

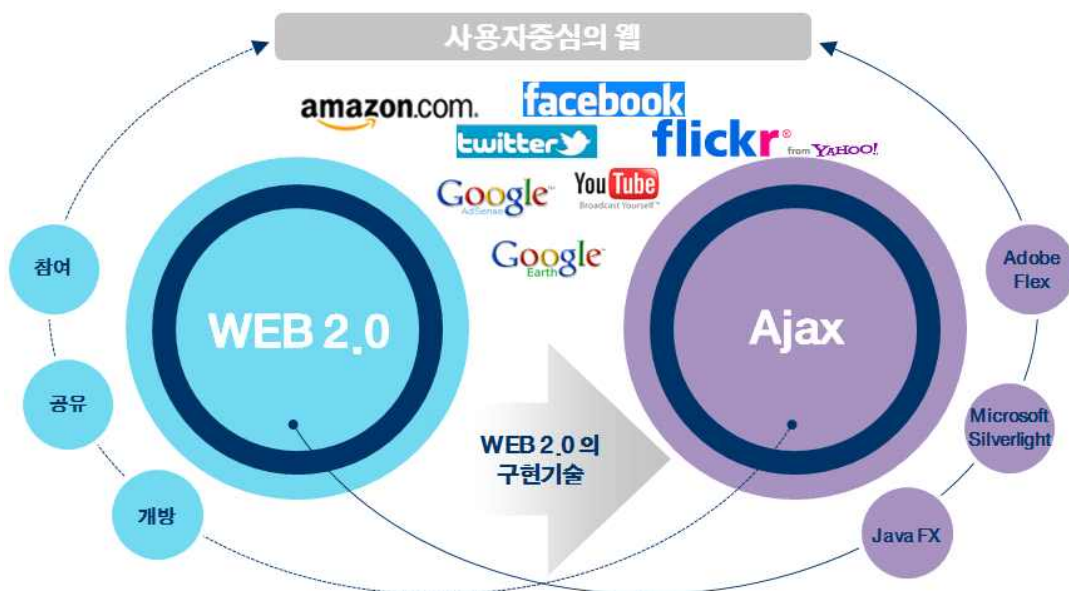
Natural-JS

Natural-JS 는 Natural-CORE, Natural-ARCHITECTURE, Natural-DATA, Natural-UI 로 구성 됩니다. **Natural-CORE**는 프레임워크에서 내부적으로 사용하는 CORE 라이브러리 패키지이고 **Natural-ARCHITECTURE**는 복잡한 웹 UI 환경을 구조화, 단순화 시켜주는 CVC Architecture Pattern 기반의 Architecture Framework를 제공 합니다. **Natural-DATA**는 Data 처리 및 데이터 동기화에 대한 모듈들을 제공 하며 **Natural-UI** 는 HTML 기반의 Grid, Form, Popup등 여러 가지 컴포넌트를 제공하여 웹 사용자경험과 개발생산성을 증대 시켜 줍니다.

1. 개발배경 및 목적

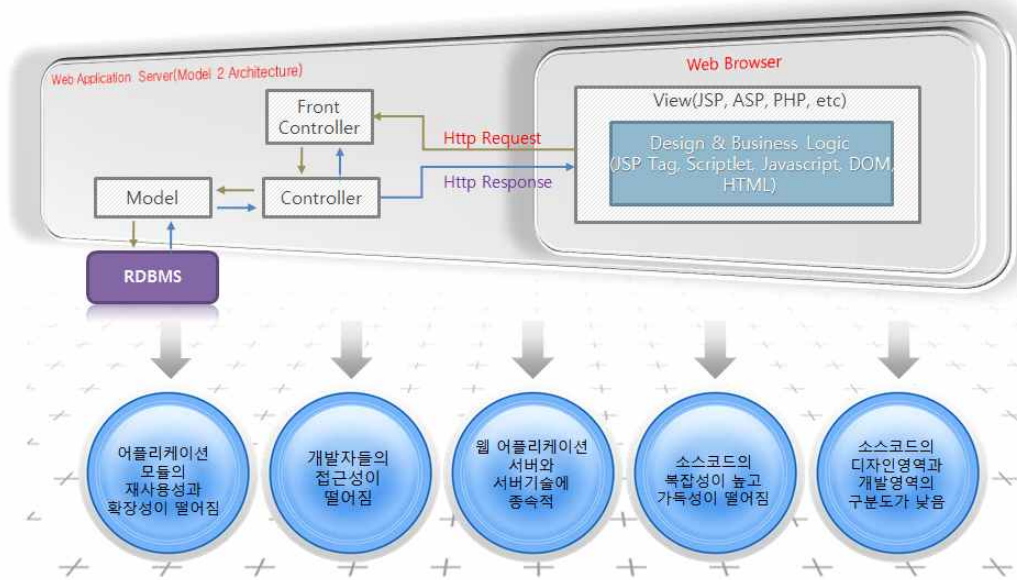
1990년대 등장한 웹 기술은 네트워크의 보급 확산과 정보공유 및 어플리케이션개발의 편의성에 힘입어 폭발적인 성장을 거듭하였습니다. 오늘날 웹 어플리케이션은 개인적 정보 공유로부터 기업의 자원 관리 및 전자정부에 이르기까지 현대인의 생활 전반에 매우 깊이 자리 잡고 있는 필수적인 부분이 되었습니다. 사용자에게 있어 이와 같은 웹 어플리케이션의 모든 기능은 사용자 인터페이스를 통해 보여 집니다. 이러한 사용자 인터페이스는 웹의 종단 사용자들이 경험하는 대화형 워크플로우의 모든 부분에 연관되며, 개발과정에서 가장 많은 시간과 노력이 요구되는 영역입니다. 한 통계자료에 따르면, 웹 어플리케이션 시스템 개발에 있어서 전체 개발 비용 중 유지보수에 관계된 비용은 80%가 사용자 인터페이스와 관련(이성혜, 마이크로소프트웨어, 7월 2003년)이 있다고 합니다.

