

01 개발 환경 구성

Mac 개발환경 구성

JDK

Java(JDK) 설치와 버전 관리

개발환경에 JDK를 설치하는 3가지 방법

1. IDE(IntelliJ)의 JDK 다운로드 기능
2. JDK 배포판 웹사이트에서 다운로드 받아서 설치
3. JDK 버전을 관리하는 도구를 이용해서 설치
 - **SDKMAN**: Java와 다양한 도구의 버전 관리
 - **jabba**: Java 버전 관리
 - **Asdf**: 다양한 프로그래밍 언어와 도구 버전 관리
 - **jvms**: 윈도우용 JDK 버전 관리자

SDKMAN

<https://sdkman.io/>

설치

```
curl -s "https://get.sdkman.io" | bash
```

터미널 사용 초기화

```
source "$HOME/.sdkman/bin/sdkman-init.sh"
```

설치 확인

```
sdk version
```

설치 가능한 JDK 목록 조회

```
sdk list java
```

Vendor, Version, Identifier를 확인하고 설치할 JDK의 Identifier를 복사

```
=====
Available Java Versions for macOS ARM 64bit
=====
Vendor          | Use | Version      | Dist   | Status      | Identifier
=====
```

Corretto		23.0.2		amzn		23.0.2-amzn
		23.0.1		amzn		23.0.1-amzn
		21.0.6		amzn		21.0.6-amzn
		21.0.5		amzn		21.0.5-amzn
		17.0.14		amzn		17.0.14-amzn
		17.0.13		amzn		17.0.13-amzn
		11.0.26		amzn		11.0.26-amzn
		11.0.25		amzn		11.0.25-amzn
		8.0.442		amzn		8.0.442-amzn
		8.0.432		amzn		8.0.432-amzn
Gluon		22.1.0.1.r17		gln		22.1.0.1.r17-gln
		22.1.0.1.r11		gln		22.1.0.1.r11-gln
GraalVM CE		23.0.2		graalce		23.0.2-graalce
		23.0.1		graalce		23.0.1-graalce
		21.0.2		graalce		21.0.2-graalce
		17.0.9		graalce		17.0.9-graalce
GraalVM Oracle		25.ea.7		graal		25.ea.7-graal
		25.ea.6		graal		25.ea.6-graal
		25.ea.5		graal		25.ea.5-graal

JDK 설치

```
sdk install java 21.0.6-librca
```

Tip

install java 뒤에서 Tab 키를 누르면 identifier를 가능한 범위까지 자동완성 해준다.
Tab을 두번 누르면 설치 가능한 목록을 보여준다.
화살표로 선택도 가능하다.

디폴트 JDK

- 터미널에 들어갔을 때 기본으로 적용되는 JDK(Java) 버전을 default로 지정해둘 수 있다.

```
sdk default java 21.0.5-librca
```

프로젝트별 JDK 버전 적용

SDKMAN을 이용해서 프로젝트에 따라 다른 JDK 버전을 적용하려면

1. JDK를 설치하고
2. 프로젝트 루트 폴더에서 sdk use로 사용할 JDK를 선택하고
3. sdk env init으로 초기화하고 선택한 JDK 정보를 저장
4. 이후에 sdk env 명으로 저장된 버전 정보에 맞는 JDK 활성화

설치된 JDK를 선택

```
sdk use java 21.0.5-librca
```

현재 선택된 JDK 버전 정보 저장

```
sdk env init
```

선택한 JDK 버전 정보는 `.sdkmanrc` 파일에 저장된다.
Git 리포지토리에 추가하면 개발팀과 버전을 공유할 수 있다.

