var socket;

var users = [];

var rooms = [];

var myid = null;

var myroom = null;

var nopm = false;

var nonot = false;

var pickedfile = null;

var power = {};

var powers = [];

var emos = [];

var dro3 = [];

var token = '';

var rbans = [];

var blocked = [];

function logout() {

send('logout', {}); close(500);

}

function sendbc(wfile) {

if (wfile) {

pickedfile = null; sendfile('d2bc', function () {

var msg = $(".tboxbc").val();

$(".tboxbc").val('');

var link = pickedfile;

pickedfile = '';

if ((msg == "%0A" || msg == "%0a" || msg == '' || msg == '\n') && (link == '' || link == null)) { return; }

send('bc', { msg: msg, link: link })

return;

});

}

else

{ pickedfile = null; }

var msg = $(".tboxbc").val();

$(".tboxbc").val('');

var link = pickedfile;

pickedfile = '';

if ((msg == "%0A" || msg == "%0a" || msg == '' || msg == '\n') && (link == '' || link == null)) { return; }

send('bc', { msg: msg, link: link })

}

var isIphone = false;

function refr() {

var r = document.referrer || '';

if (r.indexOf('http://' + location.hostname) == 0) { return ''; }

if (r.indexOf('://') != -1) { r = r.replace(/(.\*?)\:\/\//g, '').split('/')[0]; }

return r;

}

function checkupdate() {

if (needUpdate) { updateusers(); updaterooms(); needUpdate = false }

setTimeout(checkupdate, 2000);

}

function load()//d

{

isIphone = /ipad|iphone|ipod/i.test(navigator.userAgent.toLowerCase());

if (typeof $ == 'undefined' || typeof io == 'undefined') { close(5000); return; }

if ($('').tab == null) { close(5000); return; }

if (isIphone) {

$('img[data-toggle="popover"]').removeClass('nosel');

fxi();

}

checkupdate();

$('#rhtml .utopic').css('margin-left', '6px');

umsg = $("#umsg").html();

loadpro();

loadblocked();

if ($(window).width() <= 400)

{ $("meta[name='viewport']").attr('content', ' user-scalable=0, width=400'); }

if ($(window).width() >= 600)

{ $("meta[name='viewport']").attr('content', ' user-scalable=0, width=600'); }

$('#tbox').css('background-color', '#AAAAAF'); $(".rout").hide(); $(".redit").hide();

$("#u1").val(decode(getv("u1")));

$("#u2").val(decode(getv("u2")));

$("#pass1").val(decode(getv("p1")));

if (getv("isl") == "yes") {

$('.nav-tabs a[href="#l2"]').tab('show')

// $(".tlogin").tabs().tabs( "option", "active", 1 )

}

uhtml = $("#uhtml").html();

rhtml = $("#rhtml").html();

$('.ae').click(function (params) {

$('.phide').click();

})

var dbg = getUrlParameter('debug') == 'yes';

if (dbg) {

window.onerror = function (errorMsg, url, lineNumber) {

alert('Error: ' + errorMsg + ' Script: ' + url + ' Line: ' + lineNumber);

}

alert(dbg);

}

function oidbg(ev, data) {

if (dbg == false) { return; }

if (typeof data == 'object') { data = JSON.stringify(data); }

alert(ev + '\n' + data)

}

$('#tlogins button').attr('disabled', 'true');

// socket=io('http://185.65.205.58:1003');

processq();

newsock();

if (getv('refr') == '') { setv('refr', refr() || '\*') };

if (getv('r') == '') { setv('r', getUrlParameter('r') || '\*') };

$(window).on('resize pushclose pushopen', fixSize);

//$('textarea').on('blur',function(){ window.scrollTo(0,1); })

$('\*[data-toggle="tab"]').on('shown.bs.tab', function (e) { fixSize(); });

$("#tbox").keyup(function (e) {

if (e.keyCode == 13) { e.preventDefault(); Tsend() }

});

$(".tboxbc").keyup(function (e) {

if (e.keyCode == 13) { e.preventDefault(); sendbc() }

});

fixSize();

setTimeout(function () {

updateTimes();

