

# jQuery.Lazy();

## Get Started

### Installation

First of all, you will need a copy of jQuery (<http://code.jquery.com/jquery-latest.js>) or Zepto (<http://zeptojs.com/zepto.js>) to use Lazy successfully in your project. If you get this you can install Lazy by different ways. Select your favorite below.

[Self Hosted](#)[CDN](#)[Package Managers](#)

Download (<https://github.com/eisbehr-/jquery.lazy/archive/master.zip>) and save one of two available files to include Lazy to your page, either the development (<https://raw.githubusercontent.com/eisbehr-/jquery.lazy/master/jquery.lazy.js>) or the minified (<https://raw.githubusercontent.com/eisbehr-/jquery.lazy/master/jquery.lazy.min.js>) version beside of jQuery.

```
<script type="text/javascript" src="jquery.lazy.min.js"></script>
```

### Basic Usage

The basic usage of Lazy ist pretty easy. Just prepare your elements and add a little snippet to your page.

[HTML](#)[JavaScript](#)

Prepare the elements you want to lazy load. By default add a `data-src` attribute to elements containing the loadable image or a `data-loader` attribute to elements witch shall use a plugin or custom loaders.

```
<!-- load images the lazy way -->


<!-- load background images of other element types -->
<div class="lazy" data-src="path/to/image.jpg"></div>
```

### Configuration

Lazy comes with a bunch of easy to use configuration options by default, to control nearly every behavior as you like. The following parameters are available on every instance:

[General](#)[Attributes](#)[Effects](#)[Throttle](#)[Callbacks](#)[Example](#)

Configure your Lazy instances with an initial configuration object:

## jQuery.Lazy()

```
$('.lazy').Lazy({
  // your configuration goes here
  scrollDirection: 'vertical',
  effect: 'fadeIn',
  visibleOnly: true,
  onError: function(element) {
    console.log('error loading ' + element.data('src'));
  }
});
```

Or read and change the configuration live at runtime with the public `config` function. Please note the `chainable` parameter!

```
var lazy = $('.lazy').Lazy({ chainable: false });

var attributeName = lazy.config('attribute'); // get
lazy.config('effect', 'show');                // set
```

## Callbacks & Events

Lazy comes with a bunch of callbacks and events you can assign to. Just add them by initialization configuration to your instances

Callbacks / Events

Example

Simply attach a function to all callbacks you want to use by initialisation. The `element` parameter is already an jQuery or Zepto object, so you can use it directly like `element.addClass('handled')`.

```
$(".lazy").Lazy({
  beforeLoad: function(element) {
    // called before an elements gets handled
  },
  afterLoad: function(element) {
    // called after an element was successfully handled
  },
  onError: function(element) {
    // called whenever an element could not be handled
  },
  onFinishAll: function() {
    // called once all elements was handled
  }
});
```

`this` is the current Lazy instance itself, so you can access everything like public function or configuration directly.

## jQuery.Lazy()

```
jQuery.Lazy("").Lazy({
  onFinishedAll: function() {
    if( !this.config("autoDestroy") )
      this.destroy();
  }
});
```

## Public Functions & Instances

Lazy supports multiple parallel instances. Just initialize them with different selectors and possible different configurations.

Public Functions

Example

To access an instances public functions you can initialize them in an object oriented manner or grab the instance bind to every element by default.

```
// object oriented way, note the chainable parameter
var instance = $('.lazy').Lazy({ chainable: false });

// grab from elements
// only works well if you use same selector or a single instance overall
$('.lazy').Lazy();
var instance = $('.lazy').data("plugin_lazy");
```

The public functions `config` and `addItem` has some multiple functionality.

```
// 'config' can receive entries or set them to a new value
var value = instance.config('attribute');
instance.config('attribute', 'data-new-attribute');

// 'addItem' works with an jQuery object or any selector
// so these two lines have the exact same result
instance.addItem('img.new-items');
instance.addItem($('.img.new-items'));
```

## Custom Loaders

With the custom loaders option there is a powerful solution to load every contents the Lazy way. The plugin will handle everything, you just create a loading method witch got triggered whenever the element hits the visibility threshold. It's still possible to load images and custom loaders in the same Lazy instance.

Example

Response

## jQuery.Lazy():

To use a custom loader just define a name and function inside the Lazy initialisation. The first parameter is always a jQuery or Zepto DOM object, representing the element you want to lazy load.

```
$('.lazy').Lazy({
  customLoaderName: function(element) {
    element.html('element handled by "customLoaderName"');
    element.load();
  },
  asyncLoader: function(element, response) {
    setTimeout(function() {
      element.html('element handled by "asyncLoader"');
      response(true);
    }, 1000);
  }
});
```

Pass the loader name you want to use for this element to the `data-loader` attribute of those who should be lazy loaded. You can mix custom loaders and images in the same Lazy instance as well.

```
<div class="lazy" data-loader="customLoaderName"></div>

<div class="lazy" data-loader="customLoaderName"></div>
<div class="lazy" data-loader="asyncLoader"></div>
```

## Plugins

### How to use Plugins

The loader plugins for Lazy can be used whenever you want to extend the basic functionality by default or globally for many instances of Lazy.

HTML

Available Plugins

Just add the plugins you want to use or a combined file, containing all plugins, to your page and all instances can use the plugins from now on.

