

제 1 강 스프링 프레임워크의 소개 및 설치

1. 스프링 프레임워크의 정의 및 특징

자바 엔터프라이즈 개발을 편하게 해주는 오픈소스 경량급 애플리케이션 프레임워크

① 애플리케이션 프레임워크

일반적으로 라이브러리나 프레임워크는 특정 업무 분야나 한 가지 기술에 특화된 목표를 가지고 만들어집니다. 예를 들면 웹 계층을 MVC 구조로 손쉽게 만들 수 있게 한다거나, 포맷과 출력장치를 유연하게 변경할 수 있는 애플리케이션 로그 기능을 제공한다거나, 간단한 설정만으로 관계형 DB와 자바오브젝트를 매핑해주는 ORM 기술을 제공하는 것들입니다. 그래서 프레임워크는 애플리케이션의 특정 계층에서 주로 동작하는 한 가지 기술 분야에 집중됩니다. 하지만 스프링은 이와 다르게 '애플리케이션 프레임워크'라는 특징을 갖고 있습니다.

애플리케이션 프레임워크는 특정 계층이나, 기술, 업무 분야에 국한되지 않고 애플리케이션의 전 영역을 포괄하는 범용적인 프레임워크를 말합니다. 애플리케이션 프레임워크는 애플리케이션 개발의 전 과정을 빠르고 편리하며 효율적으로 진행하는데 일차적인 목표를 두는 프레임워크입니다.

여러 계층의 다양한 기술을 그저 한데 모아뒀기 때문에 애플리케이션 프레임워크라고 불리는 건 아닙니다. 애플리케이션의 전 영역을 관통하는 일관된 프로그래밍 모델과 핵심 기술을 바탕으로 해서 각 분야의 특성에 맞는 필요를 채워주고 있기 때문에, 애플리케이션을 빠르고 효과적으로 개발할 수가 있습니다. 바로 이것이 스프링이 애플리케이션 프레임워크라고 불리는 이유입니다.

② 경량급

EJB는 기술에 대한 과도한 욕심으로 인해 개발환경과 운용서버, 개발과 빌드, 테스트 과정, 작성된 코드 모두를 매우 무겁고 복잡하게 만들었습니다. EJB가 동작하려면 고가의 느리고 무거운 자바 서버(WAS)가 필요했습니다. 또한 툴의 도움 없이는 다루기 힘든 난해한 설정파일 구조와 까다로운 패키징, 불편한 서버 배치(deploy) 등으로 인한 부담 때문에 고가의 제품으로 구성된 제대로 된 개발환경을 갖추지 않고는 개발하기가 힘들었습니다.

그에 반해 스프링은 가장 단순한 서버환경인 톰캣(Tomcat)이나 제티(Jetty)에서도 완벽하게 동작합니다. 단순한 개발 툴과 기본적인 개발환경으로도 엔터프라이즈 개발에서 필요로 하는 주요한 기능을 갖춘 애플리케이션을 개발하기에 충분합니다. 서블릿 컨테이너만으로 충분하니 EJB 컨테이너를 비롯해 복잡한 기능이 잔뜩 포함된 고급 WAS를 굳이 사용하지 않아도 됩니다. 그만큼 개발 과정도 단순해 집니다. 스프링의 장점은 그런 가볍고 단순한 환경에서도 복잡한 EJB와 고가의 WAS를 갖춰야만 가능했던 엔터프라이즈 개발의 고급 기술을 대부분 사용할 수 있다는 점입니다. 코드는 더 단순하고 개발 과정은 편리하면서도 EJB에서조차 불편했던 고급 기능을 세련된 방식으로 적용할 수 있습니다.

③ 자바 엔터프라이즈 개발을 편하게

스프링 뿐 아니라 기존에 등장했던 대부분의 자바 엔터프라이즈 기술과 프레임워크는 저마다 '개발을 편하게 해 준다'고 주장하고 있습니다. 하지만 스프링이 말하는 '엔터프라이즈 개발을 편하게'라는 말은 그 무게가 다릅니다. 스프링은 근본적인 부분에서 엔터프라이즈 개발의 복잡함을 제거해내고

진정으로 개발을 편하게 해주는 해결책을 제시합니다. 단순히 편리한 몇 가지 도구나 기능을 제공하는 차원이 아닙니다. 엔터프라이즈 개발의 근본적인 문제점에 도전해서 해결책을 제시한다는 것이 기존 기술의 접근 방법과 스프링의 접근 방법의 차이점입니다.

④ 오픈소스

오픈소스란 말 그대로 소스가 모두에게 공개되고, 특별한 라이선스를 취득할 필요 없이 얼마든지 가져다 자유롭게 이용해도 된다는 뜻입니다. 소스를 자유롭게 열람하고 자신의 목적에 맞게 사용할 수 있을 뿐만 아니라, 필요하면 맘대로 수정할 수 있고, 수정된 제품과 소스를 다시 공개적으로 배포하는 자유도 허용됩니다. 물론 오픈소스도 저작권이 있기 때문에 원 저작자에 대한 정보와 라이선스는 유지한 채로 사용하거나 배포해야지, 자신이 만든 것처럼 슬며시 가져다 사용해도 좋다는 뜻은 아닙니다.

스프링에 적용된 오픈소스 라이선스는 오픈소스 라이선스 중에서도 비교적 제약이 적고 사용이 매우 자유로운 편인 아파치 라이선스 버전 2.0(Apache)입니다.

2. 스프링의 목적 및 등장 배경

① 엔터프라이즈 개발의 복잡함

2000년대 초반 각종 자바 컨퍼런스에서 자주 논의됐던 주제는 '왜 자바 엔터프라이즈(JavaEE) 프로젝트는 실패하는가?'였습니다. 당시 IT 리서치기업의 조사에 따르면 80% 이상의 자바 엔터프라이즈 프로젝트가 실패했다고 합니다. 프로젝트가 아예 중단되고 취소된 것까지는 아니더라도, 원래 정해진 기간과 계획된 예산을 맞추지 못한 경우가 그만큼 많다는 뜻입니다. 또는 원하는 만큼의 기능과 완성도를 갖춘 시스템을 못 만들고 적당히 마무리하기도 했을 것입니다. 아무튼 자바 엔터프라이즈 개발이 실패하는 이유에 대해 많은 논의가 있었습니다. 그 과정에서 밝혀진 여러 가지 원인이 있었지만, 그중 가장 대표적인 게 '엔터프라이즈 시스템 개발이 너무 복잡해져서'였습니다.

※ 복잡함의 근본적인 이유

○ 첫 번째는 기술적인 제약조건과 요구사항이 늘어가기 때문입니다.

엔터프라이즈 시스템이란 서버에서 동작하며 기업과 조직의 업무를 처리해주는 시스템을 말합니다. 엔터프라이즈 시스템은 많은 사용자의 요청을 동시에 처리해야하기 때문에 서버의 자원을 효율적으로 공유하고 분배해서 사용할 수 있어야 합니다. 또한 중요한 기업의 핵심 정보를 처리하거나 미션 크리티컬한 금융, 원자력, 항공, 국방 등의 시스템을 다루기도하기 때문에 보안과 안정성, 확장성 면에서도 뛰어나야 합니다. 따라서 뛰어난 성능과 서비스의 안정성이 요구되고 그런 점을 고려한 개발 기술이 필요합니다.

○ 두 번째는 엔터프라이즈 애플리케이션이 구현해야 할 핵심기능인 비즈니스 로직의 복잡함이 증가하기 때문입니다.

예전에는 기업 업무 중 회계처럼 복잡한 계산이나 빠른 분석 작업이 필요한 영역에서만 IT 시스템을 활용했습니다. 하지만 갈수록 엔터프라이즈 시스템을 이용해 기업의 핵심 업무를 처리하는 비율이 높아갔고, 점차 대부분의 업무 처리는 컴퓨터를 이용하지 않고는 아예 진행하기 힘들 만큼 엔터프라이즈 시스템에 대한 업무 의존도가 높아졌습니다.

※ 복잡함을 가중시키는 원인

엔터프라이즈 애플리케이션 개발이 실패하는 주요 원인은 비즈니스 로직의 복잡함과 기술적인 복잡함입니다. 복잡하다는 건 단지 양이 많고 어렵다는 뜻이 아닙니다. 세부 요소가 이해하기 힘든 방식

으로 얹혀 있고, 그 때문에 쉽게 다루기 어렵다는 의미입니다. 자칫 잘못 손을 댔다가는 더 엉망이 되기 쉬우며, 들인 노력과 시간이 허사가 될 수도 있습니다. 자바 엔터프라이즈 시스템 개발이 어려운 가장 큰 이유는 근본적인 비즈니스 로직과 엔터프라이즈 기술이라는 두 가지 복잡함이 한데 얹혀 있기 때문입니다.

② 복잡함을 해결하려는 도전 : 실패한 해결책: EJB

EJB가 처음 등장했을 때 내세웠던 목표를 봐도 알 수 있듯이 EJB의 기본 전략도 이 두 가지 종류의 복잡함을 분리하는 것이었습니다. 개발자가 로우레벨의 기술적인 복잡함에 신경 쓰지 않고 비즈니스 로직을 효과적으로 개발하는 데 더 집중할 수 있게 하자는 목표가 있었습니다.

하지만 기존 EJB는 결과적으로 그런 목표를 달성하는 데 실패했습니다. EJB는 기술적인 복잡함을 애플리케이션의 핵심 로직에서 일부분 분리하는 데 성공하긴 했습니다. 선언적 트랜잭션이나 선언적 보안, 컨테이너를 통한 리모팅 기술의 적용, 컴포넌트 단위의 배치, JNDI를 통한 서비스 검색 지원, 서비스 오브젝트의 풀링, 컴포넌트 생명주기 관리 등은 EJB의 목표를 어느 정도 충족시켰습니다. 반면에 EJB 환경에서 동작하기 위해 특정 인터페이스를 구현하고, 특정 클래스를 상속하고, 서버에 종속적인 서비스를 통해서만 접근하고 사용이 가능하게 만드는 등의 EJB 개발 방식은 잘못된 선택이었습니다. 애플리케이션 로직을 담은 핵심 코드에서 일부 기술적인 코드가 제거된 건 사실이지만, 오히려 EJB라는 환경과 스펙에 종속되는 코드로 만들어져야 하는 더 큰 부담을 안게 되었습니다.

EJB는 결국 일부 기술적인 복잡함을 덜어주려는 시도를 하다가 오히려 더 큰 복잡함을 추가하는 실수를 범했습니다. 가장 치명적인 건, EJB라는 틀 안에서 자바 코드를 만들게 강제함으로써 자바 언어가 원래 갖고 있던 장점마저 잃어버렸다는 사실입니다. EJB의 특정 클래스를 상속하게 함으로써 더 이상 상속구조를 적용하지 못하게 만들거나, 다형성 적용을 근본적으로 제한한다거나 하는 것들입니다. EJB는 결국 객체지향적인 특성을 잃어버린 밋밋한 서비스 스크립트성 코드로 변질돼갔습니다. 별다른 장점은 없는데다 개발 방식은 너무 불편했기 때문에 개발자에게 점점 외면당하는 신세가 돼버렸습니다.

③ 비 침투적인 방식을 통한 효과적인 해결책: 스프링

스프링은 EJB의 실패를 교훈으로 삼아서 출발했습니다. EJB의 처음 목표와 마찬가지로 기술적인 복잡함을 애플리케이션 핵심 로직의 복잡함에서 제거하는 데 목표를 뒀습니다. 하지만 그 과정에서 EJB 처럼 개발자의 코드에 난입해서 지저분하고 복잡한 코드를 만들어버리는 실수를 하지는 않았습니다. EJB처럼 어떤 기술을 적용했을 때 그 기술과 관련된 코드나 규약 등이 코드에 등장하는 경우를 침투적인 기술이라고 합니다. 물론 꼭 필요한 기능을 사용해야하기 때문에 특정 기술의 API를 이용하게 되는 건 어쩔 수 없습니다. 그런데 꼭 필요한 기능을 사용하는 것도 아니면서 단지 어떤 기술을 바탕으로 만들어 진다고해서 특정 클래스나 인터페이스, API 등의 코드에 마구 등장한다면 그것은 침투적인 기술이 되며 복잡함을 가중시키는 원인이 됩니다.

반면에 비 침투적인(non-invasive) 기술은 기술의 적용 사실이 코드에 직접 반영되지 않는다는 특징이 있습니다. 어딘가에는 기술의 적용에 따라 필요한 작업을 해줘야 하겠지만, 애플리케이션 코드 여기저기에 불쑥 등장하거나, 코드의 설계와 구현 방식을 제한하지는 않는다는 게 비 침투적인 기술의 특징입니다.

스프링이 성공할 수 있었던 비결은 바로 비 침투적인 기술이라는 전략을 택했기 때문입니다. 스프링을 이용하면 기술적인 복잡함과 비즈니스 로직을 다루는 코드를 깔끔하게 분리할 수 있습니다. 중

요한 점은 그 과정에서 스프링 스스로가 애플리케이션 코드에 불필요하게 나타나지 않도록 하는 것입니다. 꼭 필요한 것 같은 경우조차도 기술 코드가 직접 노출되지 않도록 만들어졌습니다.

3 POJO 프로그래밍

스프링 핵심 개발자들은 "스프링의 정수는 엔터프라이즈 서비스 기능을 POJO에 제공하는 것"이라고 했습니다. 엔터프라이즈 서비스라고 하는 것은 보안, 트랜잭션과 같은 엔터프라이즈 시스템에서 요구되는 기술을 말합니다. 이런 기술을 POJO에 제공한다는 말은, 뒤집어 생각해보면 엔터프라이즈 서비스 기술과 POJO라는 애플리케이션 로직을 담은 코드를 분리했다는 뜻이기도 합니다. "분리됐지만 반드시 필요한 엔터프라이즈 서비스 기술을 POJO 방식으로 개발된 애플리케이션 핵심 로직을 담은 코드에 제공한다"는 것이 스프링의 가장 강력한 특징과 목표입니다.

① POJO란 무엇인가?

스프링 애플리케이션 개발의 핵심인 POJO를 좀 더 자세히 알아보겠습니다. POJO는 Plain Old Java Object의 첫 글자를 따서 만든 약자입니다. 최근 몇 년간 자바에서 유행어처럼 사용되고 있는 이 단어는 마틴 파울러가 2000년에 컨퍼런스 발표를 준비하다가 만들어낸 용어라고 합니다. 그런데 POJO라는 용어를 만들어낸 이유가 재미있습니다. 마틴 파울러는 당시 인기를 끌고 있던 EJB처럼 복잡하고 제한이 많은 기술을 사용하는 것보다는 자바의 단순한 오브젝트를 이용해 애플리케이션의 비즈니스 로직을 구현하는 편이 낫다고 생각했습니다. 그럼에도 왜 개발자는 자바의 단순한 오브젝트를 사용하길 꺼리는지 궁금했습니다. 그 이유를 찾아보니 평범한 자바오브젝트에는 EJB와 같은 그럴싸한 이름이 없기 때문이었습니다. 그래서 뭔가 있어 보이도록 만든 이름이 바로 POJO였습니다. 같은 설명이지만 그냥 "간단한 자바오브젝트를 사용 하는데요"라고 말하는 것보다 "POJO 방식의 기술을 사용합니다"라고 하면 왠지 세련되고 첨단기술을 쓰는 것처럼 느껴진다는 심리를 이용한 것입니다. 평범한 자바오브젝트에 멋진 이름을 붙여줬던 시도는 기대 이상으로 성공적이었습니다.

단지 POJO라는 품 나는 이름 때문만은 아니겠지만, 아무튼 그 이후로 POJO 프로그래밍에 관한 개발자들의 관심이 높아졌고 POJO를 지원한다는 걸 장점으로 내세우는 많은 프레임워크 기술이 쏟아져 나오기 시작했습니다. 심지어 EJB조차 3.0에서는 기존의 문제점을 반성하고 POJO 프로그래밍의 장점을 적극 도입하려고 했습니다.

② POJO의 조건

그렇다면 이 POJO 프로그래밍이란 무엇일까요? 그저 EJB를 사용하지 않으면 POJO일까요? 특정 프레임워크의 클래스를 상속하지 않으면 POJO일까요? 단순히 보자면 그냥 평범한 자바오브젝트라고 할 수 있지만 좀 더 명확하게 하자면 적어도 다음의 세 가지 조건을 충족해야 POJO라고 불릴 수 있습니다.

※ 특정 규약(Contract)에 종속되지 않는다

POJO는 자바 언어와 꼭 필요한 API 외에는 종속되지 않아야 합니다. 따라서 EJB 2와 같이 특정 규약을 따라 비즈니스 컴포넌트를 만들어야 하는 경우는 POJO가 아닙니다. 스트럿츠1과 같이 특정 클래스를 상속해서 만들어야 하는 규약이 있는 경우도 마찬가지입니다. 특정 규약을 따라 만들게 하는 경우는 대부분 규약에서 제시하는 특정 클래스를 상속하도록 요구합니다. 그럴 경우 자바의 단일 상속 제한 때문에 더 이상 해당 클래스에 객체지향적인 설계 기법을 적용하기가 어려워지는 문제가 생깁니다. 또한 규약이 적용된 환경에 종속적이 되기 때문에 다른 환경으로 이전이 힘들다는 문제점이 있습니다.

스트럿츠1의 ActionForm을 상속한 폼 정보를 담는 객체는 스트럿츠의 규약에 독립적이어야 하는 서비스 레이어로 전달하기에 적합하지 않기 때문에 거의 비슷한 구조를 가진 DTO를 만들어 복사해줘야 하는 번거로움이 있습니다. 당연히 객체지향적인 설계로 만든 POJO 도메인 모델 오브젝트를 그대로 웹 레이어에서 사용할 수는 없습니다. EJB 2 또한 비즈니스 컴포넌트가 EntityBean 클래스를 상속해야만 합니다. 별다른 가치를 주지도 못하는 규약 따위에 종속되지 않아야 하고, 객체지향 설계의 자유로운 적용이 가능한 오브젝트여야만 POJO라고 불릴 수 있습니다.

※ 특정 환경에 종속되지 않는다

특정 환경에 종속적이어야만 동작하는 오브젝트도 POJO라고 할 수 없습니다. EJB 3는 분명 이전 버전의 문제점이던 규약에 따라 오브젝트를 만들어야 한다는 단점은 극복했습니다. 어떤 면에서 POJO에 가까운 설계와 구현이 가능해졌습니다. 하지만 여전히 JNDI라는 서버 서비스를 필요로 합니다. EJB 3빈의 의존 오브젝트 정보는 JNDI를 통해 가져와야하기 때문입니다. 따라서 JNDI가 없는 환경에서는 그대로 사용하기가 힘듭니다. 이렇게 JNDI와 같은 특정 환경이 의존 대상 검색 방식에 종속적이라면 POJO라고 할 수 없습니다.

어떤 경우는 특정 벤더의 서버나 특정 기업의 프레임워크 안에서만 동작 가능한 코드로 작성되기도 합니다. 또 환경에 종속적인 클래스나 API를 직접 쓴 경우도 있습니다. 예를 들면 WebLogic 서버에서만 사용 가능한 API를 직접 쓴 코드를 갖고 있거나 특정 OS에서 제공하는 기능을 직접 호출하도록 만들어진 오브젝트 등이 존재합니다. 이런 식으로 순수한 애플리케이션 로직을 담고 있는 오브젝트 코드가 특정 환경에 종속되게 만드는 경우라면 그것 역시 POJO라고 할 수는 없습니다. POJO는 환경에 독립적이어야 합니다.

※ 진정한 POJO란 객체지향적인 원리에 충실하면서, 환경과 기술에 종속되지 않고 필요에 따라 재사용될 수 있는 방식으로 설계된 오브젝트를 말합니다. 그런 POJO에 애플리케이션의 핵심 로직과 기능을 담아 설계하고 개발하는 방법을 POJO 프로그래밍이라고 할 수 있습니다.

③ POJO의 장점

POJO 프로그래밍의 장점은 무엇일까요? 바로 POJO가 될 수 있는 조건이 그대로 POJO의 장점이 됩니다.

