

# **Natural-JS API Document**

(Version : 0.8.1.7)

# 목 차

« Natural-JS ver. 0.8.0.0 API 개요 » .....	1
« Natural-CORE ver. 0.8.0.0 »	
1. N.....	2
가. 개요 .....	2
나. 함수(Methods).....	2
« Natural-ARCHITECTURE ver. 0.8.0.0 »	
2. Controller.....	5
가. 개요 .....	5
나. 생성자(Constructor).....	5
다. Contrller 기본 함수(callback default method).....	5
3. Communicator(N.comm).....	6
가. 개요 .....	6
나. 생성자(Constructor).....	6
다. 기본옵션(Default options).....	6
라. 함수(Methods).....	6
마. Communicator.request.....	7
1) 개요 .....	7
2) 생성자(Constructor).....	7
3) 기본옵션(Default options).....	7
4) 함수(Methods).....	7
4. Context(N.context).....	8
가. 개요 .....	8
나. 함수(Methods).....	8
5. natural.config.js.....	8
가. 개요 .....	8
« Natural-DATA ver. 0.8.0.0 »	
6. Formater(N.formater).....	9
가. 개요 .....	9
나. 생성자(Constructor).....	9
다. 함수(Methods).....	9
라. 포맷 룰(Format rules).....	10
7. Validator(N.validator).....	11
가. 개요 .....	11
나. 생성자(Constructor).....	11
다. 함수(Methods).....	11
라. 검증 룰(Validate rules).....	12
8. Natural-DATA Libraries.....	13
가. 개요 .....	13
나. 함수(Methods).....	13

9. Alert(N.alert).....	15
가. 개요 .....	15
나. 생성자(Constructor).....	15
다. 기본옵션(Default options).....	15
라. 함수(Methods).....	16
10. Button(N.button).....	16
가. 개요 .....	16
나. 생성자(Constructor).....	16
다. 기본옵션(Default options).....	16
라. 선언형옵션(Declarative options).....	16
마. 함수(Methods).....	17
11. DatePicker(N.datepicker).....	17
가. 개요 .....	17
나. 생성자(Constructor).....	17
다. 기본옵션(Default options).....	17
12. Popup(N.popup).....	17
가. 개요 .....	17
나. 생성자(Constructor).....	18
다. 기본옵션(Default options).....	18
라. 함수(Methods).....	19
13. Tab(N.tab).....	19
가. 개요 .....	19
나. 생성자(Constructor).....	19
다. 기본옵션(Default options).....	19
라. 선언형옵션(Declarative options).....	20
마. 함수(Methods).....	21
14. Select(N.select).....	21
가. 개요 .....	21
나. 생성자(Constructor).....	21
다. 기본옵션(Default options).....	21
라. 함수(Methods).....	21
15. Form(N.form).....	22
가. 개요 .....	22
나. 생성자(Constructor).....	22
다. 기본옵션(Default options).....	22
라. 선언형옵션(Declarative options).....	22
마. 함수(Methods).....	23
16. Grid(N.form).....	23
가. 개요 .....	23
나. 생성자(Constructor).....	23
다. 기본옵션(Default options).....	24
라. 선언형옵션(Declarative options).....	25
마. 함수(Methods).....	26

## << Natural-JS ver. 0.8.0.0 API 개요 >>

Natural-JS API Document 는 Natural-JS 의 컴포넌트와 라이브러리에서 제공하는 기능과 옵션에 대한 사용법과 설명을 제공하는 문서입니다.

문서의 큰 단락은 << 컴포넌트 나 라이브러리 >> 와 같이 구분 되어 있으며 단락안의 설명은 다음과 같이 구성 되어 있습니다.

- 개요 : 컴포넌트나 라이브러리에 대한 개요.
- 생성자(Constructor) : 컴포넌트나 라이브러리를 생성할 때 실행되는 기능과 생성자 arguments 에 대한 설명  
ex) N.grid<sup>1)</sup>(argument<sup>2)</sup>, N().grid(argument<sup>2)</sup>
- 기본옵션(Default Options) : 컴포넌트나 라이브러리를 생성할 때 생성자 argument로 지정되는 옵션에 대한 설명  
ex) N([]).grid( { resizeable : true }<sup>3)</sup> )
- 선언형옵션(Declarative Options) : 컴포넌트나 라이브러리에서 사용하는 HTML 요소(Element)들의 data-\* 속성에 정의되는 JSON Text 양식의 옵션  
ex) <input id="date" type="text" data-format='["date", 8]'<sup>4)</sup> />  
  
⚠ 선언형 옵션은 JSON 표준양식을 정확히 지켜줘야 합니다. 예를 들면 key 값에도 큰따옴표(")로 무조건 감싸줘야 하는 등 JSON 표준양식을 지키지 않으면 선언형 옵션이 인식되지 않거나 오류가 발생합니다.  
⚠ format이나 validate등 선언형 옵션으로 실행되는 룰 들의 인자들(arguments)는 배열(Array) 형태(type)으로 정의하고 인자 순서대로 나열합니다.
- 함수(Methods) : 컴포넌트나 라이브러리 인스턴스에서 제공하는 기능과 arguments 에 대한 설명  
ex) N([]).grid( { resizeable : true } ).revert<sup>5)</sup>(3<sup>6)</sup>

---

1) 생성자  
2) 생성자 arguments  
3) 기본옵션  
4) 선언형옵션  
5) 함수  
6) 함수 arguments

## << Natural-CORE ver. 0.8.0.0 >>

### 1. N

#### 가. 개요

N은 Natural-JS를 처음 시작하는 함수 입니다. jQuery 의 \$와 같은 역할을 수행하며 Natural-JS 의 N() 처럼 N 함수로 초기화하고 이후 실행되는 메서드들은 jQuery 의 \$ 함수 로 실행해도 됩니다. 반대로 jQuery 의 기능들은 모두 N() 에 상속되어 있으므로 jQuery 메서드를 호출할 때 \$() 대신 N()을 사용해도 됩니다. Natural-CORE 의 함수들은 기본적으로 Natural-JS 만을 위해서 제작 되었으나 개발할 때 공통적으로 사용되어야 할 함수들은 다음과 같이 사용할 수 있습니다.

#### 나. 함수(Methods)

Name	Arguments	Type	Return	Description
N	-	-	JWS <sup>7)</sup>	Natural 의 플러그인 형태로 제공되는 라이브러리들과 N클래스의 멤버변수(함수)가 정의되어있는 핵심 클래스입니다. N(selector <sup>8)</sup> or function, object, HTML Element(이하 element))를 호출하면 jQuery 의 init 메서드가 호출되어 Natural-JS 와 jQuery가 동시에 초기화(Initialize) 됩니다. ex) var divs = N("div");
	selector	JWS, selector or function, object, element	-	JSON data나 selector 등.
	context	JWS	-	element를 지정할 때 지정된 element 가 속해있는 element 영역을 지정할 수 있습니다.
N.locale	-	-	undefined	Natural-JS 의 현재 로케일을 설정 합니다. 설정한 로케일에 따라 프레임워크에서 사용자에게 보여주는 메시지도 다국어처리가 됩니다. 현재 등록된 다국어 메시지 리소스는 en_US, ko_KR 만 있습니다. 메시지 세트는 config에서 손쉽게 추가 할 수 있습니다.
	str	String		"en_US", "ko_KR" 등의 로케일 문자열
N.error	-	-	ErrorThrow n	에러를 발생 시키고 에러 메시지를 브라우저 콘솔창에 보여줍니다.
	msg	String		에러 텍스트 메시지
	e	ErrorThrown	-	에러 발생시 생성되는 ErrorThrown 객체를 입력 하면 Error Stack Trace 로그를 브라우저 콘솔창에 보여줍니다. (Chrome 에서만 지원)
N.warn	-	-	undefined	콘솔창에 Warning 메시지를 보여 줍니다 주) 브라우저에서 console.warn을 지원하지 않으면 메시지가 표시되지 않습니다.
	msg	String	-	텍스트 메시지
N.log	-	-	undefined	콘솔창에 로깅(Logging) 메시지를 보여 줍니다 주) 브라우저에서 console.log를 지원하지 않으면 메시지가 표시되지 않습니다.
	msg	String	-	텍스트 메시지
N.browser.cookie	-	-	undefined	쿠키를 생성 합니다.
	name	String	-	쿠키명
	value	String	-	쿠키값

