chrisix / olcc

Dismiss

Join GitHub today

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

Sign up

n-e Passage de totoz.eu en https b0b5468 Jul 30, 2018

2 contributors

734 lines (657 sloc) | 22.6 KB

```javascript
/*************************************************************
 * OnlineCoinCoin, by Chrisix (chrisix@gmail.com)
 * Définition de constantes, variables globales et fonctions utilitaires
 *************************************************************/

// Constantes globales génériques
var VERSION = '0.4.6';

// Constantes de gestion des états
var STATE_LOADED = 'loaded'; // Tribune chargée
var STATE_STOP = 'stop';     // Tribune arrêtée
var STATE_IDLE = 'idle';     // Tribune en attente
var STATE_HTTP = 'http';     // Tribune en train de rapatrier une url
var STATE_PARSE = 'parsing'; // En train de parser le backend
var STATE_POST = 'posting';  // En train de poster
var STATE_HTTP_ERROR = 'http-error';   // Erreur réseau
var STATE_PARSE_ERROR = 'parse-error'; // Erreur au parsing du backend
var STATE_POST_ERROR = 'post-error';   // Erreur au post

// Modes d'affichage des totoz
var TOTOZ_POPUP = 'popup';
var TOTOZ_INLINE = 'inline';

// Types de slip
var SLIP_TAGS_RAW = 'Raw tags';
var SLIP_TAGS_ENCODED = 'Tags encoded';

// Boss-mode
var BOSSMODE_RANDOM = 'Aléatoire';
var BOSSMODE_PTRAMO = 'Pierre Tramo';
var BOSSMODE_KERVIEL = 'J. Kerviel';
var BOSSMODE_PBPG = 'pBpG';
var BOSSMODE_DECIDEUR = 'Diçaïdor'

// Balltrap
var BALLTRAP_ONCLICK = 'Launch on click';
var BALLTRAP_AUTO = 'Auto launch';
var BALLTRAP_KILL = 'Kill on click';

// Valeurs de config par défaut
var DEFAULT_DEFAULT_UA = "onlineCoinCoin/"+VERSION;
var DEFAULT_PINNI_SIZE = 1000;
var DEFAULT_PINNI_KEEP = true;
var DEFAULT_AUTOREFRESH = true;
var DEFAULT_TIMEOUT = 30000;
var DEFAULT_ACTIVE_BOARDS = new Array();
var DEFAULT_TOTOZ_SERVER = "https://totoz.eu/";
var DEFAULT_TOTOZ_MODE = TOTOZ_POPUP;
var DEFAULT_DEFAULT_LOGIN = '';
var DEFAULT_WINDOW_TITLE = "OnlineCoinCoin";
var DEFAULT_FAVICON = "img/coin.png";
var DEFAULT_BOSS_MODE = BOSSMODE_RANDOM;
var DEFAULT_STYLE = 'default';
var DEFAULT_PLONK = '';
var DEFAULT_BALLTRAP = true;
var DEFAULT_MAX_DUCKS = 15;
var DEFAULT_BALLTRAP_MODE = BALLTRAP_AUTO;
var DEFAULT_BALLTRAP_SILENT = false;
var DEFAULT_SOUND_ENABLED = false;
var DEFAULT_SOUND_ZOO = false;
var DEFAULT_SOUND_NEW = 'notice.mp3';
var DEFAULT_SOUND_REPLY = 'msft_e-mail.mp3';
var DEFAULT_SOUND_BIGORNO = 'hail.mp3';
var DEFAULT_SOUND_VOLUME = 80;


// Notifications
var NOTIF_STATE = 'change state';
var NOTIF_NEW_POST = 'new post';
var NOTIF_ANSWER = 'new answer';
var NOTIF_BIGORNO = 'bigorno';
var NOTIF_BIGORNO_ALL = 'bigorno all';

var is_ie = true;
try {
  var plop = new ActiveXObject('Microsoft.XMLHTTP');
}
catch (err) {
  is_ie = false;
}

var favicon = {
    change: function(iconURL, optionalDocTitle) {
        if (optionalDocTitle) { document.title = optionalDocTitle; }
        this.addLink(iconURL, true);
    },
    addLink: function(iconURL) {
        var link = document.createElement("link");
        link.type = "image/x-icon";
        link.rel = "shortcut icon";
        link.href = iconURL;
        this.removeLinkIfExists();
        this.docHead.appendChild(link);
    },
    removeLinkIfExists: function() {
        var links = this.docHead.getElementsByTagName("link");
        for (var i=0; i<links.length; i++) {
            var link = links[i];
            if (link.type=="image/x-icon" && link.rel=="shortcut icon") {
                this.docHead.removeChild(link);
                return; // Assuming only one match at most.