특정한 기술과 환경에 종속되지 않는 오브젝트는 그만큼 깔끔한 코드가 될 수 있습니다. 로우 레벨의 기술과 환경에 종속적인 코드가 비즈니스 로직과 함께 섞여 나오는 것만큼 지저분하고 복잡한 코드도 없습니다. 그런 코드는 개발하기도 힘들고, 오류를 찾고 디버깅하기는 더더욱 힘듭니다. 코드를 읽고 이해기도 어려울뿐더러 검증이나 테스트 작성에도 한계가 있으므로 유지보수는 큰 부담이 됩니다.

또 POJO로 개발된 코드는 자동화된 테스트에 매우 유리합니다. 환경의 제약은 코드의 자동화된 테스트를 어렵게 합니다. 컨테이너에서만 동작을 확인할 수 있는 EJB 2는 테스트하려면 서버의 구동 및 빌드와 배치 과정까지 필요합니다. 자동화된 테스트가 불가능한건 아니지만 매우 복잡하고 번거로우므로 대부분 수동 테스트 방식을 선호합니다. 간단한 코드 수정에도 빌드, 배치, 수동 확인이라는 번거로운 사이클을 따라 작업하기가 얼마나 힘든지는 EJB 개발자들이 경험적으로 잘 알고 있습니다. 그에 반해 어떤 환경에도 종속되지 않은 POJO 코드는 매우 유연한 방식으로 원하는 레벨에서 코드를 빠르고 명확하게 테스트할 수 있습니다.

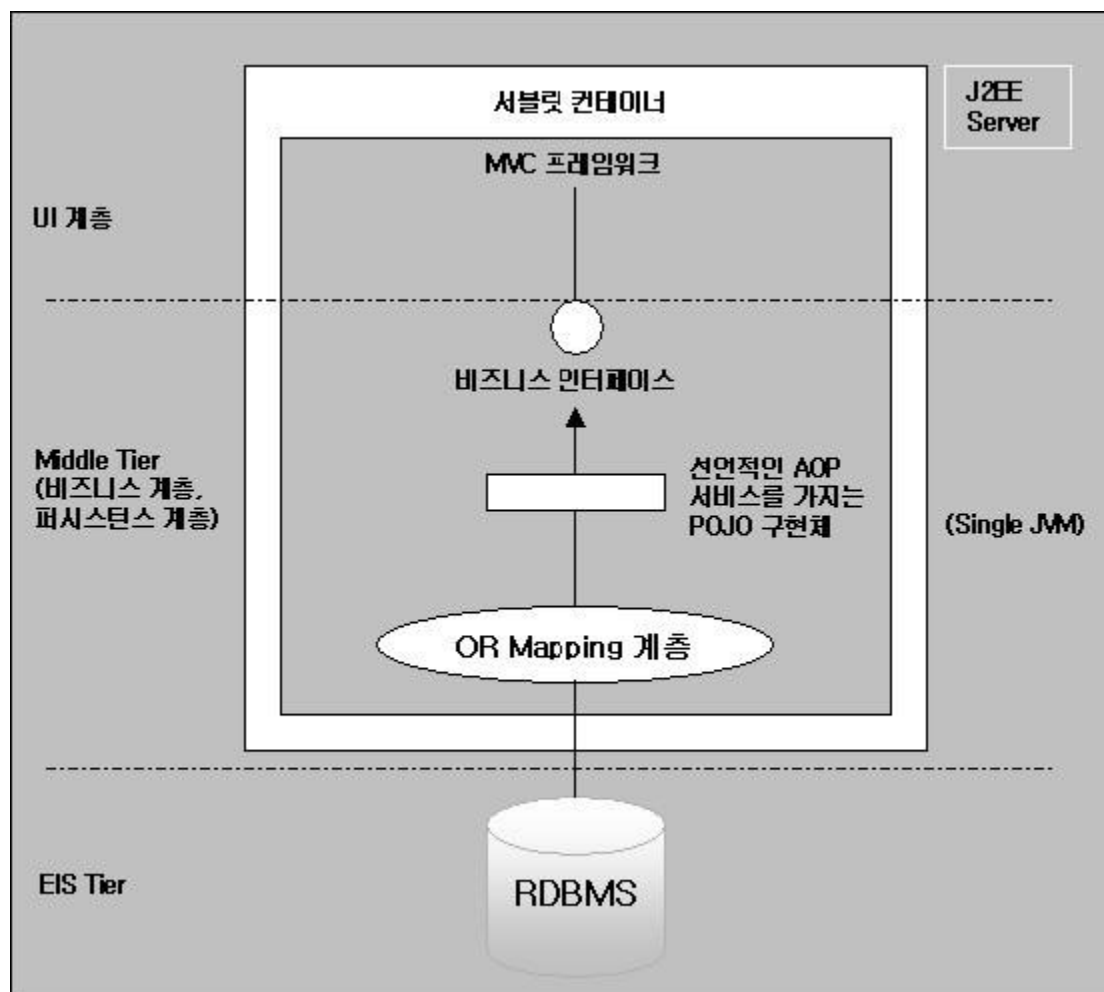
객체지향적인 설계를 자유롭게 적용할 수 있다는 것도 큰 장점입니다. 개발자들이 자바와 객체지향 프로그래밍, 모델링과 설계에 대해 배울 때 그려왔던 도메인 모델과, 오랜 경험을 통해 쌓여온 재 활용 가능한 설계 모델인 디자인 패턴 등은 POJO가 아니고는 적용하기 힘듭니다. 로드 존슨은 특정 기술과 규약, 환경보다 자바 언어와 객체지향 기술이 더 중요하다는 사실을 끊임없이 강조했습니다.

④ POJO 프레임워크

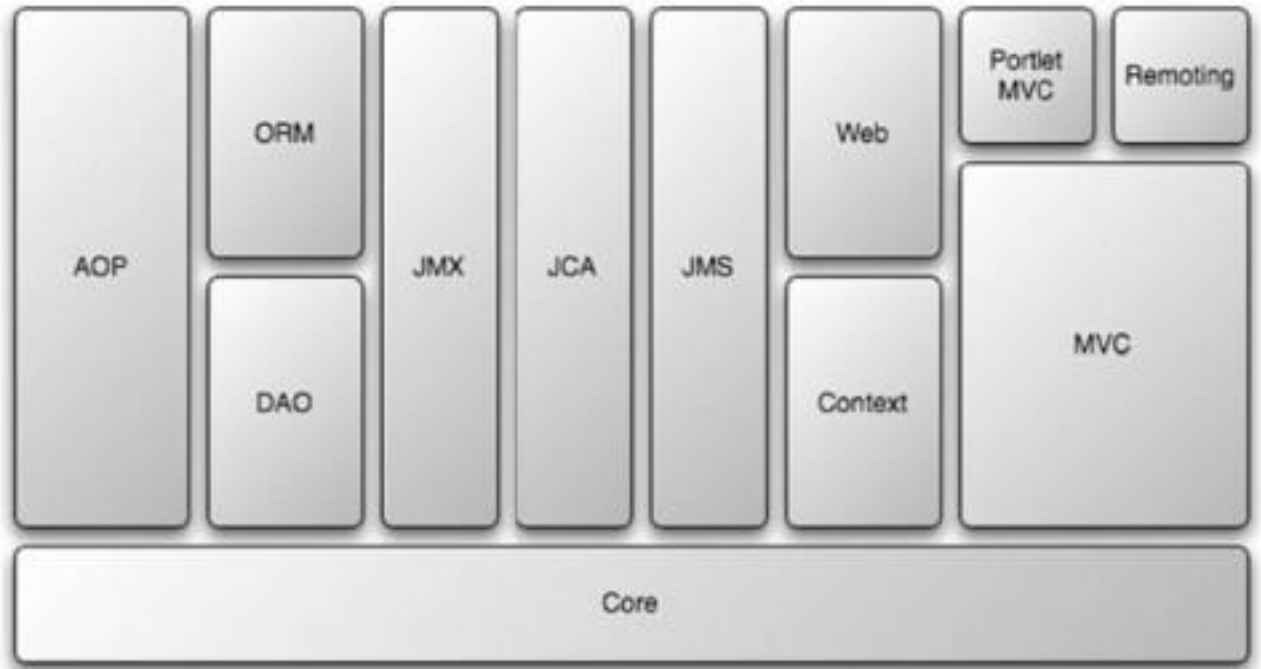
스프링은 POJO를 이용한 엔터프라이즈 애플리케이션 개발을 목적으로 하는 프레임워크라고 했습니다. POJO 프로그래밍이 가능하도록 기술적인 기반을 제공하는 프레임워크를 POJO 프레임워크라고 합니다. 스프링 프레임워크와 하이버네이트를 대표적인 POJO 프레임워크로 꼽을 수 있습니다. 주로 DB 이용 기술에 POJO를 적용하는 것을 목적으로 하는 하이버네이트와 달리, 스프링은 엔터프라이즈 애플리케이션 개발의 모든 영역과 계층에서 POJO 방식의 구현이 가능하게 하려는 목적으로 만들어졌습니다.

스프링을 이용하면 POJO 프로그램의 장점을 그대로 살려서 엔터프라이즈 애플리케이션의 핵심 로직을 객체지향적인 POJO를 기반으로 깔끔하게 구현하고, 동시에 엔터프라이즈 환경의 각종 서비스와 기술적인 필요를 POJO 방식으로 만들어진 코드에 적용할 수 있습니다.

아래 그림은 스프링이 엔터프라이즈 시스템의 복잡함을 어떻게 다루는지를 보여주는 그림입니다. 하지만 자신은 기술영역에만 관여하지 비즈니스 로직을 담당하는 POJO에서는 모습을 감춥니다. 데이터 액세스 로직이나 웹 UI 로직을 다룰 때만 최소한의 방법으로 관여합니다. POJO 프레임워크로서 스프링은 자신을 직접 노출하지 않으면서 애플리케이션을 POJO로 쉽게 개발할 수 있게 지원해줍니다.



4 Spring의 구성



① Spring Core

Core 컨테이너 기능. **Spring 프레임워크의 근간이 되는 IoC(또는 DI) 기능**을 지원하는 영역을 담당하고 있다. BeanFactory를 기반으로 Bean 클래스들을 제어할 수 있는 기능을 지원한다.

② Spring Context

Spring Core 바로 위에 있으면서 Spring Core에서 지원하는 기능 외에 **추가적인 기능**들과 좀 더 쉬운 개발이 가능하도록 지원하고 있다. 또한 JNDI, EJB등을 위한 Adaptor들을 포함하고 있다.

③ AOP (Aspect Oriented Programming) : 관점(관심) 지향 프로그래밍

Spring 프레임워크에 Aspect Oriented Programming을 지원하는 기능이다. 이 기능은 AOP Alliance 기반 하에서 개발되었다.

④ DAO (Data Access Object)

지금까지 우리들이 일반적으로 많이 사용해왔던 JDBC 기반의 DAO개발을 좀 더 쉽고, 일관된 방법으로 개발하는 것이 가능하도록 지원하고 있다. Spring DAO를 이용할 경우 지금까지 개발하던 DAO보다 적은 코드와 쉬운 방법으로 DAO를 개발하는 것이 가능하다.

⑤ ORM (Object-Relational Mapping) : OOP(객체 지향 프로그래밍)을 사용

Object Relation Mapping 프레임워크인 Hibernate, myBatis, JDO와의 결합을 지원하기 위한 기능이다. Spring ORM을 이용할 경우 Hibernate, myBatis, JDO 프레임워크와 쉽게 통합하는 것이 가능하다.

⑥ JMX (Java Management Extensions)

자바 어플리케이션의 모니터링과 관리 기능을 제공

⑦ JCA (Java EE Connector API)

엔터프라이즈 시스템들을 통합하기 위한 표준 API

⑧ Spring Web

Web Application 개발에 필요한 Web Application Context와 MultipartRequest등의 기능을 지원한다. 또한 Struts, Webwork와 같은 프레임워크의 통합을 지원하는 부분을 담당한다.

⑨ Spring Web MVC

Spring 프레임워크에서 독립적으로 Web UI Layer에 Model-View-Controller를 지원하기 위한 기능이다. 지금까지 Struts, Webwork가 담당했던 기능들을 Spring Web MVC를 이용하여 대체하는 것이 가능하다. 또한 Velocity, Excel, PDF와 같은 다양한 UI 기술들을 사용하기 위한 API를 제공하고 있다.

⑩ Portlet MVC

포탈서버에서 돌아가는 독립된 웹 어플리케이션

⑪ Remoting

원격 지원 기능

5. Spring 사용이유

① Layer 아키텍처.

② 좋은 설계가 기술 자체보다 더 중요하다.

③ Interface를 통한 느슨한 결합의 Java Beans는 훌륭한 모델이다.

④ Spring 프레임워크가 테스트하기 쉬운 구조로 코드를 개발하는 것이 가능하도록 지원한다.

⑤ 객체의 생성과 클래스간의 의존관계에 대한 고민을 덜어주고 있다는 측면이다

6. Spring 활용 시 결정 사항

① J2EE(Java, JSP, Servlet, EJB) 개발 환경에 1년 이상의 경험을 가진 개발자.

② 효율적인 애플리케이션 개발 방법을 찾고 있는 개발자.

③ 개발 생산성과 유지보수성의 향상을 높이하고자 하는 개발자.

④ 체계화된 개발을 하고자 하는 개발자.

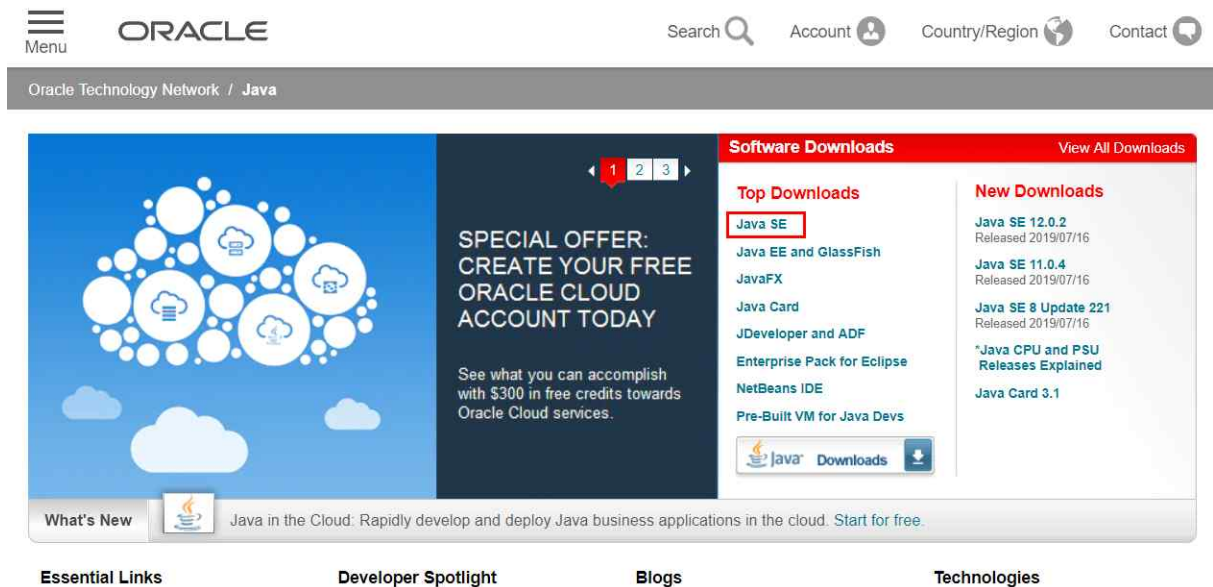
⑤ 표준 프레임워크를 도입하고자 하는 개발자.

⑥ Spring Framework는 표준 프레임워크로 채택해도 손색이 없을 만큼 모든 영역을 포괄하고 있다.

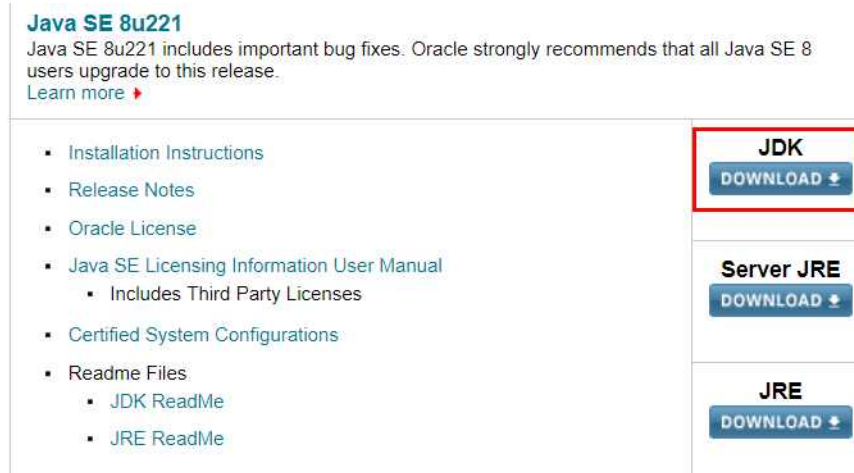
7 개발환경 설정

① JDK 설치

■ 우선 <http://java.oracle.com>에 접속한다.



■ 위 그림에서 오른쪽 위에 [Java SE] 부분을 클릭하여 다운 받을 수 있는 페이지로 이동한다.




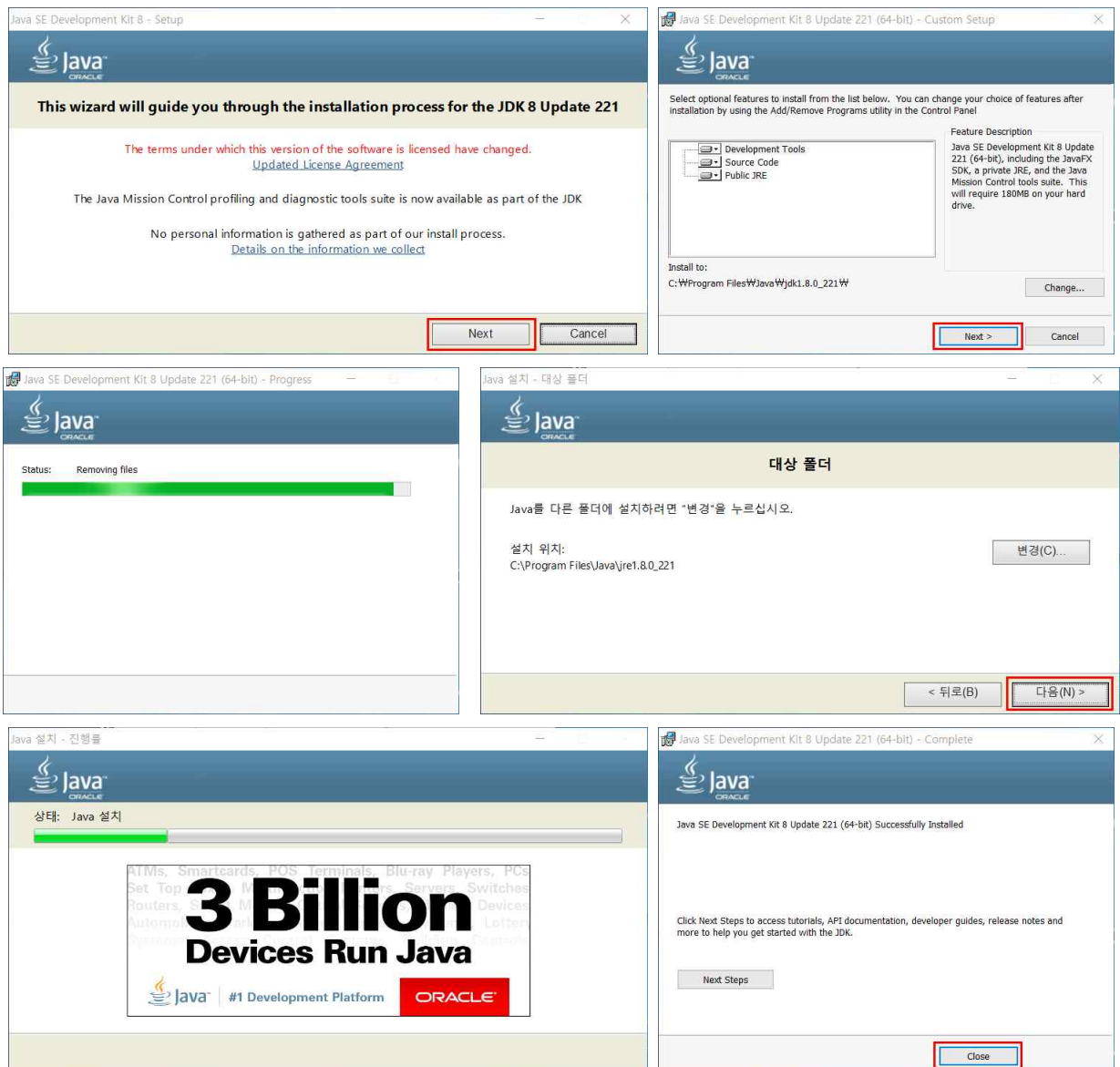
■ 위 그림에서 [JDK DOWNLOAD] 부분을 클릭하여 다운로드 페이지로 이동한다.