	expiredays	int	-	만료일
	domain	String	-	쿠키가 생성될 도메인
N.browser.removeCookie	-	-	undefined	생성된 쿠키를 제거 합니다.
	name	String	-	쿠키명
	domain	String	-	해당 쿠키가 생성된 도메인
N.browser.msieVersion	-	-	int	현재 접속한 Microsoft Internet Explorer(이하 MSIE)의 버전을 가져옵니다. MSIE가 아닐 경우 0을 반환 합니다.
N.browser.is			boolean	입력한 브라우저명과 현재 접속한 브라우저가 일치하면 true를 반환 합니다.
	name	String	-	브라우저명(opera, firefox, safari, chrome, ie)
N.message.get	-	-	-	메시지 리소스(config 에 정의)에서 현재 로케일에 맞게 메시지를 가져옵니다.
	resource	JSON Object	-	<p>메시지 리소스</p> <p>메시지 리소스는 JSON Object 형태로 만들면 되고 하위에 메시지 세트를 구성해야 함.</p> <p>ex)</p> <pre>var message = {   "ko_KR" : {     key : " 메시지{0} "</pre> <p>},</p> <pre>  "en_US" : {     key : " message{0} "</pre> <p>}</p> <p>}</p> <p>ex)</p> <pre>var msg   = N.message.get(message, "key", [" 입니다"]);</pre> <p>msg의 값 : "메시지 입니다"</p>
	key	String	-	메시지 키
	vars	Array	-	메시지 변수를 치환할 값
N().toArray	-	-	Array	JWS를 순수한 Array 객체로 반환 합니다.
N().instance	-	-	Object	<p>Natural-JS 는 기본적으로 컴포넌트나 라이브러리들에 의해 생성된 인스턴스를 대상 HTML element(이하 요소)가 있다면 대상요소에 인스턴스를 저장 합니다.</p> <p>N().instance 메서드는 이 인스턴스를 요소에 저장하고 가져오는 역할을 수행 합니다.</p> <p>인자중에 name만 입력하면 인스턴스를 가져오고 name, instance를 같이 입력하면 해당이름으로 해당요소에 instance를 저장 합니다. instance 인자로 callback function 이 입력되면 여러 인스턴스를 한번에 가져올 수 있습니다.</p> <p>ex)</p> <pre>var single = ("div#sample").instance("grid"); var multy = ("div.sample").instance("form",   function(i, inst) {     //this 는 각각의 instance     var insts = inst;   });</pre>

	name	String	-	인스턴스 명
	instance	Object or callback function(이하 callback)	-	인스턴스 or 인스턴스를 가져올 callback 주) callback의 인자는 index(arguments[0])와 각각의 instance(arguments[1])가 들어옵니다. 이 callback 안의 this 는 각각의 instance를 가리킵니다.

7) JWS : jQuery Wrapped Set(jQuery())를 실행하면 반환되는 jQuery 확장 객체) or jQuery selector

8) selector : jQuery에서 CSS selector 형식으로 지정하는 String 형태의 문자열이나 JSON Object, Array, HTML Element, Function 등의 Object

## << Natural-ARCHITECTURE ver. 0.8.0.0 >>

### 2. Controller

#### 가. 개요

Controller(이하 N.cont) 는 CVC Architecture Pattern(이하 CVC 패턴)의 Controller 영역을 구현한 클래스로서 Communicator 와 View 를 연결 해 주고 View 의 HTML 요소(element)들을 제어하는 역할을 합니다. N.cont 는 다음과 같이 페이지의 View 영역 바로 아래 선언되어야 합니다.

ex)

```
<div id="view">
  View 영역
</div>
<script type="text/javascript">
  N("div#view").cont({
    init : function(view, request) {
      //JSON 형태의 Class 로 정의
    }
  });
</script>
```

위와 같이 N.cont 를 선언하면 View 영역의 Load 가 완료된 후 N.cont의 인자로 들어온 Controller 클래스에 init 메서드가 정의 되어 있다면 init 메서드를 프레임워크에서 실행 시켜 줍니다. init 의 인자(Arguments)로는 View영역의 HTML 요소들이 담긴 JWS 객체와 Communicator.request(Communicator 참조) 객체를 넣어 줍니다. 또한 Controller Class 의 초기화(Initialize)가 완료되면 View영역의 최상위 Element 에 해당 Controller 인스턴스(instance)를 넣어줘 N("페이지 최상위 Element").instance("cont") 로 View 영역에서 참조 할 수 있습니다.

#### 나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.cont	-	-	(Controller) callback	Controller를 초기화 합니다. N.cont는 컨트롤러 클래스를 등록하고 관리 해 주는 역할을 수행 합니다.
	obj	JWS	-	View 영역을 selector로 지정 합니다.
	callback	JSON Object Class	-	Controller 클래스를 등록 합니다.
N(selector).cont	-	-	-	N.cont를 플러그인 방식으로의 초기화 방식입니다. N.cont 와 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.cont()의 첫 번째 인자로 지정 됩니다. 이와같은 방식으로 사용하면 컴포넌트를 초기화하는 여러 가지 설정들을 자동으로 세팅해 주므로 Natural-JS 에서는 이와같은 방식의 사용을 추천합니다.
	callback	JSON Object Class	-	N.cont 의 callback 인자(argument) 와 같습니다.

#### 다. Controller 기본 함수(callback default method)

Name	Arguments	Type	Return	Description
init	-	-	User define	N.cont 에 의해 등록된 callback 이 초기화 된 후 최초 실행 되는 함수입니다.
	view	JWS	-	View 영역의 요소들을 반환 합니다.



	request	Communicator.request instance	-	Communicator.request 인스턴스를 반환하여 현재 페이지의 요청 정보를 볼 수 있습니다.
--	---------	-------------------------------	---	----------------------------------------------------------

### 3. Communicator(N.comm)

#### 가. 개요

Communicator(이하 N.comm) 는 CVC 패턴의 Communicator 영역을 구현한 클래스로서 서버와의 통신을 담당하는 Natural-JS의 공통모듈입니다. Communicator 는 jQuery.ajax 모듈을 사용해서 Ajax로 서버와의 통신을 합니다. Ajax 통신외에 요청마다 Communicator.request 객체가 생성되어 웹 UI 안의 논리적인 페이지 (탭, 임베드, 팝업등)의 Controller간 데이터를 송·수신 할 수 있고 불러온 페이지의 Controller 에서 요청정보를 확인 할 수 있습니다.

#### 나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.comm	-	-	N.comm	N.comm 객체를 생성 합니다.
	obj	JWS(selector or element) or Object(JSON) or JWS(Object(JSON))	-	<ul style="list-style-type: none"> <li>JWS(selector or element) 형태의 인자로 지정하고 두 번째인자로 특정 페이지의 URL을 지정하면 가져온 HTML 콘텐츠를 지정한 요소 안에 넣어줍니다.</li> <li>Object(JSON) 형태의 인자로 지정하면 Communicator.request 의 옵션으로 지정 됩니다.</li> <li>JWS(Object(JSON)) 형태의 인자로 지정하면 지정한 값을 서버로 전송한 파라미터로 인식하여 서버에 요청을 보낼 때 전송 해 줍니다.</li> </ul> <p>ex) PAGE: N.comm(N("#contents"), "page.html").submit() DATA: N.comm("data.json").submit(function(data) {})</p>
	url	String	-	호출할 URL을 지정 합니다.
N(obj).comm	-	-	N.comm	N.comm을 플러그인 방식으로의 초기화 방식입니다. N.comm과 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.comm()의 첫 번째 인자로 지정 됩니다.
	url	String	-	N.comm 의 url 인자 와 같습니다.

#### 다. 기본옵션(Default options)

obj(arguments[0])가 JSON 오브젝트 타입의 옵션이면 Communicator.request(이하 request) 가 생성되어서 request 의 옵션으로 지정 됩니다. request 의 기본옵션 부분들과 동일하므로 "마. Communicator.request"의 기본옵션 부분을 참고하기 바랍니다.

