

jQuery: implementing a basic CMS

Two years ago I implemented a basic jQuery CMS system to save to-dos, tasks and simple text notes. Today I'd like to republish this system with some major improvements.

Two years later the browser's standard support has changed significantly. Now we can use web storage to save our data:

```
var Storage = {
    type: localStorage,

    setItem: function(item, value) {
        this.type.setItem(item, value);
    },
    getItem: function(item) {
        var value = this.type.getItem(item);
        return value;
    },
    count: function() {
        var len = this.type.length;
        return len;
    },
    remove: function(item) {
        this.type.removeItem(item);
    },
    empty: function() {
        this.type.clear();
    }
};
```

```

        },
        saveObject: function(item, obj) {
            if (typeof obj === 'object') {
                this.setItem(item,
JSON.stringify(obj));

            } else {
                this.setItem(item, 'Could not
convert to JSON string');
            }

        },
        getObject: function(item) {
            var json = this.getItem(item);
            var obj = JSON.parse(json);
            return obj;
        }

    };
};

```

Take for example the form submission handling. I've basically saved the category, the title and the description of each task/note by creating an HTML structure to be appended to the corresponding tab.

Now we could write something like this:

```

$('#message').submit(function(event) {
    var $form = $(this);

```

```

        var category = $('#category',
$form).val();
        var title = $('#title', $form).val();
        var description = $('#description',
$form).val();
        if(Storage.getItem('data') == null) {
            var data = {};
            data[category] = [];
            data[category].push({title:
title, description: description});
            Storage.saveObject('data',
data);
        } else {
            var obj =
Storage.getObject('data');
            var details = {title: title,
description: description};
            obj[category].push(details);
            Storage.saveObject('data',
obj);
        }
        // continues
    });

```

Another approach is to use web databases. You can find an excellent example and tutorial created by *Paul Kinlan* on HTML5 Rocks. This tutorial is very similar to the topic covered in this post.

[view-example url="http://codepen.io/gabrielromanato/pen/aoihm"]