Java SE Development Kit 8u221		
You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software		
<input checked="" type="radio"/> Accept License Agreement <input type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.9 MB	jdk-8u221-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.81 MB	jdk-8u221-linux-arm64-vfp-hflt.tar.gz
Linux x86	174.18 MB	jdk-8u221-linux-i586.rpm
Linux x86	189.03 MB	jdk-8u221-linux-i586.tar.gz
Linux x64	171.19 MB	jdk-8u221-linux-x64.rpm
Linux x64	186.06 MB	jdk-8u221-linux-x64.tar.gz
Mac OS X x64	252.52 MB	jdk-8u221-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	132.99 MB	jdk-8u221-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.23 MB	jdk-8u221-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.66 MB	jdk-8u221-solaris-x64.tar.Z
Solaris x64	91.95 MB	jdk-8u221-solaris-x64.tar.gz
Windows x86	202.73 MB	jdk-8u221-windows-i586.exe
Windows x64	215.35 MB	jdk-8u221-windows-x64.exe

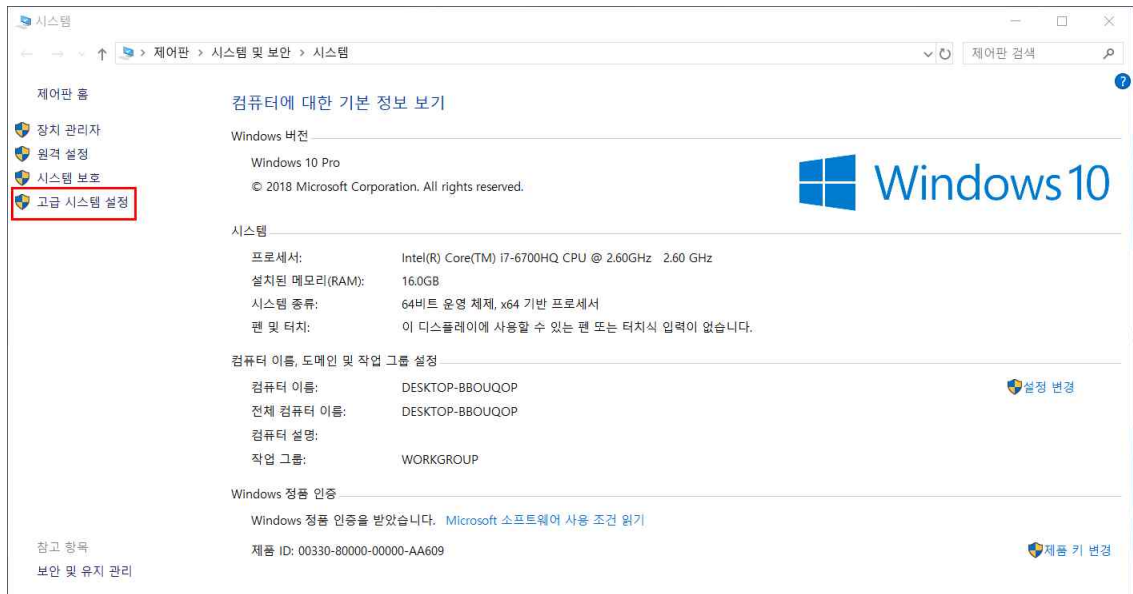
- [Accept License Agreement] 부분을 체크하고 아래와 같이 다운로드 받도록 한다.

Java SE Development Kit 8u221		
You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.		
Thank you for accepting the Oracle Technology Network License Agreement for Oracle Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.9 MB	jdk-8u221-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.81 MB	jdk-8u221-linux-arm64-vfp-hflt.tar.gz
Linux x86	174.18 MB	jdk-8u221-linux-i586.rpm
Linux x86	189.03 MB	jdk-8u221-linux-i586.tar.gz
Linux x64	171.19 MB	jdk-8u221-linux-x64.rpm
Linux x64	186.06 MB	jdk-8u221-linux-x64.tar.gz
Mac OS X x64	252.52 MB	jdk-8u221-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	132.99 MB	jdk-8u221-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.23 MB	jdk-8u221-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.66 MB	jdk-8u221-solaris-x64.tar.Z
Solaris x64	91.95 MB	jdk-8u221-solaris-x64.tar.gz
Windows x86	202.73 MB	jdk-8u221-windows-i586.exe
Windows x64	215.35 MB	jdk-8u221-windows-x64.exe

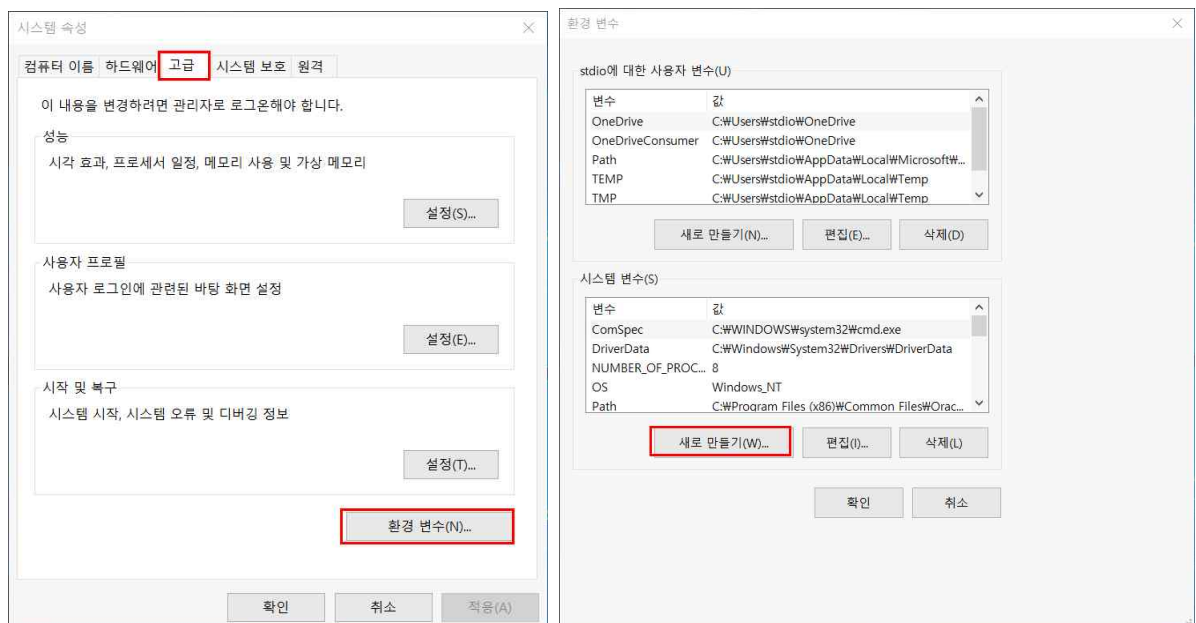
- 다운로드가 완료되면  아이콘을 더블클릭하여 설치를 진행하도록 한다.



- 설치가 완료되면 바탕화면에 [내 PC] 아이콘을 우 클릭하여 [속성]을 클릭하면 아래와 같은 화면이 나올 것이다. 이때 왼쪽의 [고급 시스템 설정]을 클릭하도록 하자.



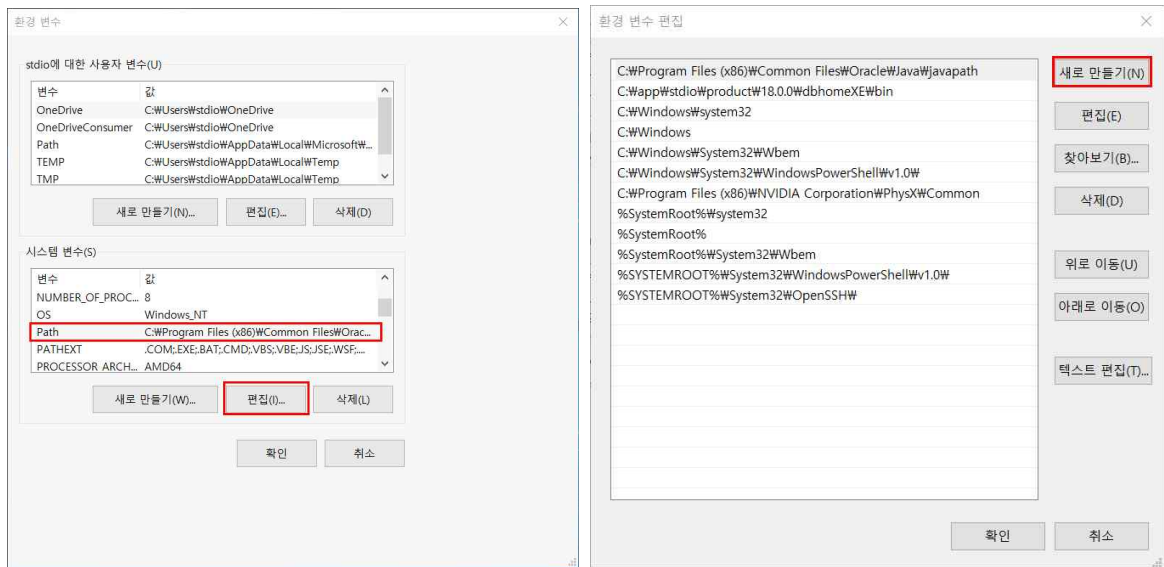
- 아래의 화면에서 [고급]탭을 클릭하고 [환경변수]버튼을 클릭하여 환경변수를 설정하도록 하자. 환경변수 설정창이 나오면 [새로 만들기]버튼을 클릭하자.



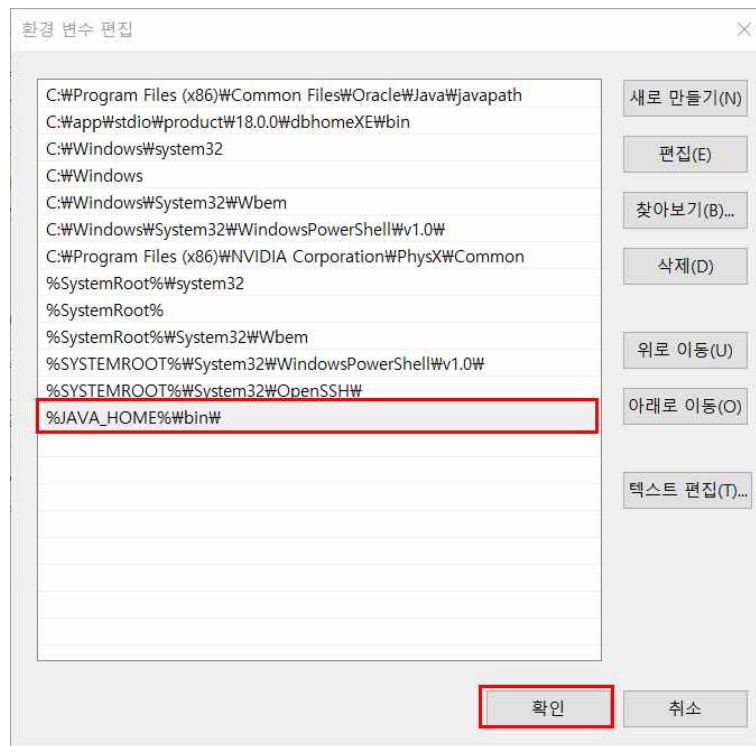
- 아래 그림과 같이 [JAVA_HOME] 이라는 환경변수의 값을 자바설치폴더로 주어 설정한다.



- 그 다음 아래와 같이 [Path]라는 환경변수를 찾아서 [편집]버튼을 클릭하고 환경변수 설정창이 나오면 우측의 [새로 만들기]버튼을 클릭한다.

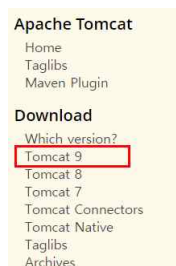


- 아래와 같이 %JAVA_HOME%\bin\를 입력하고 [확인]버튼을 클릭하여 설정을 마무리한다.

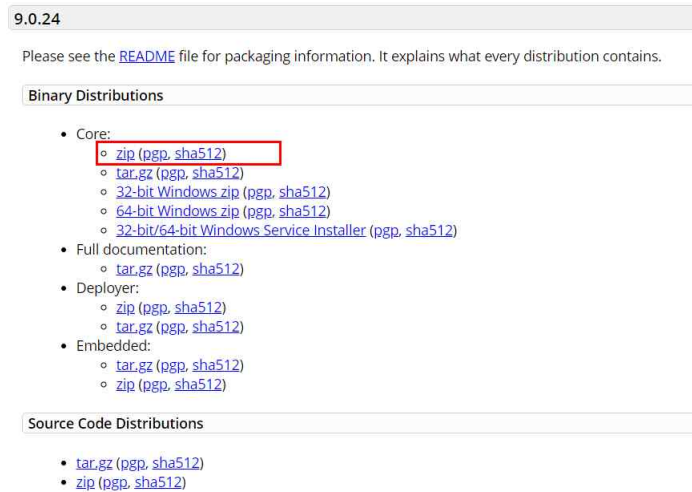


② Tomcat 설치

- 우선 <http://tomcat.apache.org>에 접속한다. 왼쪽의 [Download] 아래에 Tomcat 9를 클릭한다.



- 접속된 페이지에서 아래로 조금 내려 보면 아래와 같은 화면을 볼 수 있다. 여기에서 우리는 Tomcat 배포버전 중에서 Zip으로 압축된 버전을 다운 받도록 한다.



- 다운로드가 완료되면 다운받은 apache-tomcat-9.0.24.zip 파일을 C:\W 루트에 압축해제 하도록 한다. 그리고 압축 해제된 apache-tomcat-9.0.24 폴더의 이름을 Tomcat 으로 변경하도록 한다.



- C:\W\Tomcat\conf\context.xml 파일을 열어서 아래와 같이 Context 태그에 privileged= "true" 라는 속성 값을 추가해 주도록 하자.

```
<!-- The contents of this file will be loaded for each web application -->
<Context privileged="true">

    <!-- Default set of monitored resources. If one of these changes, the    -->
    <!-- web application will be reloaded.                                -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->
</Context>
```

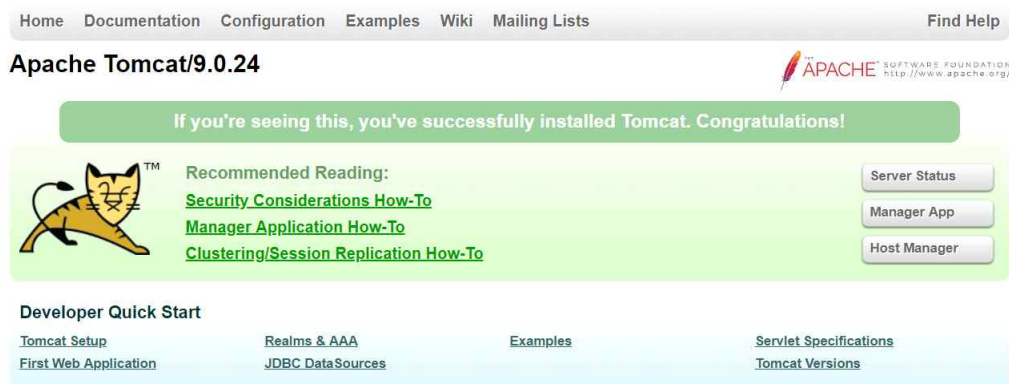
- 다음으로 Tomcat의 Encoding 설정을 UTF-8로 바꾸기 위하여 C:\W\Tomcat\conf\server.xml을 열어서 중간쯤에 있는 Connector port= "8080" 태그에 아래와 같이 URLEncoding= "UTF-8" 이라고 추가하자.

```
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" URLEncoding="UTF-8"/>
<!-- A "Connector" using the shared thread pool-->
```

- 이로써 Tomcat의 모든 설치를 마쳤다. C:\W\Tomcat\bin\startup.bat 파일을 더블클릭하여 오류 없이 실행이 되는지 확인하자. 만약 Windows 경고창이 나온다면 아래의 그림과 같이 [엑세스 허용]을 하도록 한다. 실행이 성공적으로 되었다면 도스창이 화면에 떠 있게 될 것이다.

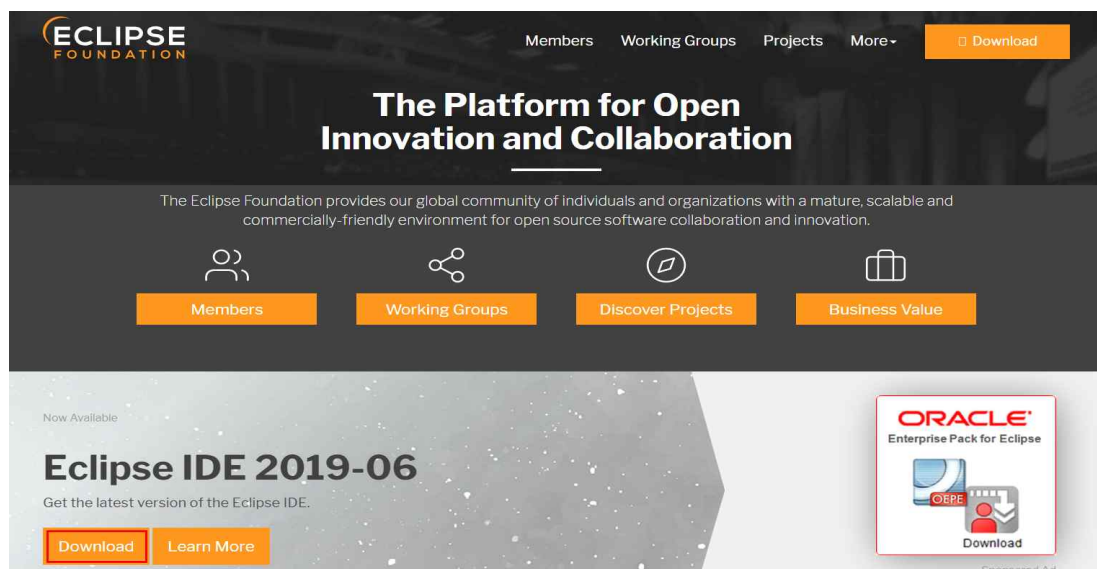


- 이제 Browser를 실행하여 `http://localhost:8080` 이라고 입력해 보자. 아래와 같은 Tomcat 실행화면이 나오면 성공이다. 성공적으로 실행된 것을 확인하였다면 `C:\Tomcat\Wbin\Wshutdown.bat` 파일을 더블클릭하여 Tomcat을 종료하도록 한다.