Ajax (Asynchronous Java Script and XML)는 클라이언트 영역에 사용되는 일련의 웹 개발 방법들을 지칭하는 용어로 이러한 사용자 인터페이스 개발을 위한 대표적 기술 중 하나입니다. Ajax를 통하여, 웹 어플리케이션들은 현재 페이지의 디스플레이나 동작에 방해받지 않고 비동기 방식으로 서버로부터 데이터를 검색하고 추출하는 것이 가능합니다. Ajax 기술은 웹 사이트의 능률적이고 역동적인 사용자 인터페이스 개발을 위해 폭넓게 채택되고 있으며, e-mail 시스템, 메시지 링킹, 주식거래 및 전자정부 시스템 등의 다양한 웹 어플리케이션에 향상된 사용자 경험을 제공하기 위해 사용되고 있습니다. 또한 Google 이나 facebook등의 초대형 글로벌사이트들은 사용자 편의성을 위해 Ajax기술을 채택하고 있어 Ajax의 중요성을 증명하고 있습니다.



[그림1] WEB 2.0을 구현하는 기술

Ajax 기반 클라이언트는 이와 같은 다양한 웹 어플리케이션에서 고급 사용자 경험의 제공을 가능하게 하는 장점이 있지만, 방대한 분량의 코드로 이루어진 복잡한 클라이언트 개발에 있어 웹-개발의 편의성과 유지보수성 등이 낮아지는 단점이 있습니다. 다음은 Ajax 기반의 웹 어플리케이션 개발의 대표적인 문제점을 나타내는 그림입니다.



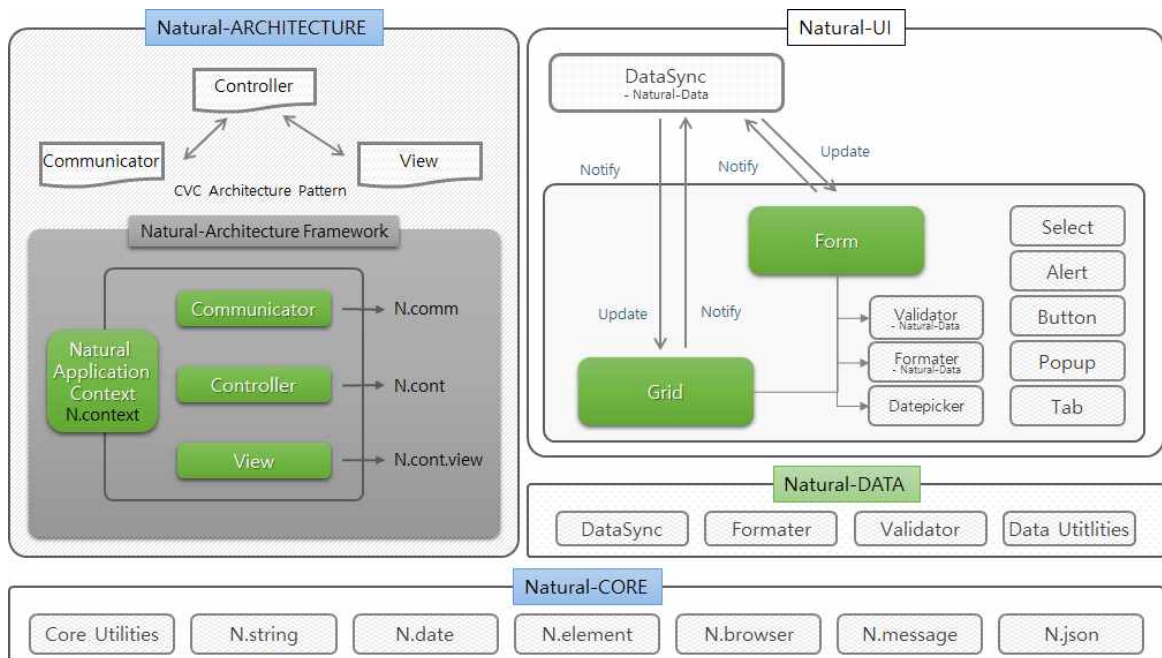
[그림2] 현행 Ajax기반의 웹 어플리케이션 개발의 문제점

위와 같은 문제점들을 해결하여 웹-개발의 편의성과 유지보수성 향상을 제공할 수 있는 기술로 MVC (Model-View-Controller) 패턴을 들 수 있는데, 이 패턴은 응용 로직을 사용자 인터페이스와 분리함으로써 구조적 디자인 복잡도를 감소시키고 이를 통해 유지보수성 향상을 제공합니다. MVC 모델은 Smalltalk80의 사용자 인터페이스 디자인에서 발전된 기술로, 다양한 웹 어플리케이션 개발에 보편적으로 적용되는 기술입니다. 하지만, 복잡한 사용자 인터페이스 코드 영역을 일반적인 서버-클라이언트 기반의 MVC 모델에서는 단순히 View 영역으로 구분함으로써 UI의 개발 편의성과 유지보수성을 개선하는데 제한적일 뿐 아니라 비대해지고 다이나믹해진 UI영역의 개발을 수용하기에도 무리가 따릅니다. 이와 같이 복잡한 Ajax 기반 클라이언트 개발에 있어, 그 개발편의성을 증대시키고 방대한 코드의 유지보수성을 개선하기 위해, Naturl-JS 를 개발하게 되었습니다. Naturl-JS 는 MVC 아키텍처 패턴을 웹 환경에 맞게 개량시킨 CVC(Communicator-View-Controller) 아키텍처 패턴(김항만, 김용구, 한국통신학회, 9월 2011년) 과 이 패턴 기반의 아키텍처 프레임워크를 제공하여 [그림2]와 같은 Ajax 기반 웹 어플리케이션 개발의 여러 가지 문제점들을 해결하고 엔터프라이즈 웹 어플리케이션에서 모바일 웹 어플리케이션까지, 웹 개발 전체의 개발 생산성과 사용자 경험을 월등하게 개선 시켜줄 수 있는 CVC 아키텍처 패턴기반의 프레임워크와 UI 컴포넌트를 제공 합니다.