기존에 저장된 버전의 JDK 활성화

```
sdk env
```

저장된 JDK 버전이 설치되어 있지 않으면 다음 명령으로 바로 설치할 수도 있다.

```
sdk env install
```

Tip

프로젝트 폴더에 들어갈 때 자동으로 저장된 JDK 버전이 활성화 되게 만들려면

`~/.sdkman/etc/config` 의 다음 줄을 수정

```
sdkman_auto_env=true
```

IntelliJ에서 JDK 설치와 선택

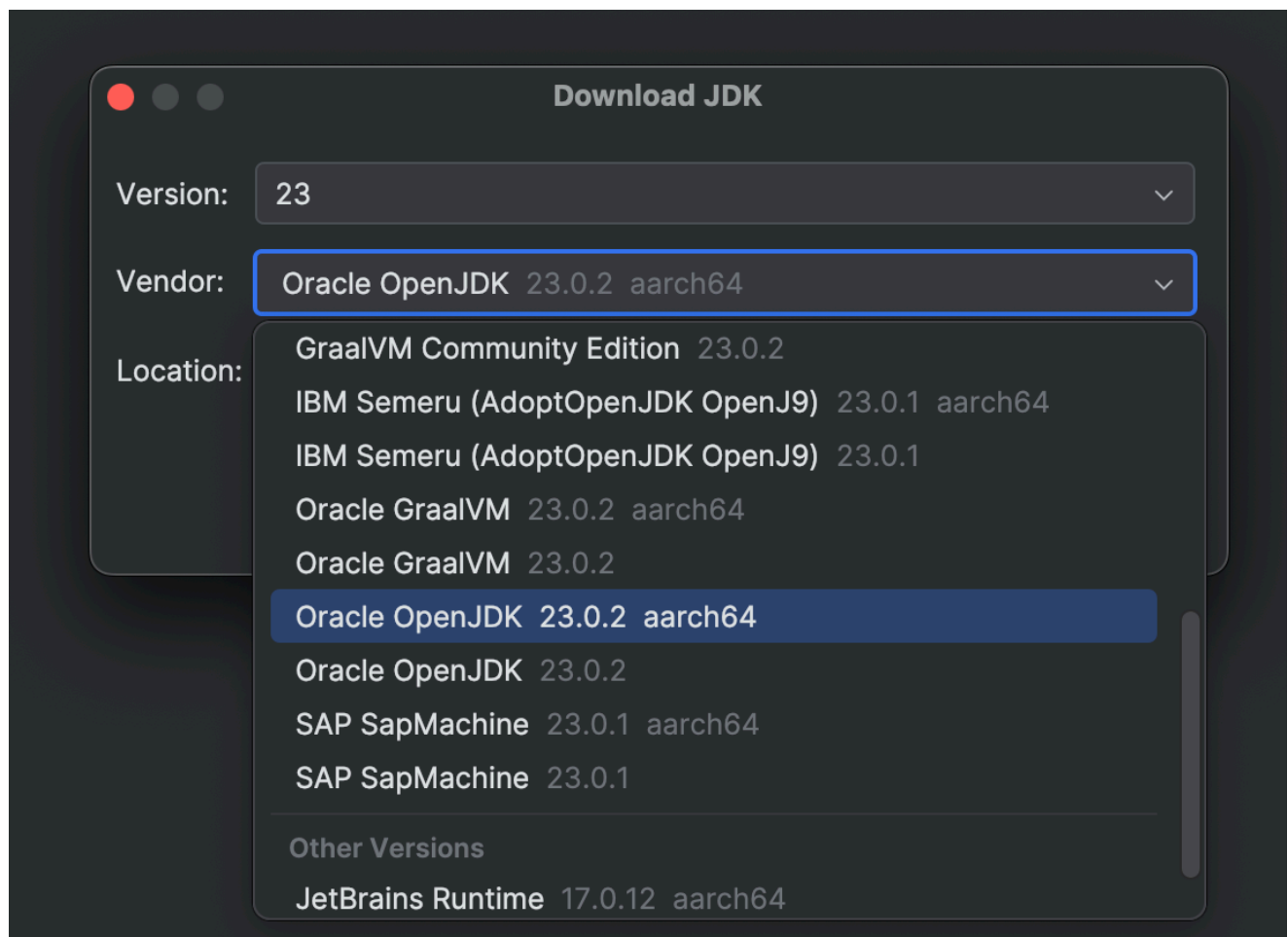
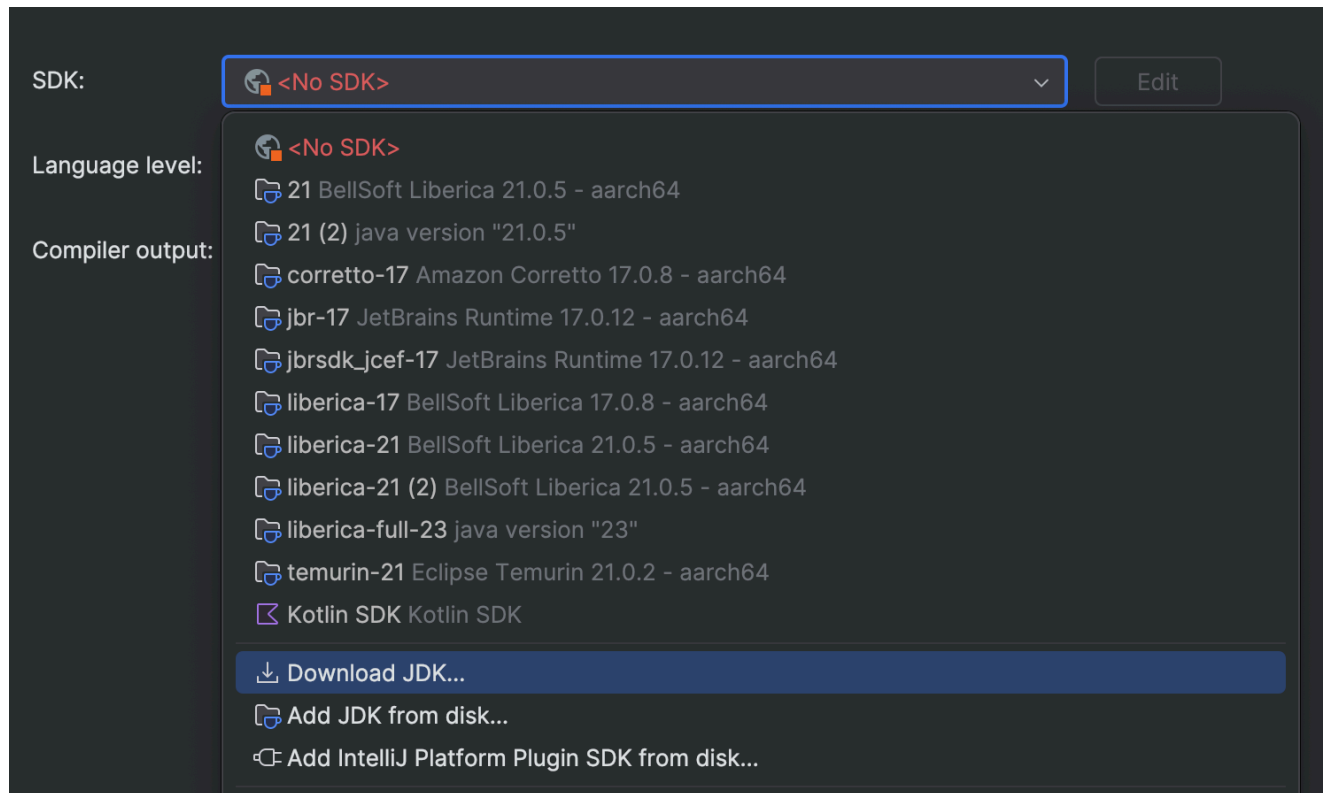
IntelliJ의 프로젝트에서 사용할 JDK는 다음 두 가지 방식으로 설치된 JDK 중에서 선택해서 사용할 수 있다.