}, 20000);

setTimeout(function () {

refreshonline();

}, 500);

}

var pending = false;

var pdata=[];

function send(cmd, data) {

if(pending){

pdata.push( { cmd: cmd, data: data });

if(pdata.length>4){pdata.splice(0,1);

}

return;

}

socket.emit('msg', { cmd: cmd, data: data });

}

function newsock()

{

socket = io('', { reconnection:false, transportstra: ['polling', 'websocket'] });

socket.on('connect', function () {

lstat('success', 'متصل | online'); $('#tlogins button').removeAttr('disabled');

if (pending) { socket.emit('re', { token: token });pending=false; }

if (getUrlParameter('enter') != null) {

$('#u1').val(hash([new Date().getTime()], 256) + '\_زائر');

login(1);

}

});

socket.on('re', function (data) { if (data.ok == true) { pending == false;pdata.forEach(function(e){socket.emit('msg',e);});pdata=[];} else { close(); }});

socket.on('msg', function (data) { onq.push(data); });

socket.on('disconnect', function (data) {

if (myid != null && pending == false) {pending = true;setTimeout(newsock,12000); return;}

lstat('danger', 'غير متصل'); close(); });

socket.on('connect\_error', function (data) {

console.log('connect\_error'); lstat('danger', 'غير متصل'); close(); });

socket.on('connect\_timeout', function (data) {console.log('connect\_timeout'); lstat('danger', 'غير متصل'); close(); });

socket.on('error', function (data) {console.log('error'); lstat('danger', 'غير متصل'); close(); });

}

function processq() {

for (var i = 0; i < onq.length && i < 20; i++) {

var data = onq[0];

onq.splice(0, 1);

ondata(data.cmd, data.data);

}

setTimeout(function () {

processq();

}, 100);

}

var onq = [];

function fxi() {

if (isIphone) {

$("textarea").on('focus', function () { fixI(this); });

$("textarea").on('blur', function () { blurI(this); });

document.addEventListener('focusout', function (e) { window.scrollTo(0, 0) });

}

}

function fixI(el) {

if (isIphone == false) { return; }

var sv = $(el).position().top - (document.body.scrollHeight - window.innerHeight) - 10;

if (sv < document.body.scrollHeight + window.innerHeight) {

// sv=0;

}

$(document.body).scrollTop(sv);

}

function blurI() {

if (isIphone == false) { return; }

$(document.body).scrollTop(0);

}

function debugI() {

var s = '';

s += window.innerHeight + '\n';

s += $(window).height() + '\n';

s += document.height + '\n';

s += document.body.height + '\n';

s += $('#tbox').position().top + '\n';

s += document.body.scrollHeight + '\n';

s += $('.dad').height() + '\n';

// alert(s);

$(document.body).scrollTop($('#tbox').position().top - (document.height - window.innerHeight));

}

function refreshonline() {

$.get('getonline', function (d) {

var data = d;

if (typeof d == 'stromg') { JSON.parse(d); }

powers = data.powers;

var lonline = $('.lonline');

lonline.children().remove();

var uhtml = $('#uhtml').html();

$('.s1').text(data.online.length + 10);

$.each(data.online, function (i, e) {

if (e.s == true) { return; }

var uh = $(uhtml);

uh.find('.u-topic').text(e.topic).css({ "background-color": e.bg, "color": e.ucol });;

uh.find('.u-msg').text(e.msg);

uh.find('.u-pic').css('background-image', 'url("' + e.pic + '")');

uh.find('.ustat').remove();

if (e.co == "--" || e.co == null || e.co == 'A1' || e.co == 'A2' || e.co == 'EU') {

uh.find(".co").remove();

}

else {

uh.find(".co").attr("src", "flag/" + e.co.toLowerCase() + ".gif")

}

var ico = getico(e);

if (ico != '') {

uh.find('.u-ico').attr('src', ico);

}

lonline.append(uh);

})

});

}

function htmljson(html) {

html = $(html);

var json = {};