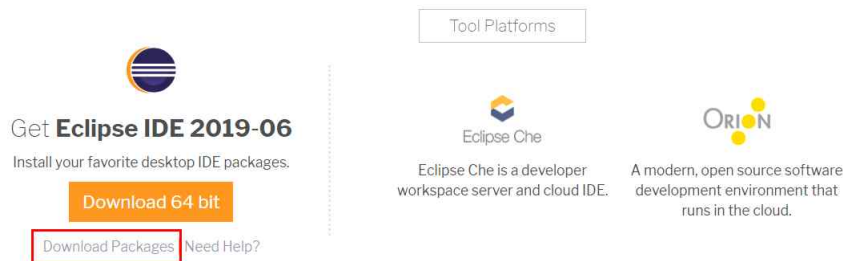


③ Eclipse 설치

- `http://www.eclipse.org`에 접속한다. 아래와 같은 화면이 나오면 오른쪽 위의 [Download] 또는 왼쪽 아래의 [Download] 버튼을 클릭하도록 한다.



- 접속된 페이지에서 조금 아래에 보면 아래와 같은 다운로드 링크가 있을 것이다. 이때 주의할 점은 바로 [Download 64 bit] 버튼을 클릭하지 말고 그 아래에 있는 [Download Packages]를 클릭하도록 하자.



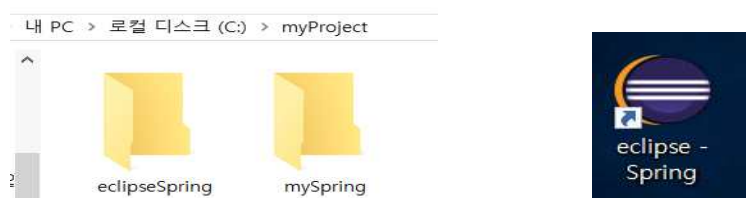
- 이어지는 화면에서 다양한 Eclipse의 종류들을 볼 수 있을 것이다. 그 중에 아래와 그림과 같이 [Eclipse IDE for Enterprise Java Developers] 버전을 찾아서 우측의 [Windows 64 bit] 버튼을 클릭하자.



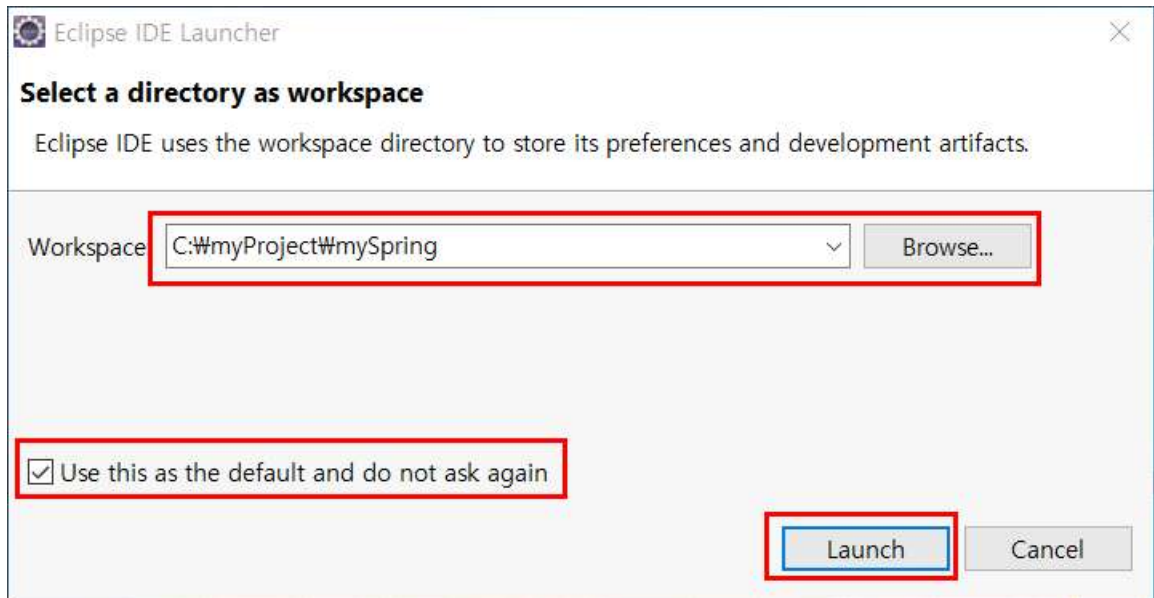
- 아래와 같은 화면이 나오면 [Download] 버튼을 클릭하여 다운받으면 된다. 만약 접속이 너무 늦다면 아래의 [>>Select Another Mirror] 부분을 클릭하여 직접 다운받을 서버를 선택할 수도 있다.



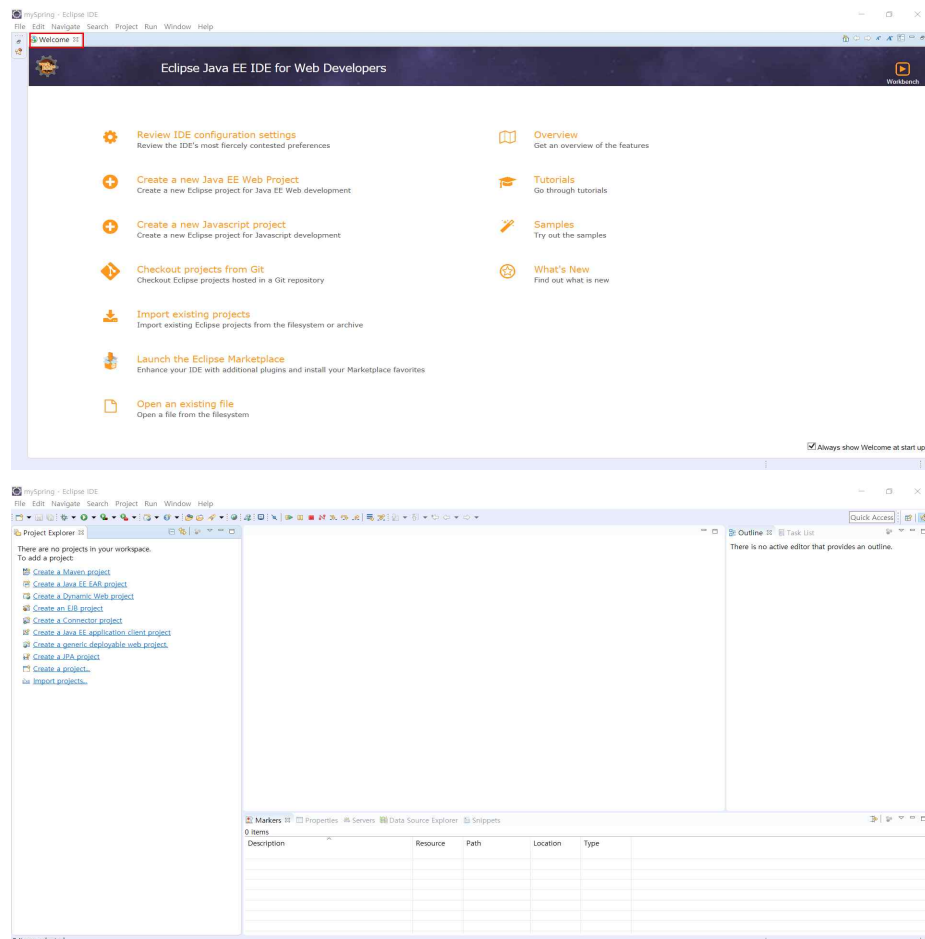
- 다운로드 받은 eclipse-jee-2019-06-R-win32-x86_64.zip 파일을 C:\WmyProject 폴더에 압축을 해제하도록 한다. 압축 해제 후 eclipse 폴더 이름을 eclipseSpring 이라고 변경하도록 하자. 그리고 C:\WmyProjectWmySpring 이라는 폴더를 생성하도록 하자. 그리고 이제 eclipseSpring 폴더 안에 eclipse.exe를 우 클릭하여 [보내기]-[바탕화면에 바로가기 만들기]를 수행하여 바탕화면에 실행 단축아이콘을 만들도록 한다. 그리고 이 아이콘의 이름을 eclipse-Spring 이라고 변경하도록 한다.



- 자 이제 실행아이콘을 더블클릭하여 Eclipse를 실행해 보자. 아래와 같은 화면이 나오면 [Browse...] 버튼을 클릭하여 이전에 우리가 만들었던 C:\WmyProject\WmySpring 폴더와 연결해 주도록 한다. 그리고 아래의 Use this as the default and do not ask again 부분을 체크하여 다음부터는 이러한 창이 뜨지 않도록 한다.

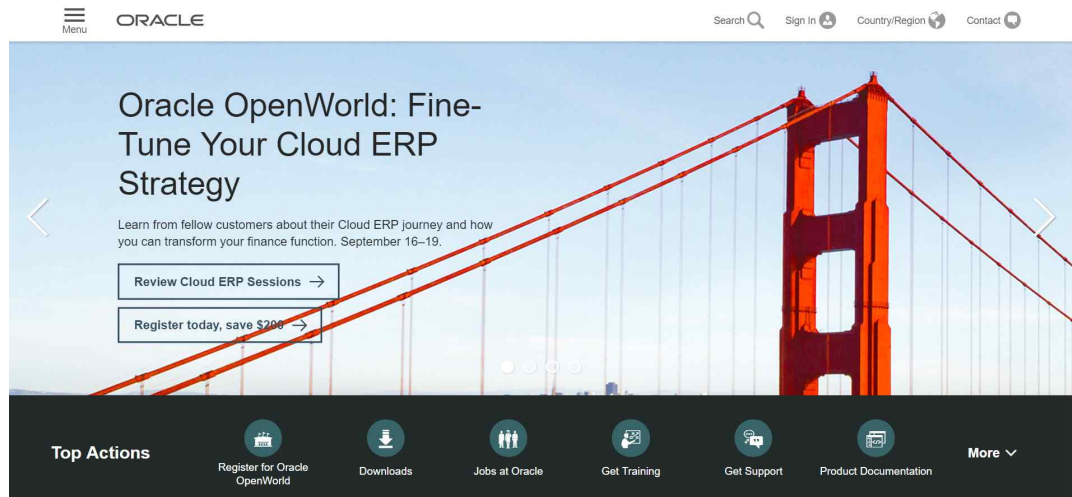


- 아래와 같은 Welcome 화면이 나오면 성공적으로 설치된 것이다. 왼쪽위의 Welcome 창의 X버튼을 클릭해서 닫으면 Eclipse 초기화면을 볼 수 있다.

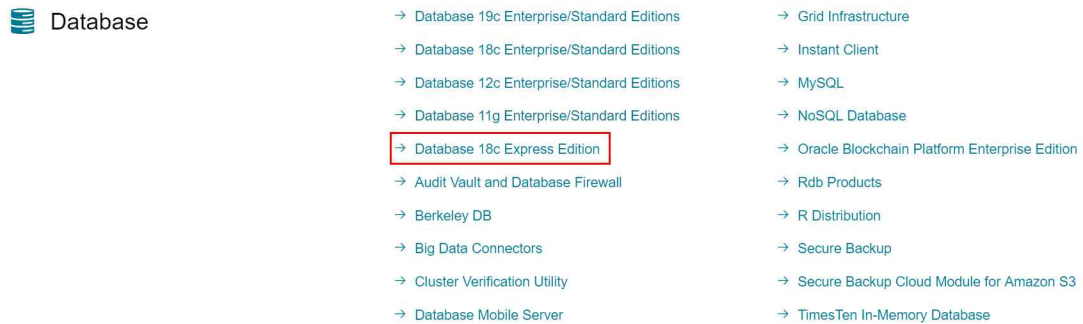


④ Database 설치 : Oracle 18c Express Edition

■ <http://www.oracle.com>에 접속하도록 하자.



■ 위와 같은 화면이 나오면 아래의 [Download] 버튼을 클릭하도록 하자. 스크롤을 내려서 중간의 Database 부분의 아래 그림과 같이 Database 18c Express Edition 부분을 클릭하도록 하자.





■ 이어지는 화면에서 본인의 운영체제에 맞는 버전을 선택하여 다운로드 받으면 된다. 우리는 윈도우 환경에서 사용할 것이므로 Oracle Database 18c Express Edition for Windows x64 버전을 다운로드 하겠다.

Database /
Oracle Database Express Edition (XE) Downloads

Oracle Database XE Downloads

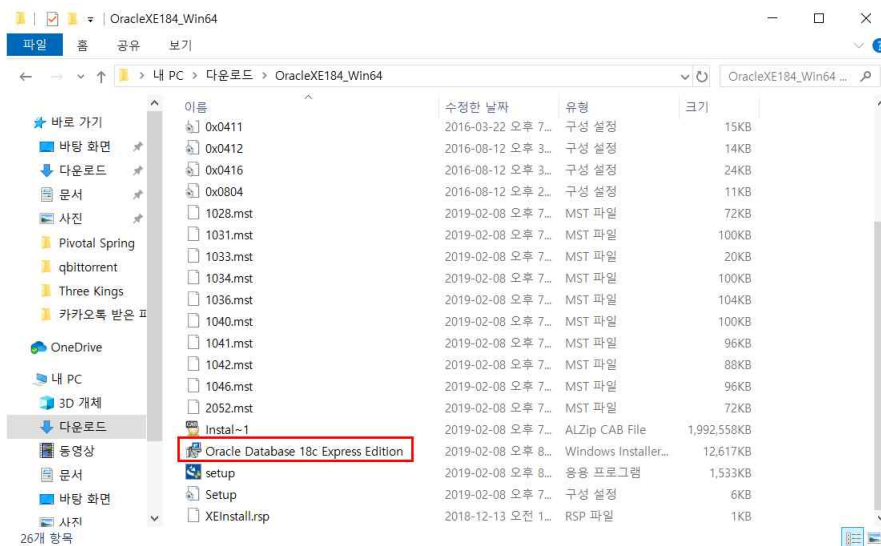
Oracle Database Express Edition (XE) Release 18.4.0.0.0 (18c)

Download	Description
 Oracle Database 18c Express Edition for Linux x64	(2,574,155,124 bytes - October 19, 2018) [Sha256sum: 308c044444342b9a3a8d332c68b12c540edf933dc8162d8eda3225e662433f1b]
 Oracle Database 18c Express Edition for Windows x64	(2,054,264,100 bytes - February 20, 2019) [Sha256sum: 7C1A85D05F6FCC5BE70E7BA4017BEEC70C8D095E685031695B8486C40C114F1A]

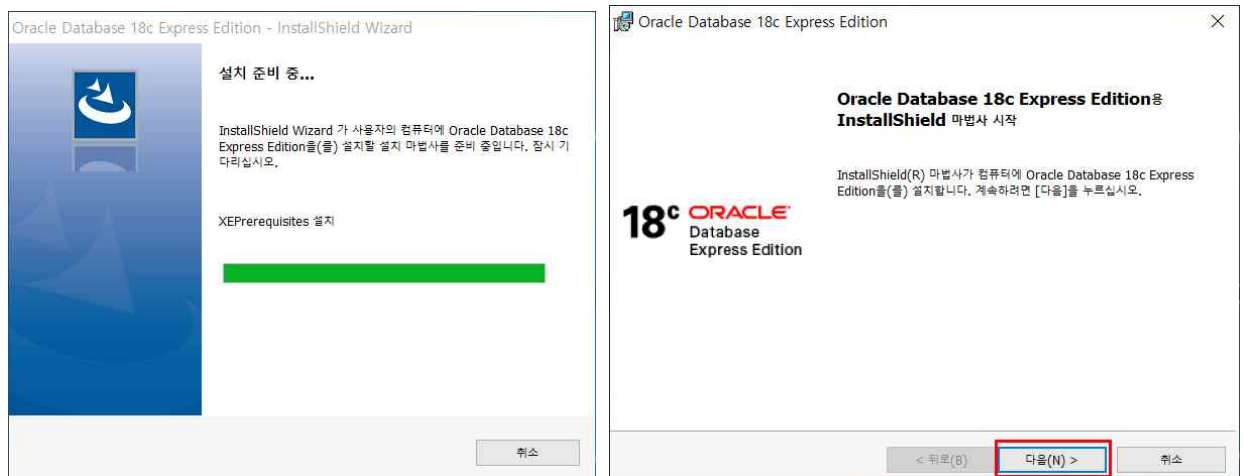
- 다운로드 페이지가 나오면 아래 그림과 같이 라이선스에 동의하고 다운받도록 한다.



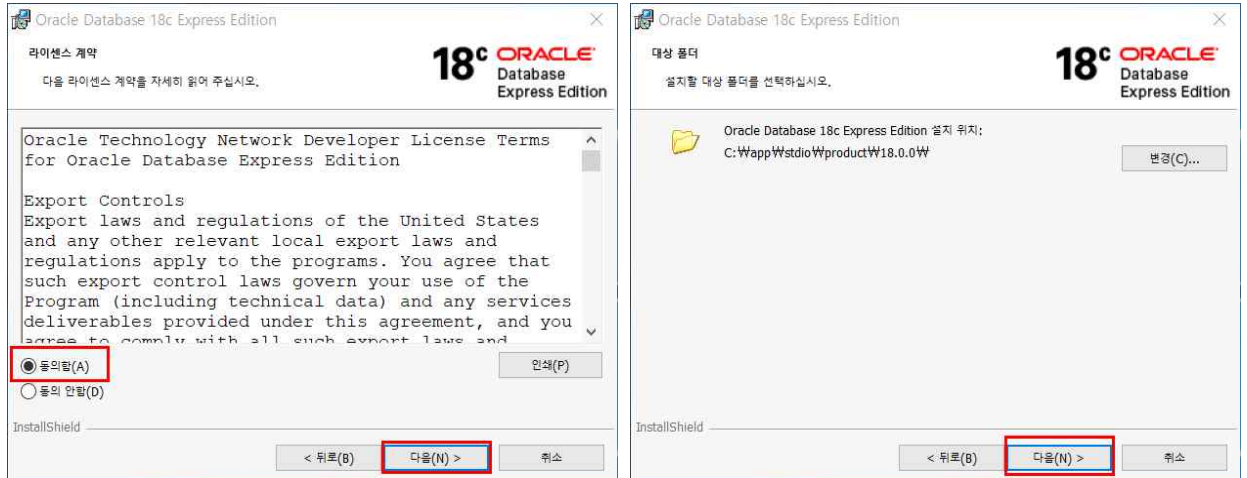
- 다운받은 OracleXE184_Win64.zip 파일을 압축을 해제하도록 하자. 압축 해제된 폴더 안에 Oracle Database 18c Express Edition.exe 파일을 더블클릭하여 설치를 진행하도록 한다.



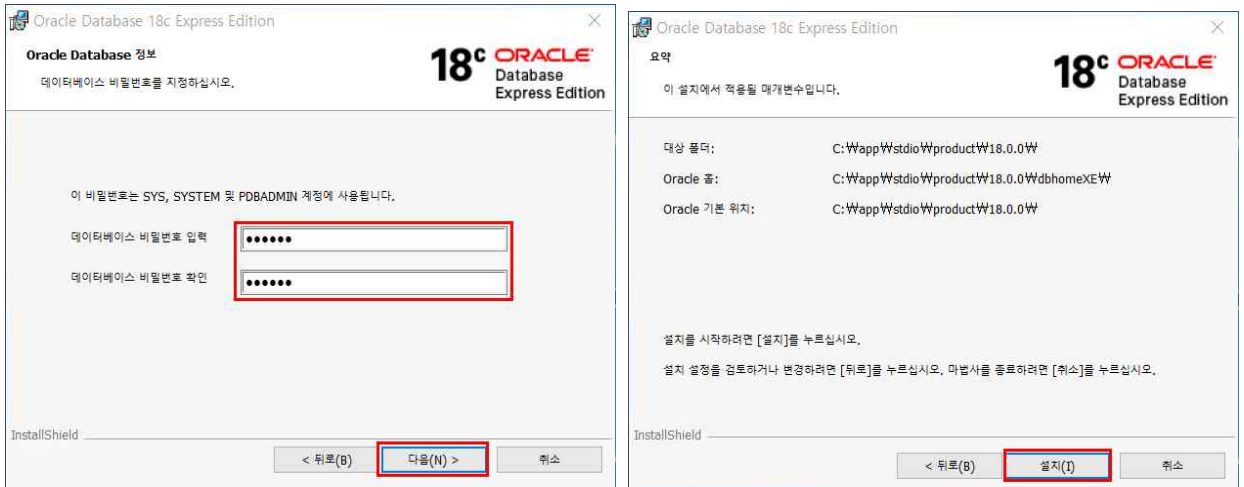
- 아래의 설치 진행 화면에 따라 설치를 진행하도록 하자. 설치 진행시 컴퓨터의 성능에 따라 시간이 지체될 수 있으니 인내심을 가지고 천천히 진행하도록 한다.