#### 라. 함수(Methods)

Name	Arguments	Type	Return	Description
submit	-	-	N.comm	컨트롤러의 실행정보가 정의되어 초기화 된 다음 서버에 요청을 보내는 메서드입니다.
	callback	Function	-	서버에서 요청이 처리되고 응답이 완료된 다음 실행되는 콜백 함수입니다.

## 마. Communicator.request

### 1) 개요

Communicator.request(이하 request) 는 단독으로 사용될 수 없고 N.comm 으로 요청을 보낼때마다 N.comm 에 의해서 생성 되는 모듈입니다. request는 요청에 대한 옵션을 지정 할 수 있고 요청정보를 가지고 있고 페이지간 데이터 전달을 위한 attr 메서드가 구현 되어 있습니다.

### 2) 생성자(Constructor)

개발시 직접 컨트롤 할 수 없고 Communicator 에 의해 생성되므로 설명을 생략 합니다.

### 3) 기본옵션(Default options)

Name	Type	Default	Required	Description
contentType	String	application/json; charset=utf-8	true	Ajax request 에 대한 contentType을 지정 합니다(jQuery.ajax 의 옵션)
cache	boolean	false		브라우저 캐시 적용 유무(jQuery.ajax 의 옵션)
async	boolean	true		Ajax 요청을 비동기 방식으로 처리할지 동기방식으로 처리할지 지정 합니다(jQuery.ajax 의 옵션)
type	String	POST		Ajax 요청에 대한 method를 지정 합니다(jQuery.ajax 의 옵션)
data	JSON Array <sup>9)</sup> or JSON object	null		요청시 파라미터를 지정 합니다(jQuery.ajax 의 옵션)
dataType	String	"json"		응답받을 데이터의 형태를 지정합니다(jQuery.ajax 의 옵션)
urlSync	boolean	true		요청할때의 페이지 URL 과 요청받을때의 페이지 URL 이 같은지 비교하여 다르면 응답을 차단 합니다.
crossDomain	boolean	false		JSONP 같은 기술을 사용하여 다른 도메인의 자원을 가져오게 할지 여부를 설정 합니다(jQuery.ajax 의 옵션).
append	boolean	false		HTML 콘텐츠를 가져올 때 지정한 영역에 더할건지 덮어 쓸건지를 결정 합니다.
effect	boolean or Array	false		HTML 콘텐츠를 가져올 때 전환 효과를 보여줄지 여부를 설정 합니다. 효과를 적용하지 않으려면 false를 입력하고 효과를 입력하려면 다음과 같이 입력합니다. ["효과명" , "시간(ms)" , "효과가 끝날 때 마다 실행될 콜백함수"] ex) ["slideDown", 300, null]

※ 위 기본옵션 외에 N.comm은 jQuery \$.ajax 모듈을 사용하여 \$.ajax에서 제공하는 옵션중 이벤트콜백 관련 부분(beforeSend, success, error, complete)을 제외한 모든 옵션을 그대로 제공합니다.

### 4) 함수(Methods)

Name	Arguments	Type	Return	Description
attr	-	-	Controller.request or Value object	호출한 페이지에 값을 전달할 때 사용합니다. ex) * 값전달 : N(selector).comm(url).request.attr("param", { paramA : "A" }).submit()

9) JSON object 가 담겨있는 Array 객체

				* 전달된 값 참고 : (불러진 페이지의 Controller) .request.attr("param")
	name	String	-	전달할 값의 key를 입력 합니다.
	obj	Object	-	전달할 값에 해당하는 object를 지정 한다. obj 인자를 지정하지 않으면 값을 가져오고 지정하면 해당 값으로 저장합니다.
param	-	-	String	현재 페이지의 URL에 표출된 GET 파라미터에서 값을 추출 합니다.
	name	String	-	GET 파라미터의 name을 지정한다. 지정하지 않으면 GET 파라미터 전체 값을 JSON Object 형태로 반환 합니다.
get	-	-	Value Object	요청마다 생성되는 request 객체에서 요청정보를 가져옵니다.
	key	String	-	key를 지정하지 않으면 전체 요청정보를 가져오고 key 값을 지정하면 해당 정보만 가져옵니다. key 값은 기본옵션과 같고 referrer(호출한 페이지 URL) 값만 추가 됩니다.

#### 4. Context(N.context)

##### 가. 개요

Context(이하 N.context)는 Natural-JS 의 생명주기(Life-Cycle, 페이지가 표시되고 다른페이지로 redirect 되기 전까지) 안 에서의 데이터 저장소 역할을 한다. Natural-JS 의 환경설정 값(Config)과 메시지등이 N.context 객체에 저장 된다.

##### 나. 함수(Methods)

Name	Arguments	Type	Return	Description
attr	-	-	N.comm	
	name	String	-	저장할 값의 key를 입력 합니다.
	obj	Object	-	저장할 값에 해당하는 object를 지정 한다. obj 인자를 지정하지 않으면 값을 가져오고 지정하면 해당 값으로 저장합니다.

#### 5. natural.config.js

##### 가. 개요

Natural-JS 의 환경설정 파일입니다. 설정값은 N.context 에 저장 되며 각 패키지 별로 따로 저장 됩니다.

N.context.attr("core") : Natural-CORE 패키지에 속한 라이브러리들의 기본 설정 값.

N.context.attr("architecture") : Natural-ARCHITECTURE 패키지에 속한 라이브러리들의 기본 설정 값.

N.context.attr("data") : Natural-DATA 패키지에 속한 라이브러리들의 기본 설정 값.

N.context.attr("ui") : Natural-UI 패키지에 속한 라이브러리들의 기본 설정 값.

세부 설정값에 대한 내용은 각 컴포넌트와 라이브러리별 기본옵션과 natural.config.js 파일 안의 설명을 참고 하기 바랍니다.

## << Natural-DATA ver. 0.8.0.0 >>

### 6. Formater(N.formater)

#### 가. 개요

Formater 는 지정한 데이터셋을 정의한 룰을 기반으로 데이터셋 전체를 양식화하여 반환 합니다. 룰(rules) 대신 입력요소가 담겨있는 블록요소(Block element)를 지정하면 입력요소에 선언형 옵션으로 지정된 룰을 기반으로 양식화하여 반환 해 주는 라이브러리입니다.

⚠ 룰 대신 입력요소를 지정하면 해당 입력요소들마다 "format", "unformat" 이벤트를 바인드(bind) 해 줍니다. 해당 입력요소를 가져다가 jQuery 의 trigger 메서드로 실행하면 해당입력요소에 입력된 값을 실시간으로 format, unformat 할 수 있습니다.

ex) N("input[name='formatEx']").trigger("format or unformat");

#### 나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.formater	-	-	N.formater	N.formater 객체를 생성 합니다.
	obj	JSON Object	-	양식화할 데이터를 지정 합니다. 데이터는 JSON Array 타입으로 입력 해야 합니다.
	rules	JSON Object or JWS(elements)	-	<p>룰을 JSON Object 형태로 지정 하거나 JSON룰 ex)</p> <pre>{   "numeric" : [{"trimtoempty"}, ["numeric", "#,###,##0000"]],   "generic" : [{"trimtoempty"}, ["generic", "@@ABCD"]],   "limit" : [{"trimtoempty"}, ["limit", "13", "..."]],   "etc" : [{"date", 12}] }</pre> <p>선언형옵션으로 룰이 지정된 입력요소나 입력요소 블록을 지정 합니다. 입력요소를 ex)</p> <pre>&lt;input id="limit" type="text" data-format=[{"trimtoempty"}, ["limit", "13", "..."] ] /&gt;</pre> <p>⚠ 룰 속성은 다음과 같이 정의 합니다. [ ["룰명", "룰arguments[0]", "룰 arguments[1]", ... ]</p>
N(obj).formater	-	-	N.formater	N.formater를 플러그인 방식으로의 초기화 방식입니다. N.formater와 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.formater()의 첫 번째 인자로 지정 됩니다.
	rules	JSON Object or JWS(elements)	-	N.formater의 rules 인자 와 같습니다.