- IntelliJ가 지원하는 다운로드 기능을 이용해서 설치된 JDK
- SDKMAN이나 다른 방식으로 설치된 JDK

터미널 환경에서 JDK 사용이 필요하지 않은 초기 개발에는 자체 다운로드 방식이 편리하다.

장기적으로는 프로젝트 JDK 벤더와 버전 정보를 함께 관리하고 설치도 지원되는 `sdkman`을 이용하는 방식으로 전환하

는 것이 좋다.



Note

SDKMAN으로 설치할 수 있는 여러 자바 툴이 있다.

대표적으로 Gradle을 시스템 레벨에 설치해서 `gradle` 명령으로 프로젝트를 초기화하는 등의 작업이 가능하다. 하지만 스프링을 이용해서 개발하는 경우 대부분 `Spring Initializr`를 이용해서 프로젝트 기본 구조와 파일을 생성하고, 이때 만들어진 프로젝트 파일에는 Gradle wrapper가 포함되어있어서 자체적으로 설정된 버전의 Gradle을 받아서 사용 가능하다. 따라서 별도로 Gradle을 설치하지 않고도 개발이 가능하다.

Gradle을 다양한 용도로 활용하거나 세밀한 설정을 해보고 싶다면 `sdkman`을 이용해서 설치해보는 것도 좋다.

Warning

애플 실리콘 칩을 이용하는 M 시리즈 맥인 경우 x86 빌드의 JDK가 아닌 aarch64로 빌드된 JDK를 받아서 설치하도록 한다. Rosetta가 설치된 경우 x86 아키텍처용으로 개발된 프로그램을 변환을 통해서 M 시리즈 맥에서 사용할 수 있게 해주지만, 일부 성능 차이가 있을 수 있으므로 ARM64 아키텍처(aarch64)로 빌드된 JDK를 사용하는 것이 권장된다.

IntelliJ에서 다운로드할 때 aarch64가 붙은 버전을 선택한다.

SDKMAN은 현재 OS에 따라서 선택해준다. 필요하면 확인을 해볼 수 있다.

현재 선택된 java의 빌드 아키텍처를 확인하고 싶으면 다음 명령을 이용한다.

```
cat `which java` | file -
```

HTTPIe

터미널에서 웹 요청을 보내고 받는 작업을 할 때 사용할 수 있는 도구가 여러가지 가있다.

기본적으로 설치되어있고 가장 많이 사용하는 `curl` 과 최근 인기를 얻고 있는 `HTTPIe` 가 있다.

HTTPIe는 복잡한 웹 요청도 좀 더 손쉽게 만들 수 있는 여러 기능을 제공한다.

<https://httpie.io/>

설치

```
brew install httpie
```

`http` 와 `https` 두 개의 명령이 설치된다.

설치후 테스트

```
https httpie.io/hello
```

사용방법

자세한 사용법은 공식 문서를 참고

<https://httpie.io/docs/cli>

IDE (IntelliJ Ultimate)

자바와 표준 엔터프라이즈 기술, 그리고 스프링을 이용하는 개발을 할 때 사용할 수 있는 여러 통합개발환경(IDE) 방식의 도구들이 있다.

그런데 본격적으로 스프링 개발을 한다면 JetBrains의 IntelliJ IDEA 사용을 강력하게 추천한다.

<https://www.jetbrains.com/idea/>

IntelliJ IDEA는 유료로 구입해야 하는 Ultimate 버전과 무료로 사용 가능한 오픈소스 Community Edition이 있다. 스프링 개발에 자주 활용되는 여러 편리한 기능을 이용하려면 Ultimate 버전의 사용이 좋지만, 불편을 조금 감수하고 무료 버전을 선택할 수도 있다.

Ultimate 버전은 기업용, 개인용으로 구분되고 구독 기간에 따라서 여러가지 가격 모델을 가지고 있다.

만약 학생이거나 교사, 오픈소스 프로젝트 메인테이너, 부트캠프 학생, 유저그룹 운영자 등이라면 ULTIMATE 버전의 무료 라이선스를 신청할 수 있다. 스타트업 할인 혜택을 받을 수도 있다.

자세한 스페셜 가격 옵션에 대해서는 다음 페이지의 내용을 참고해서 해당되는 것이 있으면 신청해보는 것이 좋다.

<https://www.jetbrains.com/idea/buy/?section=discounts&billing=yearly>

MySQL

DB로 사용할 MySQL을 설치하는 방법은 크게 두 가지가 있다.