$.each(html.find('input'), function (i, e) {

switch ($(e).attr('type')) {

case "text":

json[$(e).attr('name')] = $(e).val();

break;

case "checkbox":

json[$(e).attr('name')] = $(e).prop('checked');

break;

case "number":

json[$(e).attr('name')] = parseInt($(e).val(), 10);

break;

}

});

return json;

}

function jsonhtml(j, onsave) {

var html = $('<div style="width:100%;height:100%;padding:5px;" class="break"></div>');

$.each(Object.keys(j), function (i, key) {

switch (typeof j[key]) {

case "string":

html.append('<label class="label label-primary">' + key + '</label></br>')

html.append('<input type="text" name="' + key + '" class="corner" value="' + j[key] + '"></br>')

break;

case "boolean":

html.append('<label class="label label-primary">' + key + '</label></br>');

var checked = ''; if (j[key]) { checked = 'checked' }

html.append('<label>تفعيل<input name="' + key + '" type="checkbox" class="corner" ' + checked + '></label></br>')

break;

case "number":

html.append('<label class="label label-primary">' + key + '</label></br>')

html.append('<input name="' + key + '" type="number" class="corner" value="' + j[key] + '"></br>')

break;

}

});

html.append('<button class="btn btn-primary fr fa fa-edit">حفظ</button>');

html.find('button').click(function () { onsave(htmljson(html)) });

return html;

}

var lastfix = 0;

var lastw = 0;

function fixSize(again) { //again=1;

var w = $(document.body).innerWidth();

// $("#d2").width(w- ($("#d0").outerWidth()+4) +'px');

// $("#content").find('.tablebox,#d2').width(w + 'px');

// $("#d2").css('height',$(window).height()-$('.footer').outerHeight(true)-$('#header').outerHeight(true) -5+'px');

//$(document.body).css('height', window.innerHeight + 'px');

//$('#content').css('height',window.innerHeight-kbd+'px')

// if( $(e).hasClass('active')==false){$(e).addClass('hid')}else{$(e).removeClass('hid')}

$(document.documentElement).css('height', $(window).height() - 2 + 'px');

docss()

startcss()

// $('.lonline').css('height',$(window)[0].innerHeight-200+'px');

var lonline = $(".lonline");

if (lonline.length > 0) {

lonline.css('height', $(window).height() - lonline.position().top - 5 + 'px');

}

$(".dpnl").css("left", $('.dad').width() - ($('.dpnl').width() + 2) + 'px').css('height', $('#room').height() - ($("#d0").height() + 2) + 'px').css('top', '0px')

//$('.dpnl').css('top',($(window).height()-$('.dpnl').height())- $("#d0").height()-30 );

if (again != 1) { setTimeout(function () { fixSize(1); }, 10) } else { $('#d2').scrollTop($("#d2")[0].scrollHeight); }

}

if (getUrlParameter('x') == '1') {

dkh = 0;

setInterval(function () {

var dkk = $(document).height() - $(document.body).height();

if (dkk != dkh) {

dkh = dkk; alert(dkh);

}

}, 2000)

}

function startcss() {

$.each($('.tab-pane'), function (i, e) { if ($(e).hasClass('active')) { $(e).removeClass('hid') } else { $(e).addClass('hid') } });

$('a[data-toggle="tab"]').on('shown.bs.tab', function (e) {

$($(e.relatedTarget).attr('href')).addClass('hid')

$($(e.target).attr('href')).removeClass('hid')

})

}

function docss() {

$.each($('.filw'), function (i, e) {

var par = $(e).parent();

var wd = 0;

$.each(par.children(), function (ii, child) {

if ($(child).hasClass('filw') || $(child).hasClass('popover') || $(child).css('position') == 'absolute') { return; }

wd += $(child).outerWidth(true);

});

$(e).css('width', (par.width() - wd) - 12 + 'px');

});

$.each($('.filh'), function (i, e) {

var par = $(e).parent();

var wd = 0;

$.each(par.children(), function (ii, child) {

if ($(child).hasClass('filh') || $(child).css('position') == 'absolute') { return; }

wd += $(child).outerHeight(true);

});

$(e).css('height', (par.height() - wd) - 1 + 'px');

});

