

개발 환경 설정 가이드 - MacOS

- 1 Goal
- 2 Reader
- 3 Prerequisites
- 4 Steps
 - 4.1 Basic Set-up
 - 4.1.1 (Optional) Mouse Tracking Speed
 - 4.1.2 (Optional) Magic Mouse Right Click
 - 4.1.3 brew 패키지 매니저 설치 - Homebrew
 - 4.1.4 브라우저 설치 - Google Chrome
 - 4.1.5 Shell 설치 - ZSH
 - 4.1.6 VIM 설치
 - 4.1.7 Shell Framework 설치 - oh-my-zsh
 - 4.1.8 VIM 설정 마무리
 - 4.1.9 (Optional) Rectangle 설치
 - 4.1.10 JetBrains ToolBox 설치
 - 4.1.11 JetBrains 라이선스 적용
 - 4.1.12 git config
 - 4.2 Back-end Set-up
 - 4.2.1 SDKMAN
 - 4.2.2 Java 설치
 - 4.2.3 Gradle 설치
 - 4.2.4 IntelliJ IDEA Ultimate 설치
 - 4.2.5 IntelliJ IDEA Ultimate 설정
 - 4.3 Front-end Set-up
 - 4.3.1 NVM 설치
 - 4.3.2 Node.js 설치
 - 4.3.3 Yarn Berry를 위한 Corepack 설치
 - 4.3.4 WebStorm 설치
 - 4.3.5 WebStorm 설정

1 Goal

AM 프로젝트 수행을 위한 개발 환경을 MacOS에서 구성한다.

2 Reader

프로젝트에서 역할 하기 위한 MacOS 환경을 구성하는 개발자

3 Prerequisites

- [2. 개발 환경 설정 가이드](#)

4 Steps

4.1 Basic Set-up

4.1.1 (Optional) Mouse Tracking Speed

맥OS의 마우스 커서 이동 속도는 타 OS에 비해 느린 편입니다. 필요한 만큼 조절하길 권장드립니다.

1. Spotlight 실행 (Cmd + Space)
2. Preference 입력
3. Mouse 설정
4. 마우스 커서 이동 속도 조절

4.1.2 (Optional) Magic Mouse Right Click

기본적으로 매직 마우스는 **Ctrl + 클릭**을 통해 윈도우의 우클릭에 해당하는 기능을 사용할 수 있습니다.

다른 마우스처럼 우클릭을 사용하고 싶다면 아래와 같이 설정합니다.

1. Spotlight 실행 (Cmd + Space)
2. Preference 입력
3. Mouse 설정
4. 우측 클릭 옵션 설정

4.1.3 brew 패키지 매니저 설치 - Homebrew

맥OS 사용자들이 최근 가장 많이 사용하는 패키지 관리자입니다. 각종 패키지를 설치하고 삭제하고 업데이트하는 데에 필요한 수고를 많이 줄여줍니다

1. [macOS 용 패키지 관리자 — Homebrew](#) 방문
2. Homebrew 설치하기 항목의 스크립트를 복사
 - a. [blocked URL](#)
 - i. 드래그 후 복사 (Cmd + C)
 - ii. 또는 스크립트 우측의 복사 버튼 클릭
 - iii. `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
3. Spotlight 실행 (Cmd + Space)
4. Terminal 실행 (MacOS를 한글 Locale로 사용하고 있다면 '터미널' 실행)
5. 붙여넣기 (Cmd + V) 후 실행 (Enter)
6. 아래 명령어를 실행하여 정상적으로 동작하는지 확인

```
$ brew
```

4.1.4 브라우저 설치 - Google Chrome

- 직접 설치
 - <https://www.google.com/intl/ko/chrome/> 방문
- Homebrew를 통해 설치
 - 터미널에서 아래 명령어를 통해 설치

```
$ brew install --cask google-chrome
```

4.1.5 Shell 설치 - ZSH

oh-my-zsh 사용을 위해 설치합니다.

