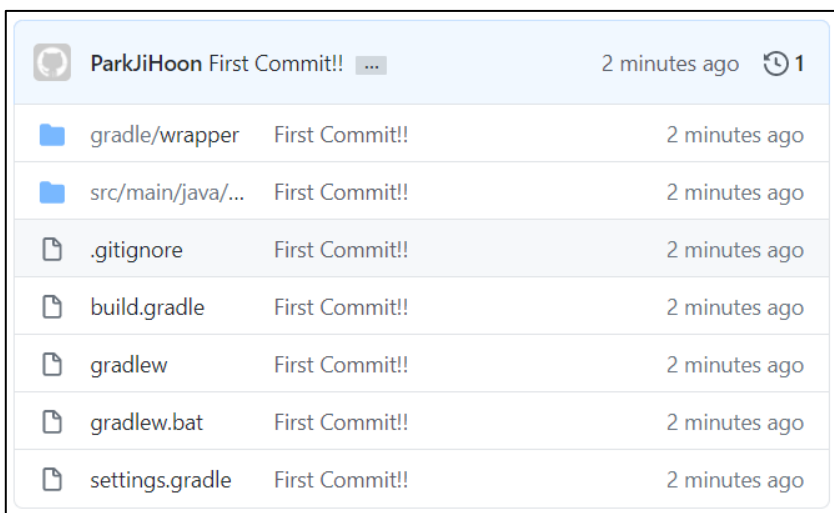
1.7. GitHub와 연동이 올바르게 진행되었다면 윈도우10같은 경우에는 아래와 같은 알림이 뜬다.

 Successfully shared project on GitHub
Studying-Spring

1.8. GitHub 웹페이지에서 올바르게 저장소가 생성되었는지 확인



(올바르게 Studying-Spring 저장소가 생성되었다.)



(올바르게 현재까지 진행된 Spring 프로젝트가 생성되었다.)

1.9. .idea 디렉토리를 모든 커밋 대상에서 제외시키기

: 이후에도 여러 번 커밋을 수행할 것임으로 .idea를 자동적으로 커밋 대상에서 제외하도록 해보자.

1.9.1. IntelliJ에 ignore plugin 부착

: git은 .gitignore파일을 통해 특정 파일 및 디렉토리를 관리대상에서 제외시킨다.

1.9.2. IntelliJ의 Actions 검색창 띄우기

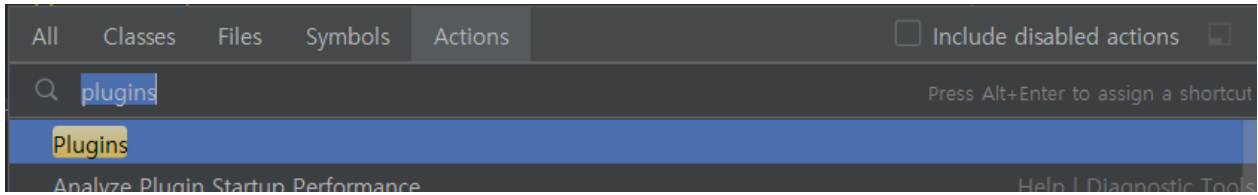
: 윈도우 >> [Ctrl + Shift + A]

: 맥 >> [Cmd + Shift + A]