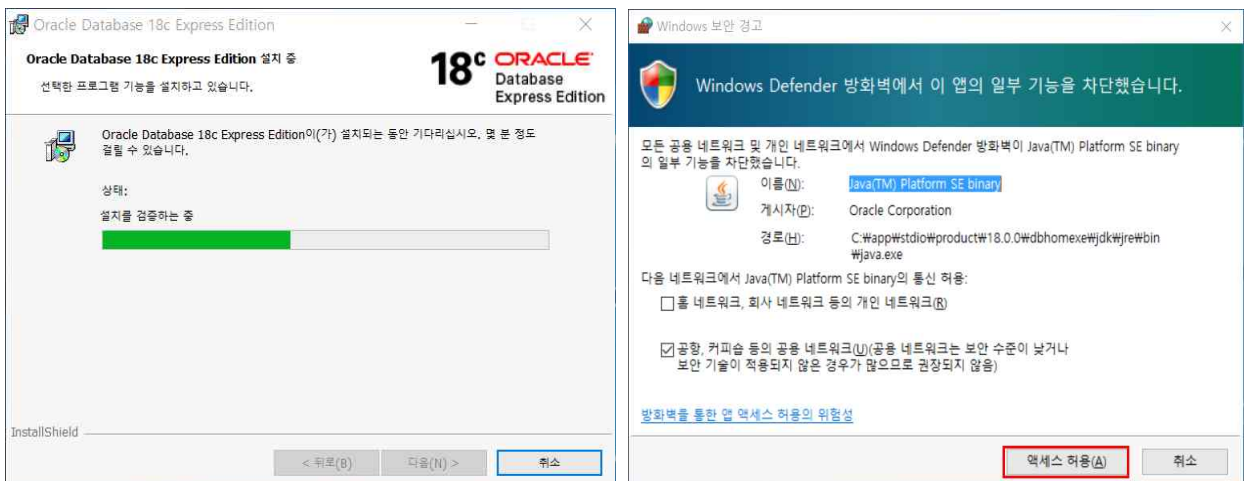
■ 사용권 계약서가 나오면 동의를 클릭하고 진행하도록 한다.



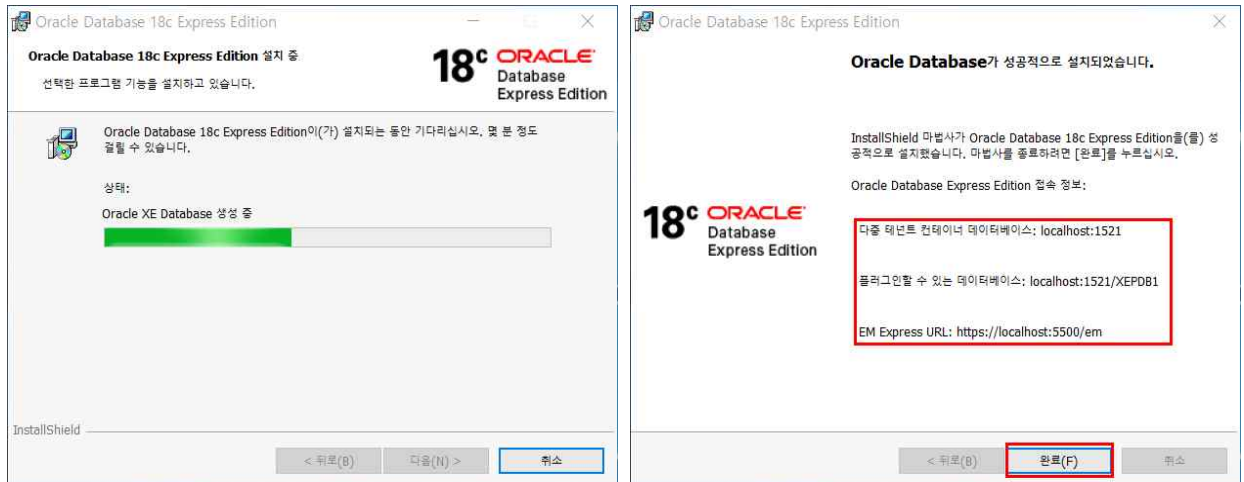
■ 아래 그림과 같이 관리자 비밀번호를 묻는 화면이 나오면 비밀번호를 입력하고 진행하도록 한다.



■ 설치 진행 중 Windows 보안 경고 창이 나오면 [엑세스 허용] 버튼을 클릭하도록 한다.



- 계속 진행이 하다 보면 아래 그림과 같이 설치가 완료된 것을 볼 수 있다. 이때 완료화면의 설치 정보를 잘 기억해 두도록 하자.



- 설치가 완료되었으면 오라클의 기본적인 환경설정을 하도록 하자. 우선 hr 계정을 사용할 수 있도록 하겠다. sqlplus를 실행하고 관리자 계정인 system 계정으로 로그인하여 아래 화면과 같이 진행하도록 하자.

```
C:\WINDOWS\system32\cmd.exe - sqlplus

SQL*Plus: Release 18.0.0.0.0 - Production on 토 8월 31 07:14:26 2019
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

사용자명 입력: system
비밀번호 입력:
마지막 성공한 로그인 시간: 토 8월 31 2019 07:13:45 +09:00

다음에 접속됨:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> alter session set container=XEPDB1;

세션이 변경되었습니다.

SQL> alter user hr account unlock;

사용자가 변경되었습니다.

SQL> alter user hr identified by hr;

사용자가 변경되었습니다.

SQL> grant create session to hr;

권한이 부여되었습니다.
```


- 이제 hr 계정으로 로그인할 수 있을 것이다. 아래 그림과 같이 우선 system 계정으로 로그인한 후에 hr 계정으로 로그인을 수행하여 보자. 성공적이라면 설정이 제대로 이루어진 것이다.

```
C:\WINDOWS\system32\cmd.exe - sqlplus
Microsoft Windows [Version 10.0.17763.678]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Wstadio>sqlplus

SQL*Plus: Release 18.0.0.0.0 - Production on 토 8월 31 07:18:01 2019
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

사용자명 입력: system
비밀번호 입력:
마지막 성공한 로그인 시간: 토 8월 31 2019 07:14:30 +09:00

다음에 접속됨:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> connect hr/hr@localhost:1521/XEPDB1;
연결되었습니다.
SQL>
```

- 앞으로 우리가 사용할 mytest라는 계정을 생성하도록 하자. 아래의 그림과 같이 진행하면 된다.

```
C:\WINDOWS\system32\cmd.exe - sqlplus

사용자명 입력: system
비밀번호 입력:
마지막 성공한 로그인 시간: 토 8월 31 2019 07:18:04 +09:00

다음에 접속됨:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> alter session set container=XEPDB1;
세션이 변경되었습니다.

SQL> create user mytest identified by mytest;
사용자가 생성되었습니다.

SQL> grant create session to mytest;
권한이 부여되었습니다.

SQL>
```

- 위의 화면처럼 권한부여에 성공하였다면 mytest로 로그인을 수행할 수 있을 것이다.

```
C:\WINDOWS\system32\cmd.exe - sqlplus
Microsoft Windows [Version 10.0.17763.678]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Wstadio>sqlplus

SQL*Plus: Release 18.0.0.0.0 - Production on 토 8월 31 07:22:54 2019
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

사용자명 입력: system
비밀번호 입력:
마지막 성공한 로그인 시간: 토 8월 31 2019 07:20:52 +09:00

다음에 접속됨:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> connect mytest/mytest@localhost:1521/XEPDB1;
연결되었습니다.
SQL> select * from tab;

선택된 레코드가 없습니다.

SQL>
```

- 그러나 테이블을 생성하려고 하면 권한이 없다고 나온다. 따라서 아래 그림과 같이 추가적으로 권한을 부여하도록 하자.

```
C:\WINDOWS\system32\cmd.exe - sqlplus
Microsoft Windows [Version 10.0.17763.678]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Wstudio>sqlplus

SQL*Plus: Release 18.0.0.0.0 - Production on 일 9월 1 01:54:56 2019
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

사용자명 입력: system
비밀번호 입력:
마지막 성공한 로그인 시간: 일 9월 01 2019 01:53:26 +09:00

다음에 접속됨:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> alter session set container=XEPDB1;

세션이 변경되었습니다.

SQL> grant connect, resource to mytest;

권한이 부여되었습니다.

SQL>
```

- 위 작업까지 성공하였다면 테이블을 생성할 수 있게 될 것이다. 그러나 여기서 insert 작업을 수행하려고 하면 USERS 테이블 스페이스에 대한 권한이 없다고 에러가 나온다. 따라서 아래와 같이 추가적인 권한을 부여하도록 한다.

```
C:\WINDOWS\system32\cmd.exe - sqlplus
Microsoft Windows [Version 10.0.17763.678]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Wstudio>sqlplus

SQL*Plus: Release 18.0.0.0.0 - Production on 일 9월 1 02:01:27 2019
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

사용자명 입력: system
비밀번호 입력:
마지막 성공한 로그인 시간: 일 9월 01 2019 01:59:51 +09:00

다음에 접속됨:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> alter session set container=XEPDB1;

세션이 변경되었습니다.

SQL> alter user mytest default tablespace users quota unlimited on users;

사용자가 변경되었습니다.

SQL>
```

⑤ SQL Developer 설치

- 오라클을 성공적으로 설치하였다 하여도 도스 창에서 작업을 수행하기에는 무리가 따른다. 따라서 편리한 작업을 위하여 클라이언트 도구를 추가로 설치하도록 하겠다. oracle 홈페이지에 접속하여 [Download] 버튼을 클릭하여 아래와 같이 [Developer Tools]에 SQL Developer를 클릭하도록 하자.

Developer Tools

- ADF Faces
- Application Express Standalone
- BI Publisher
- BPEL Process Manager
- Developer Suite 10g
- Developer Tools for Visual Studio
- Enterprise Pack for Eclipse
- Forms & Reports Services
- JDeveloper & ADF
- NetBeans IDE
- SOA Suite
- Developer Studio
- **SQL Developer**
- SQL Developer Data Modeler
- StorageTek
- Team Productivity Center
- TopLink
- VM: Pre-Built VirtualBox VMs
- VM: Oracle VM Templates
- Warehouse Builder
- WebRTC Session Controller Client SDK

■ 다운로드 페이지에서 JDK가 포함되지 않은 버전을 선택하여 아래 그림과 같이 다운받도록 한다.

SQL Developer 19.2 Downloads

Version 19.2.0.206.2117 August 1, 2019

- [Release Notes](#)
- [Bugs Fixed](#)
- [Documentation](#)

Platform	Download	Notes
Windows 64-bit with JDK 8 included	Download (490 MB)	<ul style="list-style-type: none"> • MD5: aaaa291a468b73fc5c374f9213d7456d • SHA1: f250948419d3336998fb666f8ce5dcd6203431ab • Installation Notes
Windows 32-bit/64-bit	Download (410 MB)	<ul style="list-style-type: none"> • MD5: b6633f8488d6ccee6a345d59a984d63 • SHA1: 086c800487af8dcb2599036a3b56b76f8fc0394d • Installation Notes • JDK 8 or 11 required

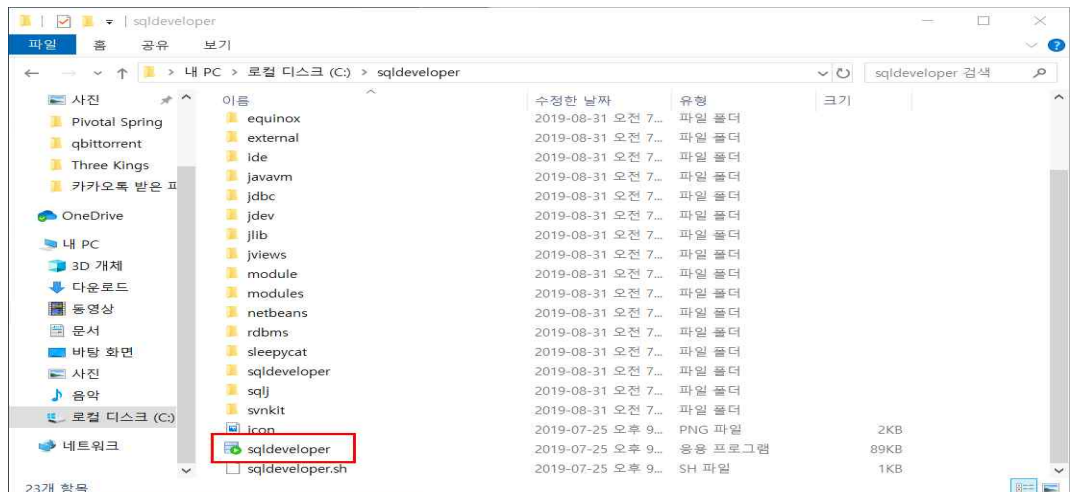
You must accept the [Oracle License Agreement](#) to download this software.

☒ I accept the Oracle License Agreement

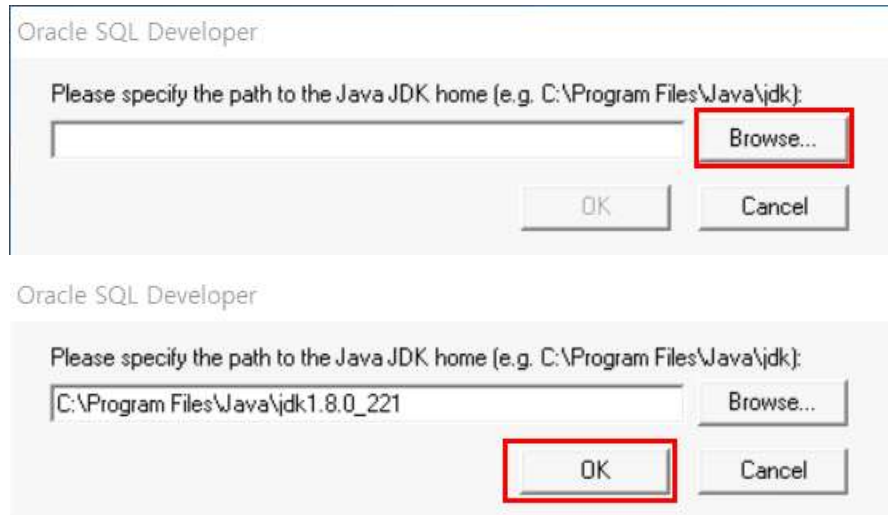
[Download sqldeveloper-19.2.0.206.2117-no-jre.zip](#)

All the content on this site is provided as samples and is not supported as a product by Oracle.

■ 다운받은 sqldeveloper-19.2.0.206.2117-no-jre.zip 파일을 아래 그림과 같이 C:\Sqldeveloper 폴더로 압축을 해제하자.



- sqldeveloper.exe 더블클릭하여 sqldeveloper를 실행하도록 한다. 아래 그림처럼 JDK 설치 위치를 물어보면 [Browse...]버튼을 클릭하여 JDK가 설치된 폴더를 지정하도록 한다.



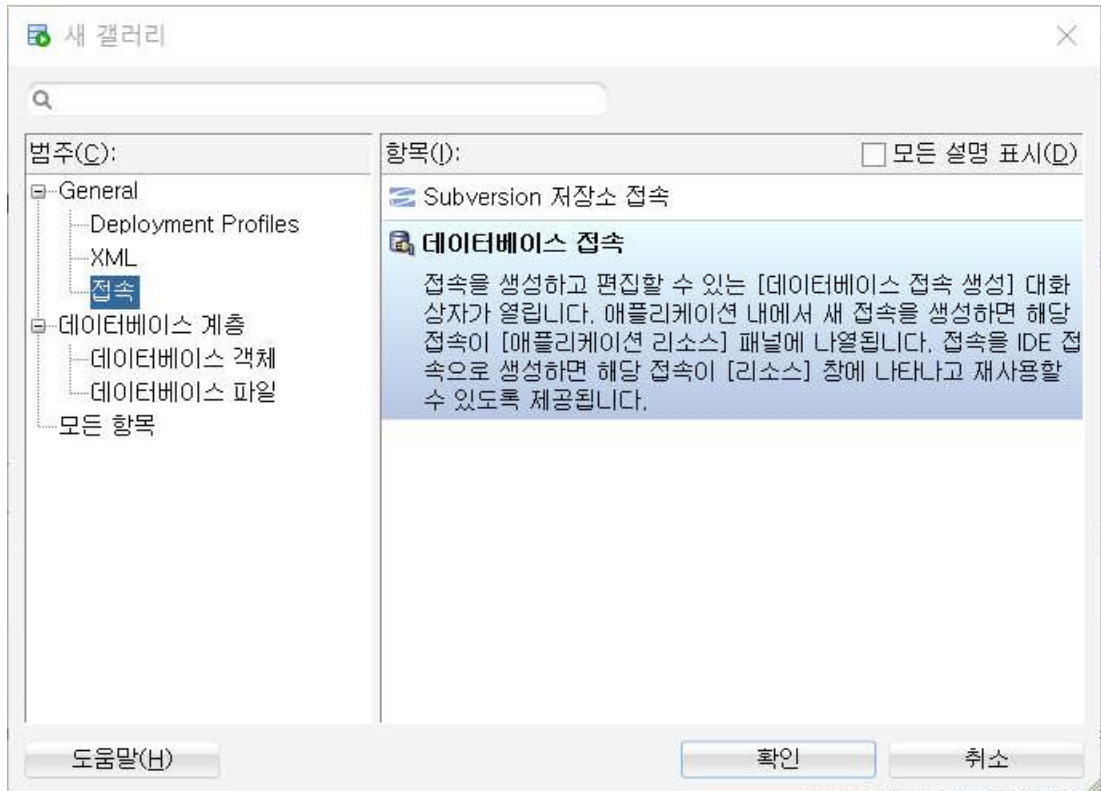
- 기존에 설치된 설정을 불러 올 것인가를 묻는 창이 뜨면 [아니오]를 선택하면 된다. 만약 이전에 sqldeveloper를 설치한 적이 있고 그 설정을 그대로 사용하고 싶다면 해당 설정을 선택하고 [예]를 클릭하면 된다.



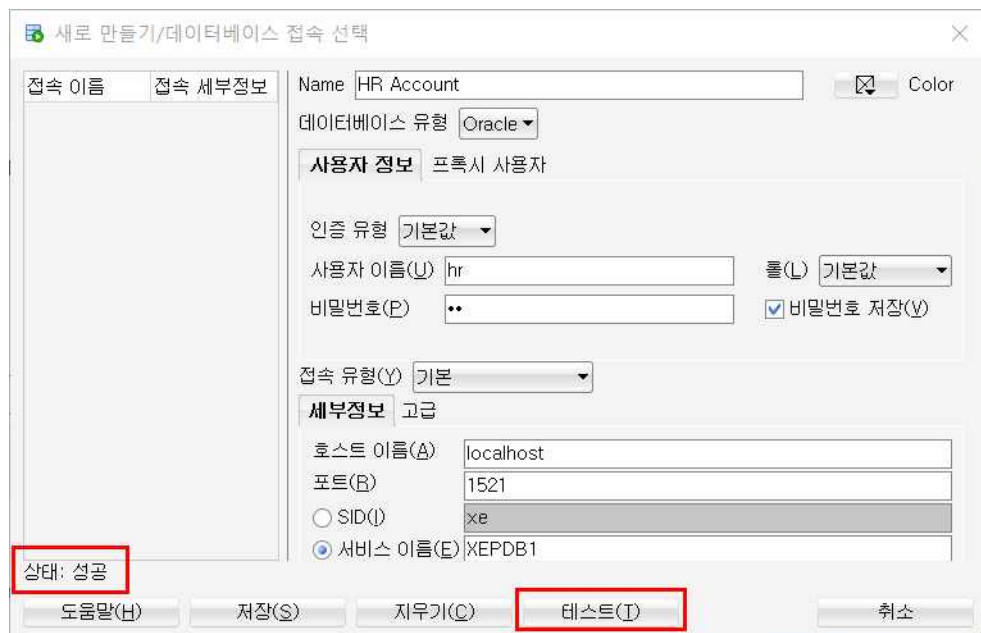
- 성공적으로 설치가 완료되었다면 아래 그림을 볼 수 있다.



