

PimemTech-scripts : jquery.jframe.js

Author : Frederic de Zorzi
Contact : fredz@nospam.pimentech.net
Revision : 1.33
Date : 2010-12-15
Copyright : 2007 PimemTech SARL
Tags : ajax javascript pimentech english jquery

Thanks to [jQuery](#) library, jFrame provides an easy way to get an HTML frame-like behaviour on DIV Elements with AJAX. It comes with *PimemTech Scripts* library. Also available on [jQuery_site](#) .

With jFrame, you can build smart, complex modern, internet apps without leaving the main page, without a single line of JavaScript !

A jFrame is a DIV tag with a src attribute. In a jFrame, click events on <a> and <input type="submit"> tags will be handled by jframe and loaded in ajax.

Recent Changes

- 2010-12-15 : removed log messages, stupid
- Fixed IE problem with form validation
- You can now activate a link outside a jframe :
- Toggle attribute on <a> tags : with toggle="off" , the jframe is loaded on the first click, hidden on the second, etc...

Requirements

The following JavaScript files must be included in the head section of your HTML page :

- jquery.js : http://docs.jquery.com/Downloading_jQuery
- jquery.form.js : <http://www.malsup.com/jquery/form/#download>
- jquery.jframes.js : <http://garage.pimentech.net/pimentech/js/jquery.jframe.js>

Usage Example

[Here](#) a demo.

By popular demand, complet demo source are located there, but it is written in [Django](#) :

- http://trac.pimentech.org/pimentech/browser/garage/site_garage/examples
- <http://trac.pimentech.org/pimentech/browser/garage/templates/examples>

Elements

jFrame : DIV with src attribute

Attributes

- id : jFrame id, required.
- src : url or "#", required. After a jFrame load, the src attribute value is replaced by the loaded url.

Links and Submit Buttons

Inside jFrames, onclick events on each link and submit button are redirected to jFrame plugin.

A link

- With target attribute, the href value is loaded by an xmlhttprequest in <div id=target>. Without target attribute, the href value is loaded in the current jframe. JFrame can be deactivated on a particular link with *jframe="no"* attribute (same for submit buttons).
- You can activate a link outside a jframe DIV with *jframe="activate"* attribute
- With *toggle="off"*, the first click will load the url link in target jframe and set *toggle* attribute to *on*. Then a second click will hide the target jframe (see demo above).

INPUT type="submit"

The submit buttons also support a target attribute, like <A> tags.
jFrame forms support get and post method, onsubmit form attribute.
BUTTON type="submit" are also supported.

jQuery API

- *\$(div element).loadJFrame(url, callback)* : like ajax.load function.
- *\$(div element).activateJFrames()* : the div element becomes a JFrame.
- *\$(element).getJFrameTarget()* : if exists, return the first JFrame parent element.
- *\$(element).waitingJFrame()* : Overload this function in your code if you want for example a "waiting" message. Example :

```
jQuery.fn.waitingJFrame = function () {
    $(this).html("<b>loading...</b>");
}
```

Source

```
// JFrame
// $Revision: 1.156 $
// Author: Frederic de Zorzi
// Contact: fredz@nospam_pimentech.net
// Revision: $Revision: 1.156 $
// Date: $Date: 2010-12-15 09:43:03 $
// Copyright: 2007-2009 PimenTech SARL
// Tags: ajax javascript pimentech english jquery

jQuery.fn.waitingJFrame = function () {
    // Overload this function in your code to place a waiting event
    // message, like : $(this).html("<b>loading...</b>");
};

function _jsattr(elem, key) {
    var res = jQuery(elem).attr(key);
    if (res == undefined) {
        return function() { return true;};
    }
    if (typeof res == 'string') {
        return function() { eval(res); };
    }
}
```

```

        return res;
    }

    jQuery.fn.preloadJFrame = function() {
        if (_jsattr(this, "beforeload")() == false) {
            return false;
        }
        jQuery(this).waitingJFrame();
        return true;
    };

    jQuery.fn.getJFrameTarget = function() {
        var target = jQuery(this).attr("target");
        if (target) {
            return jQuery("#" + target);
        }
        // Returns first parent jframe element, if exists
        return jQuery(jQuery(this).parents("div[src]").get(0));
    };

    jQuery.fn.loadJFrame = function(url, callback) {
        // like ajax.load, for jFrame. the onload->afterload attribute is supported
        var this_callback = _jsattr(this, "afterload");
        callback = callback || function(){};
        url = url || jQuery(this).attr("src");
        if (url && url != "#") {
            if (jQuery(this).preloadJFrame() == false) {
                return false;
            }
            jQuery(this).load(url,
                function(response, status, xhr) {
                    jQuery(this).attr("src", url);
                    jQuery(this).activateJFrame();
                    jQuery(this).find("div[src]").each(function(i) {
                        jQuery(this)
                    });
                    this_callback();
                    callback(response, status, xhr);
                });
        }
        else {
            jQuery(this).activateJFrame();
        }
        return true;
    };

    jQuery.fn.activateLink = function() {
        this.unbind("click");
        this.each(function () {
            var oc = this.onclick;
            this.onclick = null;
            jQuery(this).bind("click", function (event) {
                if (oc) {