2. 아키텍처 구성도

Naturl-JS 는 크게 Natural-CORE, Natural-ARCHITECTURE 와 Natural-DATA, Natural-UI 로 나뉘집니다. Natural-CORE는 프레임워크에서 전반적으로 사용되어 지는 라이브러리의 패키지이고 Natural-ARCHITECTURE는 웹 어플리케이션 UI 개발에 대한 아키텍처를 제공 해 주고 Natural-DATA는 Data 처리 및 데이터 동기화에 대한 모듈들을 제공 하며 Natural-UI 는 HTML 기반의 Grid, Form, Popup등 여러 가지 컴포넌트를 제

공 합니다.



[그림3] Natural-JS 구성도

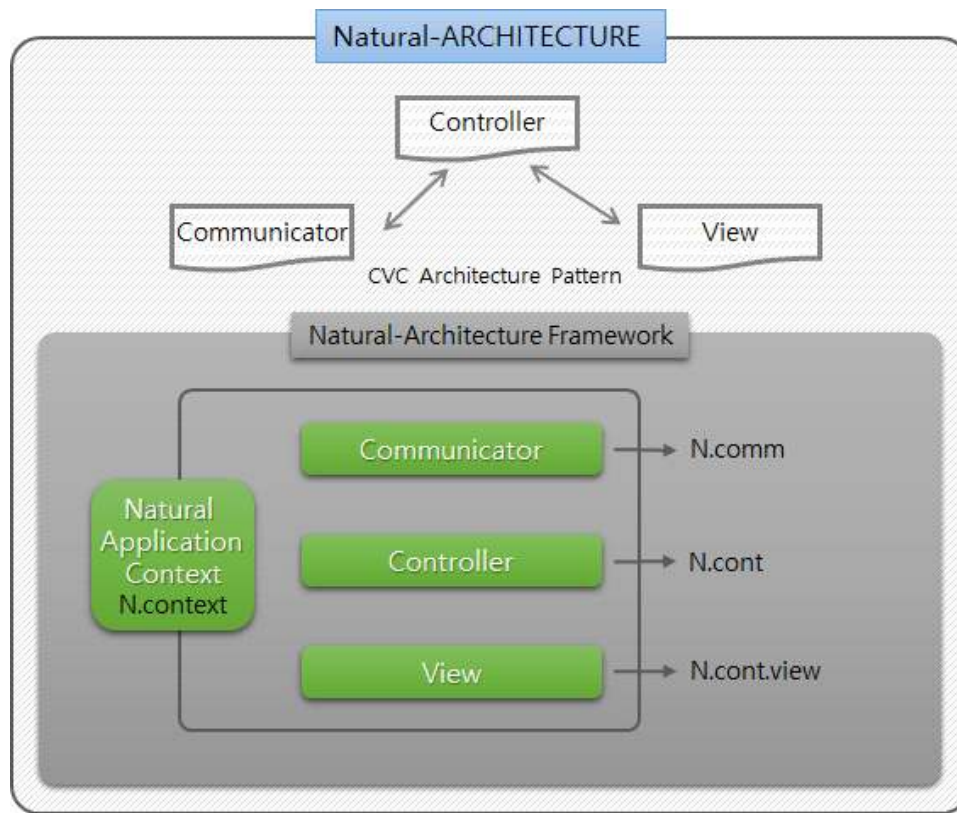
가. Natural-CORE

Natural-CORE는 Natural-JS에서 전역으로 사용하는 라이브러리 패키지입니다. Natural-Core는 다음과 같은 기능들로 구성 되어 있습니다.

- **Core Utilities** : Locale 설정, 로깅등의 코어 유틸리티들입니다.
- **N.string** : 문자열 조작 관련 메서드 집합 클래스
- **N.date** : 날짜 관련 메서드 집합 클래스
- **N.element** : HTML 요소 제어에 관련된 메서드 집합 클래스
- **N.browser** : 웹 브라우저와 관련된 메서드 집합 클래스
- **N.message** : 메시지 처리 관련 메서드 집합 클래스, 메시지 리소스의 데이터 처리 및 다국어 처리등의 역할을 수행.
- **N.json** : JSON 데이터 가공에 관련된 메서드 집합 클래스

