

개발 생산성 향상을 위한 중소기업형 웹 어플리케이션 프레임워크의 연구

임복출*, 이종혁**, 김순곤***(교신저자)

*(주)씨디네트웍스, **(주)위컴즈, ***중부대학교 컴퓨터학과
e-mail : Wiseman.Lim@gmail.com, ljh@wecoms.com, sgkim@jbm.ac.kr

A Study of Web Application Framework for Improvement of Development Productivity in SMB

Bock-Chool Lim*, Jong-Hyuk Lee**, Soon-Gohn Kim***

*CDNetworks Co., Ltd.

**Wecoms Co., Ltd.

***Dept of Computer, Joongbu University

요 약

인터넷 서비스가 발전하면서 다양한 웹 프레임워크들이 등장하였다. 웹 개발시 웹 프레임워크의 도움 없이 애플리케이션을 개발한다는 것은 상상하기 힘든 시대이다. 다양한 웹 프레임워크의 등장은 개발자에게 선택의 어려움과 생산성을 고민하도록 만들었다. 본 논문에서는 웹 프레임워크의 기본적인 구성요소를 살펴보고, 프레임워크에서 제공하는 구조적인 패턴을 비교 연구하였다. 웹 프레임워크의 장점만을 도출 및 종합하여 소형 웹서비스가 필요한 중소기업형 프레임워크 WeLink에 적용하였다.

1. 서론

기업의 생산성 및 제품 품질의 향상을 위해서 협업, 설계/해석 프로세스 및 시스템 통합, 데이터 및 정보의 체계적인 관리와 이를 통한 제품개발 시간/비용의 단축이 필요하다. 또한 기업의 글로벌화가 진행되면서 다양한 컴퓨팅 환경의 Engineering Resource를 보다 효과적으로 지원하기 위한 통합 시스템을 필요로 하게 되었다[1].

프레임워크는 사전적인 의미로, “뼈대, 판단 결정 등을 위한 틀”이라고 할 수 있지만 조금 구체적으로 “복잡한 문제를 해결하거나 서술하는데 사용하는 기본 개념 구조”라고 할 수 있다[2]. 프레임워크는 애플리케이션의 구조와 디자인을 결정하는 요소를 가지고 있다. 디자인 패턴과 마찬가지로 프레임워크는 반복적으로 발견되는 문제를 해결하기 위한 특화된 솔루션이라고 이해할 수 있다[3].

본 논문에서는 프레임워크의 일반적인 사항과 구조적인 패턴을 비교하여, 소형 웹서비스가 필요한 중소기업형 프레임워크인 WeLink에 적용하였다.

2. 관련 연구

본 장에서는 관련 연구에 대하여 기술한다. 2.1절에서는 프레임워크의 특징에 대하여 기술하고, 2.2절에서는 웹 프레임워크에 대하여 비교 기술하고, 2.3절에서는 구조적 패턴에 대하여 비교 기술한다.

2.1. 프레임워크의 특징

소프트웨어 프레임워크의 가장 큰 특징은 모듈화

(Modularity), 재사용성(Reusability), 확장성(Extendibility)을 용이하게 해주는데 있다. 즉 품질 좋은 소스코드를 한번만 쓰고 버리는 것이 아니라 안정적인 인터페이스 뒤에 적절히 모듈화를 하여 여러 시스템에 융통성 있게 재사용이 가능하게 해주고 새로운 기능을 추가할 때 유연한 확장성을 제공해 준다[2].

2.2. 웹 프레임워크 : Spring, Spring Boot

Spring Boot는 모든 개발자를 위하여 Spring을 단순화한 것이다. Spring Boot의 목적은 강력한 Spring 기반으로 상품 수준의 애플리케이션과 최소한의 수정을 통한 서비스를 개발하는 것이다.