#### 다. 함수(Methods)

Name	Arguments	Type	Return	Description
format	-	-	양식화된 데이터셋(JSON)	데이터 양식화를 실행합니다.
	row	int	-	양식화할 데이터의 row, 입력하지 않으면 무조건 첫 번째 데이터를 양식화 합니다.

unformat	-	-	양식화 되지않은 데이터(JSON)	원(元)데이터를 반환합니다.
	row	int	-	양식화 되지않은 원데이터를 반환 할 row
	key	String	-	양식화 되지않은 원데이터(JSON Object)의 key 값

#### 라. 포맷 룰(Format rules)

옵션명	설명
["ssn"]	주민등록번호 형식으로 변경
["cno"]	사업자등록번호 형식으로 변경
["upper"]	대문자로 변경
["lower"]	소문자로 변경
["capitalize"]	첫번째 영문자를 대문자로 변경
["zipcode"]	우편번호 형식으로 변경
["phoneNum"]	전화번호 형식으로 변경
["realnum"]	의미없는 0 제거 ex) 0100.0 -> 100 0100.10 -> 100.1
["trimToEmpty"]	trim 후 값이 없으면 공백으로 표시
["trimToZero"]	trim 후 값이 없으면 0 으로 표시
["trimToVal", "valStr"]	trim 후 값이 없으면 valStr 값으로 표시
["date", 4 6 8 10 12 14 "month" "date"]	날짜 형식으로 표시. 기본 날짜형식은 N.context.attr("data").formater.date에서 설정 가능 합니다. 다음은 두 번째 값들에 대한 설명 입니다. * month : Monthpicker 표시 * date : Datepicker 표시 * 4:년 * 6:년-월 * 8:년-월-일 * 10:년-월-일 시 * 12:년-월-일 시:분 * 14:년-월-일 시:분:초
["limit, cutLength, "replaceStr"]	문자열을 cutLength 만큼 자르고 replaceStr로 치환
["replace", "targetStr", "replaceStr"]	targetStr을 replaceStr로 치환
["lpad", length "fillStr"]	length만큼 왼쪽부터 replaceStr문자로 채움
["rpad", length "fillStr"]	length만큼 오른쪽부터 replaceStr문자로 채움
["generic", "사용자서식"]	엑셀 사용자지정 서식과 같이 사용자 서식을 이용하여 문자열을 양식화 * # : 숫자 * @ : 영문자 ex) * 10000123 -> ₩#,###.### -> ₩1,000.123

["numeric" "사용자서식", "option"]	엑셀 사용자지정 서식과 같이 사용자 서식을 이용하여 숫자를 양식화 option 값을 이용하여 소숫점을 다음과 같이 처리 * option 값 ceil : 무조건올림 (소수점 처리가능) floor : 무조건버림(소수점 처리가능) round : 반올림(소수점 처리가능) ex) * 1,2 -> #,##0 -> 1,200
----------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 7. Validator(N.validator)

### 가. 개요

Validator 는 지정한 데이터셋을 정의한 룰을 기반으로 데이터셋 전체를 검증하여 검증 데이터 세트를 반환해 주는 라이브러리입니다. 룰(rules) 대신 입력요소가 담겨있는 블록요소(Block element)를 지정하면 입력요소에 선언형옵션으로 지정된 룰을 기반으로 검증을 하여 검증에 통과하지 못했다면 입력요소 바로옆에 툴팁으로 검증 메시지를 보여줍니다.

### 나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.validator	-	-	N.validator	N.validator 객체를 생성 합니다.
	obj	JSON Object	-	검증할 데이터셋을 지정 합니다. 데이터는 JSON Array 타입으로 입력 해야 합니다.
	rules	JSON Object or JWS(elements)	-	룰을 JSON Object 형태로 지정 하거나 ex) { "numeric" : [{"required"}, ["integer+commas"]], "generic" : [{"required"}, ["korean"]], "limit" : [{"required"}, ["alphabet"]] } 선언형옵션으로 룰이 지정된 입력요소나 입력요소 블록을 지정 합니다. ex) <input id="numeric" type="text" data-validate=[{"required"}, ["integer+commas"]]/>  ⚠ 룰 속성은 다음과 같이 정의 합니다. [ ["룰명", "룰arguments[0]", "룰 arguments[1]", ... ]
N(obj).validator	-	-	N.validator	N.validator를 플러그인 방식으로의 초기화 방식입니다. N.validator와 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.validator()의 첫 번째 인자로 지정 됩니다.
	rules	JSON Object or JWS(elements)	-	N.validator의 rules 인자 와 같습니다.

### 다. 함수(Methods)

Name	Arguments	Type	Return	Description
validate	-	-	데이터의 컬럼별 검증결과 데이터	데이터 검증을 실행합니다.

	row	int	-	검증할 데이터의 row, 입력하지 않으면 무조건 첫 번째 데이터를 양식화 합니다.
--	-----	-----	---	-----------------------------------------------------

#### 라. 검증 룰(Validate rules)

옵션명	설명
["required"]	필수입력
["alphabet"]	영문자만 입력 가능
["integer"]	숫자(정수)만 입력가능
["Korean"]	한글만 입력 가능
["alphabet+integer"]	영문자와 숫자(정수)만 입력 가능(+로 구분하고 순서는 관계없음)
["integer+korean"]	숫자(정수)와 한글만 입력 가능(+로 구분하고 순서는 관계없음)
["alphabet+korean"]	영문자와 한글만 입력 가능(+로 구분하고 순서는 관계없음)
["alphabet+integer+korean"]	영문자, 숫자(정수), 한글만 입력 가능(+로 구분하고 순서는 관계없음)
["integer+dash"]	숫자(정수), 대시(-)만 입력가능(+로 구분하고 순서는 관계없음)
["integer+commas"]	숫자(정수), 콤마(,)만 입력가능(+로 구분하고 순서는 관계없음)
["number"]	숫자(정수), 대시(,), 점(.)만 입력가능
["decimal(length)"]	(유한)소수 length 번째 자리까지 입력 가능
["email"]	e-mail 형식에 맞는지 검사
["url"]	URL 형식에 맞는지 검사
["zipcode"]	우편번호 형식에 맞는지 검사
["frn"]	외국인등록번호 형식에 맞지 검사
["ssn"]	주민등록번호 형식에 맞지 검사
["cno"]	사업자등록번호 형식에 맞는지 검사
["cpno"]	법인번호 형식에 맞는지 검사
["date"]	날짜 형식에 맞는지 검사
["time"]	시간 형식에 맞는지 검사
["accept", "word"]	word 값만 입력 가능
["match", "word"]	word 값이 포함된 값만 입력 가능
["acceptFileExt", "ext"]	ext 값이 포함된 확장자만 입력 가능
["notAccept", "word"]	word 값만 입력 불가능
["notMatch", "word"]	word 가 포함된 값은 입력 불가능
["notAcceptFileExt", "ext"]	ext 가 포함된 확장자는 입력 불가능
["equalTo", "input"]	input 의 값과 같아야 함
["maxlength", "length"]	length 글자 이하만 입력 가능
["minlength", "length"]	length 글자 이상만 입력 가능
["rangelength", "startLength", "endLength"]	startLength 글자 에서 endLength 글자 까지만 입력 가능
["maxbyte", "byteLength"]	length 바이트 이하만 입력 가능
["minbyte", "byteLength"]	length 바이트 이상만 입력 가능
["rangebyte", "startByteLength", "endByteLength"]	startByteLength 바이트 에서 endByteLength 바이트 까지만 입력 가능
["maxvalue", "value"]	value 이하의 값만 입력 가능

["minvalue", "value"]	value 이상의 값만 입력 가능
["rangevalue", "stratValue,endValue"]	stratValue 값 에서 endValue 값 까지만 입력 가능
["regexp", "regexp", "message"]	입력한 regexp 의 조건으로 검사 후 message 값 표시

## 8. Natural-DATA Libraries

### 가. 개요

Natural-DATA 패키지에는 JSON Array 타입의 데이터를 조작하고 정렬할 수 있는 기능을 제공 합니다.