나. Natural-ARCHITECTURE

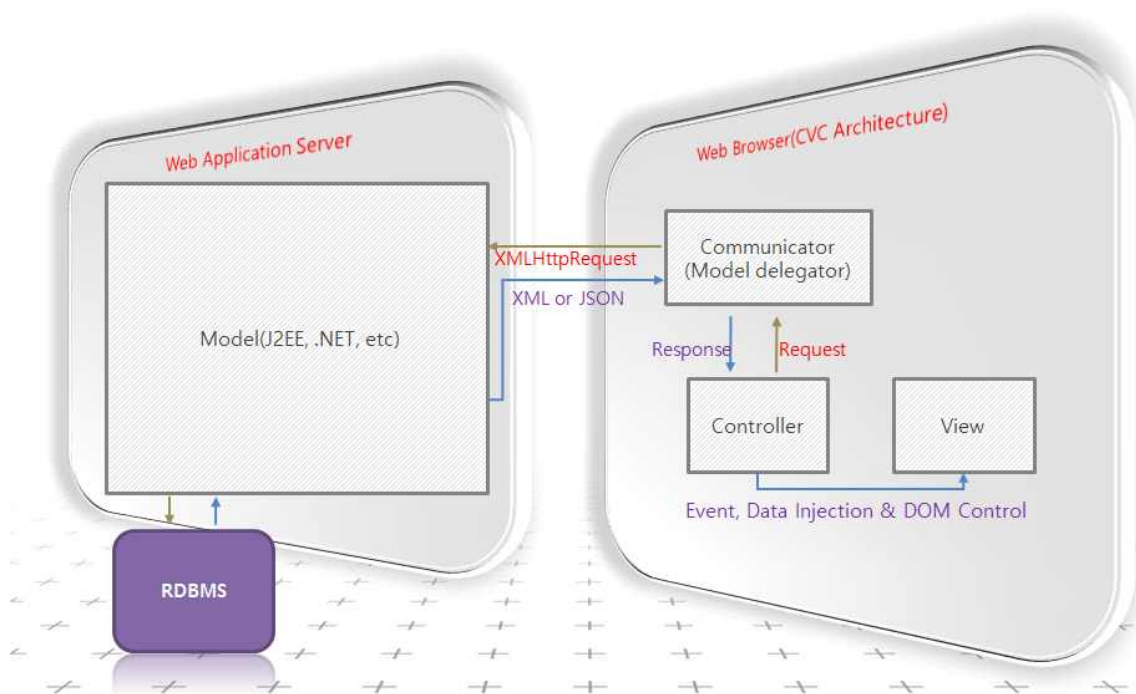
Natural-JS 는 CVC 아키텍처 패턴 기반의 Natural-Architecture Framework 를 제공하여 웹 어플리케이션 UI 개발의 복잡성을 해결하고 개발생산성을 향상 시켜줍니다. Natural-ARCHITECTURE 는 다음 다이어그램과 같이 구성되어 있습니다.



[그림4] Natural-ARCHITECTURE

1) CVC Architecture Pattern

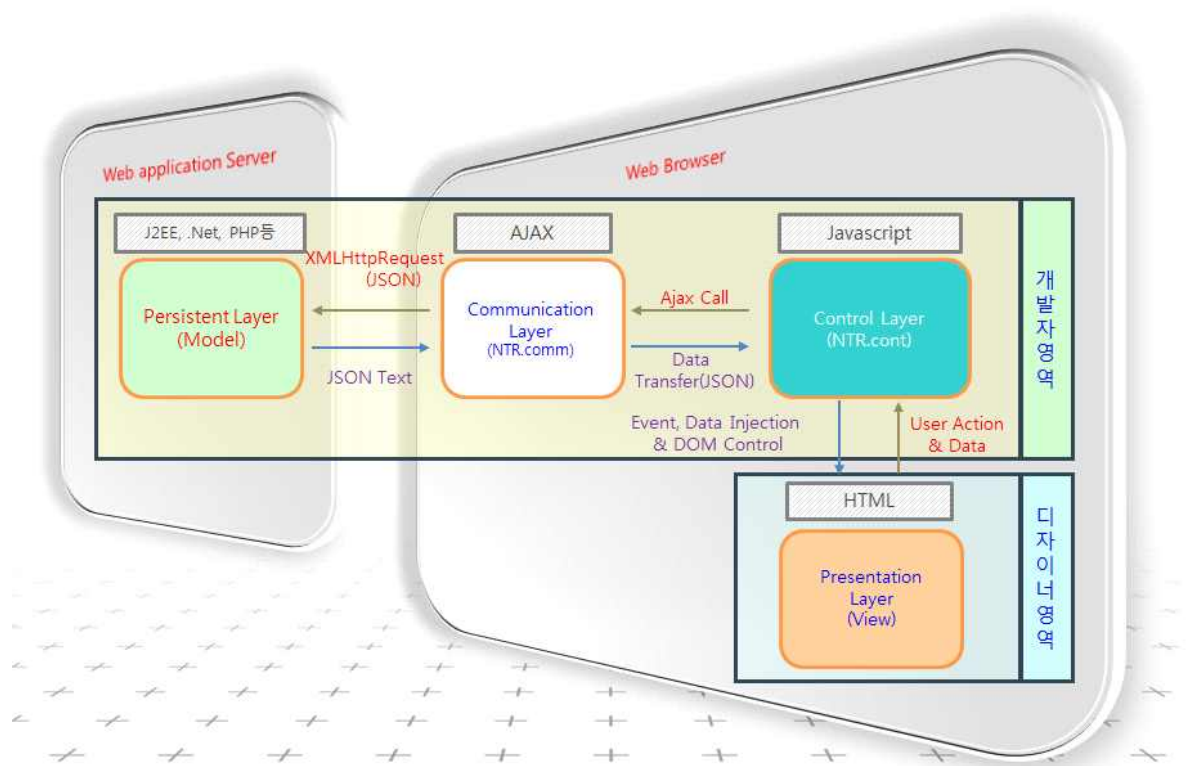
CVC 아키텍처 패턴은 기존 MVC로 구현된 서버중심 웹 어플리케이션에서 단순 뷰 (View) 영역을 표현하던 클라이언트 브라우저를 [그림5]와 같이 클라이언트중심의 MVC로 구성하고 서버를 데이터만 서비스해주는 단순 Model 영역으로 정의하여 MVC를 바라보는 관점을 서버중심에서 클라이언트 중심으로 이동 하는것이 CVC아키텍처 패턴의 핵심이론입니다. 이로서 클라이언트 브라우저를 웹 어플리케이션 서버와 서버기술의 종속성으로부터 탈피시킬 수 있고 디자인영역과 개발영역을 완벽하게 분리하여 개발의 복잡도를 낮출 수 있습니다.



