scant Documentation

Release 0.5.1

Brett Langdon

January 18, 2015

Contents

A very minimalistic javascript library.

CHAPTER 1

scant */skant/*

adjective: scant

1. barely sufficient or adequate.

   “companies with scant regard for the safety of future generations”

   synonyms: little, little or no, minimal, hardly (any), limited, negligible, barely sufficient, meager

   antonyms: abundant, ample, sufficient barely amounting to a specified number or quantity.

verb: scant; *3rd person present:* scants; *past tense:* scanted; *past participle:* scanted; *gerund or present participle:* scanting

1. provide grudgingly or in insufficient amounts.

   “he does not scant his attention to the later writings”

CHAPTER 2

Download

Github Page

- Latest: `scant-latest.min.js` `scant-latest.min.js.map`
- v0.5.0: `scant-v0.5.0.min.js` `scant-v0.5.0.min.js.map`

CHAPTER 3

Usage

```
<script type="text/javascript" src="//path/to/scant.min.js"></script>
<script type="text/javascript">
  // window.Scant === window.$
  // window.$ is only made available is $ is not already registered

  $.ready(function(){
    var $elm = $('.selector');
    $elm.on('click', '.child', function(evt){
      evt.stopPropagation();
      evt.preventDefault();

      var data = $('form').serialize();
      $.ajax({
        url: '/end-point',
        data: JSON.stringify(data),
        dataType: 'application/json',
        responseType: 'json',
      }, function(err, request){
        if(err || ~request.response){
          alert('oh no, something went wrong');
        } else {
          alert('go a good response back');
        }
      });
    });
  });
</script>
```

CHAPTER 4

API Documentation

4.1 $ Object

$ (*selector*)

Main entry point for creating an instance of `scant()`.

Arguments

- **selector** (*string|NodeElement|HTMLElement*) – Either a string css selector to use to fetch elements on the page or an element to wrap in `scant()`

Returns an instance of `scant()`.

`$.util.`**extend** (*target*, *source*)

Helper method to copy properties from one Object over to another

Arguments

- **target** (*Object*) – Object to copy properties onto
- **source** (*Object*) – Object to copy properties from

Returns a copy of *target* with all of *sources*'s properties copied to it

`$.util.`**inherits** (*parent*, *child*)

Helper method to ensure that *child*'s prototype inherits from *parent*'s

Arguments

- **parent** (*function*) – The parent class to inherit from
- **child** (*function*) – The child class to inherit to

Returns null

`$.`**fn**

The prototype for `scant()` exposed, useful for extending `scant()`.

```
$.fn.hide = function(){
  // "this" refers to the instance of scant
  this.forEach(function(elm){
    elm.style.display = 'none';
  }.bind(this);
};

$('.hidden').hide();
```

$.**ajax**(*options*, *callback*)

Wrapper around making XMLHttpRequest's.

```
// default options to $.ajax
var defaultAjaxOptions = {
    // url to fetch
    url: null,
    // http method to use
    method: 'GET',
    // additional headers e.g. {'X-Token': 'token'}
    headers: {},
    // data to send
    data: null,
    // this gets set as 'Content-Type' header with request
    dataType: 'application/x-www-form-urlencoded; charset=UTF-8',
    // an optional function(request) to call before sending the request,
    // useful to make customizations to the XMLHttpRequest object before sending
    beforeSend: null,
    // the response type expected back
    responseType: '',
};
```

Arguments

- **options** (*Object*) – Object of options to override. `url` is required
- **callback** (*function*) – `function(err, request)` will get called when the request has finished (success or fail). `err` will be true if the request failed, null otherwise `request` will be the `XMLHttpRequest()` object used for the request.

Returns null

$.**on**(*eventName*[, *selector*], *handler*)

Function for event delegation. This method will add *handler* as an event handler on *body*, optionally filtering based on the *selector* provided.

Arguments

- **eventName** (*string*) – The event to bind (e.g. 'click', 'submit', etc)
- **selector** (*string*) – Optional parameter, a css selector used to filter events
- **handler** (*function*) – The handler to bind for the event

Returns null

$.**ready**(*handler*)

Function used to add a handler when the page has finished loaded ('DOMContentLoaded').

Arguments

- **handler** (*function*) – The handler to invoke when 'DOMContentLoaded'

Returns null

4.2 scant Class

class scant([*dom*][, *selector*])

The *scant* class inherits from *Array* and properties of an *Array* are available (although *forEach* is shimmed in if it is not supported by the browser).

Arguments

- **dom** (*Object*) – *dom* is either an *Array* of elements or the result of calling *document.querySelectorAll* with a *selector*
- **selector** (*string*) – The selector used to fetch *dom*.

`scant.find(selector)`

Method used to find all children matching *selector* beloning to the elements stored in this instance of scant.

Arguments

- **selector** (*string*) – A css selector to use to match elements.

Returns `scant()`

`scant.on(eventName[, selector], handler)`

This method is the same as `$.on()` except the handler is bound to every node stored in this instance of scant as opposed to binding to *body*.

Arguments

- **eventName** (*string*) – The event to bind to (e.g. 'click', 'submit', etc)
- **selector** (*string*) – A css selector used to filter the events by
- **handler** (*function*) – The handler to call for each event.

Returns null

`scant.serialize([elm])`

Method used to serialize the data for elements stored in this instance of scant. Most useful for fetching the data from *form* elements. This method gets the name/value pairs for all *form* elements (creating an array when the same name is present multiple times) as well as appending element data attributes.

Arguments

- **elm** (*Object*) – This parameter is mostly used internally to this function when making recursive calls for serializing the data, but can be a single dom element to serialize.

Returns Object

CHAPTER 5

Indices and tables

- *genindex*
- *modindex*
- *search*

Symbols

S