1. 터미널 실행
2. `$ brew install zsh`

4.1.6 VIM 설치

맥OS에 내장된 버전은 모든 기능을 포함하지 않습니다.

또한 터미널 환경에서 Git을 활용할 때 이용하게 될 수 있습니다.

1. Terminal 실행
2. `$ brew install vim`

4.1.7 Shell Framework 설치 - oh-my-zsh

커뮤니티 주도로 발전 중인 ZSH 설정을 관리하기 위한 프레임워크입니다.

기본적으로 다양한 기능을 내장하고 있습니다.

1. <https://github.com/ohmyzsh/ohmyzsh> 방문
2. Basic Installation 항목에서 원하는 커맨드를 드래그 후 복사

Basic Installation

Oh My Zsh is installed by running one of the following commands in your terminal. You can install this via the command-line with either `curl`, `wget` or another similar tool.

Method	Command
curl	<pre>sh -c "\$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"</pre>
wget	<pre>sh -c "\$(wget -O- https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"</pre>
fetch	<pre>sh -c "\$(fetch -o - https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"</pre>

3. Terminal 실행
4. 붙여넣기 후 실행
5. `$ vim ~/.zshrc`
 - a. `plugins=()` 를 찾는다.
 - b. `plugins=(git)` 처럼 git plugin 추가

4.1.8 VIM 설정 마무리

1. Terminal 종료 후 재실행
2. \$ vi ~/.zshrc
3. 최상단에 아래 내용 추가 후 저장 및 종료

```
# VIM
export VISUAL="vim"
export EDITOR="$VISUAL"
alias vi="vim"
```

(Optional) 연습하고 싶으신 경우 아래대로 수행

1. Terminal 실행
2. \$ wget https://github.com/vim/vim/blob/master/runtime/tutor/tutor.ko.utf-8
3. \$ vi tutor.ko.utf-8

4.1.9 (Optional)Rectangle 설치

맥OS에서 창을 정리하고 유용하게 사용하기 위한 유틸리티입니다.

