

# Spring Boot

백엔드 현업에서 스프링부트가 많이 사용되는 이유

신온유  
정보보호학과  
2017111683





## 자격요건

- Java 언어, Java Framework(Spring, Spring boots, JPA) 기반 개발이 능숙하신 분
- Restful API 디자인이 능숙하신 분
- 대용량 트래픽 서비스 시스템 개발 및 운영에 능숙하신 분
- 페어프로그래밍, 코드리뷰에 거부감이 없고 테스트 코드 작성의 중요성을 아시는 분
- 멤버 간 수평 구조를 중시하고, 유연한 사고를 가지신 분
- 프로젝트 및 시간 관리 능력이 탁월하신 분
- 사업, 기획, 개발, UX 등 유관부서와 원활한 커뮤니케이션이 가능하신 분
- 신규 기술을 습득하고 업무에 적용이 가능하신 분
- 해외 출장 및 근무에 결격사유 없으신 분

## [지원자격]

- 실무 3년차 이상 또는 그에 준하는 역량 보유자
- 웹 애플리케이션과 HTTP API 개발에 익숙한 분
- 단위/통합 테스트코드 작성이 가능하신 분
- 우아한형제들의 개발 문화에 동의하시는 분
- 다음 기술에 익숙하신 분
- 자바 8 이상
- 스프링 프레임워크, 스프링 부트, 기타 자바 EE 기술
- JPA
- Git, Jira, Wiki 등의 협업 도구
- junit과 테스트코드 작성 라이브러리



우아한형제들

# 전자정부 프레임워크 + 스프링부트의 장점

# 스프링 부트란?



스프링부트는 **최소한의 설정**으로 스프링 플랫폼과 Third Party 라이브러리들을 사용 할 수 있도록 고안된 프레임워크이다.

즉, 스프링 부트는 환경 설정을 최소화하고 개발자가 비즈니스 로직에 집중할 수 있게하여 **생산성을 크게 향상시켜주는 프레임워크**이다.

## 스프링 부트의 특징

- 독립 실행이 가능한 스프링 애플리케이션 개발 (Embedded Tomcat, Jetty, Undertow를 사용)
- 통합 Starter를 제공하여 Maven/Gradle 구성을 간소화
- Starter를 통한 자동화된 스프링 설정 제공
- 번거로운 XML 설정을 요구하지 않음
- JAR을 이용하여 자바 옵션만으로도 배포 가능
- Spring Actuator 제공 (애플리케이션의 모니터링과 관리를 위해 사용)

# 스프링 부트의 장점

## 1) 라이브러리 관리 자동화

- 기존 스프링 자바 프로젝트에서는 Maven/Gradle을 이용해서 라이브러리 의존성을 관리해왔다.  
하지만 스프링 부트에서는 **스타터(Starter)**라는 것을 이용해 특정 기능에 필요한 라이브러리 의존성을 처리를 간소화 할 수 있다.

\*의존성 관리란 ? 프로젝트가 어떤 외부 라이브러리를 사용하고 있는지를 별도로 관리하는 것을 의존성 관리라고 한다.

## 2) 설정의 자동화

- 스프링 부트에서는 프로젝트에 추가된 라이브러리를 기반으로 실행에 필요한 환경을 **자동으로 설정해준다.**
- 개발에 필요한 라이브러리들을 추가하면 스프링 부트가 이 라이브러리들을 인지해서 관련된 스프링 설정을 자동으로 처리해주기 때문에 개발자들은 복잡한 설정을 하지 않고도 개발이 가능하다.

## 3) 라이브러리 버전 자동 관리

- 개발시 가장 신경쓰이는 부분이 라이브러리와 버전 관리이다. 기존의 스프링은 스프링 라이브러리만 사용하여 개발할 수 없으며, 의존관계에 있는 서드파티 라이브러들도 사용한다. 스프링 부트를 사용하면 스프링 부트 버전에 해당하는 스프링 라이브러리뿐만 아니라 **서드파티 라이브러리들도 호환되는 버전으로 다운로드해준다.**
- 라이브러리 버전이 달라 정상적으로 동작하지 않는 상황을 겪을 필요가 없고, XML설정을 이용해서 라이브러리를 매번 설정하는 과정을 줄이고 개발에만 집중할 수 있는 환경을 제공한다.

# 스프링 부트의 장점

## 4) 테스트 환경과 내장 Tomcat

- JUnit을 비롯한 테스트 관련 라이브러리들이 기본적으로 포함되어 있기 때문에 컨트롤러를 비롯한 다양한 계층의 클래스들에 대해서 **테스트 케이스를 쉽게 작성할 수 있다.**
- Tomcat 서버를 내장하고 있기 때문에 단지 main() 메소드를 가진 클래스를 실행하는 방식으로 서버를 구동하기 때문에 실행결과를 빠르게 확인할 수 있다.