### 나. 함수(Methods)

Name	Arguments	Type	Return	Description
N.data .refine	-	-	정제된 JSON Array	서버나 JSON 데이터가 담긴 파일에서 가져온 JSON 데이터를 Natural-JS에서 사용할수 있는 형태의 JSON 형태로 만들어 줍니다.
	obj	JSON Array	-	정제 할 데이터
	listId	String	-	서버나 JSON 데이터가 담긴 파일에서 여러개의 리스트를 반환했을경우의 해당 리스트의 ID 값
N.data .filter	-	-	필터링된 JSON Array	지정한 조건으로 JSON Array를 필터링 합니다.
	arr	JSON Array		필터링 할 데이터
	queryString	String	-	필터링 조건을 입력 합니다. 조건은 다음과 같이 자바스크립트 비교 문법으로 지정 합니다. ex) * queryString = 'data.키명 == "비교값"; N.data.filter([ { name : "홍길동" }, { name : "Mike" } ], 'data.name == "홍길동"); 쉽게 생각하면 queryString 바깥에 if() 문이 감싸져있고 data 라고 이름지어진 row data를 가지고 비교한다고 생각 하면 됩니다.
N.data .sort	-	-	정렬된 JSON Array	
	arr	JSON Array	-	정렬 할 데이터
	key	String	-	정렬 기준이 될 key(컬럼) 명
	reverse	boolean	-	오름차순(true), 내림차순(false) 선택
N(obj) .datarefine	-	-	정제된 JSON Array	N.data.refine을 플러그인 방식으로 초기화 하는 방식입니다. N.data.refine과 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.data.refine()의 첫 번째 인자로 지정 됩니다.
	listId	String	-	N.data.refine의 listId 인자 와 같습니다.
N(obj) .datafilter	-	-	필터링된 JSON Array	N.data.filter를 플러그인 방식으로 초기화 하는 방식입니다. N.data.filter와 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.data.filter()의 첫 번째 인자로 지정 됩니다.
	queryString	JSON Array		N.data.filter의 queryString 인자 와 같습니다.



N(obj) .datasort	-	-	정렬된 JSON Array	N.data.sort를 플러그인 방식으로 초기화 하는 방식입니다. N.data.sort와 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.data.sort()의 첫 번째 인자로 지정 됩니다.
	key	String	-	N.data.sort의 key 인자 와 같습니다.
	reverse	boolean	-	N.data.sort의 reverse 인자 와 같습니다.

## << Natural-UI ver. 0.8.0.0 >>

### 9. Alert(N.alert)

#### 가. 개요

Alert(이하 N.alert) 는 window.alert 이나 window.confirm과 같이 메시지를 다이얼로그로 표현하기 위한 컴포넌트 입니다.

#### 나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.alert	-	-	N.alert	N.alert 객체를 생성합니다.
	context	window or JWS	-	기본옵션의 context 옵션과 같습니다.
	msg	(JWS, HTML Tag String) or Object	-	기본옵션의 msg 옵션과 같습니다.
	vars	Array	Alert	기본옵션의 var 옵션과 같습니다.
N(obj).alert	-	-	N.alert	N.alert을 플러그인 방식으로의 초기화 방식입니다. N.alert과 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.alert()의 첫 번째 인자로 지정 됩니다.
	msg	(JWS, HTML Tag String) or Object	-	N.alert의 msg 옵션과 같습니다.
	vars	Array	-	N.alert의 var 옵션과 같습니다.

#### 다. 기본옵션(Default options)

Name	Type	Default	Required	Description
context	window or JWS	obj	○	Alert 의 배경이 되는 콘텐츠를 가려주는 overlay 가 적용될 부분을 jQuery selector 를 사용하여 지정 합니다. window 객체를 지정하면 화면 전체를 덮는 Modal overlay 가 표시되고 JWS를 입력하면 지정한 요소만 덮어줍니다.
msg	(JWS, HTML Tag String) or Object	undefined	○	Alert 로 표시할 메시지 JWO, HTML 문자열 이나 HTML 요소 모두 인식 합니다.
vars	Array	undefined		메시지의 변수를 vars로 지정한 값으로 치환 해줍니다. ex) msg : "{0} 저는 {1}입니다." vars : ["안녕하세요", "홍길동"] ->"안녕하세요 저는 홍길동 입니다"
width	int	0		메시지 다이얼로그의 넓이
height	int	0		메시지 다이얼로그의 높이
title	String	undefined		메시지 다이얼로그의 제목, title 옵션이 undefined 이면 타이틀바를 표시하지 않고 context 값으로 지정한 요소의 title 속성에 값이 있다면 해당 값으로 지정 됩니다.

button	boolean	true		버튼 표시 유무
modal	boolean	true		모달 적용 유무
onOk	Function	null		확인버튼을 눌렀을때 발생하는 이벤트
onCancel	Function	null		취소버튼을 눌렀을때 발생하는 이벤트
alwaysOnTop	boolean	false		메시지 다이얼로그를 항상 페이지의 최상위로 항상 유지하기 위한 옵션
confirm	boolean	false		true 지정 시 확인할 수 있도록 [확인], [취소] 버튼을 표시하고 false 지정 시 [취소] 버튼만 표시합니다.

#### 라. 함수(Methods)

Name	Arguments	Type	Return	Description
show	-	-	N.alert	메시지 다이얼로그를 보여 줍니다.
hide	-	-	N.alert	메시지 다이얼로그를 감춥니다.
remove	-	-	N.alert	메시지 다이얼로그를 표현하는 HTML Element들을 제거합니다.

### 10. Button(N.button)

#### 가. 개요

Button(이하 N.button) 은 a , input button , button 과 같은 버튼 요소들을 공통된 모양과 공통된 동작을 가진 버튼을 만들어 컴포넌트입니다.

#### 나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.button	-	-	N.button	N.button 객체를 생성합니다.
	context	window or JWS	-	기본옵션의 context 옵션과 같습니다.
	opts	JSON Object	-	기본 옵션을 JSON Object 형태로 지정 합니다.
N(obj).button	-	-	N.button	N.button을 플러그인 방식으로의 초기화 방식입니다. N.button과 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.button()의 첫 번째 인자로 지정 됩니다.
	opts	JSON Object	-	N.button의 opts 옵션과 같습니다.

#### 다. 기본옵션(Default options)

Name	Type	Default	Required	Description
context	JWS	obj	○	버튼컴포넌트화 할 버튼 요소를 지정합니다.
size	String	"medium"		크기 smaller, small, medium, large, big
color	String	"white"		색깔 white, blue, skyblue, gray
disable	boolean	false		초기화시 비활성 여부
effect	boolean	true		활성, 비활성 시 Fade 효과 적용 유무

#### 라. 선언형옵션(Declarative options)

Name	Type	Default	Description
size	String	"medium"	크기 smaller, small, medium, large, big

color	String	"white"	색깔 white, blue, skyblue, gray
disable	boolean	false	초기화시 비활성 여부
effect	boolean	true	활성, 비활성 시 Fade 효과 적용 유무

#### 마. 함수(Methods)

Name	Arguments	Type	Return	Description
enable	-	-	N.button	버튼을 활성화 합니다.
disable	-	-	N.button	버튼을 비활성화 합니다.

### 11. DatePicker(N.datepicker)

#### 가. 개요

DatePicker(이하 N.datepicker)는 달력 다이얼로그를 입력요소 주변에 표시하여 마우스를 이용하여 날짜를 입력할 수 있게 해주는 컴포넌트입니다.

#### 나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.datepicker	-	-	N.datepicker	N.datepicker 객체를 생성합니다.
	context	window or JWS	-	기본옵션의 context 옵션과 같습니다.
	opts	JSON Object	-	기본 옵션을 JSON Object 형태로 지정 합니다.
N(obj).datepicker	-	-	N.datepicker	N.datepicker을 플러그인 방식으로의 초기화 방식입니다. N.datepicker와 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.datepicker()의 첫 번째 인자로 지정 됩니다.
	opts	JSON Object	-	N.datepicker의 opts 옵션과 같습니다.