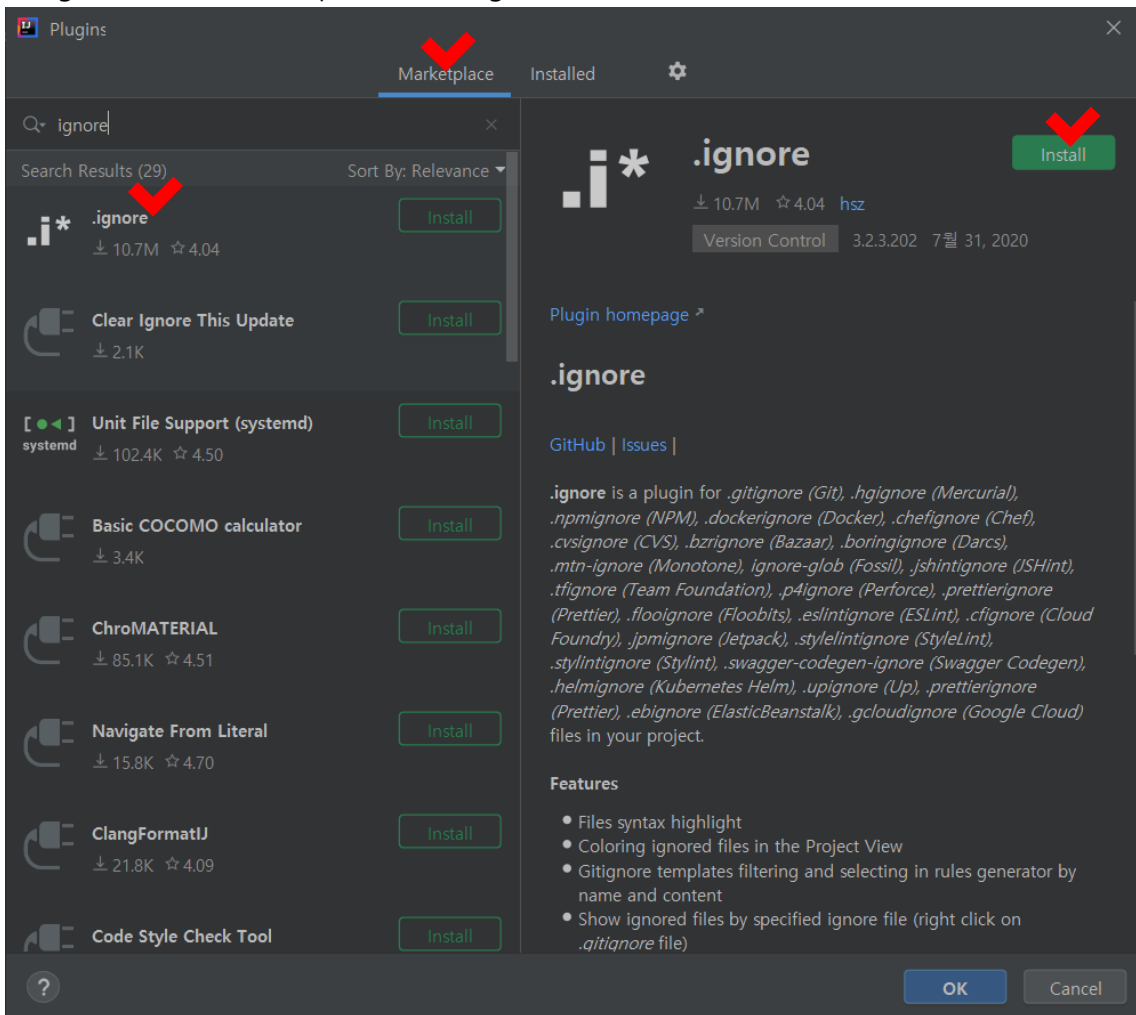
1.9.3. plugins 검색

: 입력 >> plugins >> Enter



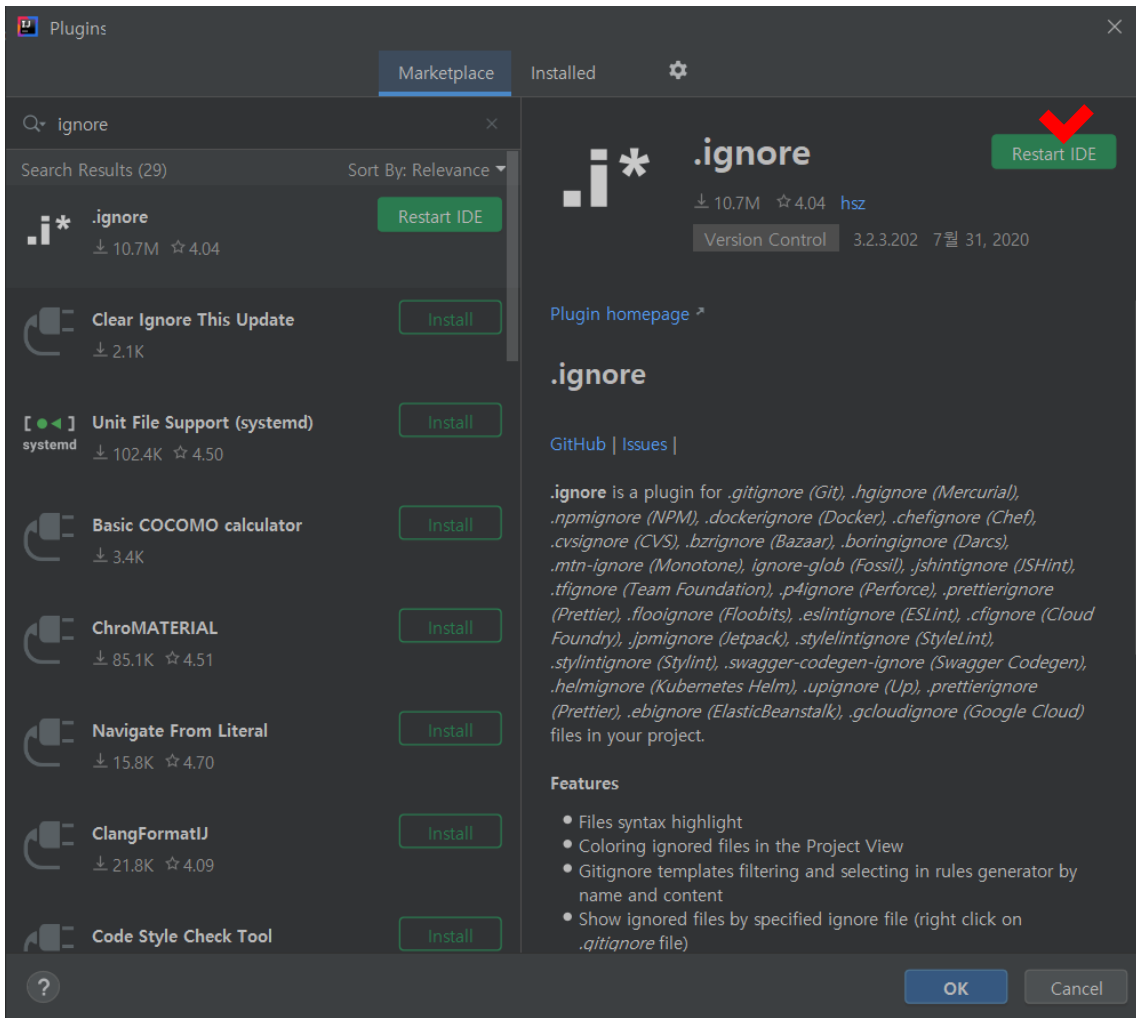
1.9.4. ignore plugin 설치