            }
        }
    },
    docHead: document.getElementsByTagName("head")[0]
}

function getAllSheets() {
  // if you want ICEbrowser's limited support, do it this way
  if( !window.ScriptEngine && navigator.__ice_version ) {
    //IE errors if it sees navigator.__ice_version when a window is closing
    //window.ScriptEngine hides it from that
    return document.styleSheets; }
  if( document.getElementsByTagName ) {
    //DOM browsers - get link and style tags
    var Lt = document.getElementsByTagName('link');
    var St = document.getElementsByTagName('style');
  } else if( document.styleSheets && document.all ) {
    //not all browsers that supply document.all supply document.all.tags
    //but those that do and can switch stylesheets will also provide
    //document.styleSheets (checking for document.all.tags produces errors
    //in IE [WHY?!], even though it does actually support it)
    var Lt = document.all.tags('LINK'), St = document.all.tags('STYLE');
  } else { return []; } //lesser browser - return a blank array
  //for all link tags ...
  for( var x = 0, os = []; Lt[x]; x++ ) {
    //check for the rel attribute to see if it contains 'style'
    if( Lt[x].rel ) { var rel = Lt[x].rel;
    } else if( Lt[x].getAttribute ) { var rel = Lt[x].getAttribute('rel');
    } else { var rel = ''; }
    if( typeof( rel ) == 'string' && rel.toLowerCase().indexOf('style') + 1 ) {
      //fill os with linked stylesheets
      os[os.length] = Lt[x];
    }
  }
  //include all style tags too and return the array
  for( var x = 0; St[x]; x++ ) { os[os.length] = St[x]; } return os;
}

function changeStyle() {
  for( var x = 0, ss = getAllSheets(); ss[x]; x++ ) {
    //for each stylesheet ...
    if( ss[x].title ) {
      //disable the stylesheet if it is switchable
      ss[x].disabled = true;
    }
    for( var y = 0; y < arguments.length; y++ ) {
      //check each title ...
      if( ss[x].title == arguments[y] ) {
        //and re-enable the stylesheet if it has a chosen title
        ss[x].disabled = false;
      }
    }
  }
}

// Renvoie une classe CSS (permet de modifier ensuite le style de la classe
// dans sa globalité, affectant ainsi tous les éléments de cette classe)
function getStyleClass (className) {
    if (document.styleSheets.length < 1) {
        return null;
    }
    if (document.styleSheets[0].cssRules) {
        var cssRules = 'cssRules';
    } else {
        var cssRules = 'rules';
    }
    for (var s=document.styleSheets.length; s--;) {
        if (!document.styleSheets[s].disabled) {
            for (var r=document.styleSheets[s][cssRules].length; r--;) {
                if (document.styleSheets[s][cssRules][r].selectorText == '.' + className) {
                    return document.styleSheets[s][cssRules][r];
                }
            }
        }
    }
    return null;
}

// Ajoute dynamiquement de nouvelles définitions de classes CSS globales
function addCSSClass(ruleName, cssText) {
    if (document.styleSheets) {
        if (!getStyleClass(ruleName)) {
            for (var s=document.styleSheets.length; s--;) {
                // if (!document.styleSheets[s].disabled) {
                    if (document.styleSheets[s].addRule) {
                        document.styleSheets[s].addRule('.'+ruleName, cssText, 0); // IE style
                    } else {
                        document.styleSheets[s].insertRule('.'+ruleName+' {'+cssText+'}', 0); // Moz style.
