

**Wind Lidar 개발 환경 구성**



2017.02

본 문서는 한진정보통신에서 문서의 저작권을 소유하고 있으며, 제공 목적 이외의 용도로 사용할 수 없습니다. 본 문서와 관련하여 전체 또는 부분적으로 복제하거나 복사 또는 한진정보통신의 승인 없이는 어떤 사람에게 전송 될 수 없습니다

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 개 정 현 황 | | | | | |
| 개정번호 | 일 자 | 개 정 사 유 | 작 성 | 검 토 | 승 인 |
| V 1.0 | 2017.02.08 |  | 정민섭 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

목 차

[1. FTP 설치 ４](#_Toc478032662)

[1.1 FileZilla Server 설치 ４](#_Toc478032663)

[1.2 FileZilla Server 실행 ６](#_Toc478032664)

[1.2.1 FTP Server 설정 ７](#_Toc478032665)

[1.2.2 FTP 사용자 등록 ８](#_Toc478032666)

[1.2.3 FTP 사용자 홈 디렉토리 설정 ９](#_Toc478032667)

[2. MariaDB 10.1 Series 설치 １０](#_Toc478032668)

[2.1 MariaDB Download １０](#_Toc478032669)

[2.2 MariaDB 설치 １０](#_Toc478032670)

[2.2.1 다운로드 받은 파일 실행 １０](#_Toc478032671)

[2.2.2 설치 완료 테스트 ２１](#_Toc478032672)

[2.3 MariaDB 설정 ２１](#_Toc478032673)

[2.3.1 Maria Db UTF-8 설정 ２２](#_Toc478032674)

[2.3.2 사용자 데이터베이스 생성 ２３](#_Toc478032675)

[2.3.3 사용자 계정 생성 ２４](#_Toc478032676)

[2.3.1 사용자 DB 권한 설정 ２４](#_Toc478032677)

[2.3.2 사용자 DB 접속 테스트 ２５](#_Toc478032678)

[3. MariaDB Table 생성 ２５](#_Toc478032679)

[3.1 MariaDB UTF-8 설정 ２５](#_Toc478032680)

[3.1.1 데이터베이스 문자셋 확인 ２５](#_Toc478032681)

[3.1.2 UTF-8 문자셋 설정 ２６](#_Toc478032682)

[3.2 테이블 생성 ２７](#_Toc478032683)

[4. MariaDB 개발 환경 구성 ２８](#_Toc478032684)

[4.1 Mysql-connector-net 설치 ２８](#_Toc478032685)

[4.2 Visual Studio에서 Mysql 참조하기 ２９](#_Toc478032686)

[5. INI 파일 구성 ３０](#_Toc478032687)

[5.1 WindLidar 서버 ３０](#_Toc478032688)

[5.2 WindLidar 클라이언트 ３２](#_Toc478032689)

[6. Visual Studio 2013에서 GitHub 연결 ３３](#_Toc478032690)

[6.1 GitHub 레파지토리 URL 확인 ３４](#_Toc478032691)

[6.2 Visual Studio에서 GitHub 연결 ３４](#_Toc478032692)

[6.3 Source Tree tool for GitHub ４０](#_Toc478032693)

[7. 웹 개발 환경 구성 ４０](#_Toc478032694)

[7.1 프로젝트 디렉토리 정의 ４０](#_Toc478032695)

[7.2 웹 개발 환경 버전 ４１](#_Toc478032696)

[7.2.1 Eclipse 다운로드 받기 ４１](#_Toc478032697)

[7.2.2 JDK 다운로드 하기 ４１](#_Toc478032698)

[7.2.3 Tomcat 다운로드 ４５](#_Toc478032699)

[7.2.4 Maven 다운로드 ４６](#_Toc478032700)

[7.3 웹 개발 환경 설정 ４７](#_Toc478032701)

[7.3.1 Tomcat Installation Identification ４７](#_Toc478032702)

[7.3.2 Maven Setup ５１](#_Toc478032703)

[7.3.3 STS (Spring Tool Suite) 설치 ５４](#_Toc478032704)

[7.3.4 Maven Integration Installation ５８](#_Toc478032705)

[7.3.5 Environment Variable Setup ５９](#_Toc478032706)

[8. Web Development with spring framework ６１](#_Toc478032707)

[8.1 Spring framework 개요 ６１](#_Toc478032708)

[8.2 Eclipse에서 Spring Project 생성 ６１](#_Toc478032709)