#### 다. 기본옵션(Default options)

Name	Type	Default	Required	Description
context	JWS	obj	○	데이트피커를 적용할 입력 요소를 지정합니다.
monthonly	boolean	false		true 이면 Monthpicker를 표시합니다.
focusin	boolean	true		날짜 입력요소에 자동으로 focusin 이벤트를 bind 하여 DatePicker를 표시할 것인지 여부를 지정 합니다.
onSelect	Function	null		날짜나 월(monthonly:true)이 선택되었을 때 발생하는 이벤트입니다.

#### 라. 함수(Methods)

Name	Arguments	Type	Return	Description
show	-	-	N.datepicker	DatePicker를 표시 합니다.
hide	-	-	N.datepicker	DatePicker를 감춥니다.

### 12. Popup(N.popup)

#### 가. 개요

Popup(이하 N.popup) 은 지정한 요소나 다른페이지를 레이어팝업 형태로 만들어 주는 컴포넌트입니다.  
url 옵션으로 지정한 페이지를 가져올때 가져온 페이지의 Controller 에는 caller(호출한 팝업의

instance(N.popup)) 와 opener(팝업을 호출한 페이지(부모페이지)의 Controller) 속성이 추가 되어 부모 페이지와 자신을 호출한 팝업 객체를 직접 제어할 수 있습니다.

## 나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.popup	-	-	N.popup	N.popup 객체를 생성합니다.
	context	window or JWS	-	기본옵션의 context 옵션과 같습니다.
	opts	JSON Object	-	기본 옵션을 JSON Object 형태로 지정 합니다.
N(obj).popup	-	-	N.popup	N.popup을 플러그인 방식으로의 초기화 방식입니다. N.popup과 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.popup()의 첫 번째 인자로 지정 됩니다.
	opts	JSON Object	-	N.popup의 opts 옵션과 같습니다.

## 다. 기본옵션(Default options)

Name	Type	Default	Required	Description
context	JWS	obj		같은 페이지 내에 있는 블록(Block) 요소를 지정 합니다.
url				
title	String	undefined		팝업의 제목, title 옵션이 undefined 이면 타이틀바를 표시하지 않고 context 값으로 지정한 요소의 title 속성에 값이 있다면 해당 값으로 지정 됩니다.
button	boolean	true		버튼 표시 유무
modal	boolean	true		모달 적용 유무
height	int	0		팝업의 높이
width	int	0		팝업의 넓이
alwaysOnTop	boolean	false		팝업을 항상 페이지의 최상위로 항상 유지하기 위한 옵션
onOk	Function	null		확인버튼을 눌렀을때 발생하는 이벤트
onCancel	Function	null		취소버튼을 눌렀을때 발생하는 이벤트
				팝업이 열릴 때마다 발생하는 이벤트
onOpen	String	null		이벤트 콜백 메서드는 팝업페이지의 Controller 에 구현되어야 합니다. 메서드명은 Controller 에 정의한 메서드명칭만 String 으로 지정 합니다. 이벤트 콜백의 메서드는 인자로 onOpenData 가 들어옵니다.
				⚠ url을 지정하여 다른 페이지를 가져왔을때만 사용가능
onOpenData	Object	null		onOpen 이벤트의 인자로 들어가 팝업에 전달될 데이터
				⚠ url을 지정하여 다른 페이지를 가져왔을때만 사용가능
onClose	Function	null		팝업이 닫힐 때마다 발생하는 이벤트. 이벤트 콜백의 메서드는 인자로 onCloseData 가 들어옵니다.  팝업 페이지에서 자신을 닫을때는 Controller에서 다음과 같이 처리 합니다. this.caller.colse(onCloseData)

				⚠ url을 지정하여 다른 페이지를 가져왔을때만 사용가능
onCloseData	Object	null		onClose 이벤트의 인자로 들어가 팝업의 부모페이지에 전달될 데이터
				⚠ url을 지정하여 다른 페이지를 가져왔을때만 사용가능

#### 라. 함수(Methods)

Name	Arguments	Type	Return	Description
show	-	-	N.popup	팝업을 보여 줍니다.
hide	-	-	N.popup	팝업을 감춥니다.
remove	-	-	N.popup	팝업을 표현하는 HTML Element들을 제거합니다.

### 13. Tab(N.tab)

#### 가. 개요

Tab(이하 N.tab) 지정한 HTML 탭 템플릿을 기반으로 탭을 만들어 주는 컴포넌트입니다.

#### 나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.tab	-	-	N.tab	N.tab 객체를 생성합니다.
	context	window or JWS	-	기본옵션의 context 옵션과 같습니다.
	opts	JSON Object	-	기본 옵션을 JSON Object 형태로 지정 합니다.
N(obj).tab	-	-	N.tab	N.tab을 플러그인 방식으로의 초기화 방식입니다. N.tab과 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.tab()의 첫 번째 인자로 지정 됩니다.
	opts	JSON Object	-	N.tab의 opts 옵션과 같습니다.

#### 다. 기본옵션(Default options)

Name	Type	Default	Required	Description
context	JWS	obj	○	같은 페이지 내에 있는 탭과 탭 콘텐츠를 감싸고 있는 "DIV" 요소를 지정 합니다. 탭플릿 요소는 다음과 같이 작성 합니다. ex) <pre>&lt;div id="tab"&gt;   &lt;ul&gt;     &lt;!-- 탭요소는 li 태그 로 표현 --&gt;     &lt;li&gt;&lt;a href="#tab1"&gt;Tab1&lt;/a&gt;&lt;/li&gt;     &lt;li data-opts='{ "url" : "prototype/tab1.html" }'&gt;       &lt;!-- 탭 콘텐츠와의 연결은 a TAG 로 표현             href 속성에 해당 콘텐츠의 #id 입력 --&gt;       &lt;a href="#tab2"&gt;Tab2&lt;/a&gt;     &lt;/li&gt;     &lt;li data-opts='{ "url" : "prototype/tab2.html" }'&gt;       &lt;a href="#tab3"&gt;Tab3&lt;/a&gt;&lt;/li&gt;     &lt;/ul&gt;   &lt;!-- 탭컨텐츠는 div 태그 로 표현 --&gt;   &lt;div id="tab1"&gt;     Tab1   &lt;/div&gt;   &lt;div id="tab2"&gt;&lt;/div&gt;   &lt;div id="tab3"&gt;&lt;/div&gt; &lt;/div&gt;</pre>

dataOpts	Array	[]	<p>선언형옵션을 명령형으로 지정할때 탭별로 탭의 순서대로 JSON Obejct 형태로 서브옵션을 지정 합니다.</p> <p>ex)</p> <pre>[   { width: "auto", url: "tab1.html", preload: false,     active: false, onOpen: undefined }, //Tab1,   { width: "auto", url: "tab2.html", preload: false,     active: false, onOpen: undefined }, //Tab2   { width: "auto", url: "tab3.html", preload: false,     active: false, onOpen: function() { } } //Tab3 ]</pre> <p>⚠ 선언형옵션은 탭요소(LI 태그)에 지정한다.</p>
randomSel	boolean	false	<p>true 이면 탭이 초기화 될 때 탭과 탭컨텐츠를 랜덤으로 보여줍니다. false 이면 첫 번째 탭을 보여줍니다.</p> <p>⚠ 탭에 선언형옵션으로 active 가 true 로 설정되어있으면 해당 옵션이 적용 됩니다.</p>
effect	boolean or Array	false	<p>탭 컨텐츠가 표시될때 전환 효과를 보여줄지 여부를 설정 합니다.</p> <p>효과를 적용하지 않으려면 false를 입력하고 효과를 입력하려면 다음과 같이 입력합니다.</p> <p>["효과명"  , "시간(ms)"  , "효과가 끝날 때 마다 실행될 콜백함수"]</p> <p>ex) ["slideDown", 300, null]</p>
onActive	Function	null	<p>탭이 활성화 될 때마다 실행되는 이벤트입니다. 이벤트 콜백 함수의 인자로 다음과 같은 요소가 반환 됩니다.</p> <p>arguments[0] : 현재 활성화된 탭요소  arguments[1] : 전체 탭요소  arguments[2] : 전체 컨텐츠요소</p>

#### 라. 선언형옵션(Declarative options)

Name	Type	Default	Description
url	String	undefined	특정 페이지를 탭컨텐츠로 만들경우 해당 페이지의 URL을 지정 합니다.
width	int or "auto"	auto	탭의 넓이를 지정 합니다.,
active	boolean	false	탭이 생성될 때 최초 선택될 탭에 true 로 설정 합니다.
preload	boolean	false	탭컨텐츠들을 탭이 초기화 모두 불러올지 탭이 최초열릴 때 불러올지 유무를 지정 합니다.
onOpen	String	undefined	onOpen 이벤트를 지정한 탭이 열릴때마다 불러온 컨텐츠에서 실행될 콜백 평선의 이름을 문자열로 지정 합니다. 콜백평선은 불러올 페이지의 Controller 에 지정한 이름으로 정의 되어 있어야 합니다.