}

function pickedemo(e) {

e = $(e);

var ei = e.attr('title');

var par = $(e.attr('eid'));

par.parent().find('.tbox').val($(par).parent().find('.tbox').val() + ' ف' + ei);

par.popover('hide').blur()

}

function roomChanged(isme) {

$("#users").find(".inroom").removeClass("inroom");

$("#rooms").find(".inroom").removeClass("inroom");

var r = getroom(myroom);

$('.bord').removeClass('bord')

if (r != null) {

$('.inr,.ninr,.rout').show();

if ($("#room.active").length == 0 && isme == true) { $("[data-target='#room']").trigger('click'); }

if (isme == true) { $("[data-target='#room']").show(); }

$.each(rusers(r.id), function () { $('#users').find('.uid' + this.id).addClass('inroom'); });

$('#rooms').find('.r' + r.id).addClass('inroom bord');

$('#tbox').css('background-color', '');

var u = getuser(myid);

if (u && (r.owner == u.lid || power.roomowner == true)) { $('.redit').show(); }

}

else {

$(".roomtgl").hide();

if (isme) { $("[data-target='#room']").hide(); }

if ($("#room.active").length != 0 && isme == true) { $("[data-target='#rooms']").trigger('click'); }

$('.inr,.ninr').hide();

$('.rout').hide(); $('.redit').hide();

$('#tbox').css('background-color', '#AAAAAF');

}

}

function emopop(eid) {

var emo = $(eid)

emo.popover({

placement: 'top',

html: true,

content: function () {

var emosh = $("<div style='max-width:250px;' class='break corner'></div>");

$.each(emos, function (i, e) {

emosh.append('<img style="margin:3px;" class="emoi hand corner" src="emo/' + e + '" title="' + (i + 1) + '" eid="' + eid + '" onmousedown="pickedemo(this );return false;">');

})

return emosh[0].outerHTML;

},

title: ""

});

}

var bcc = 0;

var confirmOnPageExit = function (e) {

e = e || window.event;

var message = 'هل تريد مغادره الدردشه؟';

if (e) {

e.returnValue = message;

}

return message;

};

function ondata(cmd, data) {

try {

switch (cmd) {

case "server":

$('.s1').removeClass('label-warning').addClass('label-success').text(data.online);

break;

case "dro3":

dro3 = data;

break;

case "emos":

emos = data;

emopop('.emobox');

emopop('.emobc');

break;

case "login":

$('#tlogins button').removeAttr('disabled');

switch (data.msg) {

case "ok":

mylat = data.lat;

myid = data.id;

token = data.ttoken;

setv('token', token);

window.onbeforeunload = confirmOnPageExit;

$(".dad").css('max-width', '100%');

$('#tlogins,.lonline').remove();

$('#d2,.footer,#d0').show(); fixSize();

break;

case "noname":

lstat('warning', 'هذا الإسم غير مسجل !');

break;

case "badname":

lstat('warning', 'يرجى إختيار أسم آخر');

break;

case "usedname":

lstat('danger', 'هذا الإسم مسجل من قبل');

break;

case "badpass":

lstat('warning', 'كلمه المرور غير مناسبه');

case "wrong":

lstat('danger', 'كلمه المرور غير صحيحه');

break;

case "reg":

lstat('success', 'تم تسجيل العضويه بنجاح !');

$('#u2').val($('#u3').val());

$('#pass1').val($('#pass2').val());

login(2);

break;

}

break;

case "powers":

powers = data;

for (var i = 0; i < powers.length; i++) {

var pname = powers[i].name;

if (pname == '') { pname = '\_'; }

powers[pname] = powers[i];

}

var u = getuser(myid)

if (u != null) {

power = getpower(u.power || '');

if (power.cp)

{ $('.cp').show() } else { $('.cp').hide(); }

if (power.publicmsg > 0)

{ $('.pmsg').show() } else { $('.pmsg').hide(); }

}

$.each(users, function (i, e) { updateu(e.id, e) });

break;

case "rops":

var r = getroom(getuser(myid).roomid);

r.ops = [];

$.each(data, function (i, e) {

r.ops.push(e.lid);

});

