AXValidator

new AXValidator()

AXValidator.js, line 1

axisj form validator

Version:
- v1.53

Author:
- tom@axisj.com, hwshin@superjump.co.kr

Example

```
var myValidator = new AXValidator();
myValidator.setConfig({
    targetFormName : "myForm"    //{String} - validate를 수행하려는 form "name", ID가 아닙니다.
});
```

Extends

- AXJ

Methods

static AXValidator.add()

AXValidator.js, line 198

validate 대상 아이템을 추가합니다.

Type	Description
JSObject	example code 참고

Example

```

var jsObjectSample = {
    id: "userID",           //{string} - 아이템식별자
    label: "아이디",        //{string} - 아이템라벨
```

```
    config: {               //필요한 조합을 object로 정의합니다.
        required: true,         //[boolean=true] - 필수입력 체크
        number: true,           //[boolean=true] - 숫자입력 체크
        email: true,            //[boolean=true] - 이메일형식 체크
        hangul: true,           //[boolean=true] - 한글형식 체크
        engonly: true,          //[boolean=true] - 영문형식 체크
        residentno: true,       //[boolean=true] - 주민등록번호형식 체크
        foreignerno: true,      //[boolean=true] - 외국인번호형식 체크
        bizno: true,            //[boolean=true] - 사업자등록번호형식 체크
        phone: true,            //[boolean=true] - 전화번호형식 체크
        isdate: true,           //[boolean=true] - 날짜형식 체크
        zip: true,              //[boolean=true] - 우편번호형식 체크
        money: true,            //[boolean=true] - 숫자에 , 포함 체크
        earlierThan:{
            id: "targetId",             //{string} - 대상의 아이디. 현재 아이템의 값이 대상보다 커야함
            label:    "targetLabel"     //{string} - 대상의 라벨
        },
        laterThan:{
            id: "targetId",             //{string} - 대상의 아이디. 현재 아이템의 값이 대상보다 작아야함
            label:    "targetLabel"     //{string} - 대상의 라벨
        },
        min: true,              //[boolean=true] - 최소값
        max: true,              //[boolean=true] - 최대값
        minbyte: true,          //[boolean=true] - 최소바이트값
        maxbyte: true,          //[boolean=true] - 최대바이트값
```

```
        minlength: true,     //[boolean=true] - 최소길이
        maxlength: true      //[boolean=true] - 최대길이
    },
    realtime:{                //특정이벤트 발생시 액션정의
        event: "keydown",    //{String} - 발생하는 이벤트 종류
        response: function(){   //{fn} - 정의된 이벤트에 따른 실시간 이벤트 콜백함수
            //trace(this);
        }
    },
    onblur: function(){      //[fn] - 대상 아이템에 블러 이벤트 발생 콜백함수
        //trace(this);
    }
};

var myValidator = new AXValidator();
myValidator.add({
    id: "userID",            //{string} - 아이템식별자
    label: "아이디",         //{string} - 아이템라벨
    config: {
        required: true,      //[boolean=true] - 필수입력 체크
        minbyte:10,          //[boolean=true] - 최소바이트값
        maxbyte:20           //[boolean=true] - 최대바이트값
    },
    realtime: {
        event: "keydown",            //{String} - 발생하는 이벤트 종류
```

```
        response: function () {     //{fn} - 정의된 이벤트에 따른 실시간 이벤트 콜백함수
            if(this.result){
                $("#userID_realtime").html("OK");
            }else{
                $("#userID_realtime").html(this.message);
           }
            if (this.validateKey == "maxbyte" || this.validateKey == "maxlength"
) {
                return false; //키 입력 중지
            } else {
                return true; //키 입력 제어 안함
            }
        }
    },
    onblur: function(){ //[fn] - 대상 아이템에 블러 이벤트 발생 콜백함수
        trace(this);
    }
});

myValidator.add({
    id:"enddate",   //{string} - 아이템 식별자
    label:"종료일",  //{string} - 아이템 라벨
    config:{
        isdate:true,    //[boolean=true] - 날짜형식 체크
        laterThan:{
            id:"regdate",   //{string} - 대상의 아이디. 현재 아이템의 값이 대상보다 작아야함
```

```
            label:"등록일"   //{string} - 대상의 라벨
        }
    },
    onblur: function(){      //[fn] - 대상 아이템에 블러 이벤트 발생 콜백함수
        trace(this);
    }
});
```

static AXValidator.del()

AXValidator.js, line 517

validate *대상 아이템을 제거합니다.*

Type	Description
JSObject	example code *참고*

Example

```
myValidator.del( {id:"userID"});    //아이템식별자
```

static AXValidator.getErrorElement() {HTMLElement}

AXValidator.js, line 1012

에러가 발생된 엘리먼트를 리턴합니다.

Returns:

발생된 엘리먼트

Example

```
var validateResult = myValidator.validate();
if (!validateResult) {
    var msg = myValidator.getErrorMessage();      //에러메세지를 리턴합니다.
    alert(msg);
    myValidator.getErrorElement().focus();        //에러가 발생된 엘리먼트를 리턴합니다.
    return false;
```

```
}else{
    alert( validateResult );
}
```

static AXValidator.getErrorMessage() {string}

AXValidator.js, line 979

에러메세지를 리턴합니다.

Example

```
var validateResult = myValidator.validate();
if (!validateResult) {
    var msg = myValidator.getErrorMessage();     //에러메세지를 리턴합니다.
    alert(msg);
    myValidator.getErrorElement().focus();       //에러가 발생된 엘리먼트를 리턴합니다.
    return false;
}else{
    alert( validateResult );
}
```

static AXValidator.validate() {boolean}

AXValidator.js, line 568

validate *처리결과를 리턴합니다.*

Returns:

규칙에 따라 (true|false) *로 결과를 리턴합니다.*

Example

```
var validateResult = myValidator.validate();
```