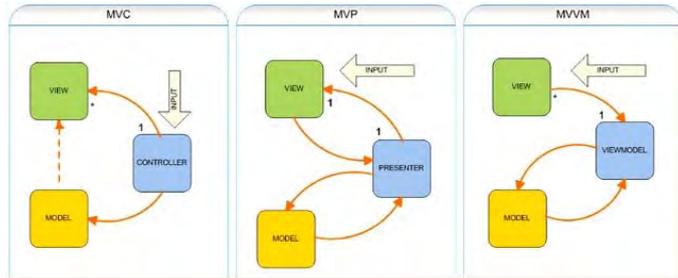
[그림 1] Spring vs. Spring Boot

상기 [그림 1]은 Spring의 초점으로 Spring Boot를 보여준다. 사용자는 경량화된 부분으로 접근하고 Spring의 나머지 부분으로 결과를 확장한다.

Spring Boot의 주요 목표는 모든 Spring 개발을 위해 근본적으로 빠르고 광범위한 접근을 제공하고, 새로운 요구사항에 대해서 빠르게 분기할 수 있고, 프로젝트의 큰 클래스를 공통화할 수 있는 비기능적 특징을 제공한다[4].

2.3. 구조적 패턴 : MVC, MVP, MVVM

프레임워크는 개발생산성을 제공하기 위하여 구조적 패턴을 내포하고 있다. 많은 구조적 패턴 중 MVC, MVVM, MVP를 비교하였으며, 다음 [그림 2]는 구조적 패턴을 비교한 것이다[5].



[그림 2] MVC, MVP, and MVVM

MVC 패턴은 Controller에 입력이 들어오면 Controller는 Model에 있는 Data를 조작하고, View는 Model에서 조작된 Data를 참조하여 View를 수정한다. MVP 패턴은 들어오면 Presenter에 Data를 요청하고, Presenter는 자신이 참조하는 Model에 업데이트를 요청하는 방식으로 동작한다. MVVM 패턴은 View에 입력이 들어오면 View가 참조하고 있는 ViewModel에서 Binding된 객체를 찾아 업데이트를 한다. MVP 패턴에서는 Presenter는 전적으로 View의 형태에 따라 달라지지만, MVVM 패턴에서는 ViewModel이 View를 참조하지 않으므로 Model의 형태를 따른다고 할 수 있다. View는 Model과 완벽히 분리되며 ViewModel과도 Binding을 통해 자동 업데이트 되므로 Data와도 완벽히 분리된다.

3. WeLink : 중소기업형 웹 어플리케이션

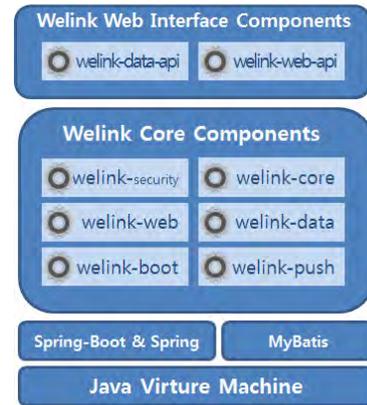
3.1. WeLink Stack 구조

WeLink는 Spring-Boot 기반, Embedded WAS형태의 경량 Web Application Framework이며, Database Tunneling을 통한 Domain Logic 구현의 최소화, 자동화된 Database Handling 지원한다. 또한 Restful Web Architecture 적용, 완벽한 클라이언트 Stateless를 구현함으로써 손쉬운 Scale-Out을 지원한다.

WeLink Stack의 주요 기능과 역할은 다음과 같다.

- Welink-core : Welink Framework 의 공통 모듈, 파일, 네트워크 처리, 비동기 처리 등을 담당
- Welink-data : 각종 Data처리, Database Handling, NoSQL 등을 처리
- Welink-web : web front-end 모듈
- Welink-web-api : File upload, Excel file handler 등

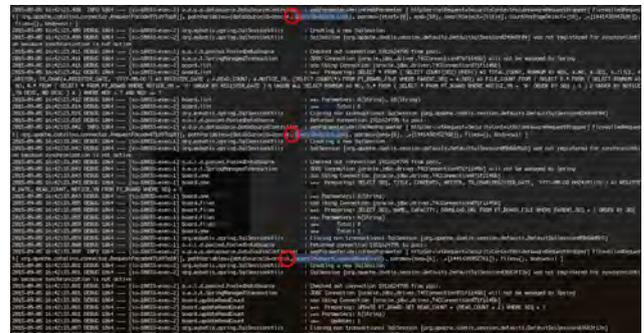
다음 [그림 3]은 WeLink Stack 구성도이다.