- 첫 화면 왼쪽위의 [새로만들기] 아이콘을 클릭하여 아래 그림처럼 [접속]-[데이터베이스 접속]을 클릭하여 새로운 접속을 만들어 보자.



- 아래와 같은 창이 나오면 [Name] 부분에 HR Account 라고 입력을 하고 [사용자 이름]에 hr [비밀 번호]를 입력하고 아래의 [서비스 이름]을 체크하고 XEPDB1을 입력하고 [테스트] 버튼을 클릭하자. 상태부분에 성공이라는 메시지가 나오면 성공적으로 접속이 이루어진 것이다. 그렇다면 아래의 [저장] 버튼을 클릭하여 설정을 저장하도록 하자.

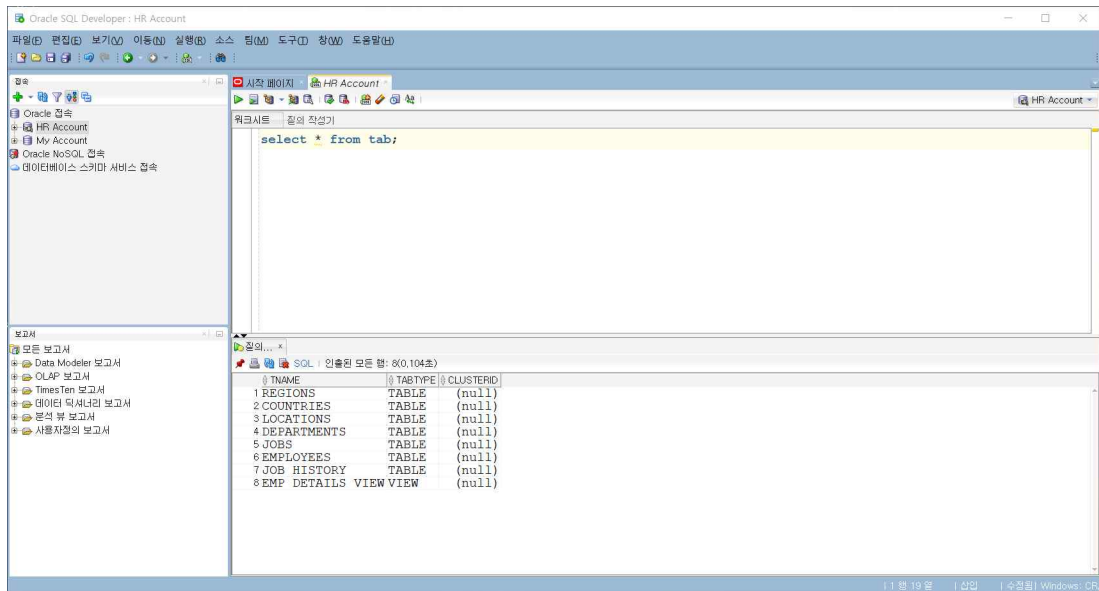


- 위와 같은 원리로 My Account 라는 이름으로 앞서 우리가 만든 mytest 계정의 접속환경도 아래 그림과 같이 설정하도록 하자.

새로 만들기/데이터베이스 접속 선택

접속 이름 HR Account	접속 세부정보 hr@//localh...	Name My Account	<input type="checkbox"/> Color
		데이터베이스 유형 Oracle	
사용자 정보 프록시 사용자			
인증 유형 기본값			
사용자 이름(U) mytest		롤(L) 기본값	
비밀번호(P)		<input checked="" type="checkbox"/> 비밀번호 저장(Y)	
접속 유형(Y) 기본			
세부정보 고급			
호스트 이름(A) localhost			
포트(B) 1521			
<input type="radio"/> SID(I) xe			
<input checked="" type="radio"/> 서비스 이름(E) XEPDB1			
상태: 성공			
<input type="button" value="도움말(H)"/>		<input type="button" value="저장(S)"/>	<input type="button" value="지우기(Q)"/>
		<input type="button" value="테스트(T)"/>	<input type="button" value="취소"/>

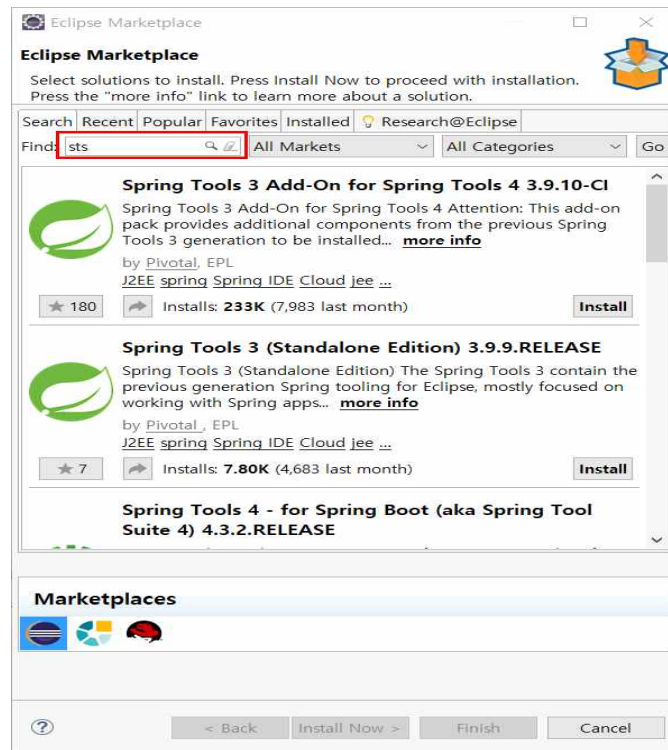
- 저장된 hr 계정으로 더블클릭하여 쿼리창에 `select * from tab;`을 실행하여 아래 그림과 같이 나오는 지 확인해 보자.



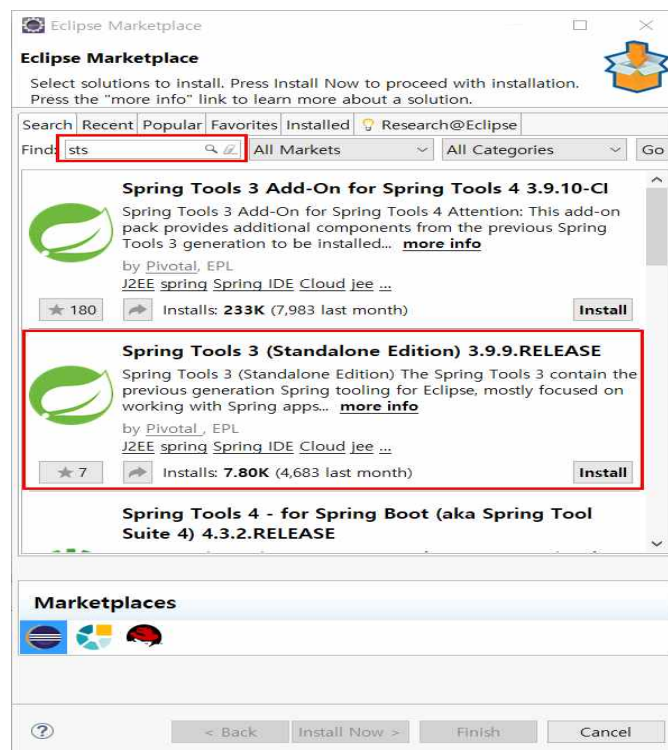
- 이로써 데이터베이스 환경설정을 마무리 하였다.

⑥ STS 플러그인 설치

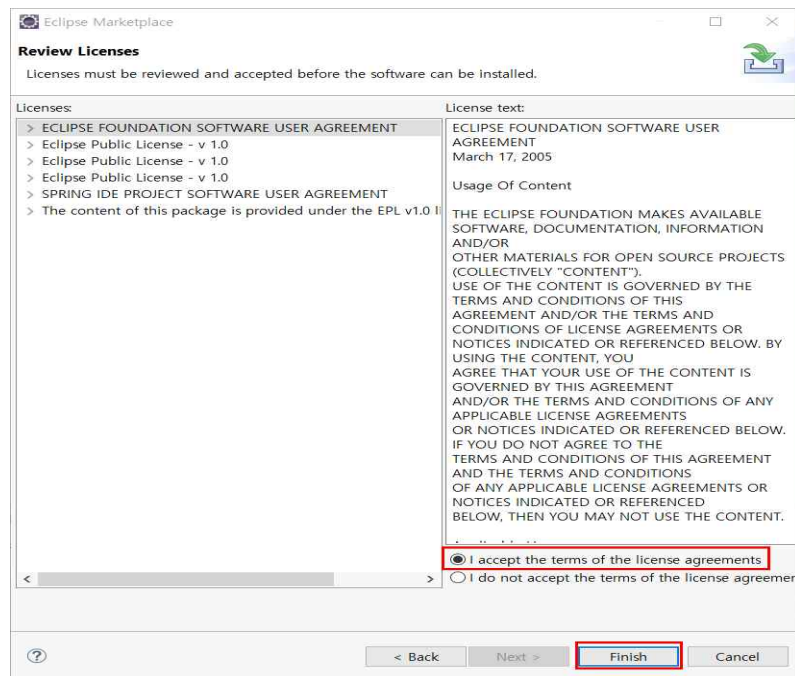
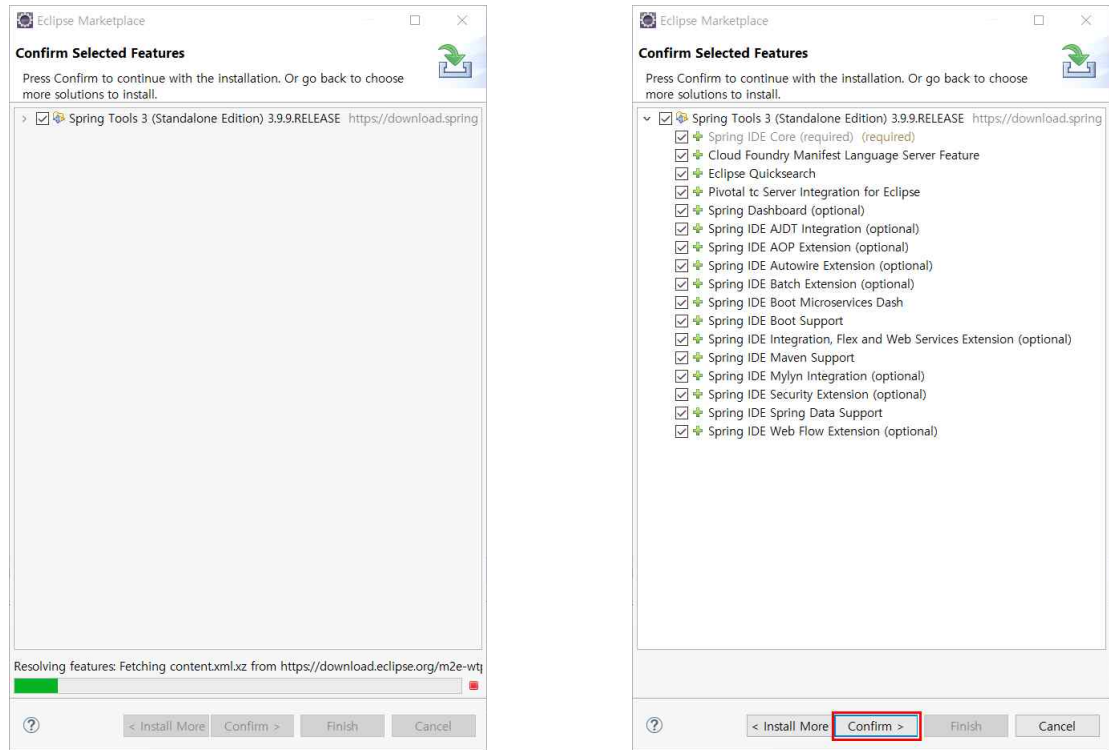
- Eclipse에서 Spring을 개발하기 위해서는 Plugin 설치가 필수적이다. 우선 Eclipse를 실행한다. 메뉴에서 [Help]-[Eclipse Marketplace...]를 선택하도록 하자. 아래와 같은 화면이 나오면 검색창에 STS를 입력하고 검색을 수행하도록 한다.



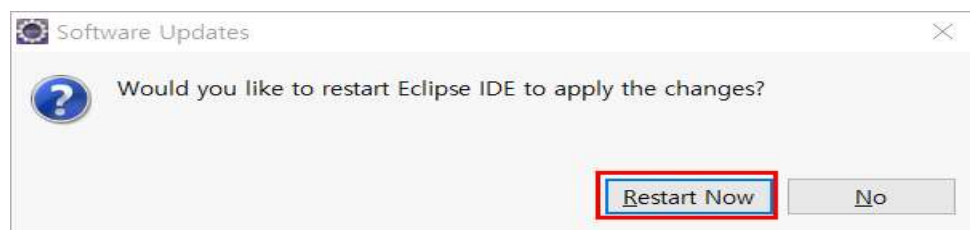
- 검색 결과 중에서 Spring Tools 3를 선택하여 설치를 진행하도록 하자.



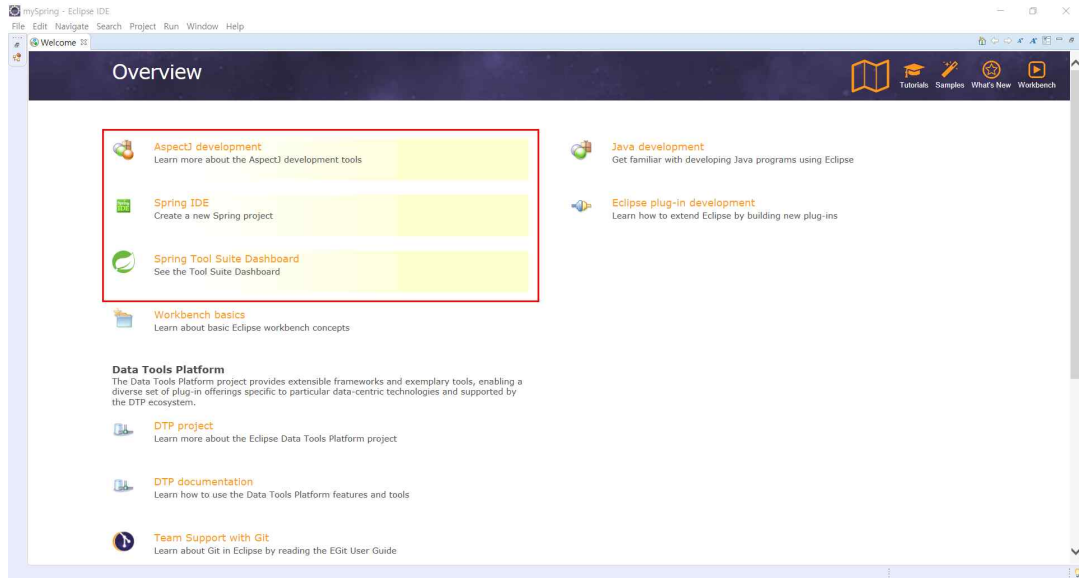
■ 아래의 그림과 같이 설치를 진행하도록 하자.



■ 설치가 완료되면 아래와 같이 재시작하라는 버튼이 나올 것이다. [Restart Now] 버튼을 클릭하자.



- Eclipse가 재시작이 되면 아래 화면처럼 Spring 관련 기능이 추가된 것을 확인할 수 있을 것이다.

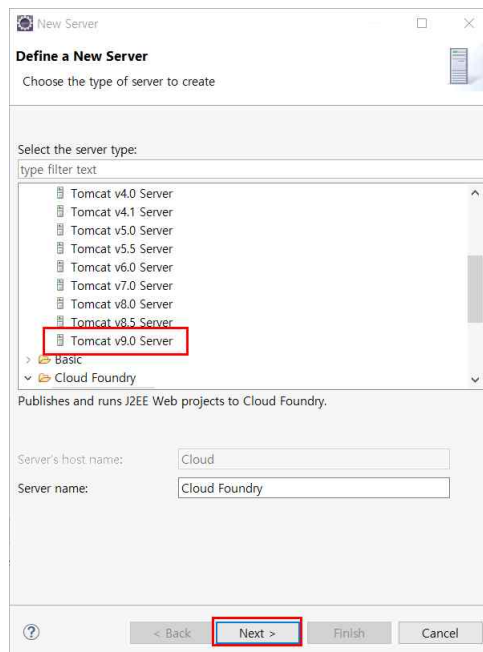


⑦ Eclipse와 Tomcat의 연동

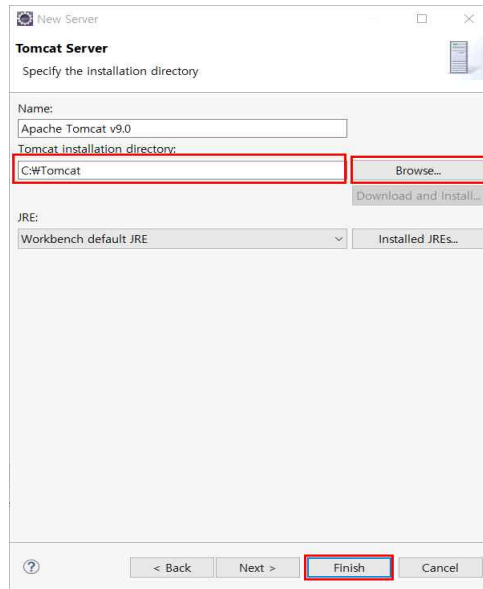
- 이클립스를 실행하여 Eclipse의 Welcome창을 닫고 아래의 Servers 탭을 클릭한 후 Click this link create new server 부분을 클릭한다.



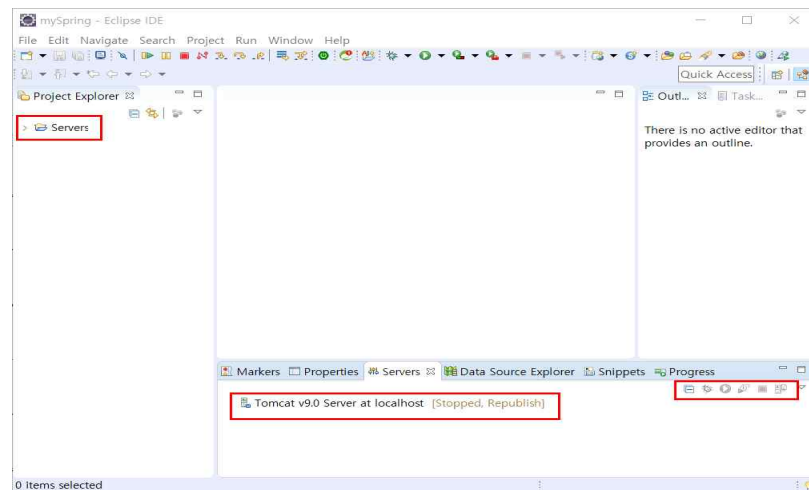
- 아래와 같은 화면이 나오면 Tomcat v9.0 Server를 선택하고 [Next] 버튼을 클릭하여 진행하도록 한다.