[8.2.1 Spring Project 생성 ６１](#_Toc478032710)

[8.2.2 Maven Dependencies ６４](#_Toc478032711)

[8.2.3 POM.xml ６５](#_Toc478032712)

[8.2.4 UTF-8 설정 ７２](#_Toc478032713)

[8.2.5 생성된 프로젝트 실행 ７２](#_Toc478032714)

[8.3 Mybatis 연동하기 ７６](#_Toc478032715)

[8.3.1 Mybatis개요 ７６](#_Toc478032716)

[8.3.2 Mybatis 라이브러리 ７７](#_Toc478032717)

[8.3.3 MyBatis와 데이터베이스 연결 설정 ７７](#_Toc478032718)

[8.3.4 MyBatis와 Spring 연결 ７９](#_Toc478032719)

[8.3.5 DAO 작성 ８０](#_Toc478032720)

[8.4 Application Development with MyBatis ８２](#_Toc478032721)

[8.4.1 List 데이터 가져오기 ８２](#_Toc478032722)

[8.4.2 HandlerMethodArgumentResolver 적용 ９０](#_Toc478032723)

[8.4.3 사용자 관리 프로그램 개발 ９６](#_Toc478032724)

# Git 사용하기

GitHub에 올라간 프로젝트 폴더를 GitHub에서 삭제를 할 수가 없다. 삭제를 위해서는 SoruceTree나 기타 Git Console에서 삭제를 수행해야 한다.

## SourceTree 사용하기

윈도우 환경에서 Git을 자동으로 업로드 할 수 있다. Source Tree는 버전관리 및 업로드에 최적화 되어 있다.

## Git 설치하기

### Git 설치

Git을 사용하려면 설치를 해야 하는데 다양한 방법으로 Git을 설치할 수 있지만 두가지 방법이 가장 일반적이다. 하나는 소스코드로부터 컴파일해서 설치하는 방법이고 다른 하나는 각 운영체게(혹은 플랫폼)의 패키지를 사용하여 설치하는 방법이다.

#### 소스코드로 설치하기

소스코드로 설치하면 Git의 가장 최시버전을 설치 할 수 있기 때문에 컴파일하여 설치할 시간이 있으면 소스코드로 Git을 설치하는 것이 좋다. Git은 계속 UI를 개선하고 있기 때문에 최신 버전을 사용하면 좋은 기능을 빨리 사용할 수 있다. 리눅스 패키지는 보통 최신 버전이 아니고 예전 버전이다. 그래서 Backport를 사용하거나 소스코드로 설치하는 것도 좋은 대안이다.

Git을 설치하려면 아래와 같은 라이브러리들이 필요하다. Git은 curl, zlib, opens니, expat, libiconv를 필요로 한다. 예를 들어 Fedora처럼 yum을 사용하는 시스템이나 apg-get이 있는 데비안류 시스템이면 아래 명령어를 실행하여 의존 패키지를 설치할 수 있다.

|  |
| --- |
| **$yum install curl-devel expat-devel gettext-devel \**  **Openssl-devel zlib-devel**  **$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \**  **Libz-dev libssl-dev** |

필요한 라이브러리를 모두 설치하고 다음 단계를 진행한다. Git 웹사이트에서 최시 스냅샷을 가져온다.

|  |
| --- |
| http://git-scm.com/download |

그리고 컴파 일하고 설치한다

|  |
| --- |
| $tar –zxf git-1.7.2.2.tar.gz  $cd git-1.7.2.2  $make prefix=/usr/local all  $sudo make prefix=/usr/local install |

설치한 다음부터는 Git을 사용하여 Git 소스코드를 수정할 수 있다.

|  |
| --- |
| $git clone git://git.kernel.org/pub/scm/git/git.git |

#### 리눅스에 설치

리눅스에서 패키지로 Git을 설치할 때에는 보통 각 배포판에서 사용하는 패키지 관리도구를 사용하여 설치한다. Fedora에서는 아래와 같이 한다.

|  |
| --- |
| # yum install git-core |

Ubuntu같은 데비안류 배포판에선느 apt-get을 사용한다.

