* Jump To … +  
    
  [<<< back to documentation](http://docs.google.com/index.html)  [base.js](http://docs.google.com/base.html)   [constraint.js](http://docs.google.com/constraint.html)   [defaults.js](http://docs.google.com/defaults.html)   [factory.js](http://docs.google.com/factory.html)   [field.js](http://docs.google.com/field.html)   [form.js](http://docs.google.com/form.html)   [main.js](http://docs.google.com/main.html)   [multiple.js](http://docs.google.com/multiple.html)   [pubsub.js](http://docs.google.com/pubsub.html)   [remote.js](http://docs.google.com/remote.html)   [ui.js](http://docs.google.com/ui.html)   [utils.js](http://docs.google.com/utils.html)   [validator.js](http://docs.google.com/validator.html)   [validator\_registry.js](http://docs.google.com/validator_registry.html)

# base.js

* [¶](#gjdgxs)  
  import $ from 'jquery';  
  import Utils from './utils';  
    
  var Base = function () {  
   this.\_\_id\_\_ = Utils.generateID();  
  };  
    
  Base.prototype = {  
   asyncSupport: true, // Deprecated  
    
   \_pipeAccordingToValidationResult: function () {  
   var pipe = () => {  
   var r = $.Deferred();  
   if (true !== this.validationResult)  
   r.reject();  
   return r.resolve().promise();  
   };  
   return [pipe, pipe];  
   },  
    
   actualizeOptions: function () {  
   Utils.attr(this.element, this.options.namespace, this.domOptions);  
   if (this.parent && this.parent.actualizeOptions)  
   this.parent.actualizeOptions();  
   return this;  
   },  
    
   \_resetOptions: function (initOptions) {  
   this.domOptions = Utils.objectCreate(this.parent.options);  
   this.options = Utils.objectCreate(this.domOptions);
* [¶](#30j0zll)  
  Shallow copy of ownProperties of initOptions:  
   for (var i in initOptions) {  
   if (initOptions.hasOwnProperty(i))  
   this.options[i] = initOptions[i];  
   }  
   this.actualizeOptions();  
   },  
    
   \_listeners: null,
* [¶](#1fob9te)  
  Register a callback for the given event name Callback is called with context as the first argument and the this The context is the current parsley instance, or window.Parsley if global A return value of false will interrupt the calls  
   on: function (name, fn) {  
   this.\_listeners = this.\_listeners || {};  
   var queue = this.\_listeners[name] = this.\_listeners[name] || [];  
   queue.push(fn);  
    
   return this;  
   },
* [¶](#3znysh7)  
  Deprecated. Use on instead  
   subscribe: function(name, fn) {  
   $.listenTo(this, name.toLowerCase(), fn);  
   },
* [¶](#2et92p0)  
  Unregister a callback (or all if none is given) for the given event name  
   off: function (name, fn) {  
   var queue = this.\_listeners && this.\_listeners[name];  
   if (queue) {  
   if (!fn) {  
   delete this.\_listeners[name];  
   } else {  
   for (var i = queue.length; i--; )  
   if (queue[i] === fn)  
   queue.splice(i, 1);  
   }  
   }  
   return this;  
   },
* [¶](#tyjcwt)  
  Deprecated. Use off  
   unsubscribe: function(name, fn) {  
   $.unsubscribeTo(this, name.toLowerCase());  
   },
* [¶](#3dy6vkm)  
  Trigger an event of the given name A return value of false interrupts the callback chain Returns false if execution was interrupted  
   trigger: function (name, target, extraArg) {  
   target = target || this;  
   var queue = this.\_listeners && this.\_listeners[name];  
   var result;  
   var parentResult;  
   if (queue) {  
   for (var i = queue.length; i--; ) {  
   result = queue[i].call(target, target, extraArg);  
   if (result === false) return result;  
   }  
   }  
   if (this.parent) {  
   return this.parent.trigger(name, target, extraArg);  
   }  
   return true;  
   },  
    
   asyncIsValid: function (group, force) {  
   Utils.warnOnce("asyncIsValid is deprecated; please use whenValid instead");  
   return this.whenValid({group, force});  
   },  
    
   \_findRelated: function () {  
   return this.options.multiple ?  
   $(this.parent.element.querySelectorAll(`[${this.options.namespace}multiple="${this.options.multiple}"]`))  
   : this.$element;  
   }  
  };  
    
  export default Base;