

(Git) Submodule 사용하기

이 페이지는 SVN의 external link와 유사한 기능인 submodule 사용법을 정리한다.

- [개요](#)
- [Submodule 생성](#)
 - Submodule 추가
 - 적용 상태 확인
 - 로컬 저장소에 반영
 - Bitbucket 저장소에 반영
- [Submodule Clone 하기](#)
 - Clone 후 소스코드 가져오기
 - Clone 할 때 소스코드 가져오기
- [Submodule 갱신 내용 가져오기](#)
 - Fetch 후 merge
 - Update 명령 사용
- [Submodule 브랜치 변경하기](#)
 - Branch 변경하기
 - Commit 지점 변경하기
- [기타 시나리오](#)
 - 로컬 변경 취소하기

개요

Submodule은 다른 repository를 참조할 수 있는 기능으로 하나의 저장소가 여러가지 저장소의 소스코드를 참조할 수 있다.

[blocked URL](#)

Submodule 생성

Submodule 추가

예) 저장소 repo1이 repo2 (<https://bitbucket.curvc.com/project/repo2.git>)를 참조

```
$ git submodule add https://bitbucket.curvc.com/project/repo2.git
```

기본 연결:

- branch: master
- commit: HEAD

적용 상태 확인

.gitmodules 파일과 repo2 파일이 생성된다.

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   .gitmodules
    new file:   repo2
```

```
[submodule "repo2"]
  path = repo2
  url = https://bitbucket.curvc.com/project/repo2.git
```

로컬 저장소에 반영

```
$ git commit -am "submodule repo2 "
[master fb9093c] added repo2 module
2 files changed, 4 insertions(+)
create mode 100644 .gitmodules
create mode 160000 repo2
```

Bitbucket 저장소에 반영

```
$ git push origin master
```

Submodule Clone 하기

Submodule은 clone 후 추가 명령을 통해 다운로드 받거나 옵션을 추가해서 clone 해야 한다.

Clone 후 소스코드 가져오기

```
$ git submodule init
$ git submodule update
```

Clone 할 때 소스코드 가져오기

Submodule의 소스코드도 함께 가져오기 위해 --recurse-submodules 옵션을 추가한다.

```
$ git clone --recurse-submodules https://bitbucket.curvc.com/project/repo1.git
```

Submodule 갱신 내용 가져오기

몇 가지 방법으로 upstream (Bitbucket) repo2의 변경된 내용을 로컬 저장소에 가져올 수 있다.

Fetch 후 merge

```
$ cd repo2
$ git fetch
$ git merge origin/master
```

Update 명령 사용

```
$ git submodule update --remote [submodule ]
```

Submodule 브랜치 변경하기

연결된 submodule의 commit 지점을 변경하거나 branch를 변경할 수 있다.

Branch 변경하기

예) submodule repo2를 stable branch로 변경

```
$ git config -f .gitmodules submodule.repo2.branch stable
$ git submodule update --remote
```

Commit 지점 변경하기

Submodule로 연결된 upstream의 commit를 반영하여 연결점을 갱신할 수 있다.

```
# Upstream
$ git submodule update --remote
# repo1 submodule
$ git commit -am "repo2  "
# Bitbucket repo1
$ git push origin master
```

기타 시나리오

로컬 변경 취소하기

```
$ git submodule update [--recurse] --init
```