: Plugins >> 상단 Marketplace 탭 >> ignore 검색 >> Install 클릭



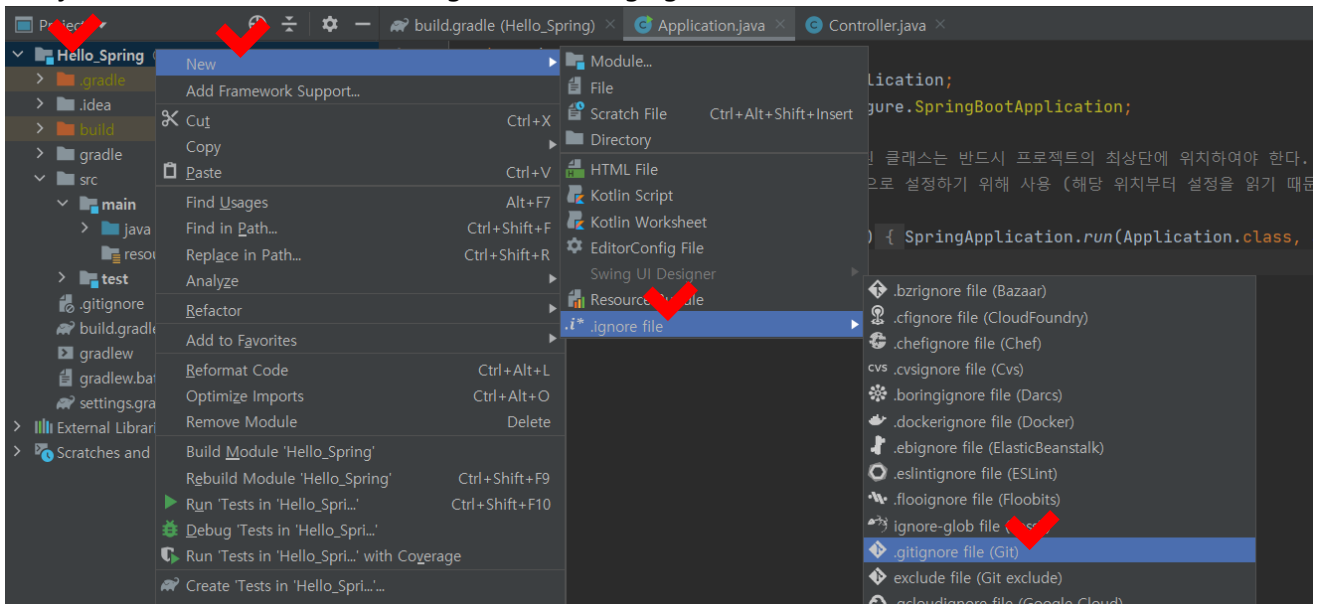
1.9.5. 설치완료 시 IntelliJ 재시작

: ignore설치가 완료되면 Install 버튼이 Restart 버튼으로 자동 변경된다.