1. Terminal 실행
2. \$ brew install --cask rectangle
3. Spotlight 실행
4. rectangle 실행
5. 안내를 따라 진행하여 rectangle 설정

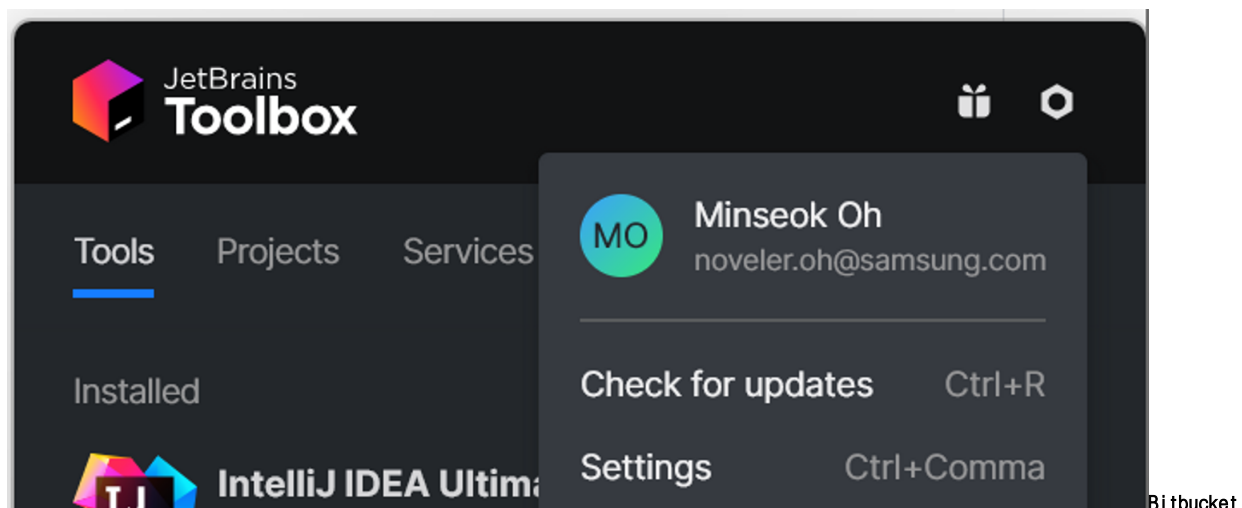
4.1.10 JetBrains ToolBox 설치

1. <https://www.jetbrains.com/ko-kr/toolbox-app/> 방문
2. 안내에 따라 설치

4.1.11 JetBrains 라이선스 적용

1. JetBrains ToolBox 실행
2. Settings 실행
3. Login

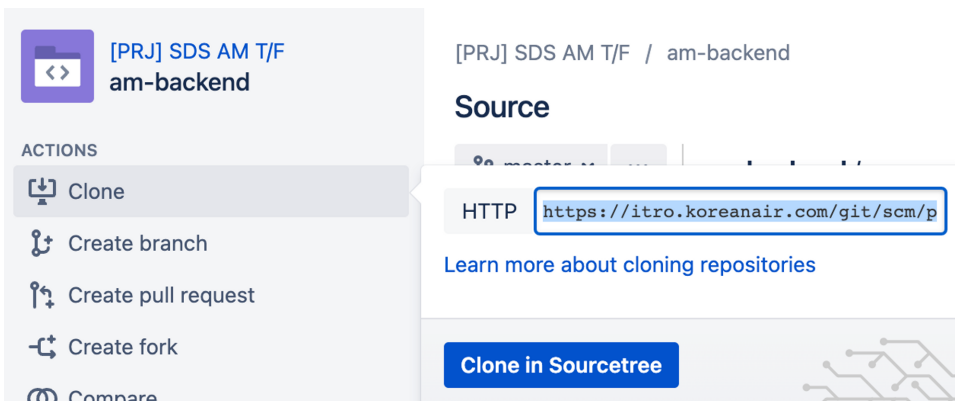
4.1.12 git config



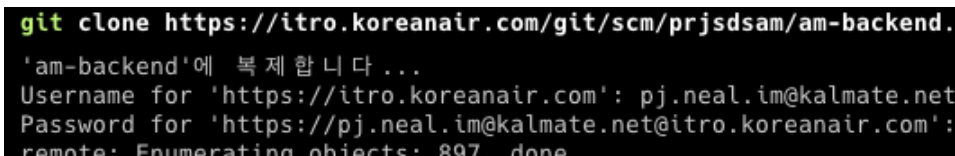
대한항공 내 AM 프로젝트의 경우 소스코드는 <https://itro.koreanair.com/git/projects/PRJSDSAM> 에서 관리합니다.

프로젝트 Git Clone 방법

1. Bitbucket 접근 및 로그인. (google 연동)
2. Terminal 실행
3. mkdir ~/workspace; cd workspace
4. clone할 repository url 획득



5. \$ git clone {4에서 copy한 url}
6. Username과 Password에는 Google 인증정보를 입력



7. (옵션) username, email설정

여기에서는 예시로 전역으로 username과 email을 설정한다. 필요하다면 필요 repository 내에서 --global 을 제외한다.

```
#
git config --global user.name "xx"
git config --global user.email "pj.xxxx@kalmate.net"
```

만약 Google 로그인만 되지만 Bitbucket clone 시 인증이 실패하는 경우,

높은 확률로 기존 인증이 꼬인 상태이므로 기존 credential을 초기화할 필요가 있습니다.

<http://manage.kalmate.net/pwd/resetpassword.aspx> 에서 비밀번호 변경을 하면 AD와 Google 계정이 동시에 반영된다

Mac에서의 credential은 keychain Access를 참조한다. 아래 itro.koreanair.com정보를 검색해서 지운다.