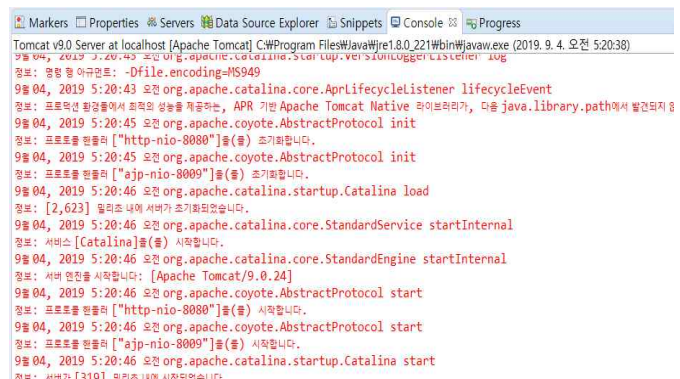
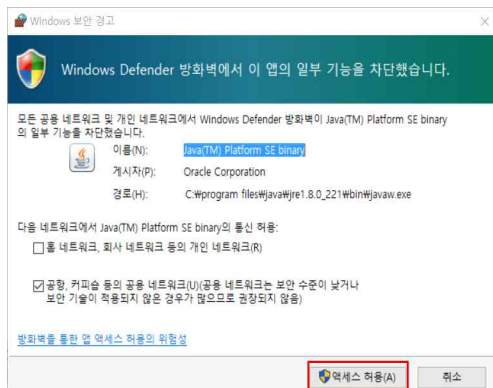
- 다음 화면에서 [Browse...]을 클릭하여 Tomcat을 설치한 폴더(C:\WTomcat)를 선택하여 지정하여 주고 아래의 [Finish] 버튼을 클릭하여 마무리한다.



■ 진행이 완료되면 아래와 같이 새로운 Server가 생긴 것을 확인할 수 있다.



■ 연동이 완료되었다면 Servers 탭의 우측의  버튼을 클릭하여 서버를 실행해 보자. 실행 시 Windows 경고 창이 나오면 [엑세스 허용]을 클릭한다. Console 탭을 확인하여 오류가 없는지 확인하고(콘솔 창 메시지에 정보만 있어야 한다.) 오류가 없다면  버튼을 클릭하여 서버를 종료하자.



8. Spring 모듈 다운로드 및 구성정보

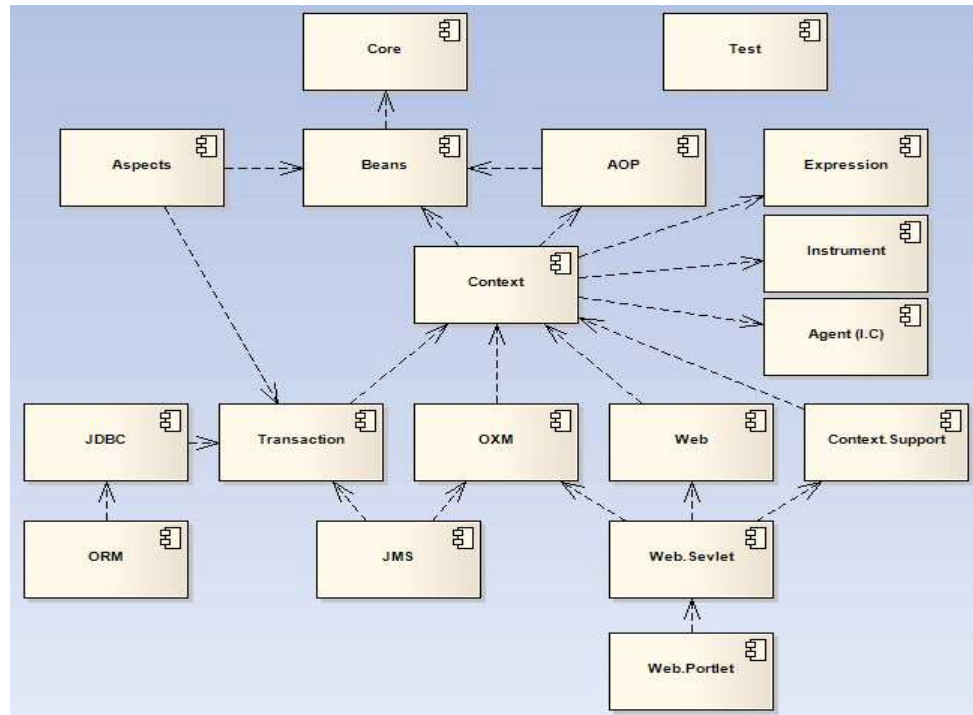
① 스프링 모듈 다운받기

- <http://repo.spring.io/release> 에 접속하여 org - springframework - spring - 5.1.9.RELEASE으로 들어가서 spring-framework-5.1.9.RELEASE-dist.zip 다운로드해서 압축을 해제하면 된다.

② 스프링 모듈의 구성

모듈명 (jar파일)	설 명
core	DI 기능을 비롯한 프레임워크의 기반을 제공한다.
beans	BeanFactory 인터페이스를 통해 구현된다.
expression	객체에 접근하고 객체를 조작하기 위한 표현 언어를 제공한다. JSP 2.1 규약에 명시된 통합 EL을 확장하였다.
context	spring-core와 spring-beans 모듈을 확장해서 국제화, 이벤트 처리, 리소스 로딩, 서블릿 컨테이너를 위한 컨텍스트 생성 등의 기능을 추가로 제공한다. ApplicationContext 인터페이스를 통해 구현된다.
context.support	cache, 메일, 스케줄링, UI의 Velocity 지원기능을 제공한다.
context.indexer	후보 구성요소의 색인 생성 기능을 지원한다. 컴파일 시간에 정적 후보 목록을 작성하여 대규모 응용 프로그램의 시작 성능을 향상시킬 수 있습니다.
aop	AOP Alliance에 호환되는 AOP 구현을 제공한다.
aspects	AspectJ와의 통합을 지원한다.
web	파일업로드, locale 처리 등 웹을 위한 통합 기능을 제공한다. 또한 원격 지원 기능 중 웹 관련 기능을 제공한다.
webmvc	스프링 MVC를 제공한다. JSP, Velocity에 대한 뷰 연동을 지원
web.flux	비동기 Non Blocking Reactive System 개발에 사용한다.
web.socket	스프링에서 WebSocket 기능을 지원한다.
jcl	로그 기능을 지원한다.
transaction : tx	AOP를 이용한 선언적 트랜잭션 관리 및 코드를 이용한 트랜잭션 관리 기능을 제공한다.
jdbc	JDBC프로그래밍을 위한 추상 레이어를 제공한다. JDBC 템플릿을 제공함으로써 간결한 코드로 JDBC프로그래밍을 할 수 있도록 지원한다.
orm	하이버네이트, JPA, iBatis, JDO등 ORM API를 위한 통합 레이어를 제공한다. 스프링이 제공하는 트랜잭션 관리와의 연동을 지원
oxm	객체와 XML 사이의 매핑을 처리하기 위한 추상 레이어를 제공한다. JAXB, Castor, XMLBeans, JiBX, XStream과의 연동을 지원
jms	JMS의 메시지를 생성하고 수신하는 기능을 제공한다.
messaging	메시지 처리 통합기능을 제공한다.
test	JUnit이나 TestNG를 이용한 스프링 컴포넌트의 테스트를 지원
instrument	instrumentation 지원 클래스를 제공

③ 스프링 모듈간의 의존 관계

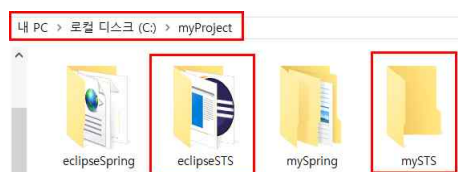


9.참고 : 스프링 프레임워크 설치 다른 방법

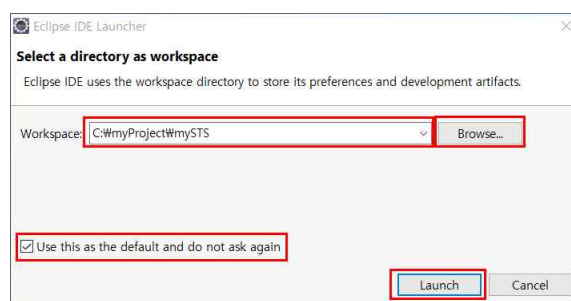
① 수동으로 플러그인 설치하기

■ 앞에서 우리는 Eclipse에 스프링 플러그인을 설치하였다. 앞의 방식대로 Eclipse Market Place에서 플러그인을 설치하는 것이 보편적인 방법이나 가끔 Market Place에서 검색이 안 되거나 하는 오류가 나오면 수동으로 플러그인을 설치하는 방법을 고려해 볼 수 있다.

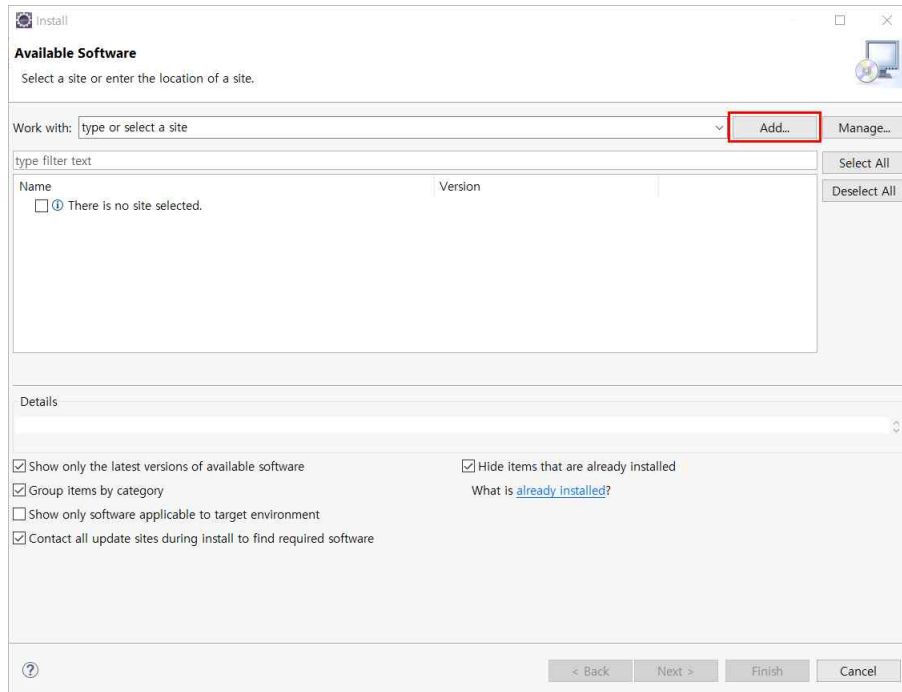
■ 우선 다운받은 eclipse-jee-2019-06-R-win32-x86_64.zip 파일을 C:\myProject 폴더에 압축을 해제하고 eclipse 폴더의 이름을 eclipseSTS로 변경한다. 이렇게 하면 기존에 설치한 eclipseSpring과 충돌 없이 사용할 수 있다. 그리고 eclipseSTS와 연동해서 사용할 C:\myProject\mySTS 폴더를 하나 생성하도록 하자.



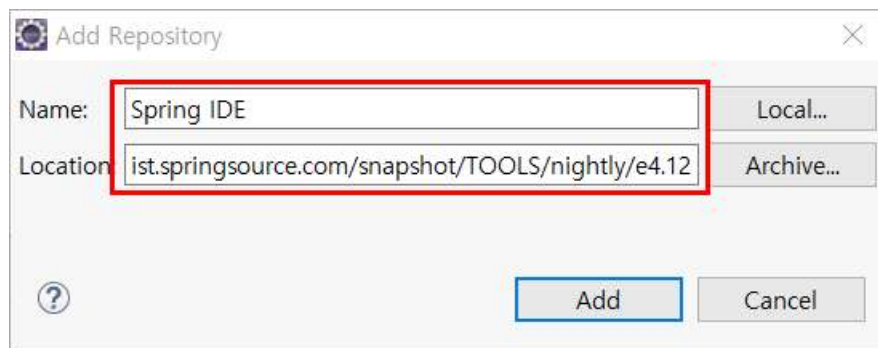
■ 위에서 새로 압축을 해제한 eclipseSTS 폴더 안에 eclipse.exe를 실행한다. (아래 참조)



- 이클립스가 실행되면 메뉴에서 [Help]-[Install New Software...]를 클릭하고 아래와 같은 화면이 나오면 [ADD] 버튼을 클릭한다.



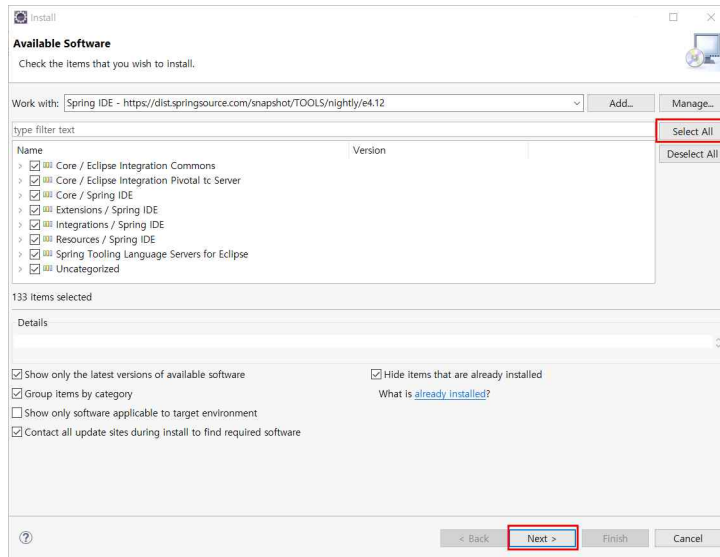
- 아래와 같은 화면이 나오면 [Name]에 Spring IDE라고 입력하고 아래 [Location] 부분에 <https://dist.springsource.com/snapshot/T00LS/nightly/e4.12> 주소를 입력한다.



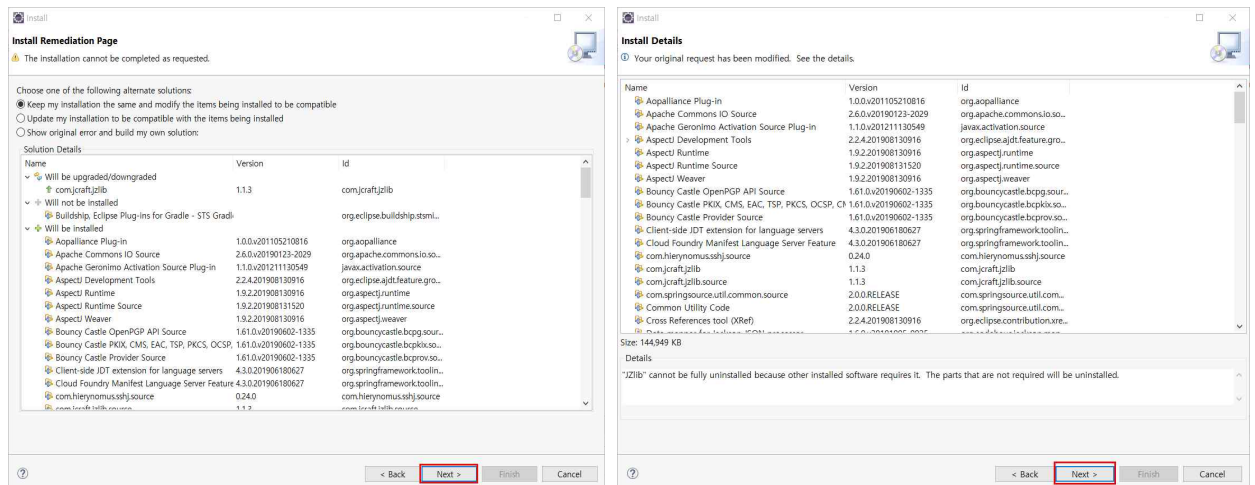
- ※ 위의 주소 입력 시 본인이 사용하는 Eclipse 버전과 맞추어야 한다. 자세한 정보는 <http://dist.springsource.com/snapshot/STS/nightly-distributions.html> 에서 확인할 수 있다.

for Eclipse 2019-09 (4.13): <https://dist.springsource.com/snapshot/T00LS/nightly/e4.13>
 for Eclipse 2019-06 (4.12): <https://dist.springsource.com/snapshot/T00LS/nightly/e4.12>
 for Eclipse Photon (4.8): <http://dist.springsource.com/snapshot/T00LS/nightly/e4.8>
 for Eclipse Oxygen (4.7): <http://dist.springsource.com/snapshot/T00LS/nightly/e4.7>
 for Eclipse Neon (4.6): <http://dist.springsource.com/snapshot/T00LS/nightly/e4.6>
 for Eclipse Mars (4.5): <http://dist.springsource.com/snapshot/T00LS/nightly/e4.5>
 for Eclipse Luna (4.4): <http://dist.springsource.com/snapshot/T00LS/nightly/e4.4>
 for Eclipse Kepler (4.3): <http://dist.springsource.com/snapshot/T00LS/nightly/e4.3>
 for Eclipse Juno (4.2): <http://dist.springsource.com/snapshot/T00LS/nightly/e4.2>
 for Eclipse Juno (3.8): <http://dist.springsource.com/snapshot/T00LS/nightly/e3.8>
 for Eclipse Indigo (3.7): <http://dist.springsource.com/snapshot/T00LS/nightly/e3.7>

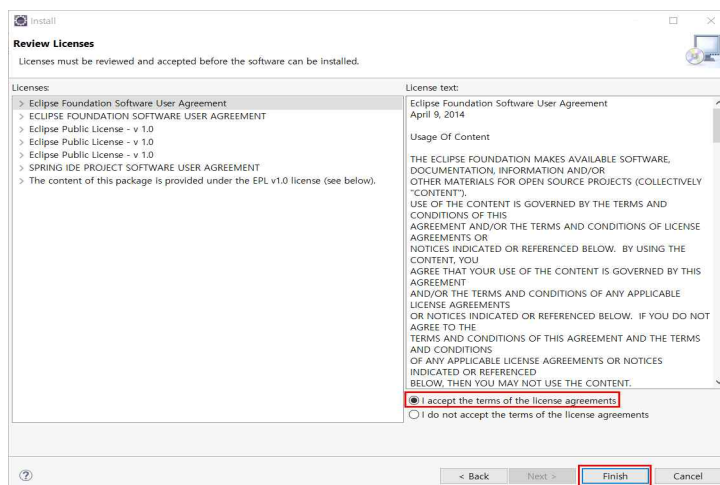
- 오타 없이 잘 입력하였다면 아래와 같은 화면이 나올 것이다. 여기서 [Select All]을 클릭하여 전체 선택을 한 후 [Next] 버튼을 클릭하여 진행하도록 한다.



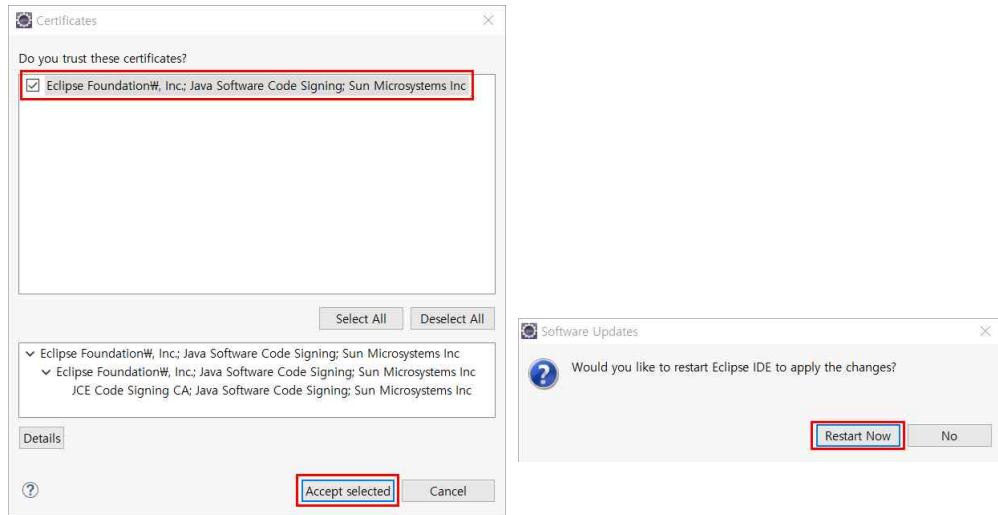
- 아래와 같은 화면이 나오면 [Next] 버튼을 클릭하여 계속 진행하도록 한다.