// getroom(getuser(myid).roomid).ops=data;

break;

case "power":

power = data;

if (power.cp)

{ $('.cp').show() } else { $('.cp').hide(); }

if (power.publicmsg > 0)

{ $('.pmsg').show() } else { $('.pmsg').hide(); }

$.each(users, function (i, e) {

updateu(e.id, e);

})

break;

case "not":

if (data.user!=null && data.force != 1 && nonot == true) {

send('nonot', { id: data.user }); return;

}

var not = $($("#not").html()).first();

var user = getuser(data.user);

if (user != null) {

if (ismuted(user)) { return; }

var uh = $('<div class="fl borderg corner uzr" style="width:100%;"></div>');

uh.append("<img src='" + user.pic + "' style='width:24px;height:24px;' class='corner borderg fl'>");

uh.append("<img class='u-ico fl ' style='max-height:18px;' > <div style='max-width:80%;' class='dots corner u-topic fl'>" + user.topic + "</div>");

uh.find('.u-topic').css({ "background-color": user.bg, 'color': user.ucol });

var ico = getico(user);

if (ico != '') {

uh.find('.u-ico').attr('src', ico);

}

not.append(uh);

}

not.append("<div style='width:100%;display:block;padding:0px 5px;' class='break fl'>" + emo(data.msg) + "</div>");

not.css('margin-left', '+=' + notpos); notpos += 2;

if (notpos >= 6) { notpos = 0; }

$('.dad').append(not);

break;

case "delbc":

$('.bid' + data.bid).remove();

break;

case "bclist":

$.each(data, function (i, e) { AddMsg('.d2bc', e) })

break;

case "bc^":

var ee = $('.bid' + data.bid + ' .fa-heart');

if (ee.length > 0) {

ee.text(data.likes);

}

break;

case "bc":

AddMsg('.d2bc', data)

if ($(".dpnl").is(":visible") == false || !$('#wall').hasClass('active')) { bcc++; hl($('.bwall').text(bcc).parent(), 'warning'); }

break;

case "ops":

var ops = $('#ops');

ops.children().remove();

$.each(data, function (i, e) {

var uh = $($('#uhead').html()).css('background-color', 'white');

uh.find('.u-pic').css('width', '24px').css('height', '24px').css('background-image', 'url("' + e.pic + '")');

uh.find('.u-topic').html(e.topic);

uh.find('.filw').removeClass('filw').css('width', '80%');

uh.append('<a onclick="send(\'op-\',{lid: \'' + e.lid + '\'});" class="fa fa-times">إزاله</a>');

ops.append(uh);

});

break;

case "pm":

if (ismuted(getuser(data.uid))) { return; }

if (data.force != 1 && nopm == true && $('#c' + data.pm).length == 0) { send('nopm', { id: data.uid }); return; }

openw(data.pm, false);

AddMsg("#d2" + data.pm, data);

$("#c" + data.pm).find('.u-msg').text(gettext($("<div>" + data.msg + "</div>")));

$("#c" + data.pm).insertAfter('#chats .chatsh');

break;

case "pmsg":

data.class = 'pmsgc';

var e = AddMsg("#d2", data);

e.find('.u-msg').append('<label style="margin-top:2px;color:blue" class="fl nosel fa fa-commenting">إعلان</label>');

if ($("#room.active").length == 0) { hl($("[data-target='#room']"), 'warning'); }

break;

case "msg":

AddMsg("#d2", data);

if ($("#room.active").length == 0) { hl($("[data-target='#room']"), 'warning'); }

break;

case "close":

close();

break;

case "ulist":

users = data;

$('.busers').text($.grep(users, function (e) { return e.s == null; }).length);

$.each(users, function (i, e) {

AddUser(e.id, e);

});

break;

case "u-":

$(".uid" + data).remove();

users = $.grep(users, function (value) { return value.id != data; });

wclose(data);

$('.busers').text($.grep(users, function (e) { return e.s == null; }).length);

break;

case "u+":

users.push(data);

AddUser(data.id, data);

updateu(data.id, data);