1. MySQL Installer를 이용해 시스템 레벨에 설치

- <https://dev.mysql.com/doc/refman/8.4/en/macos-installation.html>

2. Docker를 이용한 설치

- https://hub.docker.com/_/mysql

개발용 DB를 프로젝트에 맞게 DB 버전을 맞추고, 간편하게 구성하고 싶다면 Docker를 이용하는 것이 편리하다.

IntelliJ는 Docker 컨테이너 관리 기능을 제공한다.

개발하는 애플리케이션에서 필요로 하는 서비스, 네트워크, 포트 매핑, 환경 변수 등을 정의해서 설정 파일에 담을 수 있는 docker compose를 사용하면 일관된 개발환경을 구성하기 편리하다.

docker-compose.yml 파일에 DB 컨테이너와 관련된 정보를 넣어두고, Spring Boot의 Docker Compose Support 기능을 이용하면 DB 접속 정보 등을 손쉽게 지정할 수도 있다.

Docker 환경 구성

1. Docker Desktop을 설치해서 사용

- 개인 장비에 학습 또는 개인적인 목적을 위해서 설치해서 사용하거나 일정 조건을 만족하는 기업인 경우에는 무료 개인 버전을 사용할 수 있다
- 그 외에는 조건에 맞는 구독 플랜을 구매해서 사용해야 한다.

2. Docker Desktop을 대체할 수 있는 도구 선택

- Rancher Desktop
- Podman Desktop
- OrbStack

Rancher Desktop

데스크톱에서 컨테이너와 쿠버네티스를 사용하는 작업에 필수적인 기능을 제공하는 오픈소스 애플리케이션

설치

<https://rancherdesktop.io/> 에서 다운로드 후 설치

처음 실행할 때 Kubernetes

Windows 개발환경 구성

Chocolatey

윈도우용 패키지 매니저

오픈소스 버전은 무료로 사용할 수 있다.

<https://community.chocolatey.org/>

설치

<https://chocolatey.org/install>

PowerShell에서 설치

- HTTPie

SDKMAN

두 가지 설치 방법

- Git Bash
- WSL (Windows Subsystem for Linux)

프로젝트 초기 설정

SpringBoot Docker Compose

`ConnectionDetails` 추상화를 기반으로 Docker Compose를 이용해서 스프링부트 애플리케이션을 개발하는 동안에 필요한 서비스를 도커 위에 실행하고, 그 설정 정보를 스프링부트의 프로퍼티 정보로 활용하게 해주는 도구이다.

참고 문서

- <https://spring.io/blog/2023/06/21/docker-compose-support-in-spring-boot-3-1>
- <https://docs.spring.io/spring-boot/how-to/docker-compose.html>

mysql docker compose 설정

스프링이 기본적으로 만들어주는 compose.yml 파일은 다음과 같다.

```
services:
  mysql:
    image: 'mysql:latest'
    environment:
      - 'MYSQL_DATABASE=mydatabase'
      - 'MYSQL_PASSWORD=secret'
      - 'MYSQL_ROOT_PASSWORD=verysecret'
      - 'MYSQL_USER=myuser'
    ports:
      - '3306'
```

개발에 사용할 DB 이름과 접속 정보를 수정하고 사용하면 된다.

Note

ports 설정을 위와 같이 하나만 넣으면 container 쪽의 3306 포트를 개발에 사용하는 장비인 호스트쪽 포트에 매핑할 때 현재 사용중이지 않은 임의의 포트를 선택해서 사용한다.

이때 선택된 랜덤 포트는 SpringBoot의 Docker Compose 지원 기능을 통해서 스프링의 DataSource 설정을 만들 때 사용되므로 애플리케이션에서 연결하는 데는 문제가 없지만, 외부 DB 툴에서 접속하려면 매번 호스트 포트 번호를 확인해야 하는 번거로움이 있다.

개발 장비에 3306 포트가 사용 가능하다면 다음과 같이 호스트 포트도 3306으로 매핑하는 것이 편리하다.

```
# compose.yml
ports:
  - '3306:3306'
```

호스트 매핑에 관해서는 Docker의 네트워크 설정에 관한 다음 문서를 참고해보자.

<https://docs.docker.com/compose/how-tos/networking/>

```
- HOST_PORT:CONTAINER_HOST
```