                    }
                // }
            }
        }
    }
    /*
    else {
        // alert("here i am");
        for( var x = 0, ss = getAllSheets(); ss[x]; x++ ) {
            if (ss[x].addRule) {
                ss[x].addRule('.'+ruleName, cssText, 0); // IE style
            } else {
                ss[x].insertRule('.'+ruleName+' {'+cssText+'}', 0); // Moz style.
            }
        }
    } */
    return getStyleClass(ruleName);
}

// Ajoute de nouvelles classes CSS à un élément particulier
function addClass(node, newClass) {
    var theClass = node.className; // getAttribute('class');
    if (!theClass) {
        node.className = newClass; // setAttribute('class', newClass);
    }
    else if (!theClass.match(newClass)) {
        node.className = theClass + ' ' + newClass; // setAttribute('class', theClass + ' ' + newClass);
    }
}

// Retire une classe CSS à un élément particulier
function removeClass(node, oldClass) {
    if (!node) return;
    var exp = new RegExp(oldClass, "g");
    var classes = node.className.replace(exp, "").split(/ +/);
    node.className = classes.join(" ");
}

// Ajout d'une méthode à l'objet Date pour retourner un timestamp au format YYYYMMDDhhmmss
function pad0(i) { return (i<10) ? '0'+i : ''+i ; }
Date.prototype.timestamp = function () {
    return this.getFullYear() + pad0(this.getMonth()+1) + pad0(this.getDate())
         + pad0(this.getHours()) + pad0(this.getMinutes()) + pad0(this.getSeconds());
}

// Une chtite fonction map pour les tableaux, fait suer de pas en avoir par défaut
function map(fun) {
    var len = this.length;
    var res = new Array(len);
    for (var i=0; i<len; i++) {
        res[i] = fun(this[i]);
    }
    return res;
}
Array.prototype.map = map;

// Pareil avec each
Array.prototype.each = function (fun) {
    for (var i=0, l=this.length; i<l; i++) {
        fun(this[i]);
    }
}

// Idem avec contains
Array.prototype.contains = function (value) {
    for (var i=this.length; i--;) {
        if (this[i] == value) {
            return true;
        }
    }
    return false;
}

// Et pis remove passque quand même
Array.prototype.remove = function (item) {
    for (var i=this.length; i--;) {
        if (this[i] === item) this.splice(i, 1);
    }
}

// Ca c'est un hack foireux pour rendre compatible les arrays avec les résultats de queries xpath
Array.prototype.getLength = function () { return this.length; };

// Gestion des cookies
function getCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';');
    for (var i=ca.length;i--;) {
        var c = ca[i].strip();
        // while (c.charAt(0)==' ') c = c.substring(1, c.length);
        if (c.indexOf(nameEQ) == 0) {
            return unescape(c.substr(nameEQ.length));
        }
    }
    return null;
}

function setCookie(name, value, days) {
    if (days) {
        var date = new Date();
        date.setTime(date.getTime()+(days*24*60*60*1000));
        var expires = "; expires="+date.toGMTString();
    }
    else { var expires = "" };
    document.cookie = name+"="+escape(value)+expires+"; path=/";
}

function IE_selectNodes(classes, notclasses) {
    var all = GlobalPinni.getElementsByTagName("div") || [];
    if (!notclasses) notclasses = [];
    var res = new Array();
    for (var i=all.length; i--;) {
        var curdiv = all[i];
        // var curclasses = curdiv.className.split(/\s+/);
        var bool = true;
        for (var j=classes.length; j--;) {
            if (curdiv.className.indexOf(classes[j]) == -1) {
               bool = false;
               break;
            }
        }
        if (bool) {
            for (var j=notclasses.length; j--;) {
                if (curdiv.className.indexOf(notclasses[j]) != -1) {
                   bool = false;
                   break;
                }
            }
        }
        if (bool) res.push(curdiv);
    }
    return res;