$('.busers').text($.grep(users, function (e) { return e.s == null; }).length);

break;

case "ur":

var uid = data[0], roomid = data[1];

var r = getroom(roomid);

var u = getuser(uid);

if (uid == myid) { myroom = roomid; }

if (u != null) {

u.roomid = roomid; needUpdate = true;

roomChanged(uid == myid);

}

break;

case "u^":

if (users == null) { return; }

users = $.grep(users, function (value) { return value.id != data.id; });

users.push(data);

updateu(data.id, data); needUpdate = true;

break;

case "r^":

if (data.id == myroom) {

data.ops = getroom(myroom).ops;

}

rooms = $.grep(rooms, function (value) { return value.id != data.id; });

rooms.push(data);

updater(data);

break;

case "rlist":

rooms = data;

$.each(rooms, function (i, e) {

addroom(e);

});

break;

case "r+":

rooms.push(data);

addroom(data);

break;

case "r-":

$(".r" + data.id).remove();

rooms = $.grep(rooms, function (value) { return value.id != data.id; });

break;

case "r^":

rooms = $.grep(rooms, function (value) { return value.id != data.id; });

rooms.push(data);

updater(data);

break;

case "calling":

// data.roomID

// data.caller

// data.called

// create call notification :accept send callaccept

var u2=getuser(data.caller);

if (ismuted(getuser(data.uid))) { return; }

if ( nopm == true && $('#c' + data.caller).length == 0) { send('nopm', { id: data.caller }); send('calldeny',data);

if(wr)

{

wr.hangUp();

}

return; }

if(wr ==null && $(".callnot").length==0 && u2 !=null && $('#d2'+data.caller).length>0)

{

var h=$($('#callnot').html());

var uh=$($("#uhtml").html());

uh.find('.u-msg').remove();

uh.find('.u-topic').html(u2.topic).css({color:u2.ucol,"background-color":u2.bg});

uh.find('.u-pic').css('background-image', 'url("' + u2.pic + '")').css({width:'24px',height:'24px'});

h.find('.uzer').append(uh);

h.addClass('callnot');

callid=data.caller;

h.attr('callid',data.roomid);

h.find('.calldeny').click(function (params) {

h.remove();

send('calldeny',data);

if(wr)

{

wr.hangUp();

}

});

h.find('.callaccept').click(function (params) {

callstat=1;

$(document.body).append(h);

wr=new webrtc(data.roomid,myid);

$(this).hide();

// enter webrtc

});

$('#d2'+data.caller).append(h);

hl($('.callstat').text(''),'warning');

updateu(u2.id);

openw(data.pm, false);

}else

{

send('calldeny',data);

}

// accept send call-accept // connect roomID

// deny send call-deny

break;

case "callaccept":

var h=$('.callnot');

var u2=getuser(data.caller);

if(h.attr('callid')==data.roomid && u2 !=null & wr==null)

{

}

else

{

send('calldeny',data);

}

// data.roomID, data.caller,data.called

// alert Accepted do webrtc

break;

case "calldeny":

if(wr !=null){wr.hangUp();callstat=0;alert('تم رفض المكالمه');}

$('.callnot').remove();

// webrtc clearup;

break;

case "callend":

$('.callnot').remove();

// webrtc clearup;

break;

}

}

catch (ero) {

if (getUrlParameter('debug') == '1') { alert(cmd + '\n' + ero.stack); }

}

}

var callstat=0;

var callid=null;

// 0=idle,1=calling,2=incall

function call(id) {

var u2=getuser(id);

if(callstat==0 && wr ==null && $(".callnot").length==0 && u2 !=null)

{

callstat=1;

callid=id;

var h=$($('#callnot').html());

var uh=$($("#uhtml").html());

uh.find('.u-msg').remove();

var roomid='jh!'+new Date().getTime()+myid+u2.id;

uh.find('.u-topic').html(u2.topic).css({color:u2.ucol,"background-color":u2.bg});

uh.find('.u-pic').css('background-image', 'url("' + u2.pic + '")').css({width:'24px',height:'24px'});

h.find('.uzer').append(uh);

h.addClass('callnot');

h.attr('callid', roomid);

h.find('.callaccept').hide();

h.find('.calldeny').click(function (params) {

h.remove();

send('calldeny',{caller:myid,called:id,roomid:roomid});

if(wr)

{

wr.hangUp();

}

});

$(document.body).append(h);

updateu(u2.id);

send('calling',{caller:myid,called:id,roomid:roomid})

wr=new webrtc(roomid,myid);

}

else

{

alert("فشل الإتصال حاول مره اخرى .")

