

# 스프링 프레임워크의 특징을 통한 분석

이제호<sup>1</sup>

## 요약

스프링 프레임워크(영어: Spring Framework)는 자바 플랫폼을 위한 오픈 소스 애플리케이션 프레임워크로서 동적인 웹 사이트를 개발하기 위한 여러 가지 서비스를 제공하고 있다. 대한민국 공공기관의 웹 서비스 개발 시 사용을 권장하고 있는 전자정부 표준프레임워크의 기반 기술로서 쓰이고 있다. 또한, 대부분의 IT 서비스 기업은 대용량 서버 처리를 위해 활용하고 있으며, 스프링 프레임워크와 프레임 워크 구조를 탐색하므로써 그에 대한 분석을 하고자 한다.

## 1 서론

스프링 프레임워크(Spring Framework)는 자바 플랫폼을 위한 오픈 소스 애플리케이션 프레임워크로서 간단히 스프링(Spring)이라고도 한다. JVM에서 작동하며, 아파치 라이선스 2.0을 따른다. 한국 전자정부 프레임워크의 기반 기술이며 한국정보화진흥원에서는 공공기관의 웹 서비스 제공 시 스프링을 권장하고 있다. '스프링'이라는 이름의 유래는 이전에 Java EE(엔터프라이즈 에디션)의 스펙을 구현한 EJB가 기술의 복잡도가 증가해서 성능이 느렸던 것을 탈피하여, EJB시절을 "겨울"에 빗대어 겨울 후의 "봄"으로 새로운 시작한다는 것을 의미하는 스프링(봄)이 되었다. Java는 한

어떤 웹 프레임워크를 사용하시나요?

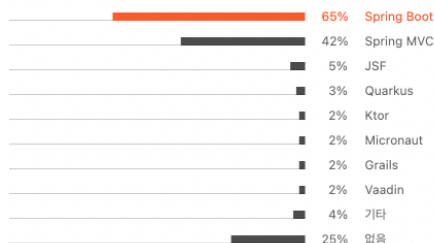


그림 1: "What web frameworks do you use, if any?", jetbrains.com, <https://www.jetbrains.com/lp/devecosystem-2021/java/>

국, 중국 및 독일에서 가장 많이 사용되는 언어이다. 한국의 Java 점유율은 53%, 중국은 47%, 독일은 33%이다. Java 플랫폼을 위한 프레임워크 중 Spring Boot의 점유율은 2021년에 약 4% 포인트 약간 증가했으며, 65%의 대다수 개발자가 사용하고 있다. Spring이 어떻게 높은 점유율을 차지하게 되었는지 Spring 특징을 분석하고 기존 EJB(Enterprise Java Beans)<sup>2</sup>와 비교한다.

## 2 본론

### 2.1 스프링 프레임워크의 특징

[1]

2) 당시 자바로 엔터프라이즈 애플리케이션을 개발하는 가장 일반적인 방법

스프링은 다른 프레임워크에 비해 다음과 같은 특징을 가진다.

- 스프링은 경량의 빈 컨테이너(Factory)이다. 경량 컨테이너로서 자바 객체를 직접 관리한다. 각각의 객체 생성, 소멸과 같은 라이프 사이클을 관리하며 스프링으로부터 필요한 객체를 검색하여 사용할 수 있다.
- 스프링은 Plain Old Java Object를 지원한다. 일반적인 J2EE 프레임워크에 비해 구현을 위해 특정한 인터페이스를 구현하거나 상속을 받을 필요가 없어 기존에 존재하는 라이브러리, 클래스 등을 지원하기에 용이하다.
- 스프링은 제어 반전(IoC : Inversion of Control)을 지원한다. 컨트롤의 제어권이 사용자가 아니라 프레임워크에 있어서 필요에 따라 스프링에서 사용자의 코드를 호출한다.
- 스프링은 의존성 주입(DI : Dependency Injection)을 지원한다. 각각의 계층이나 서비스들 간에 의존성이 존재할 경우 프레임워크가 서로 연결시켜준다.
- 스프링은 관점 지향 프로그래밍(AOP : Aspect-Oriented Programming)을 지원한다. 따라서 트랜잭션이나 로깅, 보안과 같이 여러 모듈에서 공통적으로 사용하는 기능의 경우 해당 기능을 분리하여 관리할 수 있다.
- 스프링은 영속성과 관련된 다양한 서비스를 지원한다. iBATIS나 하이버네이트 등 이미 완성도가 높은 데이터베이스 처리 라이브러리와 연결할 수 있는 인터페이스를 제공한다.
- 스프링은 확장성이 높다. 스프링 프레임워크에 통합하기 위해 간단하게 기존 라이브러리를 감싸는 정도로 스프링에서 사용이 가능하기 때문에 수많은 라이브러리가 이미 스프링에서 지원되고 있고 스프링에서 사용되는 라이브러리를 별도로 분리하기도 용이하다.