[그림5] CVC Architecture Pattern

2) Natural Architecture Framework

Natural Architecture Framework는 CVC Architecture Pattern(이하 CVC 패턴)의 Communicator영역은 N.comm 클래스로 구현하였고, Controller 영역은 N.cont 라는 클래스로 구현되어 있습니다. 각 페이지별로 구현되는 N.cont 는 데이터를 가져와 가공하거나 View 에 데이터를 뿌려주거나 View로부터 데이터를 가져와 N.comm 에 전달 해 주는 역할을 수행 합니다. N.cont 는 페이지 별로 구현된 Controller 들의 라이프사이클과 Scope을 관리 해 줍니다. View 영역은 CVC 패턴에 정의된바 와같이 개발스크립트가 첨가되지 않은 순수 HTML만으로 구현하는 방법을 프레임워크에서 제공하고 있습니다. 특히 N.comm 는 논리페이지(물리적으로는 같이 있지만 논리적 Scope에 의해 분리된 페이지(팝업, 탭등)) 간 데이터를 전송하고 요청 정보등을 가져올 수 있는 N.comm.request 객체를 제공 합니다. 또한 N.comm에서는 서버에 요청(Request)과 응답(Response)의 각 단계별로 공통된 기능을 수행시킬 수 있는 Java Servlet Filter나 Spring의 Interceptor와 같은 기능을 수행하는 Natural Filter 기능을 제공 합니다.



[그림6] Natural Architecture Framework

Natural Application Context(N.context)는 Natural-JS 의 생명주기(Life-Cycle) 안의 데이터 저장소 역할을 수행 하며 설정 데이터, 메시지 리소스 등을 저장 하고 관리할 수 있는 기능을 제공 합니다.

다. Natural-DATA

Natural-DATA는 서버에서 가져온 데이터나 직접 작성한 JSON 데이터를 가공(정제, 조작)할 수 있는 기능과 데이터를 양식화(Formatting), 검증(Validate)을 할 수 있는 기능과 동일한 데이터를 사용하는 컴포넌트나 라이브러리 간 실시간 데이터 동기화(Synchronous)를 보장 해 주는 기능을 수행합니다. Natural-JS 는 데이터와 데이터가 표현될 HTML 요소(Element)의 값이 분리되어 있습니다. 원(元) 데이터는 메모리에 저장되고 HTML 요소에 표시되는 값들은 단지 보여주는 값들로만 처리될 뿐 표시되어 있는 데이터는 참고하지 않습니다(입력요소에 의한 입력 값 제외). HTML 입력 요소를 통한 입력 값이 수정되거나 생성되면 이와 연결되어 있는 메모리의 데이터(이하 DataSet)도 자동으로 수정 생성 됩니다. 때문에 서버와의 데이터 송수신시 메모리의 DataSet 만 다루면 되어 개발이 깔끔해지고 간단(Simple)해 집니다.

다음은 Natural-DATA 패키지의 각각의 모듈에 대한 설명입니다.

1) DataSync

DataSync는 동일한 데이터를 사용하는 컴포넌트나 라이브러리간 데이터 동기화를 보장해주는 Data Observer 역할을 수행하는 모듈 입니다.

2) Formater

Formater는 주어진 데이터를 양식화된 데이터로 변환하거나 HTML 입력요소에 양식화된 데이터를 표시 해 줍니다. 제공되는 이벤트들을 활용하면 입력요소(Input Element)에 사용자가 입력한 값에 대한 실시간 Formatting 기능도 간단하게 구현 할 수 있습니다. 또한 유용한 Formatting Rule 들이 기본으로 탑재 되어 있어 있습니다. 물론 사용자 정의 Rule 들도 손쉽게 추가 할 수 있습니다.

3) Validator

Validator는 주어진 데이터나 HTML 입력요소의 값을 검증하여 결과 데이터를 반환하거나 HTML 입력요소에 메시지를 표시 해 줍니다. 또한 유용한 Validation Rule 들이 기본으로 탑재 되어 있습니다. 물론 사용자 정의 Rule 들도 손쉽게 추가 할 수 있습니다.

4) Utilities

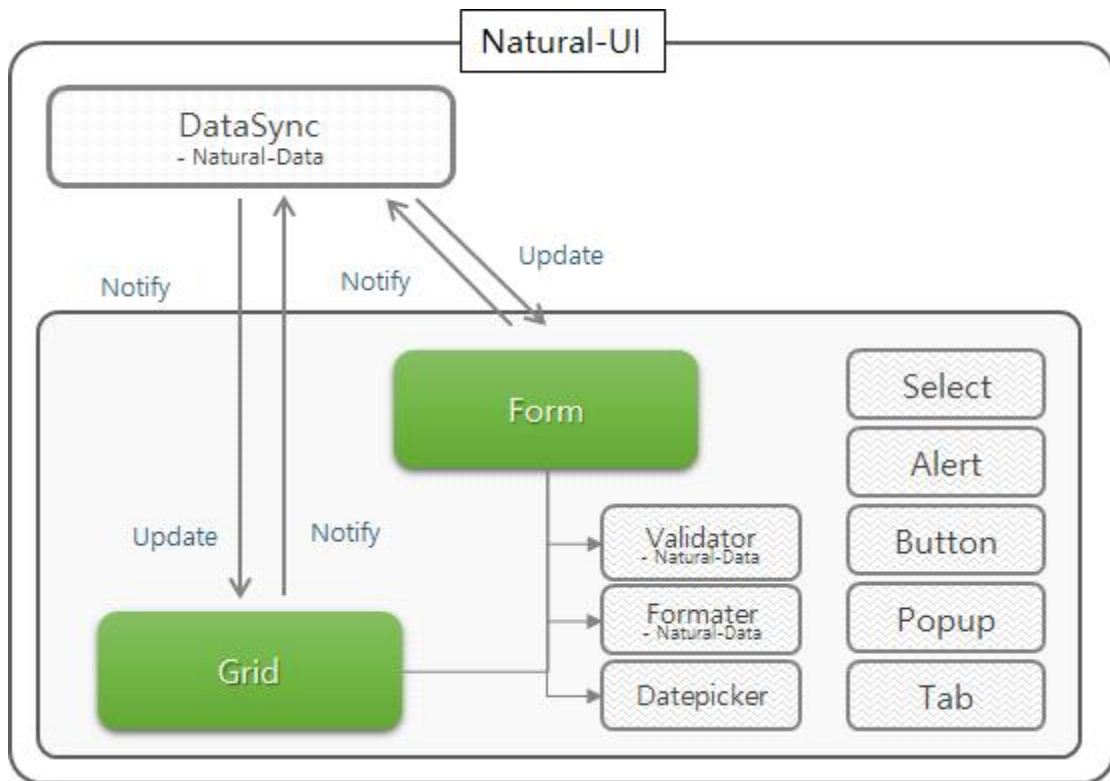
Natural-JS는 어플리케이션 개발의 편의성을 제공하기 위한 여러 가지 유틸리티를 제공합니다.