```

```

        if (!oc()) {
            event.stopImmediatePropagation();
            return false;
        }
    }
    return true;
});

});

this.click(function() {
    var target = jQuery(this).getJFrameTarget();
    if (target.length) {
        var href = jQuery(this).attr("href");
        var toggle = jQuery(this).attr("toggle");
        if (href && href.indexOf('javascript:') != 0) {
            if (toggle == "on") {
                target.hide();
                jQuery(this).attr("toggle", "off");
                return false;
            }
            if (toggle == "off") {
                target.show();
                jQuery(this).attr("toggle", "on");
            }
            target.loadJFrame(href);
            return false;
        }
    }
    return true;
} )
.attr("jframe", "activated");
};

function jFrameSubmitInput(input) {
    var target = jQuery(input).getJFrameTarget();
    if (target.length) {
        var form = input.form;
        if (form) {
            if (form.onsubmit && form.onsubmit() == false) {
                return false;
            }
            if (!$.browser.msie && target.preloadJFrame() == false) {
                return false;
            }

            var submit_events = jQuery(form).data("events");
            if (submit_events) {
                submit_events = submit_events.submit;
                if (submit_events) {
                    jQuery.each(submit_events, function(i, submit) {
                        submit.handler();
                    });
                }
            }

            jQuery(form).ajaxSubmit({

```

```

        target: target,
        beforeSubmit: function(formData, theform,
            formData.push({ name:"submit", value:
        },
        success: function() {
            target.attr("src", jQuery(form).attr(
            _jsattr(target, "afterload") ();
            target.activateJFrame ();
        }
    });

    return false;
}
}
return true;
}

jQuery.fn.activateSubmitButton = function() {
    this.unbind("click")
        .click(function() {
            return jFrameSubmitInput(this);
        } )
        .attr("jframe", "activated");
};

jQuery.fn.activateJFrame = function() {
    // Add an onclick event on all <a> and <input type="submit"> tags
    jQuery(this).find("a")
        .not("[jframe='no' ]")
        .not("[jframe=' activated' ]")
        .activateLink ();

    jQuery(":image, :submit, :button", this)
        .not("[jframe='no' ]")
        .not("[jframe=' activated' ]")
        .activateSubmitButton ();

    if ($.browser.msie && $.browser.version.substr(0,1)<7) {
        // Only for IE6 : enter key invokes submit event
        jQuery(this).find("form")
            .unbind("submit")
            .submit(function() {
                return jFrameSubmitInput(jQuery(":image, :submit, :button", this
            } ) );
    }
};

jQuery(document).ready(
    function() {
        jQuery(document).find("div[src]").each(
            function(i) {
                jQuery(this).loadJFrame(undefined, undefined, true);
            } );
        jQuery(document).find("a[jframe=' activate' ]").activateLink ();
        jQuery(":image, :submit, :button", document).find("[jframe=' activate' ]").act

```

```
} );
```

Tips

Loading two jframes with one link

The base idea of jframe is “iframe alternative with ajax”, and I don’t want to add a non html behaviour like ‘ a href=“url1url2” ’. If you want to load 2 jframes with one link, you have to do it in JavaScript, like :

```
<a href="javascript:$ (' #target1' ).loadJFrame (' url1' );$ (' #target2' ).loadJFrame (' ur
```