

Crazy Bob의 스프링 비판

I don't get Spring

이 글은 티스토리 블로그인 '용호의 작은 이야기' (<http://madeinkorea.tistory.com>)에서 발췌 인용한 내용이다.

구글의 유명한 블로거 미친 밥(Crazy Bob)은 자기 블로그에 올린 스프링 비판(I don't get Spring)으로 스프링 개발자들 사이에 유명하다. 그는 로드 존슨과 스프링 개발자들에게 「Effective Java」를 읽고 자바의 기초를 좀 배우라고 비아냥거리기도 했을 만큼 스프링과 그 개발자들을 좋아하지 않는 것 같다. 그의 스프링 비판은 나름 논리를 가지고 있는 것처럼 보이지만 그의 글을 자세히 읽어보면 스프링의 절대적인 인기에 불만인 것처럼 보이기도 한다. 어쩌면 대중으로부터 절대적인 인기가 있는 기술에 대해서는 일부러 더 뼈뺏하게 보는 것을 즐기는 사람일지도 모르겠다.

그가 비판한 내용을 보면서 그의 이야기를 나름 이해할 수 있었다(물론 그렇다고 그의 스프링 비판에 동의한다는 의미는 아니다). 다만 스프링에 대해 잘 모를 때 생길 수 있는 자연스러운 반응이라는 점에서 이해할 수 있을 뿐이다. 많은 사람들은 스프링을 단지 IoC/DI(Dependency Injection) 컨테이너라고 생각한다. 물론 모든 스프링의 기능은 IoC/DI 기반 위에서 동작하게 되어 있다. 하지만 그렇다고 스프링이 하나의 IoC/DI 컨테이너라고 생각하는 것은 마치 윈도우를 파일 관리 프로그램 정도로 여기는 것과 다를 바 없다. 결국 Bob Lee는 자신의 주장을 증명하기 위해서였는지 자기 스스로 심플한 IoC/DI 프레임워크를 만들어 공개했다. 구글 주스(Google Guice)라는 이름의 이 IoC 프레임워크는 Annotations, Generics를 적극적으로 활용하고 심플한 Interceptor도 사용할 수 있게 구성되어 있다. 단지 IoC/DI 기능만 요구하는 특별한 환경과 조건에서라면 어쩌면 스프링의 IoC 컨테이너보다 부분적으로 더 나을지도 모르겠다.

Joe Walker처럼 구글 주스가 스프링의 경쟁자라고 생각하는 것은 Hibernate 3(JPA engine)가 WebLogic 10(JEE5 WAS)의 경쟁자라고 생각하는 것과 다를 바 없다. 어떤 사람이 오해하는 것처럼 스프링은 단순히 IoC 컨테이너가 아니다. IoC는 스프링의 가장 기초가 되는 기술일 뿐이다. 로드 존슨은 한번도 스프링이 IoC 컨테이너 또는 AOP 프레임워크라고 얘기한 적이 없다. 스프링은 그 문서와 웹사이트에서 정의하듯이 'Full Stack Application Framework'이다.

한 가지만 더 얘기하자면 Bob Lee의 스프링 비판 중에 API Doc이 쓸데없이 복잡하다는 지적도 있다. 사용자들이 사용할 스펙에 해당하는 퍼블릭 API만 나와 있으면 그만이지, 굳이 그 구현까지 복잡하게 설명할 필요가 있었냐는 비판이다. 필자 또한 비슷한 생각을 해 본 적이 있다. 왜 스프링의 내부 구조까지 상세히 설명하기 위해 기반이 되는 인터페이스 하나하나에 대해 그토록 상세한 API Doc을 달았을까? 이유는 다음과 같다. 스프링은 사용자들에게 공개된 외부 API와 구현을 위해 사용한 내부 구현 코드의 구분이 없다는 것이다. 스프링은 그 코드 베이스를 거의 수정하지 않고 3년이 넘도록, 2.0이 되도록 계속 발전해 왔다. 스프링의 거의 모든 API는 사용자들이 필요하면 사용할 수 있는 Public API이다.

Bob Lee의 글에서 한 가지 공감 가는 내용은 스프링에 대해 비판적인 자세를 가지라는 주문이다. 사람들이 좋다고 하니 의심 없이 무조건 특정 기술과 제품을 선호하는 것은 결코 바람직한 태도가 아니다. 스프링에 대한 애정이 담긴 비판과 개선의 요청이 지금의 Spring 2.0을 만들어온 원동력일 것이다. 로드 존슨은 그의 책에서 자신의 말도 무조건 믿지 말라고 강조했다. 스스로 검증하고 확인하라는 것이다. 그와 스프링 개발팀은 아직까지도 어노테이션을 통한 DI 설정이 XML을 이용하는 것에 비해 그다지 장점이 없을 뿐더러 그다지 바람직하지 않다는 입장을 가지고 있다.

어느 분야에서나 완벽한 솔루션과 개발 방법론은 존재하지 않는다. 하지만 계속해서 부족한 부분은 보충하고 강점은 더욱 확대해 나가게 마련이다. 그러한 부분에서 본다면 스프링 프레임워크는 매우 큰 결단을 내렸으며 Spring-OSGi는 J2EE 엔터프라이즈 환경에서 오랫동안 사실상의 표준으로 자리매김하고 영향력을 더욱 확대해 나갈 가능성이 매우 크다. 늦어도 2-3년 내에는 유비쿼터스 시대에 걸맞은 u-시대가 본격적으로 도래할 것이다. 눈에 보이는 작은 센서노드(아니 어쩌면 철저하게 숨겨져 있는지도 모르겠다)에서부터 내 손안의 휴대폰과 책상 위의 PC, 거실과 방안의 TV, 그리고 회사 어디엔가 놓여 있을 냉장고보다 더 큰 서버에 이르기까지 OSGi는 장착되고 운영되고 있을 것이다. 끊임없이 발전하고 무한으로 치닫는 경쟁 속에서 OSGi는 큰 힘을 발휘할 것으로 기대된다. ● 김석우 | suhgoo.kim@samsung.com