[blocked URL](#)

Bitbucket 로그인 인증 캐시는 기본 30초로 Google 비밀번호 변경 후 최소 30초가 지난 뒤에 계정 인증 정보를 변경해야한다.

4.2 Back-end Set-up

4.2.1 SDKMAN

1. <https://sdkman.io/install> 방문
2. 설치 커맨드를 확인하고 복사
- [blocked URL](#)
3. Terminal 실행
4. 붙여넣기 후 실행
5. Terminal 재기동
6. 아래 명령어를 실행하여 동작 확인

```
$ sdk version
```

4.2.2 Java 설치

1. Terminal 실행
2. \$ sdk install java 21.0.3-amzn
3. 아래 명령어를 실행하여 동작 확인

```
$ java -version
```

4.2.3 Gradle 설치

일반적으로는 필요하지 않으나 필요할 수도 있기에 설치해 두면 좋습니다.

1. Terminal 실행
2. \$ sdk install gradle
3. 아래 명령어를 실행하여 동작 확인

```
$ gradle -V
```

4.2.4 IntelliJ IDEA Ultimate 설치

1. JetBrains ToolBox 실행
2. Available 항목 중 IntelliJ IDEA Ultimate를 Install

4.2.5 IntelliJ IDEA Ultimate 설정

1. IntelliJ IDEA Ultimate 실행
2. Open

[blocked URL](#)

3. ~/workspace/servers 선택
4. Project View에서 Project Root 선택

[blocked URL](#)

5. Project Structure 실행 (Command + ; or Ctrl + Alt + Shift + S)
 - a. Platform Settings – SDKs
 - 클릭
 - ii. sdkman으로 설치한 Java 선택
 - b. Project Settings – Project
 - i. SDK – 위에서 추가한 SDK 선택
6. Preferences 실행 (Cmd + ,)
 - a. Editor – General
 - i. Remove trailing spaces on Modified lines 체크
 - ii. Remove trailing blank lines at the end of saved files 체크
 - iii. Ensure every saved file ends with a line break 체크
 - b. Editor – General – Appearance
 - i. Show whitespaces 체크
 - c. Build, Execution, Deployment – Build Tools – Gradle
 - i. Gradle JVM: Project SDK 선택
 - d. Tools – Shared Indexes
 - i. (Optional) Wait for shared indexes 선택
 - ii. (Optional) JDKs: Download automatically 선택
 - iii. (Optional) Maven Libraries: Download automatically 선택

4.3 Front-end Set-up

4.3.1 NVM 설치

1. <https://github.com/nvm-sh/nvm> 방문
2. Install & Update Script 항목의 스크립트 복사

[blocked URL](#)

3. Terminal 실행
4. 붙여넣기 후 엔터
5. Terminal 재기동
6. 아래 명령어를 통하여 동작 확인

```
$ nvm --help
```

4.3.2 Node.js 설치

1. Terminal 실행
2. \$ nvm install v20
3. 아래 명령어를 통해 동작 확인

```
$ node --version
```

4.3.3 Yarn Berry를 위한 Corepack 설치

1. Terminal 실행
2. \$ corepack enable

4.3.4 WebStorm 설치

1. JetBrains ToolBox 실행
2. Available 항목 중 WebStorm를 Install

4.3.5 WebStorm 설정

1. WebStorm 실행
2. Open

[blocked URL](#)

3. ~/workspaces/ceed 선택
4. Preferences 실행 (Cmd + ,)
 - a. Editor – General
 - i. Remove trailing spaces on Modified lines 체크
 - ii. Remove trailing blank lines at the end of saved files 체크
 - iii. Ensure every saved file ends with a line break 체크
 - b. Editor – General – Appearance
 - i. Show whitespaces 체크
 - c. Languages & Frameworks – Node.js
 - i. Package Manager