#### 윈도우에 설치

윈도우에서도 Git를 쉽게 설치할 수 있다. 구글 코드 페이지에서 msysGit 인스톨러를 내려받고 실행하면 된다.

|  |
| --- |
| **http://mysgit.github.com/** |

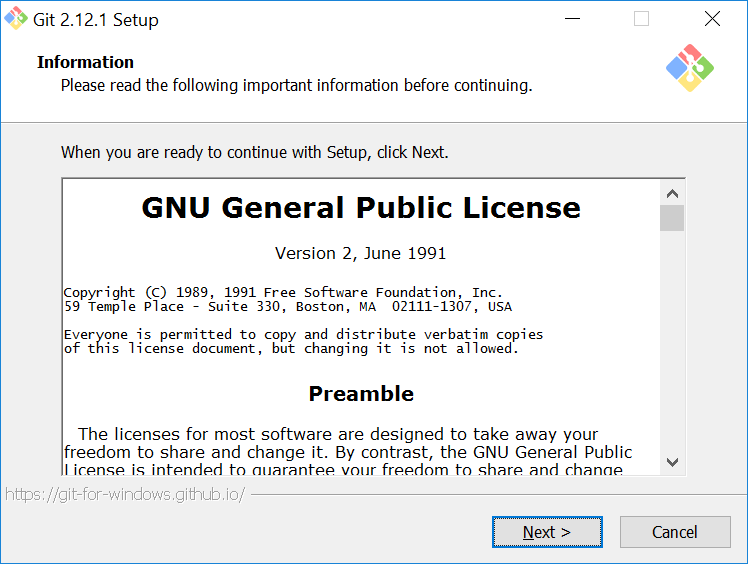
설치가 완료되면 CLI 프로그램과 GUI 프로그램을 둘 다 사용할 수 있다 CLI 프로그램에서는 SSH 클라이언트가 포함되어 있기 때문에 유용한다.

Windows 사용자가 Git의 다양한 명령어를 사용하려면 유닉스 스타일의 msysGit쉘을 사용하는 것이 좋다. 어쩔 수 없이 Windows에 포함된 기본 쉘(Command Prompt, 명령 프롬프트)을 꼭 써야 하면 공백이 포함된 파라미터를 Git 명령어에 넘길 때 작은 따옴표(‘’) 대신 큰 따옴표(“”)를 사용해야 한다. 파라미터 끝에 ^기호가 있을 때도 큰 따옴표로 파라미터를 감싸야 한다. Windows 쉘에서 ^기호는 다음줄로 명령어가 이어짐을 나타낸다.

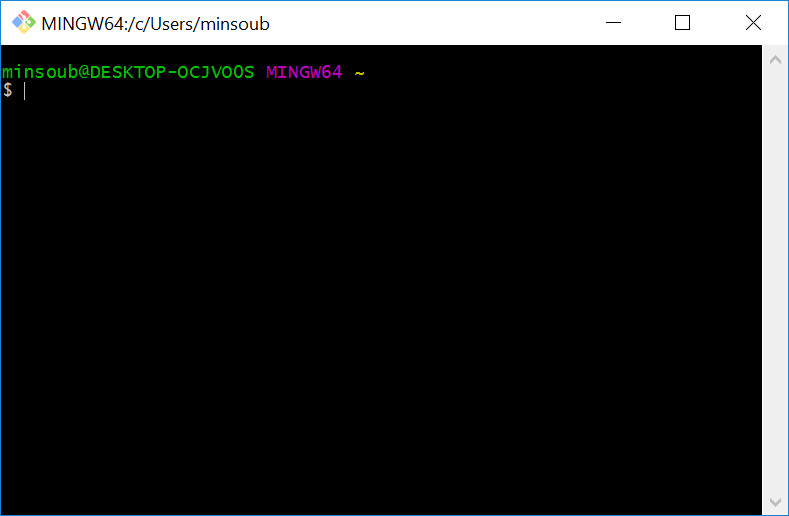


다운로드 버튼을 클릭하여 git를 다운로드 받는다.

설치버전은 Git-2.12.1-64-bit.exe 파일을 다운로드 받아서 설치한다.



Git command 창 실행모습



## Git Bash 사용하기

### git config

Git을 설치했을 때 가장 먼저해야 하는 것은 GitHub 사이트에 등록된 user name과 e-mail 주소이다. 이것은 git commit 시마다 사용하는 정보로서 초기에 설정해 놓아야 한다.