[그림 3] WeLink Stack

3.2. WeLink Framework을 활용한 게시판

WeLink를 활용하여 게시판을 기능을 개발하면 개발자는 사용할 게시판의 아이디와 Select, Insert, Update, Delete 등의 Query만을 고려하면 된다. Welink-core와 Welink-data에서 Data base와의 Connection의 관리는 물론, 표준화된게시판 Table 및 Common controller를 제공한다. View는 템플릿 형태로 구성되어 있어 별도의 디자인을 고려하지 않아도 되며, Bootstrap과 같은 템플릿을 활용도 가능하다. 다음 [그림 4]는 게시판을 임의 선택하고, 리스트와 하나의 게시물의 선택했을 때의 실행 로그이다.



[그림 4] WeLink 실행 로그

①은 임의의 게시판을 선택했을 때, 단순하게 board_id만을 전달하는 것을 볼 수 있으며, ②는 게시물 리스트중 1개의 게시물을 선택한 것을 볼 수 있다. ③은 게시물의 조회수를 업데이트하는 기능의 Query의 아이디이다. 모든 구조는 mybatis의 xml query기반으로 실행이 된다. 단순히 환경 설정을 통해 다양한 Data base를 동시에 지원도 가능하다.

WeLink는 구조적 패턴이 MVVM에 가깝다고 할 수 있다. 하지만 ViewModel를 통한 Data와 ViewModel Binding외에 MVC의 Controller를 통한 Data 조작과 이 결과를 통한 View를 통한 표현이나 MVP의 Presenter를 통한 Data 요청 및 표현 후 Model를 통한 업데이트 방식 상의 Data Binding이나 Data 조작 간편화 및 표준화 등의

장점 또한 포함하여 설계가 되었다. 구조적으로 Controller와 Model의 표준화를 통한 View 독립적 구성을 지원하도록 구현되었다. 이는 개발자가 View나 ViewModel, Presenter에 대해서 고민을 적게 하고 실제 비즈니스 로직만을 구성하여 자유로운 화면에 개발생산성을 높일 수 있는 구조이다. Controller와 Model의 표준화를 통하여 다양한 Database의 접근이 쉬우며, 개발자는 단순히 Query만 고민하면 되기 때문에 개발자의 업무 역할도 나눌 수 있다. 또한 View, ViewModel, Presenter를 별도로 고민하지 않기 때문에 자유로운 화면을 구현할 수 있게 된 것이다.

4. 결론

본 논문에서는 개발 생산성 향상을 위한 중소기업형 웹 어플리케이션 프레임워크인 WeLink를 구성하는 웹 프레임워크 Spring Boot와 구조적 패턴에 대하여 비교 기술하였다. 또한 WeLink의 Stack구조와 게시판 기능을 살펴 보았다.

향후 연구과제는 WeLink를 기반으로 한 소형 웹서비스가 필요한 분야에 대하여 프레임워크가 아닌 통합 개발 환경을 제공할 것이다.

감사의 글

이 연구는 중소기업청 2015년도 이공계전문가 기술개발서포터즈사업 기술애로 해결 과제(과제번호: SJ2015309)의 결과임.

참고문헌

- [1] Donguk Kim, Seunghak kuk, and Hyeonsoo Kim, "Recovery Policies on a Web Services-based Engineering Framework for Reliability Improvement", 2007 한국컴퓨터종합학술대회 논문집 Vol. 34, No. 1(A), pp.41-42
- [2] ByungHa Choi, Jongmoo Choi, "A Framework For Efficient Development of XML WebServices", 2011 한국컴퓨터종합학술대회 논문집 Vol.38, No.1(B), pp.132-135
- [3] ZDNet Korea, "http://www.zdnet.co.kr/news/news_view.asp?artice_id=00000039160910", 2008.09.10
- [4] Spring.io, "<https://spring.io/blog/2013/08/06/spring-boot-simplifying-spring-for-everyone>".
- [5] 뽀뽀's Blog, "<http://secretroute.tistory.com/entry/디자인패턴MVC-MVP-and-MVVM>"