}

function evalexp(query) {
    // return document.evaluate(query, document.getElementById("pinnipede"), null, XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null
    return evalXPath(document, query);
}

// retourne la position left et top d'un élément
function findPos(obj) {
    var curleft = curtop = 0;
    if (obj.offsetParent) {
        curleft = obj.offsetLeft;
        curtop = obj.offsetTop;
        while (obj = obj.offsetParent) {
            curleft += obj.offsetLeft;
            curtop += obj.offsetTop;
        }
    }
    return [curleft, curtop];
}


// Gestion des widgets des panneaux de configuration
function CheckBox(c_id, txt, val) {
    tr_el = document.createElement('tr');
    tr_el.className = 'subpanel'; // setAttribute('class','subpanel');

    td_el = document.createElement('td');
    td_el.className = 'subpanel'; // setAttribute('class','subpanel');
    td_el.innerHTML = txt;
    tr_el.appendChild(td_el);

    td_el = document.createElement('td');
    td_el.className = 'subpanel'; // setAttribute('class','subpanel');

    but = document.createElement('input');
    but.setAttribute('id',c_id);
    but.setAttribute('type','checkbox');
    if (val) {
        but.setAttribute('checked','yes');
    }

    td_el.appendChild(but);
    tr_el.appendChild(td_el);
    return tr_el;
}

function TextBox(c_id, txt, val, size) {
    tr_el = document.createElement('tr');
    tr_el.className = 'subpanel'; // setAttribute('class','subpanel');

    td_el = document.createElement('td');
    td_el.className = 'subpanel'; // setAttribute('class','subpanel');
    td_el.innerHTML = txt;
    tr_el.appendChild(td_el);

    td_el = document.createElement('td');
    td_el.className = 'subpanel'; // setAttribute('class','subpanel');

    but = document.createElement('input');
    but.setAttribute('id',c_id);
    but.setAttribute('type','textbox');
    if (size != -1) {
      but.setAttribute('size',size);
    }
    else {
      but.style.width = "98%";
    }
    but.setAttribute('value',val);

    td_el.appendChild(but);
    tr_el.appendChild(td_el);
    return tr_el;
}

function ButtonBox(txt, val, onclick) {
    tr_el = document.createElement('tr');
    tr_el.className = 'subpanel'; // setAttribute('class','subpanel');

    td_el = document.createElement('td');
    td_el.className = 'subpanel'; // setAttribute('class','subpanel');
    td_el.innerHTML = txt;
    tr_el.appendChild(td_el);

    td_el = document.createElement('td');
    btn = document.createElement('input');
    btn.setAttribute('type','button');
    btn.setAttribute('onclick',onclick);
    btn.setAttribute('value',val);
    td_el.appendChild(btn);
    tr_el.appendChild(td_el);

    return tr_el;
}

function FileBox(txt, val, action, onsubmit) {
    tr_el = document.createElement('tr');
    tr_el.className = 'subpanel'; // setAttribute('class','subpanel');

    td_el = document.createElement('td');
    td_el.className = 'subpanel'; // setAttribute('class','subpanel');
    td_el.innerHTML = txt;
    tr_el.appendChild(td_el);

    td_el = document.createElement('td');
    form = document.createElement('form');
    form.setAttribute('method','post');
    form.setAttribute('action',action);
    form.setAttribute('enctype','multipart/form-data');
    form.setAttribute('onsubmit',onsubmit);
    input = document.createElement('input');
    input.setAttribute('type','file');
    input.setAttribute('name','attachment');
    form.appendChild(input);
    input = document.createElement('input');
    input.setAttribute('type','submit');
    input.setAttribute('value',val);
    form.appendChild(input);
    td_el.appendChild(form);
    tr_el.appendChild(td_el);

    return tr_el;
}

function InfoBox(txt, val) {
    tr_el = document.createElement('tr');
    tr_el.className = 'subpanel'; // setAttribute('class','subpanel');

    td_el = document.createElement('td');