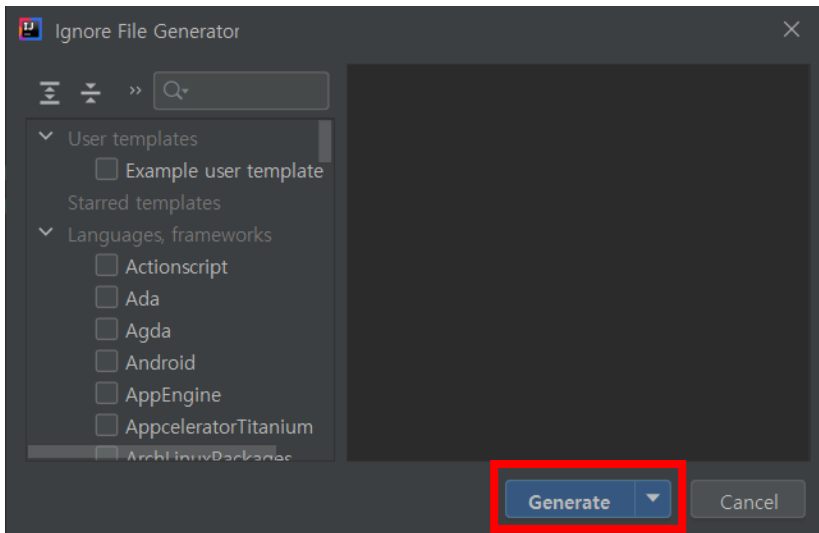


1.9.6. .gitignore 파일 생성

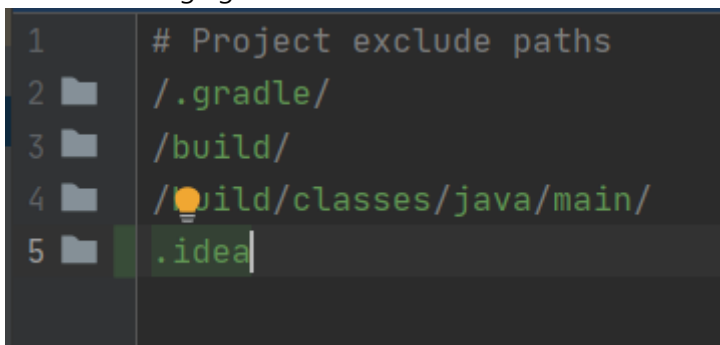
: Project 오른쪽 클릭 >> New >> .ignore file >> .gitignore file (Git) 왼쪽 클릭