마. 함수(Methods)

Name	Arguments	Type	Return	Description
open	idx	int	N.tab	지정한 index 번째 탭과 콘텐츠를 표시한다.

14. Select(N.select)

가. 개요

Select(이하 N.select) 는 SELECT, INPUT[TYPE=CHECKBOX], INPUT[TYPE=RADIO] 에 데이터를 주입하여 선택 요소를 만들어 주고 기능을 확장시켜주는 컴포넌트입니다.

나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.select	-	-	N.select	N.select 객체를 생성합니다.
	data	JWS(JSON Array)	-	기본옵션의 data 옵션과 같습니다.
	opts	JSON Object	-	기본 옵션을 JSON Object 형태로 지정 합니다.
N(obj).select	-	-	N.select	N.select을 플러그인 방식으로의 초기화 방식입니다. N.select와 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.select()의 첫 번째 인자로 지정 됩니다.
	opts	JSON Object	-	N.select의 opts 옵션과 같습니다.

다. 기본옵션(Default options)

Name	Type	Default	Required	Description
data	JWS(JSON Array)	undefined	○	입력요소에 바인드 할 데이터
context	JWS	null	○	N.select를 적용할 입력요소(SELECT, INPUT[TYPE=CHECKBOX], INPUT[TYPE=RADIO])를 지정합니다.
key	String	null	○	name 과 매핑(Mapping)될 데이터의 key 컬럼명을 지정 합니다.
val	String	null	○	value 과 매핑될 key 컬럼명을 지정 합니다.
append	boolean	true		SELECT 요소에서 에서 기본 입력값(기본으로 입력되어 있는 OPTION 요소)을 비우고 바인드할 건지, 더할건지 지정 합니다.
direction	String	h		INPUT[TYPE=CHECKBOX], INPUT[TYPE=RADIO]에서 요소생성을 오른쪽으로 이어서 가로로 만들건지 줄넘김하여 세로로 만들건지 지정합니다. * h : 가로 * v : 세로

라. 함수(Methods)

Name	Arguments	Type	Return	Description
data	-	-	JSON Array	선택입력요소에 바인드 되어 있는 데이터를 반환 합니다.
	selfFlag	boolean	-	true 이면 현재 선택된 값에 해당하는 데이터만 가져오고 false 이면 바인드 되어 있는 전체 데이터를 가져 옵니다.
context	-	-	JWO	바인드한 선택입력요소를 가져옵니다.
	sel	String	-	jQuery selector를 입력하면 해당 context 하위 요소에서 선택된 요소를 반환 합니다.

bind	-	-	N.select	context 에 데이터를 바인드 합니다.
val	-	-	String or Array	선택된 값을 가져옵니다.
	val	String	-	val 인자가 undefined 가 아니면 val 값에 해당하는 요소를 선택해 줍니다.
reset	-	-	N.select	선택입력요소의 선택을 초기화 합니다.
	selfFlag	boolean	-	true 이면 아무것 선택하지 않고 false 이면 SELECT 요소의 경우 기본 입력값으로 선택 됩니다.

## 15. Form(N.form)

### 가. 개요

Form(이하 N.form)은 폼 입력, 조회 양식을 구성하는 폼 블록(Block)요소(DIV, TABLE, SPAN 등)에 데이터를 바인드(bind)하여 폼을 간단하게 구현할 수 있도록 지원하는 컴포넌트입니다.

### 나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.form	-	-	N.form	N.form 객체를 생성합니다.
	data	JWS(JSON Array)	-	기본옵션의 data 옵션과 같습니다.
	opts	JSON Object	-	기본 옵션을 JSON Object 형태로 지정 합니다.
N(obj).form	-	-	N.form	N.form을 플러그인 방식으로의 초기화 방식입니다. N.form과 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.form()의 첫 번째 인자로 지정 됩니다.
	opts	JSON Object	-	N.form의 opts 옵션과 같습니다.

### 다. 기본옵션(Default options)

Name	Type	Default	Required	Description
data	JWS(JSON Array)	undefined	○	입력요소에 바인드 할 데이터
row	int	-1		입력한 data에서 바인드할 row index 컴포넌트의 기본값은 -1 이지만 입력하지 않으면 0으로 세팅 됩니다.
context	JWS	null	○	폼을 적용할 폼 블록(Block)요소를 지정합니다.
validate	boolean	true		입력요소의 실시간 검증을 할건지 유무를 지정 합니다.
html	boolean	false		bind 시 HTML 태그를 인식할건지 그대로 표시할건지 유무를 지정 합니다.
addTop	boolean	false		add 메서드를 호출했을 때 새로운 데이터셋을 입력된 데이터의 상단에 생성할건지 하단에 생성할건지를 결정 합니다. true 이면 상단에 생성하고 false 이면 하단에 생성 합니다.
fRules	Array	null		포맷룰을 지정 합니다. 6. Formater 참고
vRules	Array	null		검증룰을 지정 합니다. 7. Validator 참고

### 라. 선언형옵션(Declarative options)

Name	Type	Default	Description
format	Array[Array]	undefined	포맷룰을 지정 합니다. 6. Formater 참고
validate	Array[Array]	undefined	검증룰을 지정 합니다. 7. Validator 참고

마. 함수(Methods)

Name	Arguments	Type	Return	Description
data	-	-	JSON Array	폼에 바인드 되어 있는 데이터를 반환 합니다.
	selfFlag	boolean	-	true 이면 현재 바인드되어있는 데이터만 가져오고 false 이면 바인드 되어 있는 전체 데이터를 가져 옵니다.
row	-	-	int	폼에 현재 바인드되어 있는 데이터의 row index를 가져옵니다.
context	-	-	JWO	폼을 적용한 폼 블록(Block)요소를 가져옵니다.
	sel	String	-	jQuery selector를 입력하면 해당 context 하위 요소에서 선택된 요소를 반환 합니다.
bind			N.form	데이터를 폼요소에 데이터를 바인드 합니다.
	row	int	-	row를 지정하면 입력된 데이터의 지정한 row index 의 데이터로 바인드 합니다.
	data	JWS(JSON Array)	-	data를 지정하면 데이터 전체를 지정된 데이터로 치환 합니다.
add	-	-	N.form	새로운 로우 데이터를 생성 합니다. 로우 데이터 생성 시 폼요소안의 입력요소의 id 와 값들로 데이터를 생성합니다.  신규입력 페이지와 같이 입력된 데이터를 서버에 보낼 때 사용 하면 됩니다. N.form 초기화 후 add 메서를 호출하기만 하면 폼요소의 값들이 새로 생성된 데이터와 연동되므로 저장 버튼에 data 메서드를 호출하여 서버로 보내기만 하면 됩니다.
revert	-	-	N.form	최초 수정하기 전 데이터로 되돌려 줍니다.
validate	-	-	N.form	수정되거나 입력된 데이터 전체를 검증하여 검증결과를 반환 합니다. 검증 메시지는 입력요소 옆에 툴팁으로 표시 됩니다.
val	-	-	Value object or N.form	선택한 값을 가져오거나 입력 합니다. val 인자를 입력하지 않으면 선택한 값을 반환하고 입력하면 선택한 컬럼의 값을 입력한 값으로 바꿔준다.  ⚠ 값 입력 시 key 에 해당하는 HTML 요소가 없으면 데이터만 생성한다.
	key	String	-	가져올 값의 key 컬럼명
	val	Value object	-	저장할 값

16. Grid(N.grid)

가. 개요

Grid(이하 N.grid)는 테이블(TABLE) 요소에 데이터를 바인드하여 그리드형태로 데이터를 표현해 주는 컴포넌  
트입니다. 헤더고정형, 리스트형의 두 가지 타입을 제공 합니다.