## jQuery.Lazy()

```

<!-- single plugin files -->
<script type="text/javascript" src="jquery.lazy.min.js"></script>
<script type="text/javascript" src="plugins/jquery.lazy.ajax.min.js"></script>
<script type="text/javascript" src="plugins/jquery.lazy.av.min.js"></script>
<script type="text/javascript" src="plugins/jquery.lazy.iframe.min.js"></script>
<script type="text/javascript" src="plugins/jquery.lazy.noop.min.js"></script>
<script type="text/javascript" src="plugins/jquery.lazy.picture.min.js"></script>
<script type="text/javascript" src="plugins/jquery.lazy.script.min.js"></script>
<script type="text/javascript" src="plugins/jquery.lazy.vimeo.min.js"></script>
<script type="text/javascript" src="plugins/jquery.lazy.youtube.min.js"></script>

<!-- or combined in one plugin file -->
<script type="text/javascript" src="jquery.lazy.min.js"></script>
<script type="text/javascript" src="jquery.lazy.plugins.min.js"></script>

```

Using a plugin on an element is just the same as with custom loaders. Add the attribute `data-loader` (by default) with the plugin name to the specific elements.

```

<!-- element will be handled by 'ajax' plugin -->
<div data-loader="ajax" data-src="file.html"></div>

```

If you create or use a plugin which is set to specific html element types by default, like our AV plugin, there is no need to add a `data-loader` attribute.

```

<!-- this element will automatically be handled by the 'av' plugin -->
<audio>
  <data-src src="file.ogg" type="audio/ogg"></data-src>
  <data-src src="file.mp3" type="audio/mp3"></data-src>
  <data-src src="file.wav" type="audio/wav"></data-src>
</audio>

```

## Create own Loader Plugins

If you want to, you can easily create own loader plugins. Just use jQuery or Zepto's public function `Lazy` to create and register them.

JavaScript   Using Zepto

The first parameter is the name of your plugin as `string`, or an `array` of names / aliases. Second parameter is the loader function, which is the same as in custom loaders. Best practice is to wrap everything by an IIFE ([https://en.wikipedia.org/wiki/Immediately-invoked\\_function\\_expression](https://en.wikipedia.org/wiki/Immediately-invoked_function_expression)).

## jQuery.Lazy()

```
(function($) {
  $.Lazy('pluginName', function(element, response) {
    // add your whole logic here

    // 'this' is the current instance of Lazy
    // so it's possible to access all public functions, like:
    var imageBase = this.config('imageBase');
  });
})(window.jQuery || window.Zepto);
```

The optional second parameter gives you the ability to register a plugin by default to html element types. When you do this, there is no need to set the `data-loader` attribute on each element you want to use this loader on.

But keep in mind, if you register an plugin on often used elements, like `<div>`, Lazy will try to handle each of them, every time! If you want to do so anyway, use a most specific selector for jQuery or Zepto.

```
(function($) {
  $.Lazy('av', ['audio', 'video'], function(element, response) {
    // this plugin will automatically handle '<audio>' and '<video>' elements,
    // even when no 'data-loader' attribute was set on the elements
  });
})(window.jQuery || window.Zepto);
```

## Examples

**BASIC USAGE ([HTTP://JQUERY.EISBEHR.DE/LAZY/EXAMPLE\\_BASIC-USAGE](http://jquery.eisbehr.de/lazy/example_basic-usage))**

**BOTTOM-TO-TOP SCROLLING ([HTTP://JQUERY.EISBEHR.DE/LAZY/EXAMPLE\\_BOTTOM-TO-TOP-SCROLLING#BOTTOM](http://jquery.eisbehr.de/lazy/example_bottom-to-top-scrolling#bottom))**

**DELAYED LOADING ([HTTP://JQUERY.EISBEHR.DE/LAZY/EXAMPLE\\_DELAYED-LOADING](http://jquery.eisbehr.de/lazy/example_delayed-loading))**

**COMBINE BASIC & DELAYED LOADING ([HTTP://JQUERY.EISBEHR.DE/LAZY/EXAMPLE\\_COMBINE-BASIC-AND-DELAYED-LOADING](http://jquery.eisbehr.de/lazy/example_combine-basic-and-delayed-loading))**

**SHOW A LOADING IMAGE ([HTTP://JQUERY.EISBEHR.DE/LAZY/EXAMPLE\\_SHOW-A-LOADING-IMAGE](http://jquery.eisbehr.de/lazy/example_show-a-loading-image))**

**DISABLED JAVASCRIPT FALLBACK ([HTTP://JQUERY.EISBEHR.DE/LAZY/EXAMPLE\\_DISABLED-JAVASCRIPT-FALLBACK](http://jquery.eisbehr.de/lazy/example_disabled-javascript-fallback))**

**THROTTLE DEMONSTRATION ([HTTP://JQUERY.EISBEHR.DE/LAZY/EXAMPLE\\_THROTTLE-DEMONSTRATION](http://jquery.eisbehr.de/lazy/example_throttle-demonstration))**

**HEAVY USAGE ([HTTP://JQUERY.EISBEHR.DE/LAZY/EXAMPLE\\_HEAVY-USAGE](http://jquery.eisbehr.de/lazy/example_heavy-usage))**