1.9.7. Generate 클릭



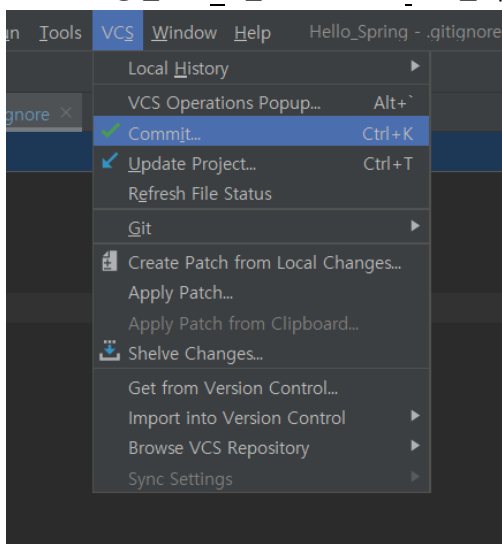
1.9.8. 생성된 .gitignore 파일의 기존내용에 .idea추가



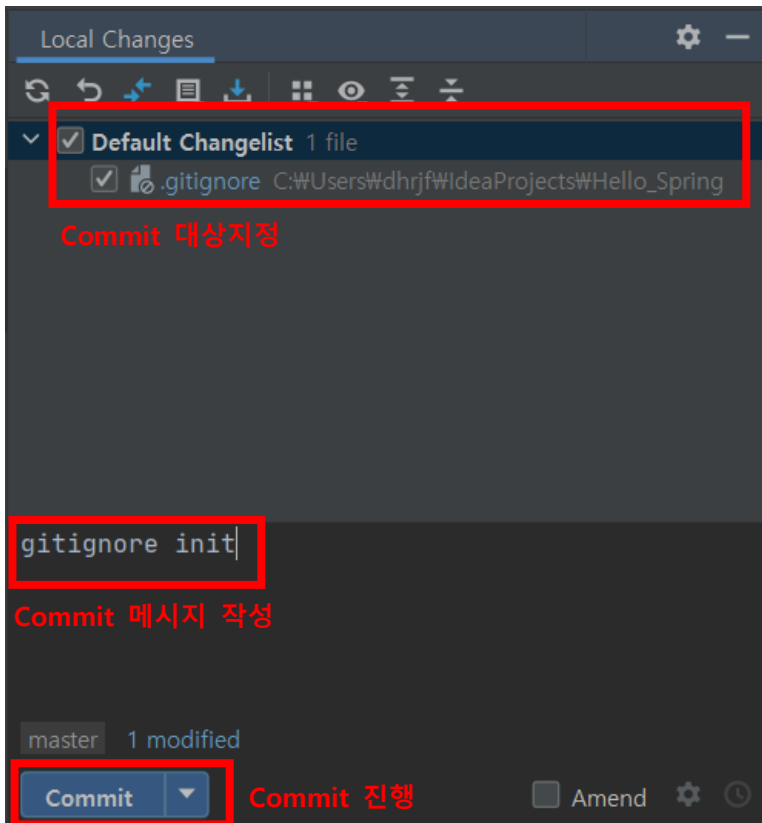
(2,3,4번째 줄은 기존에 작성되어 있었으며 마지막 5번째 줄이 실습에서 추가된 내용이다.)
 (!! 이제 git은 .idea 디렉토리를 관리대상에서 자동 제외시킨다.)