나. 생성자(Constructor)

Name	Arguments	Type	Return	Description
N.grid	-	-	N.grid	N.grid 객체를 생성합니다.

	data	JWS(JSON Array)	-	기본옵션의 data 옵션과 같습니다.
	opts	JSON Object	-	기본 옵션을 JSON Object 형태로 지정 합니다.
N(obj).grid	-	-	N.grid	N.grid를 플러그인 방식으로의 초기화 방식입니다. N.grid와 동작되는 로직은 같으며 실제 N() 의 인자로 들어온 값이 N.grid()의 첫 번째 인자로 지정 됩니다.
	opts	JSON Object	-	N.grid의 opts 옵션과 같습니다.

#### 다. 기본옵션(Default options)

Name	Type	Default	Required	Description
data	JWS(JSON Array)	undefined	○	입력요소에 바인드 할 데이터
context	JWS	null	○	그리드를 적용할 그리드 블록(Block)요소(TABLE)를 지정합니다.
height	int	0		그리드 바디(body)의 높이를 지정 합니다. 입력한 값이 0보다 크면 헤더고정형의 일반적인 그리드형태로 초기화 되고 0을 입력하면 전체데이터 한번에 표현하는 리스트형의 그리드로 초기화 됩니다.
validate	boolean	true		입력요소의 실시간 검증을 할건지 유무를 지정 합니다.
html	boolean	false		bind 시 HTML 태그를 인식할건지 그대로 표시할건지 유무를 지정 합니다.
addTop	boolean	false		add 메서드를 호출했을 때 새로운 로우요소와 데이터세트를 그리드의 상단에 추가할건지 하단에 추가할건지를 결정 합니다. true 이면 상단에 추가하고 false 이면 하단에 추가 합니다.
resizable	boolean	false		컬럼 리사이징 기능 활성화 유무
vResizable	boolean	false		그리드 높이 리사이징 기능 활성화 유무
sortable	boolean	false		컬럼정렬 기능 활성화 유무
windowScrollLock	boolean	false		그리드 바디 영역에 mouseover 시 윈도우 스크롤을 감출건지 유무  ⚠ 그리드안에서 마우스 스크롤 시 그리드의 끝 영역이나 시작영역에 가면 브라우저 윈도우 스크롤이 자동으로 올라가거나 내려가서 그리드가 표시된 부분이 넘어가게 됩니다. 때문에 그리드 바디영역에 마우스가 올라가면 브라우저 윈도우 스크롤을 감춥니다. false로 지정하면 브라우저 기본모드로 동작 됩니다.
select	boolean	false		특정 로우를 선택하는 효과 적용 유무, 한건만 선택 됨.  ⚠ 로우 선택시 해당로우의 TBODY요소에 "grid_selected_" 스타일클래스가 생성되고 선택이 해제되면 생성된 스타일클래스가 제거 됩니다.
multiselect	boolean	false		특정 로우를 선택하는 효과 적용 유무, 여러건 선택 가능  ⚠ 로우 선택시 해당로우의 TBODY요소에

				"grid_selected_" 스타일클래스가 생성되고 선택이 해제되면 생성된 스타일클래스가 제거 됩니다.
hover	boolean	false		그리드 로우들의 HOVER 효과 적용 유무
createRowDelay	int(ms)	1		로우 요소들이 생성될 때마다의 딜레이를 지정 할수 있습니다.  ⚠ 0지정시 모든로우를 한번에 만들어 표현 합니다. 이렇게 되면 브라우저는 모든로우가 생성될때까지 멈춰있게 됩니다.
scrollPagination.size	int	100		헤더고정형 그리드에서는 스크롤페이징이 기본으로 적용됩니다. 페이징시 한번에 바인드하는 로우수를 지정 합니다.  ⚠ 너무 크게 지정하면 스크롤 페이징이 작동될때 브라우저에 부하가 발생되므로 그리드에 입력요소나 이미지가 들어 있다면 200 이하로 그렇지 않다면 1000 이하로 설정하는 것을 권장합니다.
fRules	Array	null		포맷룰을 지정 합니다. 6. Formater 참고
vRules	Array	null		검증룰을 지정 합니다. 7. Validator 참고
rowHandler	Function	null		그리드 데이터가 바인드될때 로우가 생성될때마다 발생하는 이벤트입니다. 이벤트 콜백 메서드의 인자로 생성될 당시의 row index, 로우 요소, 로우 데이터를 넣어줍니다. ex) rowHandler : function(index, rowEle, rowData) { // this 는 rowEle }
onSelect	Function	null		그리드의 행이 선택될때마다 발생하는 이벤트입니다. 이벤트 콜백 메서드의 인자로 생성될 당시의 row index, 로우 요소, 로우 데이터를 넣어줍니다. ex) onSelect : function(index, rowEle, rowData) { // this 는 rowEle }  ⚠ select 옵션이 true 일 때만 발생합니다.
onBind	Function	null		그리드의 데이터가 바인드 될 때마다 발생하는 이벤트입니다. 이벤트 콜백 메서드의 인자로 context, data를 넣어줍니다. ex) onBind : function(context, data) { // this 는 context }  ⚠ 스크롤페이징에 의해 페이징 될 때도 페이지가 완성 될 때마다 발생합니다.

#### 라. 선언형옵션(Declarative options)

Name	Type	Default	Description
format	Array[Array]	undefined	포맷룰을 지정 합니다. 6. Formater 참고
validate	Array[Array]	undefined	검증룰을 지정 합니다. 7. Validator 참고

마. 함수(Methods)

Name	Arguments	Type	Return	Description
data	-	-	JSON Array	그리드에 바인드 되어 있는 데이터를 반환 합니다.
	rowStatus	String	-	undefined 이면 전체 데이터를 가져오고 modified 이면 추가, 수정, 삭제 된 데이터를 가져옵니다. delete 이면 삭제된 데이터, update 이면 수정된 데이터, insert 이면 추가된 데이터를 가져옵니다.
context	-	-	JWO	그리드를 적용한 그리드 블록(Block)요소를 가져옵니다.
	sel	String	-	jQuery selector를 입력하면 해당 context 하위 요소에서 선택된 요소를 반환 합니다.
bind			N.grid	데이터를 그리드요소에 데이터를 바인드 합니다.
	data	JWS(JSON Array)	-	data를 지정하면 데이터 전체를 지정된 데이터로 치환 합니다.
add	-	-	N.grid	새로운 로우 데이터를 생성 합니다. 로우 데이터 생성 시 그리드요소안의 입력요소의 id 와 값들로 데이터를 생성합니다.
remove	-	-	N.grid	선택한 행 요소와 데이터를 삭제 합니다.
	row	int	-	제거할 row를 지정 합니다.
revert	-	-	N.grid	최초 수정하기 전 데이터로 되돌려 줍니다.
	row	int	-	row index를 입력하면 해당 로우만 되돌려 주고 입력하지 않으면 전체다 되돌려 줍니다.
validate	-	-	N.grid	수정되거나 입력된 데이터 전체를 검증하여 검증결과를 반환 합니다. 검증 메시지는 입력요소 옆에 툴팁으로 표시 됩니다.
	row	int	-	row index를 입력하면 해당 로우데이터만 검증하고 입력하지 않으면 전체 데이터를 검증합니다.
val	-	-	Value object or N.grid	선택한 값을 가져오거나 입력 한다. val 인자를 입력하지 않으면 선택한 값을 반환하고 입력하면 선택한 컬럼의 값을 입력한 값으로 바꿔준다.  ⚠ 값 입력 시 지정한 로우의 key 에 해당하는 HTML 요소가 없으면 데이터만 생성한다.
	row		-	가져올 값의 row index
	key		-	가져올 값의 key 컬럼명
	val		-	저장할 값