    td_el.className = 'subpanel'; // setAttribute('class','subpanel');
    td_el.innerHTML = txt;
    tr_el.appendChild(td_el);

    td_el = document.createElement('td');
    td_el.className = 'subpanel config-info';
    td_el.innerHTML = val;
    tr_el.appendChild(td_el);

    return tr_el;
}

function SelectBox(c_id, txt, tab, val, size) {
    tr_el = document.createElement('tr');
    tr_el.className = 'subpanel';

    td_el = document.createElement('td');
    td_el.className = 'subpanel';
    td_el.innerHTML = txt;
    tr_el.appendChild(td_el);

    td_el = document.createElement('td');
    td_el.className = 'subpanel';

    but = document.createElement('select');
    but.setAttribute('id',c_id);

    for(i=0; i<tab.length; i++) {
        opt = document.createElement('option');
        opt.innerHTML = tab[i];
        if (val == tab[i]) {
            opt.setAttribute('selected','yes');
        }
        but.appendChild(opt);
    }

    td_el.appendChild(but);
    tr_el.appendChild(td_el);
    return tr_el;
}

function SoundBox(c_id, txt, val, size) {
    tr_el = document.createElement('tr');
    tr_el.className = 'subpanel';

    td_el = document.createElement('td');
    td_el.className = 'subpanel';
    td_el.innerHTML = txt;
    tr_el.appendChild(td_el);

    td_el = document.createElement('td');
    td_el.className = 'subpanel';

    but = document.createElement('select');
    but.setAttribute('id',c_id);

    for(i=0; i<GlobalSoundList.length; i++) {
        opt = document.createElement('option');
        opt.innerHTML = GlobalSoundList[i];
        if (val == GlobalSoundList[i]) {
            opt.setAttribute('selected','yes');
        }
        but.appendChild(opt);
    }
    td_el.appendChild(but);
    td_el.innerHTML += ' <div id="'+c_id+'_play" class="sndbutton play" onclick="playSample('+"'"+c_id+"'"+')"></div>'
                     + ' <div id="'+c_id+'_stop" class="sndbutton stop" onclick="stopSample('+"'"+c_id+"'"+')"></div>';

    tr_el.appendChild(td_el);
    return tr_el;
}
GlobalSoundList = ["(aucun)", "bugle.mp3", "notice.mp3", "twang.mp3"];
function getSoundList() {
    var xhr = new XMLHttpRequest();
    xhr.open('GET', 'listdir.php?dir=sound', true);
    xhr.onreadystatechange = function() {
        switch (xhr.readyState) {
          case 4:
            soundListResult(xhr);
            break;
          default:
            // inprogress(xhr);
            break;
        }
      }
    xhr.send(null);
}
function soundListResult(xhr) {
    try {
        if (xhr.status == 200) {
            var res = eval(xhr.responseText);
            GlobalSoundList = res['Files'];
        }
    }
    catch(err) {
        // nothing
    }
}
function playSample(c_id) {
    settings.set('sound_volume', document.getElementById('config-sound_volume').value || 80);
    var sound = document.getElementById(c_id).value;
    if (sound != "(aucun)") {
        stopSample();
        document.getElementById(c_id+"_play").style.display = "none";
        document.getElementById(c_id+"_stop").style.display = "block";
        sound_play("sound/"+sound);
        GlobalIsPlaying = c_id;
    }
}
function stopSample() {
    if (GlobalIsPlaying) {
      document.getElementById(GlobalIsPlaying+"_play").style.display = "block";
      document.getElementById(GlobalIsPlaying+"_stop").style.display = "none";
    }
    sound_stop();
    GlobalIsPlaying = null;
}

var GlobalIsPlaying = null;
var GlobalSoundQueue = new Array();

function sound_play(sound) {
    if (sound.substr(-7) != "(aucun)" && !GlobalSoundQueue.contains(sound)) {
        GlobalSoundQueue.push(sound);
        if (document.getElementById('player').paused)
          sound_queue();
    }
}
function sound_queue() {
    if (GlobalSoundQueue.length == 0) return;
    var player = document.getElementById("player");
    player.volume = settings.value('sound_volume')/100.0;