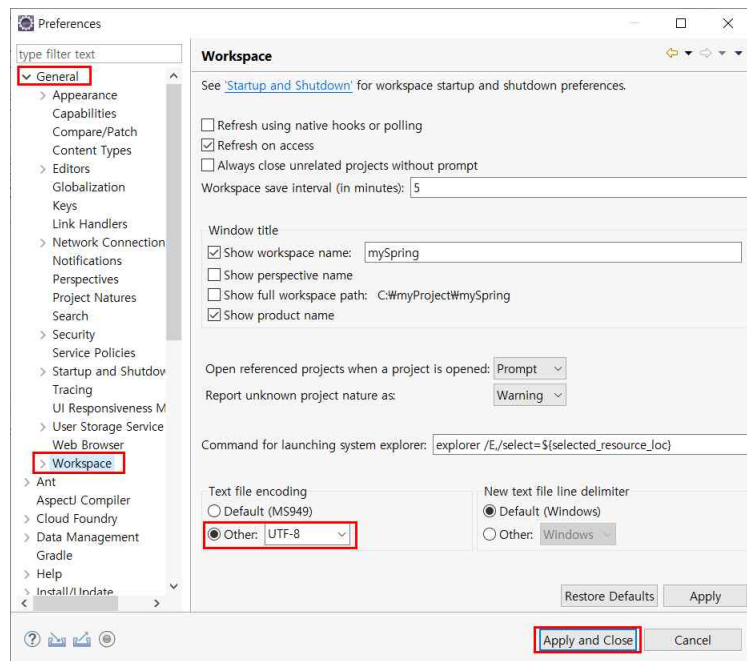
- 사용권 계약 동의서가 나오면 동의를 체크하고 [Finish] 하도록 한다.



- 진행 중에 Sun Micro Systems를 신뢰할 것인지 묻는 창이 나온다면 체크하고 진행하자. 설치가 완료되면 Eclipse를 재시작하라는 창이 나올 것이다. [Restart Now]를 클릭하여 Eclipse를 재시작하고 나면 Spring 플러그인이 설치 완료된 것을 확인할 수 있다.

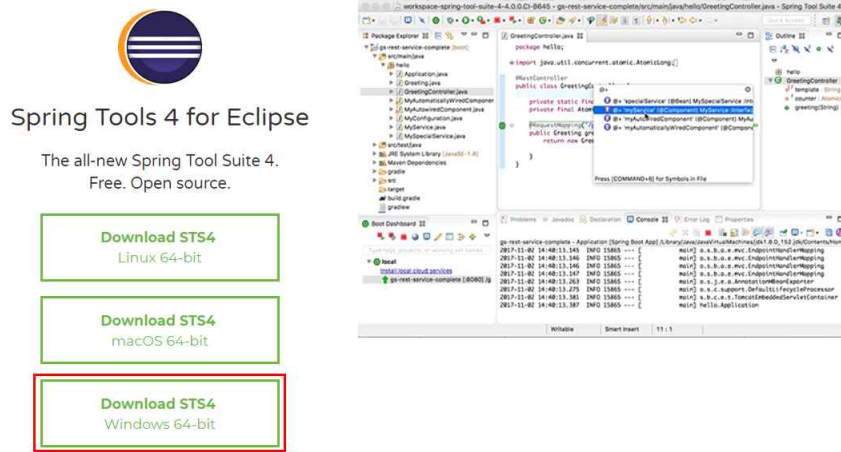


- 참고 : Eclipse의 Encoding을 UTF-8로 변경하려면 메뉴에서 [Windows]-[Preperences]를 선택하고 나오는 화면에서 [General]-[Workspace]를 선택한 후 아래에서 [Other]에서 Encoding 설정을 UTF-8로 변경해 주면된다.

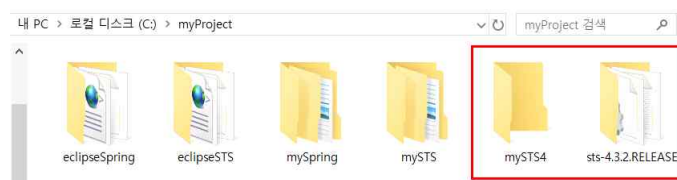


② STS 설치

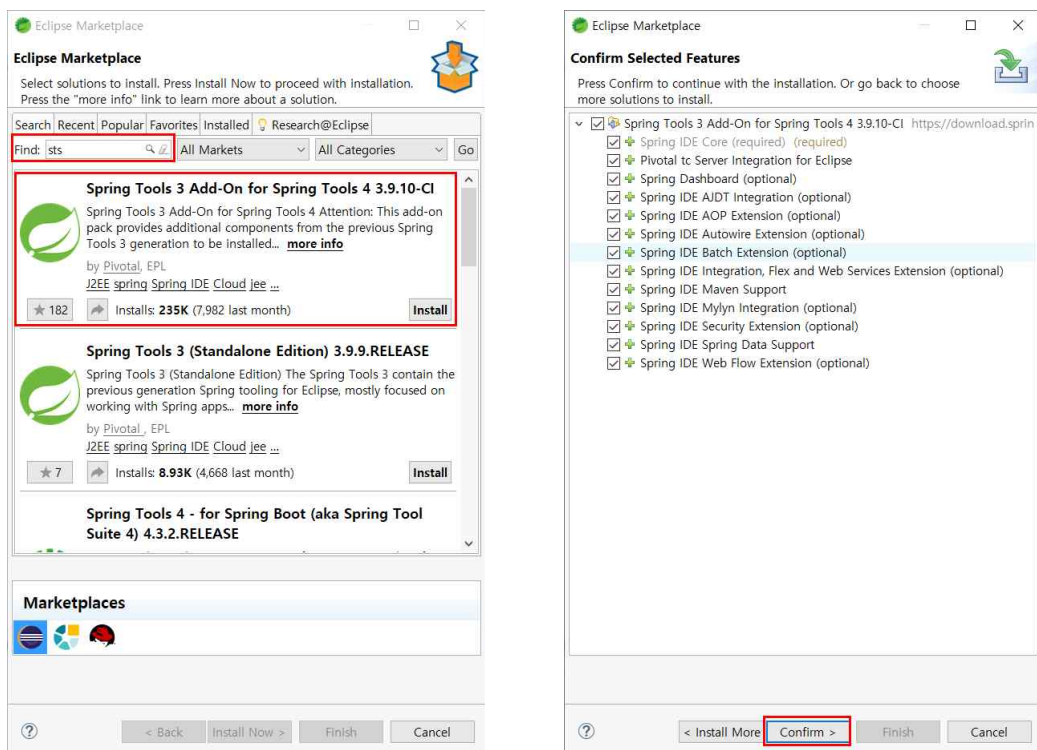
- <http://spring.io>에 접속하여 맨 아래 [Tools]를 클릭하면 아래와 같은 화면이 나온다. 여기에서 STS를 다운받아 C:\WmyProject 압축을 해제하면 설치 완료이다.

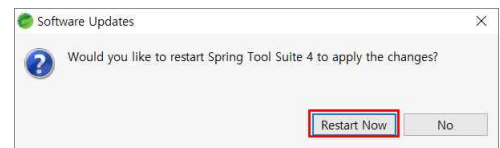
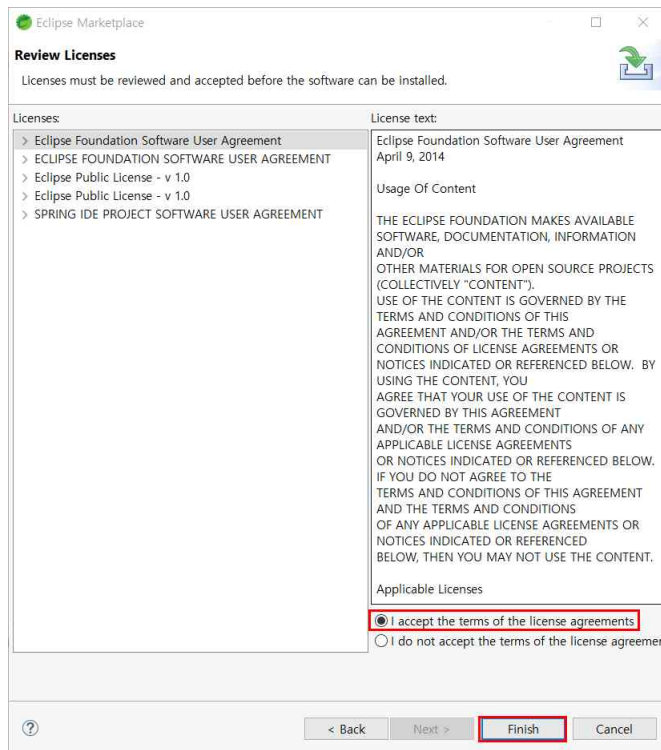


- C:\WmyProject\WmySTS4라는 폴더를 만들고 다운받은 STS와 연결하여 사용하면 된다.

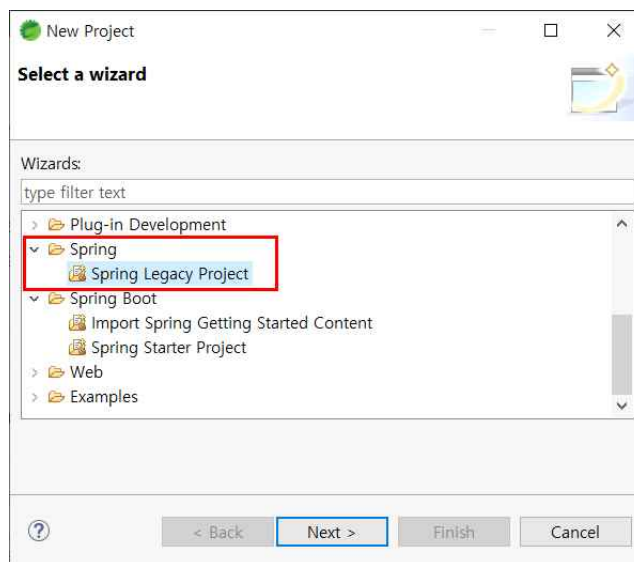


- 참고 : STS4 버전의 경우에 처음에 실행하고 [New]-[Project]에서 [Spring] 탭을 보면 Spring Legacy Project 메뉴가 존재하지 않는다. 이를 해결하려면 메뉴에서 [Help]-[Eclipse Marketplace...]를 선택하고 나오는 화면에서 STS를 검색한 후 Spring Tool 3 Add On을 설치하면 된다. (아래 진행 순서를 따라하면 된다.)





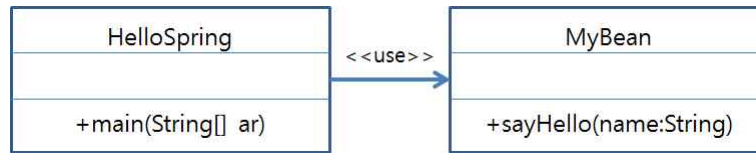
- 설치 완료 후 메뉴에서 [File]-[New]-[Project]를 선택해 보면 아래와 같이 Spring Legacy Project가 생긴 것을 확인할 수 있다.



- ※ 우리는 지금까지 다양한 스프링 설치 방법에 대해서 알아보았다. 어떠한 것이 좋고 나쁘다 보다 여러분만의 개발환경에 맞게 사용하면 된다.

10. 스프링 첫 번째 예제 따라 하기

① 제 1단계 : 일반적인 자바 애플리케이션 구조



■ Eclipse를 실행 후 [myfirst]라는 Dynamic Web Project를 생성한다.

■ MyBean 클래스 작성 : src/tommy/spring/exam01/MyBean.java

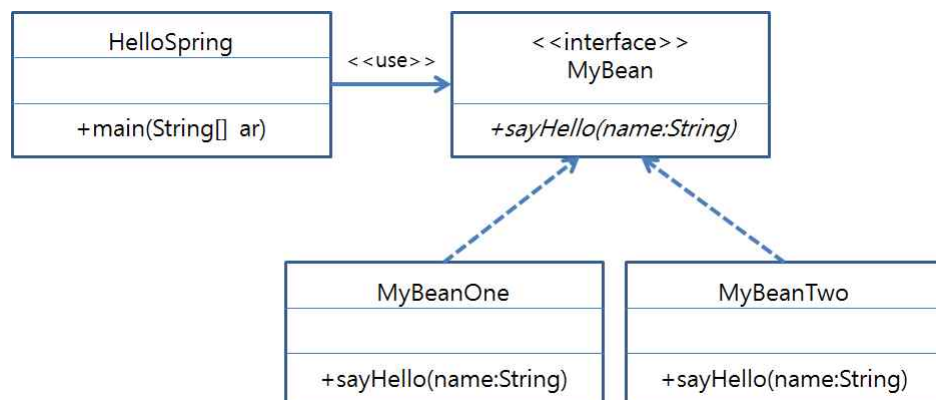
```
1 package tommy.spring.exam01;
2 public class MyBean {
3     public void sayHello(String name){
4         System.out.println("Hello, " + name + "!!!");
5     }
6 }
```

■ MyBean을 테스트할 HelloSpring 작성 : src/tommy/spring/exam01/HelloSpring.java

```
1 package tommy.spring.exam01;
2 public class HelloSpring {
3     public static void main(String[] args) {
4         MyBean bean = new MyBean();
5         bean.sayHello("Spring");
6     }
7 }
```

■ 코드 창에서 우 클릭하여 [Run As]-[Java Application]을 선택하여 실행하여 보자. 화면에 Hello Spring!!! 이라고 출력되는 것을 확인할 수 있을 것이다. 여기에서 생각해 볼 것은 이러한 구조는 우리가 일반적으로 볼 수 있는 코드이나 **결합도가 강하다. 따라서 자주 변경해야 하는 어플리케이션에 위와 같은 구조를 사용한다면 코드를 대폭 수정하는 문제점이 발생할 수 있다.**

② 제 2단계 : 인터페이스를 이용하여 의존관계를 약하게 설정하기



■ MyBean 인터페이스 작성 : src/tommy/spring/exam02/MyBean.java

```
1 package tommy.spring.exam02;
2 public interface MyBean {
3     void sayHello(String name);
4 }
```

■ 인터페이스를 구현한 클래스 작성하기 : src/tommy/spring/exam02/MyBeanOne.java

```
1 package tommy.spring.exam02;
2 public class MyBeanOne implements MyBean {
3     @Override
4     public void sayHello(String name) {
5         System.out.println("Hello, " + name + "!!!");
6     }
7 }
```

■ 테스트용 메인클래스 작성하기 : src/tommy/spring/exam02/HelloSpring.java

```
1 package tommy.spring.exam02;
2 public class HelloSpring {
3     public static void main(String[] args) {
4         MyBean bean = new MyBeanOne();
5         bean.sayHello("Spring");
6     }
7 }
```

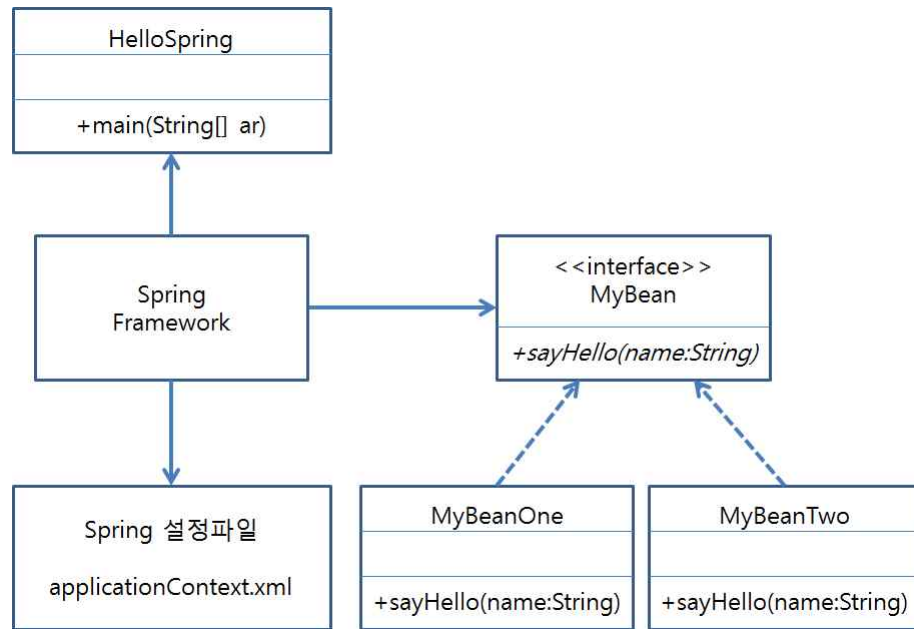
■ 이제 [Run As]-[Java Application]을 선택하여 실행하여 보자. 이전과 같이 Hello Spring!!!이 출력되는 것을 확인할 수 있다. 그런데 만약 이때 Client가 영어대신 한국어로 바꾸어 달라고 요구했다고 해 보자. 그러면 어떻게 수정할 수 있을까?

■ 인터페이스를 구현한 클래스 작성하기 : src/tommy/spring/exam02/MyBeanTwo.java

```
1 package tommy.spring.exam02;
2 public class MyBeanTwo implements MyBean {
3     @Override
4     public void sayHello(String name) {
5         System.out.println("안녕하세요, " + name + "!!!");
6     }
7 }
```

■ 이제 HelloSpring의 빨간색 부분을 "new MyBeanTwo();"로 변경하여 실행해 보자. 인사말이 변경되어 출력되는 것을 확인할 수 있다. 이로써 첫 번째 예제보다 결함도가 약간 프로그램을 제작할 수 있게 되었다. 하지만 첫 번째 보다는 약하지만 아직도 new를 이용한 객체생성을 함으로 방대한 어플리케이션의 경우에 역시 수정해야할 코드 또한 많아진다.

③ 제 3단계 : 스프링을 이용한 의존성 주입하기



■ src/tommy/spring/exam03라는 Package를 만든다. 두 번째 예제에서 MyBean, MyBeanOne, MyBeanTwo는 그대로 사용할 것이므로 그냥 복사 하자.

■ 관련 jar파일을 WEB-INF/lib 안에 넣는다.

- spring-beans-5.1.9.RELEASE
- spring-core-5.1.9.RELEASE
- spring-context-5.1.9.RELEASE
- spring-expression-5.1.9.RELEASE
- commons-logging-1.2 : <http://www.mvnrepository.com>에 접속하여 common-logging을 검색.

Apache Commons Logging » 1.2
 Apache Commons Logging is a thin adapter allowing configurable bridging to other, well known logging systems.

License	Apache 2.0
Categories	Logging Frameworks
HomePage	http://commons.apache.org/proper/commons-logging/
Date	(Jul 05, 2014)
Files	pom (18 KB) jar (60 KB) View All
Repositories	Central Apache Releases IBiblio ImageJ Public Redhat GA
Used By	8,830 artifacts

■ 스프링 설정파일 작성 : src/applicationContext.xml

- src에서 우 클릭한 후 [New]-[Other]-[Spring]-[Spring Bean Configuration File] 선택

1	<?xml version="1.0" encoding="UTF-8"?>
2	<beans xmlns="http://www.springframework.org/schema/beans"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://www.springframework.org/schema/beans
	https://www.springframework.org/schema/beans/spring-beans-4.3.xsd">
3	<bean id="myBean" class="tommy.spring.exam03.MyBeanOne" />
4	</beans>

■ 테스트용 메인클래스 작성하기 : src/tommy/spring/exam03/HelloSpring.java

1	package tommy.spring.exam03;
2	import org.springframework.context.support.AbstractApplicationContext;
3	import org.springframework.context.support.GenericXmlApplicationContext;
4	public class HelloSpring {
5	public static void main(String[] args) {
6	AbstractApplicationContext factory =
	new GenericXmlApplicationContext("applicationContext.xml");
7	MyBean bean = (MyBean) factory.getBean("myBean");
8	bean.sayHello("Spring");
9	factory.close();
10	}
11	}

■ 예제를 실행하여 결과를 확인하고 스프링 설정파일(applicationContext.xml)에서 MyBeanOne 대신에 MyBeanTwo 바꾸어서 실행해 보자. 인터페이스 기반의 설계와 스프링을 활용함으로써 소스파일은 변경하지 않고 스프링 설정파일만 변경해서 다양한 객체를 생성하는 유연한 어플리케이션을 작성할 수 있다.