- N.refine : 서버에서 가져온 데이터를 Natural-JS UI 컴포넌트들에서 쓸 수 있도록 데이터 정제
- N.filter : 조건에 의한 JSON Array 데이터 필터링
- N.sort : JSON Array 데이터의 컬럼별 정렬

라. Natural-UI

Natural-JS는 표준웹 기반의 RIA를 손쉽게 구현 할 수 있도록 필수 컴포넌트를 제공합니다. 모든 컴포넌트는 개별로 사용할수도 있지만 결합되어 사용되어질 경우 Natural-CORE와 Natural-DATA 기반하에 모두 유기적으로 연동되어 좀더 편리하고 강력한 개발 환경을 제공 해 줍니다. 또한 컴포넌트들에 대한 기본 옵션을 Global Config 에 지정할 수 있어 통일된 UI 환경을 손쉽게 구성 할 수 있습니다. 물론 컴포넌트 생성 시 컴포넌트 옵션을 따로 지정 할 수 있습니다. Data 와 관련된 컴포넌트들(Select, Form, Grid)은 실제 데이터(메모리에 저장)와 HTML 요소로 보여지는 데이터가 따로 관리 되어 집니다. Grid를 예를 들면 화면에서 새로운 행을 생성하면 메모리의 데이터도 새로운 행이 생성되고 값을 수정하면 메모리의 값도 실시간으로 수정됩니다. Formatting, Validate 등 Data 조작과 검증에 관련된 라이브러리들도 이 메모리의 데이터를 기반으로 동작 됩니다. 수정, 입력, 삭제된 데이터를 서버로 보낼때는 화면의 입력요소에서 가져올 필요 없이 자동으로 동기화되어진 메모리의 데이터만 보내면 되어서 UI 개발이 단순 해 집니다. 또한 UI 컴포넌트들은 Controller 영역에서 컴포넌트의 기본 옵션을 직접 적용할 수 있는 명령형 옵션과 컴포넌트의 기본옵션과 서브옵션을 view 영역의 HTML 요소의 class 속성에 JSON Text 형태로 지정할 수 있는 선언형 옵션을 지원합니다. 선언형 옵션과 명령형 옵션의 지원 항목은 컴포넌트 기능의 속성에 따라 일치 하지 않을 수 있습니다.

Natural-UI는 다음과 같은 컴포넌트들과 구조로 구성 되어 있습니다.



[그림7] Natural-UI

1) Grid(N.grid)

Grid(이하 N.grid)는 메모리에 저장되어 있는 리스트 데이터(JSON Array)를 HTML 기반의 데이터 그리드 형태로 표현 해 줍니다. 다른 그리드 컴포넌트들과 달리 미리 정의한 HTML Table을 그대로 그리드 컴포넌트화 해줍니다. 때문에 그리드 디자인에 대한 제약이 전혀 없어 퍼블리싱된 HTML Table 요소에 Grid 컴포넌트만 실행시켜주면 아주 손쉽게 데이터 그리드를 완성 할 수 있습니다. 컬럼 리사이징, 스크롤 페이징, 행(row)의 추가·삭제, 컬럼정렬(Sort)등 일반적인 데이터그리드의 기능 외에 표시 데이터와 입력 데이터에 대한 실시간 양식화(Formatting)와, 검증(Validate) 기능을 제공 해 줍니다. 또한 그리드의 각 행의 요소 및 데이터를 제어할 하기위한(RowHandler) 기능을 제공 함으로서 그리드 조작에 대한 한계를 극복할 수 있습니다.

다음은 N.grid에서 제공하는 기능 리스트입니다.

- 헤더고정, 리스트 두 가지 타입의 그리드 제공
- 대용량 데이터 처리를 위한 스크롤페이징 기능 제공
- 입력한 데이터에 대한 실시간 검증(Validate) 기능 제공
- 입력한 데이터에 대한 실시간 양식화(Formatting) 기능 제공
- 컬럼 소트 기능 제공
- 컬럼 넓이, 그리드 높이에 대한 실시간 조절 기능 제공
- 행이 선택 되었을 경우의 효과와 이벤트 제공
- 행이 Hover 되었을 경우의 효과 제공
- CSS(Cross Site Scripting) 공격 차단을 위한 HTML TAG 적용 유무 옵션 제공

- 그리드에서 행 생성시마다 발동되는 효과 지정 가능
- 각 행들이 생성될 때마다 실행되는 rowHandler 이벤트 제공
- 데이터 바인드가 완료될때마다 실행되는 onBind 이벤트 제공
- 그리드안에 새로운 행을 생성할 수 있는 add 메서드 제공
- 새로운 행이 생성 되었을 경우 그리드 상단에 생성 할 건지 하단에 생성 할 건지 결정 할 수 있는 옵션제공(Data 생성도 같이 적용됨)
- 특정 행을 제거 할 수 있는 remove 메서드 제공
- 그리드의 특정 행 또는 전체 데이터를 바인드 할 때 당시의 데이터로 되돌려 주는 revert 메서드 제공
- 그리드의 특정 행 또는 전체 입력 데이터를 한번에 검증 할 수 있는 validate 메서드 제공
- 특정 행(row)의 특정 컬럼(key) 값을 가져오고 변경할 수 있는 val 메서드 제공

2) Form(N.form)

Form(이하 N.form)은 리스트 데이터(JSON Array)의 특정 로우(Row) Data를 HTML 요소에 값을 바인드(Bind) 해 주는 컴포넌트입니다. 그리드와 연계된 상세화면 보거나 그리드의 조회조건에 해당하는 입력요소의 값을 손쉽게 컨트롤 하는데 사용됩니다. 또한 신규 입력페이지 등 에서도 손쉽게 입력요소를 기반으로 데이터를 생성하고 입력 데이터 및 표현 데이터에 대한 실시간 양식화(Formatting)와 검증(Validate)할 수 있는 기능을 제공합니다.

