

# Fixing broken plugins in jQuery 1.9

As of jQuery 1.9, many plugins stopped working. The main reason behind this side effect is related to the changes recently introduced in the jQuery library itself. These changes were actually announced several months before the final release of this new version but many developers simply ignored them or, more likely, they probably didn't know how their plugins were designed. That's why the copy-and-paste approach should be blamed.

## **\$.browser is gone**

The `$.browser` property was deprecated before jQuery 1.9 was released. Many obsolete plugins (i.e. plugins no longer updated) rely on this feature to detect the browser's version currently in use (IE above all).

You can fix this problem by simply replacing this property with a custom detection routine:

```
if(typeof $.browser === 'undefined') {
    $.browser = function() {
        var ua =
navigator.userAgent.toLowerCase();
        var ie = /msie/.test(ua); // add a
better check for browser's version, etc.
        // detect other browsers

        return {
            msie: ie
            // other browsers
        }
    }
}
```

```
}
```

This is a quick and dirty solution to be used within your plugin's code:

```
if($.browser().msie) {  
    //...  
}
```

A typical example of this case are lightbox plugins. Colorbox is currently one of the few plugins which are not affected by this problem, mainly because it's been constantly updated.

## **live() and die() are gone**

The `live()` and `die()` event methods should be replaced by `on()` and `off()`, respectively. This is an easy replacement because their syntax is almost identical.

## **AJAX events now attached to the document object**

The global AJAX events (`ajaxStart`, `ajaxStop`, `ajaxSend`, `ajaxComplete`, `ajaxError`, and `ajaxSuccess`) are only triggered on the document element. Change the program to listen for the AJAX events on the document. For example, if the code currently looks like this:

```
$("#status").ajaxStart(function(){
```

```
$(this).text("Ajax started");  
});
```

Change it to this:

```
$(document).ajaxStart(function(){  
    $("#status").text("Ajax started");  
});
```

## The \$() wrapper now requires well-formed HTML strings

If you're using AJAX with HTML strings, you've probably stumbled on the *Unrecognized expression* error. This error is due to the stricter policy by which HTML strings must be passed to the \$( ) wrapper. Simply put, an HTML string is now considered valid only if it begins with a less-than character (<).

Consider the following code:

```
$('#form').submit(function(e) {  
    e.preventDefault();  
    var $form = $(this);  
  
    $.ajax({  
        url: 'ajax.php',  
        type: 'POST',  
        dataType: 'html',  
        data: $form.serialize(),
```

```
                success: function(html) {
                    $(html).appendTo('#output');
// error-prone!

                }

            });

});
```

This should be changed to:

```
$('#form').submit(function(e) {
    e.preventDefault();
    var $form = $(this);

    $.ajax({
        url: 'ajax.php',
        type: 'POST',
        dataType: 'html',
        data: $form.serialize(),
        success: function(html) {
            $('#output').html(html);
        }
    });
});

});
```

## Documentation

You can find more useful information to fix your plugins on the official jQuery's documentation page.