## 5) 독립적으로 실행 가능한 JAR

- 애플리케이션을 개발하고 테스트까지 마쳤으면 애플리케이션을 실제 운영 서버에 배포하기 위해서 패키징을 해야하는데, 프로젝트가 일반 자바 프로젝트라면 간단하게 JAR파일로 패키징하면 되지만 웹 프로젝트라면 WAR 파일로 패키징 해야한다.
- 스프링 부트는 **독립적으로 실행 가능한 애플리케이션을 빠르게 개발하는 것을 목표**로 하기 때문에 웹 애플리케이션도 WAR가 아닌 JAR파일로 패키징 하여 사용할 수 있다.

# 전자정부 프레임워크

## 등장배경 및 목적

개발프레임워크는 정보시스템 개발을 위해 필요한 기능 및 아키텍처를 미리 만들어 제공함으로써 효율적인 어플리케이션 구축을 지원합니다. (후략)

※ 표준프레임워크는 기존 다양한 플랫폼(.NET, php 등) 환경을 대체하기 위한 표준은 아니며, java 기반의 정보시스템 구축에 활용하실 수 있는 개발·운영 표준 환경을 제공하기 위한 것입니다.

즉, 전자정부프레임워크란 정부 및 공공기관, 공기업 등의 웹사이트에 자주 쓰이는 공통 기능들을 Java의 Spring 프레임워크와 유명 Java 라이브러리(iBatis/MyBatis, Jackson, Apache Commons 등)를 가지고 미리 만들어 놓은 공통컴포넌트와 이를 개발하는 개발환경, 실행환경, 운영환경, 관리환경 등으로 구성된 것을 말합니다.



# 전자정부 프레임워크

## 왜 이것을 사용할까?

과거에는 공공기관에서 전산 시스템을 구축할 때 **각 기업별로 자체 제작한 별개의 개발 프레임워크를 사용했다.**

모든 기업들이 독특한 개발 프레임워크를 보유한 것은 아니며, 대개 공공기관의 시스템 개발 프로젝트에 대한 경험이 많은 **대형 SI 기업들이 개발 프레임워크를 자체 구축해 사용했다.** 이들은 자체 개발 프레임워크를 통해 개발 과정을 효율화할 수 있었기에, 이러한 프레임워크를 보유하지 못한 중소기업들에 비해 효과적인 개발 계획을 세울 수 있었다.

하지만 표준이 정립되지 않은 채로 각자의 프레임워크를 사용하다보니 각각의 **공공사업에서 사업자별로 동일한 기능을 중복해서 개발하는 일이 많았다.** 서로 다른 업체들이 각각의 프레임워크를 통해 개발하다보니 시스템 간의 연동에도 많은 공수와 시간이 소요됐다. 특정 기업의 프레임워크로 개발한 시스템을 유지·관리하기 위해 해당 기업에 대한 **종속성이 발생**한다는 이슈도 있었다.

출처 : 컴퓨터월드(<http://www.comworld.co.kr>)



# 전자정부 프레임워크

## 왜 이것을 사용할까?

공공사업 시 표준프레임워크를 사용하면 다양한 이점을 얻을 수 있다. 대다수의 기업들이 무료로 사용할 수 있는 표준프레임워크로 공공기관의 시스템을 개발하게 되므로 프레임워크 보유 여부에 따른 대형 SI기업과 중소기업간의 개발역량 격차를 해소할 수 있고, 공통된 기능과 모듈을 사용하다보니 유지보수도 용이해 특정 기업에 대한 종속성 문제도 해결할 수 있다.

또한 자주 사용되는 핵심 기능들을 표준프레임워크 상에 탑재함으로써 같은 기능을 중복 개발하며 발생하는 역량과 예산 낭비도 줄일 수 있다. 표준화된 기능과 모듈을 사용하니 서로 다른 시스템간의 연동 또한 용이하다.

출처 : 컴퓨터월드(<http://www.comworld.co.kr>)

# 전자정부 프레임워크

- 기업에 대한 종속성 탈피
- 핵심기능을 탑재함으로서 중복되는 기능 개발을 막을 수 있음
- 표준화 된 기능과 모듈을 사용함으로서 서로 다른 시스템과의 연동성이 높음

# 스프링부트

- 기본제공 되는 Starter를 이용하여 손쉽게 의존성 관리를 할 수 있다.
- Test Case 구현의 용이성
- 라이브러리 관리 자동화

즉,스프링 부트로 구현된 전자정부 프레임워크를 이용함으로서 이러한 이점을 얻을 수 있어 백엔드 프레임워크로 스프링부트를 채택하는 기업이 많다.

감사합니다