데이터 레이아웃을 표현하는 HTML 요소(DIV, SPAN, TABLE등)를 지정하면 지정한 요소의 하위요소(Children Elements)중에서 Data의 컬럼명과 일치하는 식별자(태그의 id 속성)를 가진 요소를 찾아 값을 바인드 해 주어 데이터 폼을 쉽게 구현 해 줍니다.

다음은 N.form에서 제공하는 기능 리스트입니다.

- 입력한 데이터에 대한 실시간 검증(Validate) 기능 제공
- 입력한 데이터에 대한 실시간 양식화(Formatting) 기능 제공
- CSS(Cross Site Scripting) 공격 차단을 위한 HTML TAG 적용 유무 옵션 제공
- 폼의 입력요소를 기반으로 새로운 데이터를 생성할 수 있는 add 메서드 제공(신규 입력페이지 등에서 간편하게 데이터를 생성 할 수 있음)
- 새로운 행이 생성 되었을 경우 데이터 리스트의 상단에 생성 할 건지 하단에 생성 할 건지 결정 할 수 있는 옵션제공(Grid 연동 개발 시 Grid와 동기화를 위함)
- 특정 행을 제거 할 수 있는 remove 메서드 제공
- Form 에 바인드 되어 있는 데이터를 바인드 할 때 당시의 데이터로 되돌려 주는 revert 메서드 제공
- Form의 전체 입력 데이터를 한번에 검증 할 수 있는 validate 메서드 제공
- 특정 컬럼(key) 값을 가져오고 변경할 수 있는 val 메서드 제공

3) Select(N.select)

Select(이하 N.select)는 select, input 태그의 type이 radio, checkbox인 선택 입력 요소들에 데이터 리스트를 바인드(Bind)하여 선택요소의 선택항목들을 자동으로 구현해주는 컴포넌트 입니다. 데이터를 바인딩 하는 기능 외에 값을 가져올 때 선택된 값만 가져오

는 것이 아니라 선택한 값이 포함된 row data set 을 가져올 수 있는 기능도 제공 해 줍니다.

다음은 N.select에서 제공하는 기능 리스트입니다.

- 선택요소로 생성 될 데이터리스트의 name 컬럼과 value 컬럼을 지정 할 수 있는 기능 제공.
- 추가될 선택요소를 기존 선택항목에 덮어쓸건지 더할건지 지정 할 수 있는 기능 제공.
- radio나 checkbox 의 경우 표시 방향을 가로인지 세로인지 지정 할 수 있는 기능 제공.
- 지정한 값으로 실시간으로 선택 해 주는 val 메서드를 제공.
- 선택을 초기화하는 reset 메서드를 제공.

4) Tab(N.tab)

Tab(이하 N.tab)은 여러 콘텐츠 패널을 하나의 콘텐츠 영역으로 표현해 주는 컴포넌트입니다.

다음은 N.tab에서 제공하는 기능 리스트입니다.

- 탭이 초기화 되고 첫 번째 탭을 랜덤으로 선택해서 표시 해 주는 기능 제공.
- 탭이 활성화 될 때마다 실행되는 이벤트 제공.
- 각 탭 단위별로 각각 옵션을 지정할 수 있는 기능 제공(data-opts 속성에 JSON 형태로 지정)
 - 탭의 넓이를 지정(auto 지정 시 자동) 할 수 있는 기능 제공
 - 탭의 콘텐츠 URL 지정 할 수 있는 기능 제공
 - 탭의 콘텐츠를 탭이 초기화 되면서 로드할건지 탭이 최초열릴 때 로드할건지 지정 할 수 있는 기능 제공
 - 탭 컴포넌트가 이 초기화 됐을 경우 최초 선택될 콘텐츠와 탭을 지정 할 수 있는 기능 제공
 - 탭이 열렸을 때마다 실행되는 이벤트를 지정 할 수 있는 기능 제공

5) Popup(N.popup)

Popup(이하 N.popup)은 레이어 팝업을 손쉽게 구현할 수 있는 컴포넌트 입니다. 레이아웃을 표현할 수 있는 특정 요소(DIV, SPAN TABLE 등의 BLOCK 요소)를 팝업으로 만들 수 있고 특정페이지를 가져와 팝업으로 만들 수도 있습니다. 또한 Grid, Form과 연동하여 사용하면 Grid와 상세조회 팝업이 연동된 페이지를 쉽게 만들 수 있습니다.

다음은 N.popup에서 제공하는 기능 리스트입니다.