$ git config -–global user.name “Joung minsoub”

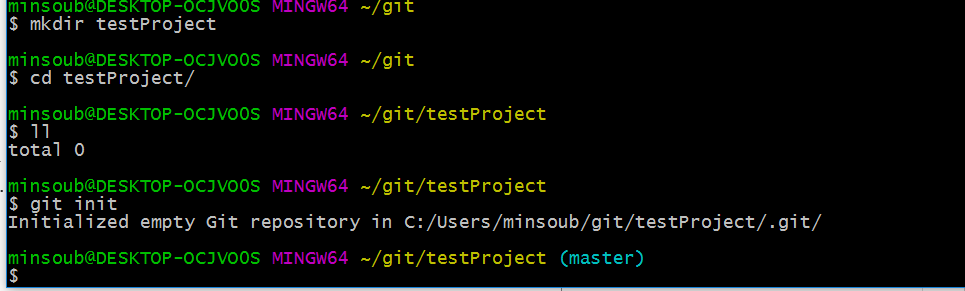
$git config –-global user.email [minsoub@gmail.com](mailto:minsoub@gmail.com)



### git init

우선 로컬저장소로 사용할 폴더를 하나 생성한다. 윈도우 탐색기로 D 드라이버에다 폴더를 생성했다. ( D:\WindLidarRepo )

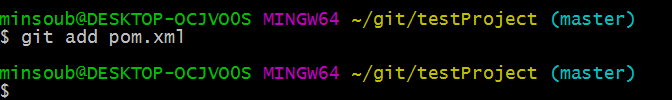
그리고 나서 Git Bash 실행창에서 로컬저장소로 이동한 후 **git init** 명령어를 실행한다.



### git add

GitHub에 올릴 테스트 아무 파일 하나를 로컬저장소에 붙여 넣는다. testProject 폴더에 아무 파일을 하나 복사해서 넣는다. 로컬저장소에 파일을 복사한다고 해서 git 저장소에 바로 연결(tracked)되는 것은 아니다. SVN에서도 새로운 파일을 인식하기 위해서 add 하는 절차가 있듯이 git도 add관련 명령어가 있다. 그리고 Git에서는 **준비영역(Staging area)**으로 add한다라고 설명하고 있다.

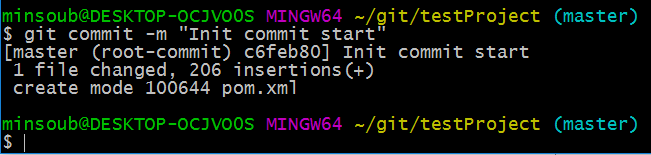
$git add pom.xml



### git commit

git add 명령어도 새로운 파일을 인식했다면 commit으로 버전관리를 시작하게 된다. 하지만 git commit은 원격저장소인 GitHub에 반영되는 것이 아닌, 로컬저장소에만 저장되는 것이다.

$git commit –m “Init commit”

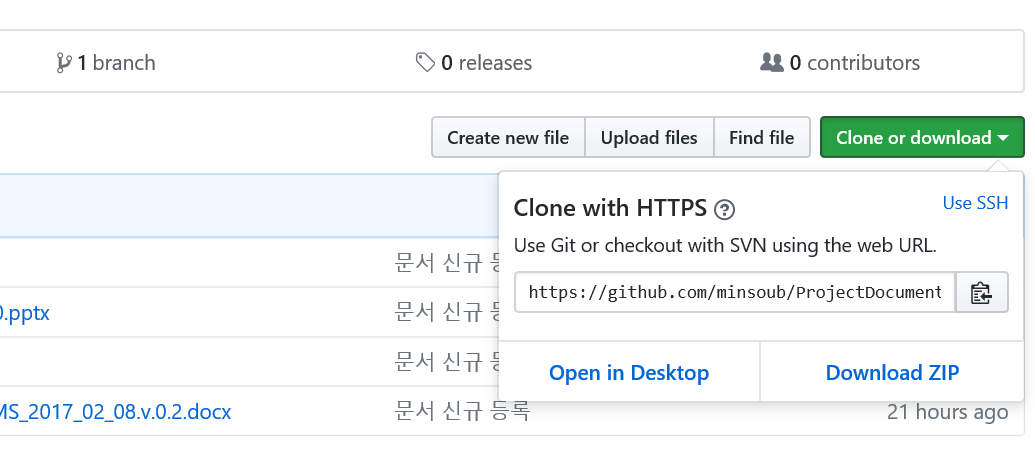


### git remote

remote 레파지토리(repository)를 셋팅한다. 여기서는 https로 등록한다.

$git remote add origin <https://github.com/minsoub/testProject.git>

위의 주소는 GitHub 사이트에 가서 자신이 만든 repository 화면에 들어가면 아래와 같이 URL 정보가 나온다.



위의 명령을 수행한 후 서버에 파일이 올려져 있는지 확인한다.