1.9.9. .gitignore를 새로운 커밋에 추가

: IntelliJ 상단 VCS 탭 >> Commit... 왼쪽 클릭

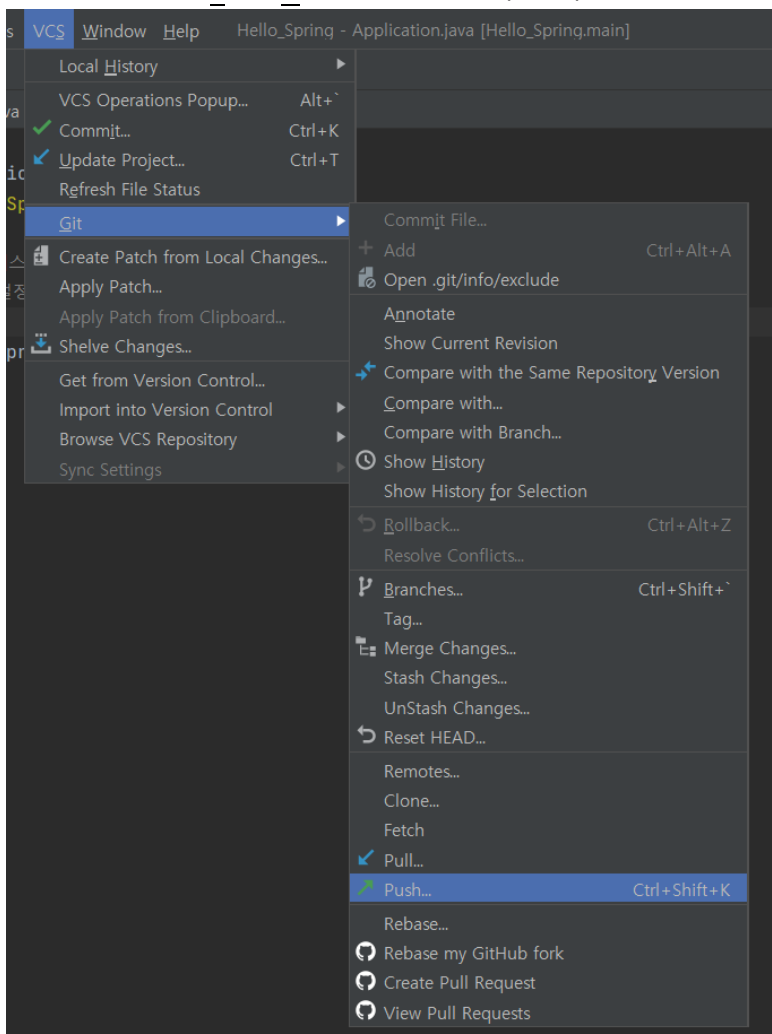


1.9.10. 커밋 파일 선택 및 커밋 메시지 작성 후 Commit 진행

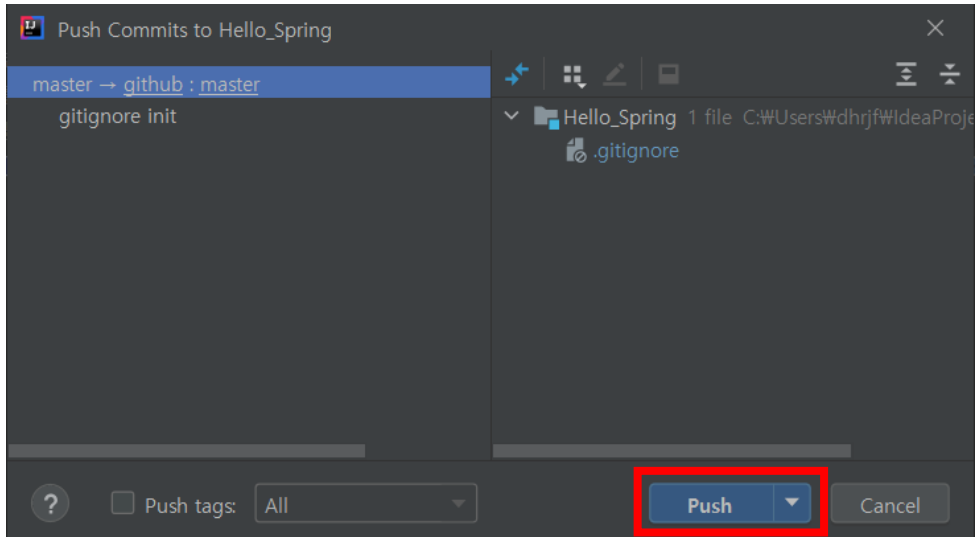


1.9.11. 커밋된 대상 Push 진행

: IntelliJ 상단 VCS >> Git >> Push... 왼쪽 클릭



1.9.12. Push 수행



1.9.13. Push가 올바르게 진행되었는지 확인

	ParkJiHoon gitignore init ...	4 minutes ago	🕒 2
📁	gradle/wrapper	First Commit!!	34 minutes ago
📁	src/main/java/...	First Commit!!	34 minutes ago
📄	.gitignore	gitignore init	4 minutes ago
📄	build.gradle	First Commit!!	34 minutes ago
📄	gradlew	First Commit!!	34 minutes ago
📄	gradlew.bat	First Commit!!	34 minutes ago
📄	settings.gradle	First Commit!!	34 minutes ago

(.gitignore 파일의 Commit 메시지가 처음 시도된 Commit 메시지들과 다른 내용임을 확인할 수 있다.)

참고문헌

- 이동욱 (2019). 스프링 부트와 AWS로 혼자 구현하는 웹 서비스. 39-49.