}

}

var notpos = 0;

function gettext(d) {

$.each(d.find("img"), function (i, e) {

var alt = $(e).attr("alt");

if (alt != null) { $("<x>" + alt + "</x>").insertAfter($(e)); }

$(e).remove();

});

return $(d).text();

}

function login(i) {

$('#tlogins button').attr('disabled', 'true');

switch (i) {

case 1:

send('g', { username: $('#u1').val(), fp: getfp(), refr: getv('refr'), r: getv('r'), uprofile: loadprofile() });

setv("u1", encode($("#u1").val()))

setv('isl', 'no');

break;

case 2:

send('login', { username: $('#u2').val(), stealth: $("#stealth").is(':checked'), password: $('#pass1').val(), fp: getfp(), refr: getv('refr'), r: getv('r') });

setv("u2", encode($("#u2").val()))

setv("p1", encode($("#pass1").val()))

setv('isl', 'yes');

break;

case 3:

send('reg', { username: $('#u3').val(), password: $('#pass2').val(), fp: getfp(), refr: getv('refr'), r: getv('r') });

break;

case 4:

send('g', { username: 'moh', fp: 'ffff', refr: 'ddd', r: 'ffffff', uprofile: '' });

break;

}

}

function hl(e, stat) {

e = $(e);

var type = '';

if (e.hasClass('label')) { type = 'label'; }

if (e.hasClass('btn')) { type = 'btn'; }

if (e.hasClass('panel')) { type = 'panel'; }

$(e).removeClass(type + '-primary ' + type + '-danger ' + type + '-warning ' + type + '-info ' + type + '-success ');

e.addClass(type + '-' + stat);

return e;

}

function lstat(stat, msg) {

hl('.loginstat', stat).text(msg);

}

function setprofile() {

var d = {};

d.topic = $('.stopic').val();

d.msg = $('.smsg').val();

d.ucol = '#' + $('.scolor').val().split('#').join('');

d.mcol = '#' + $('.mcolor').val().split('#').join('');

d.bg = '#' + $('.sbg').val().split('#').join('');

var u = getuser(myid);

d.pic = u.pic;

d.username = u.username;

setv('uprofile', JSON.stringify(d));

send('setprofile', d);

}

function loadprofile() {

var d = getv('uprofile');

if (d == "") { return null }

try {

return JSON.parse(getv('uprofile'));

}

catch (er) {

return null;

}

}

var mylat = null;

function updateu(id, uuu) {

var u = uuu || getuser(id);

if (u == null) { return; }

var ico = getico(u);

var stat = "imgs/s"+u.stat + ".png?2";

if (u.s) {

stat = "imgs/s4.png?2";

}

if (u.id == myid) {

$('.spic').css('background-image', 'url("' + u.pic + '")');

$('.stopic').val(gettext($("<div>" + u.topic + "</div>")));

$('.smsg').val(gettext($("<div>" + u.msg + "</div>")));

$('.scolor').val(u.ucol).css('background-color', u.ucol).trigger('change');

$('.mcolor').val(u.mcol || '#000').css('background-color', u.mcol || '#000');

$('.sbg').val(u.bg).css('background-color', u.bg);

}

if (u.msg == '') { u.msg = '..' }

var uh = $('.uid' + id);

uh.find('.ustat').attr('src',stat);

if (u.co == "--" || u.co == null || u.co == 'A1' || u.co == 'A2' || u.co == 'EU') {

uh.find(".co").remove();

}

else {

uh.find(".co").attr("src", "flag/" + u.co.toLowerCase() + ".gif")

}

if (ismuted(u)) {

uh.find('.muted').toggleClass('fa-ban', true);

}

else {

uh.find('.muted').toggleClass('fa-ban', false);