...

위와 같은 이유로 스프링은 다른 프레임워크에 비교하여 장점을 갖는다. 개발자가 기본적인 디자인 패턴(DI, AOP, 서비스 추상화 등)

을 사용하도록 하기 때문에 어느 정도 정형화된 개발이 가능해진다. 각 모듈을 조립해서 사용하는 것으로 기능을 쉽게 구현 가능하다. 또한 모듈(기능) 추가 및 제거가 용이하다. 그러나, 스프링으로 인한 단점들도 존재한다. 습득시간이 오래 걸리며, 기존 jsp, java bean을 통한 개발보다 무겁다.

## 2.2 스프링 프레임워크와 EJB의 비교

[2]

스프링이 처음 나올 당시, 자바로 엔터프라이즈 애플리케이션을 개발하는 가장 일반적인 방법은 EJB(Enterprise Java Beans)를 사용하는 것이었다. 그러나 당시 EJB에는 단점이 있었다.

1. 단위테스트가 어렵다.
2. 불필요한 메서드를 구현해야 한다.
3. 예외 처리가 번거롭다.
4. 배포가 불편하다.

그러나 스프링은 다음과 같은 특징으로 스프링을 사용하게 만들었다.

### 1. 단순화된 단위 테스트

- EJB는 컨테이너 외부에서 실행하는 것이 어려웠다. 그러므로 테스트를 위해서 반드시 컨테이너에 배포해야 했다. 그러나 스프링 프레임워크는 의존성 주입(DI)라는 개념을 도입하였다.
- 의존성 주입을 도입하면서, 단위 테스트를 위해 전체 애플리케이션을 배포할 필요가 없게 됐다.

### 2. 복잡한 코드의 감소

- 모든 메서드에서 예외 처리해야 하는 EJB와 달리 관점 지향 프로그래밍(AOP)을 사용하여 코드를 간단히 할 수 있다.

### 3. 아키텍처의 유연성

- 스프링 프레임워크는 모듈 방식이다. 스프링 코어 모듈 위에 독립적인 모듈을 올려 완성한다. 스프링 프레임워크는 스프링 애플리케이션의 서로 다른 부분들 간의 결합을 줄이고 (모듈화를 통해), 이를 테스트할 수 있게 만드는 것에 중점을 두면서 사용자가 선택한 프레임워크와의 통합을 제공한다

### 4. 빠르게 변화

- 스프링은 트렌드를 Java EE보다 빠르게 반영한다.

...

스프링은 경량 컨테이너 라는 점에서 이점을 갖는다. Tomcat과 같은 일반 서블릿 컨테이너에서 구동이 가능하다는 점은 더 이상 비싼 Java EE 서버를 구매하지 않아도 EJB보다 훨씬 간편한 방식으로 EJB가 제공하던 선언적 트랜잭션 및 보안 처리, 분산 환경 지원 등 주요 기능을 모두 사용할 수 있게 되었음을 뜻하며, 무엇보다 이제는 더 이상 각 Java EE 서버 제품에 특화된 설정을 따로 공부하거나 서버 제품을 바꿀 때마다 이식 작업이 필요 없이 Spring만 이용하면 Tomcat이든 레진( Resin )이든 기존의 풀스택 Java EE 서버이든 관계없이 간단하게 배포가 가능하다는 뜻이다.

## 3 결론

스프링 프레임워크가 전자정부 표준 프레임워크의 기반 기술로 쓰이고 있을 뿐만 아니라, 많은 IT 기업들로 부터 사용되는 이유를 특징을 통해 분석할 수 있다. 먼저 스프링은 Java EE(엔터프라이즈 에디션)를 개선하기 위해 나온 프레임 워크이다. 그런 배경이 있는 만큼 기업이 애플리케이션을 만들 때 필요한 기능이 갖추어져 있다. DI, AOP, 제공되는 서비스 인터페이스, 확장성으로 구현에 이점을 갖고 있다. 이 전에 일반적으로 개발하던 방법인 EJB에 비교해 보았고 실제로 단순화된 단위 테스트, 복잡한 코드의 감소, 아키텍처의 유연성으로 장점을 가지고 있었다. 이처럼 스프링 프레임워크는 개발자로 하여금 구현에 이점을 제공하여 개발에 편리함과 유연성을 제공한다. 이로 인해 스프링 프레임워크는 전자정부 표준 프레임워크의 기반 기술로 채택되어 사용될 수 있었으며 대부분의 IT기업이 선호하고 고려하게 되었다.

## 참고 문헌

- [1] wikipedia, “스프링 프레임워크,” [https://ko.wikipedia.org/wiki/스프링\\_프레임워크](https://ko.wikipedia.org/wiki/스프링_프레임워크).
- [2] S. A. John Arthur, “Spring framework for rapid open source j2ee web application development: A case study,” *IEEE*, accessed June. 6, 2005.