    player.src = GlobalSoundQueue.shift();
    player.play();
}
function sound_stop() {
    document.getElementById("player").pause();
}
function sound_ended() {
    if (GlobalIsPlaying) stopSample();
    sound_queue();
}

function to_url(chaine) {
    return str_replace('&', '%26', chaine);
}
function str_replace(p, r, s) {
    return s.replace(RegExp(encodeRE(p), 'g'), encodeRE(r));
}
function encodeRE(s) { // Theodor Zoulias, http://simon.incutio.com/archive/2006/01/20/escape#comment14
    return s.replace(/([.*+?^${}()|[\]\/\\])/g, '\\$1');
}
// ]/

String.prototype.strip = function () {
    return this.replace(/^\s+|\s+$/g, '');
}

// Evenements cross-browser
function addEvent(object, event, callback, flag) {
    if (is_ie) {
        object.attachEvent("on"+event, callback);
    }
    else {
        object.addEventListener(event, callback, flag);
    }
}
// Evenements cross-browser
function removeEvent(object, event, callback, flag) {
    if (is_ie) {
        object.detachEvent("on"+event, callback);
    }
    else {
        object.removeEventListener(event, callback, flag);
    }
}

// Divers ajouts pour compatibilité IE/reste du monde
var srz = null;
var no_xpath = false;
if (is_ie) {
    // Array.prototype.snapshotItem = function (i) { return this[i]; };
    window.XMLHttpRequest = function () {
        return new ActiveXObject('Microsoft.XMLHTTP');
    }
}
else {
    try {
        XPathResult.prototype.__defineGetter__('length', function () { return this.snapshotLength; });
    }
    catch (err) {
        no_xpath = true;
    }
    srz = new XMLSerializer();
}

// Retourne un objet xml dom, fonction cross-browser
function loadXML(text) {
    if (is_ie) {
        var xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
        xmlDoc.async = "false";
        xmlDoc.setProperty("SelectionLanguage", "XPath");
        // xmlDoc.evaluate = function (query) { return this.selectNodes(query); };
        xmlDoc.loadXML(text);
    }
    else {
        var parser = new DOMParser();
        var xmlDoc = parser.parseFromString(text, "text/xml");
    }
    return xmlDoc;
}

function evalXPath(node, query) {
    if (is_ie) {
        return node.selectNodes(query);
    }
    else {
        var target = (node == document) ? GlobalPinni : node;
        try {
            return node.evaluate(query, target, null, XPathResult.ORDERED_NODE_SNAPSHOT_TYPE, null);
        }
        catch (err) {
            return [];
        }
    }
}

function getItem(nodes, i) {
    if (is_ie || no_xpath) {
        return nodes[i];
    }
    else {
        return nodes.snapshotItem(i);
    }
}

function getLength(nodes) {
    if (is_ie || no_xpath) {
        return nodes.length;
    }
    else {
        return nodes.snapshotLength;
    }
}

function serializeNodes(node, tag) {
    var content = "";
    if (is_ie || no_xpath) {
        content = node.xml || node.innerHTML;
        // return content.replace(new RegExp('<'+tag+'>(.*)<\/'+tag+'>'), "$1");
    }
    else {
        /*
        node.childNodes.map = map;
        return node.childNodes.map(function(x){return srz.serializeToString(x);}).join(""); */
        content = srz.serializeToString(node)
    }
    return content.replace(new RegExp('<'+tag+'>(.*)<\/'+tag+'>'), "$1");
    /* try {
        node.childNodes.map = map;
        return node.childNodes.map(function(x){return srz.serializeToString(x);}).join("");
    }
    catch(err) {
        var content = node.xml || node.innerHTML;
        return content.replace(new RegExp('<'+tag+'>(.*)<\/'+tag+'>'), "$1");
    } */
}

function getNodeText(node) {
    if (is_ie) {
        if (node.firstChild) {
            return node.firstChild.nodeValue.strip();
        }
        else {
            return '';
        }
    }
    else {
        return node.textContent.strip();
    }
    // AFR: webkit utilise probablement innerHTML
}
```