- 페이지 안의 특정 요소를 팝업으로 만들기 위해 특정요소를 지정할 수 있는 기능 제공
- 특정 페이지를 팝업으로 만들기 위해 해당페이지의 URL을 지정 할 수 있는 기능 제공
- 팝업의 제목을 지정 할 수 있는 기능 제공(팝업 요소의 title 속성으로도 지정 가능)
- 팝업의 높이와 넓이를 지정 할 수 있는 기능 제공
- 팝업의 버튼을 취소버튼만 표시할건지 Confirm처럼 확인, 취소 버튼을 표시할 건지 지정 할 수 있는 기능 제공
- 팝업의 버튼들을 표시할건지 유무를 지정 할 수 있는 기능 제공
 - 버튼을 표시하지 않고 팝업의 콘텐츠에 버튼을 포함할 경우 사용, 특정페이지를 불러

- 을 경우 해당페이지의 Controller(N.cont) 에 opener(팝업을 호출한 부모페이지의 Controller)와 caller(팝업을 호출한 팝업 인스턴스)를 자동으로 세팅 해 줌
- 확인 버튼에 대한 이벤트 콜백함수를 지정 할 수 있는 기능 제공
 - 취소 버튼에 대한 이벤트 콜백함수를 지정 할 수 있는 기능 제공
 - 팝업이 열릴때마다 실행되는 이벤트를 지정 할 수 있는 기능 제공
 - 해당 이벤트가 실행될때마다 이벤트 콜백의 인자 형태로 데이터를 팝업의 Controller로 넘길 수 있음
 - 팝업이 닫힐때마다 실행되는 이벤트를 지정 할 수 있는 기능 제공
 - 해당 이벤트가 실행될때마다 콜백의 인자 형태로 데이터를 부모페이지의 Controller로 넘길 수 있음
 - 팝업의 컨텐츠를 팝업이 초기화 되면서 로드할건지 팝업이 최초열릴 때 로드할건지 지정 할 수 있는 기능 제공(특정 페이지를 로드했을때만 적용됨)
 - 팝업을 열고(open), 닫고(close), 팝업 요소를 제거(remove) 할 수 있는 메서드 제공

6) DatePicker(N.datepicker)

DatePicker(이하 N.datepicker)는 날짜를 선택 하여 입력 할 수 있는 컴포넌트입니다. 날짜 포맷은 전역(Config)으로 설정한 포맷을 따르며 Grid나 Form 에 연계되어 사용되어 질 경우 Format 룰의 세 번째 인자로 date(Datepicker) 나 month(Monthpicker)를 지정 하면 Grid와 Form 에 자동으로 연동되어 표시되며 선택한 값이 메모리의 데이터에 실시간 반영 됩니다.

다음은 N.popup에서 제공하는 기능 리스트입니다

- 년월만 표시하여 Monthpicker 로 만들어 주는 기능 제공.
- 선택되었을 때 발생하는 이벤트를 지정 할 수 있는 기능 제공.
- 현재는 입력 요소가 readonly 상태로 변해 DatePicker 에 의해서만 입력할 수 있지만 직접 숫자를 입력해도 DatePicker와 연동되도록 추가 구현 할 예정.

7) Button(N.button)

Button(이하 N.button)은 HTML a, button, input중 타입이 button인 요소들을 기본버튼모양으로 만들어 줍니다. 명령형으로 옵션을 지정할 수 있지만 편의성을 위해 선언형 옵션으로 버튼 요소의 data-opts 속성(attribute)에 옵션으로 넓이, 기본크기, 색상, 비활성화 유무를 지정 할 수 있습니다. Button 은 유일하게 여러 버튼요소를 한번에 초기화 할 수 있습니다.

다음은 N.button에서 제공하는 기능 리스트입니다.

- 크기(smaller, small, medium, large, big)를 지정할 수 있는 기능 제공.
- 색상(white, blue, skyblue, gray)을 지정할 수 있는 기능 제공.
- 초기화시 버튼을 비활성화 할 수 있는 기능 제공.
- 비활성, 활성, 마우스 오버, 클릭 시의 페이드 효과를 적용 할 건지 지정 할 수 있는 기능 제공.
- 버튼을 비활성화(disable), 활성화(enable) 할 수 있는 메서드 제공.

8) Alert(N.alert)

Alert(이하 N.alert)은 Javascript 의 window.alert 이나 window.confirm 같은 메시지를 표현을 위한 컴포넌트입니다. N.alert은 메시지 다이얼로그를 모달 레이어나 일반 레이어 형태로 표시 해 줍니다.

다음은 N.alert에서 제공하는 기능 리스트입니다.

- Modal overlay 가 표시될 영역을 지정 할 수 있는 기능 제공(window 객체 지정 시 전체 화면을 가리고 특정영역(Element) 지정 시 지정된 영역만 가려줌.
- Modal overlay 의 스타일을 지정 할 수 있는 기능 제공.
- 메시지의 특정 변수({} 형태)를 특정 값으로 치환하여 표현할 수 있는 기능 제공.
- 메시지 다이얼로그의 높이와 넓이를 설정 할 수 있는 기능 제공.
- 메시지 다이얼로그의 제목을 설정 할 수 있는 기능 제공.
- 메시지 다이얼로그의 버튼들을 감출 수 있는 기능 제공.
- 모달 다이얼로그 유무를 지정 할 수 있는 기능 제공.
- window.confirm처럼 확인, 취소 버튼을 표시 할 수 있는 기능 제공.
- 확인 버튼에 대한 클릭(Click) 이벤트 제공.
- 취소 버튼에 대한 클릭(Click) 이벤트 제공.
- 다이얼로그의 z-index를 항상 최상위로 유지시켜 다른 요소들보다 항상 위에 표시되도록 설정 할 수 있는 기능 제공(기본적으로 최상위에 유지되지만 다른 요소들이 간섭할 때 사용).

3. 개발언어

- Javascript 2.0
- HTML4(HTML5 지원)
- CSS2(CSS3 지원)
- jQuery 1.11.1 : Natural-JS는 기본적으로 jQuery 의 확장객체(Wraped Object)를 상속받아 사용함.

4. 사용시스템

- 개발시스템 : Intel CPU 계열의 PC
- 개발 툴 : eclipse
- 지원 브라우저 :
 - * Internet Explorer 8이상
 - * Chrome, Firefox, InternetExplorer, Safari, Opera 최신 브라우저
 - * 안드로이드 브라우저, IOS Safari 등 모바일 브라우저

5. 참여인원 및 업무분장

- 김항만 : 기획, 분석/설계, 개발, 테스트, 릴리즈