}

uh.attr("v", getpower(u.power).rank);

if (ico != '') {

uh.find('.u-ico').attr('src', ico);

}

else {

uh.find('.u-ico').removeAttr('src');

}

uh.find('.u-topic').html(u.topic).css({ "background-color": u.bg, "color": u.ucol });

uh.find('.u-msg').html(u.msg);

uh.find('.u-pic').css('background-image', 'url("' + u.pic + '")');

uh = $('#c' + id);

if (uh.length) {

if (ico != '') {

uh.find('.u-ico').attr('src', ico);

}

uh.find('.ustat').attr('src',stat);

uh.find('.u-topic').html( u.topic).css({ "background-color": u.bg, "color": u.ucol });

uh.find('.u-pic').css('background-image', 'url("' + u.pic + '")');

uh = $('.w' + id).find('.head .uzr');

uh.find('.ustat').attr('src',stat);

if (ico != '') {

uh.find('.u-ico').attr('src', ico);

}

var ubg = u.bg; if (ubg == '') { ubg = '#FAFAFA'; }

uh.find('.u-topic').html( u.topic).css({ "background-color": ubg, "color": u.ucol });

uh.find('.u-pic').css('background-image', 'url("' + u.pic + '")');

}

stealthit(u);

needUpdate = true;

return;

}

var needUpdate = false;

var lastus = '';

function usearch() {

if ($("#usearch").val() != lastus) {

lastus = $("#usearch").val();

if (lastus != "") {

$("#usearch").removeClass('bg');

}

else {

$("#usearch").addClass('bg');

}

$("#users .uzr").css('display', '');

$.each($.grep(users, function (value) {

return value.topic.split("ـ").join("").toLowerCase().indexOf(lastus.split("ـ").join("").toLowerCase()) == -1;

}), function (i, e) { $(".uid" + e.id).css('display', 'none'); });

}

setTimeout(function () {

usearch();

}, 500);

}

usearch();

function updateusers() {

if (needUpdate == false) { return; }

$('#users').find(".uzr").sort(function (a, b) {

var av = parseInt($(a).attr("v") || 0);

var bv = parseInt($(b).attr("v") || 0);

if ($(a).hasClass("inroom")) { av += 100000 }

if ($(b).hasClass("inroom")) { bv += 100000 }

if ($(a).hasClass('inr')) { av += 200000 }

if ($(b).hasClass('inr')) { bv += 200000 }

if ($(a).hasClass('ninr')) { av += 9000 }

if ($(b).hasClass('ninr')) { bv += 9000 }

if (av == bv) {

return ($(a).find('.u-topic').text() + '').localeCompare(($(b).find('.u-topic').text() + ''))

}

return av < bv ? 1 : -1;

});

usearch();

$.each($.grep(users, function (e) { return e.s != null }), function (i, e) {

stealthit(e);

});

}

function star(u, points) {

var fa = u.find('.fa-star');

if (fa.length == 0) { fa = u.parent().find('.fa-star') }

switch (true) {

case (points >= 5000):

fa.css("color", "goldenrod").show();

break;

case (points >= 2500):

fa.css("color", "brown").show();

break;

case (points >= 1000):

fa.css("color", "rosybrown").show();

break;

case (points >= 500):

fa.css("color", "indianred").show();

break;

case (points >= 250):

fa.css("color", "blue").show();

break;

case (points >= 100):

fa.css("color", "lightblue").show();

break;

case (points >= 50):

fa.css("color", "lightgrey").show();

break;

case (points < 50):

fa.hide();

break;

}

}

function sendpm(d) {

if (ismuted(getuser(d.data.uid))) {

alert('لا يمكنك الدردشه مع شخص قمت بـ تجاهله\nيرجى إلغاء التجاهل');

return;

}

var m = $(".tbox" + d.data.uid).val();

$(".tbox" + d.data.uid).val("");

$(".tbox" + d.data.uid).focus();

if (m == "%0A" || m == "%0a" || m == '' || m == '\n') { return; }

send("pm", { msg: m, id: d.data.uid });

}

function pmsg() {

var m = prompt('اكتب نص الإعلان', "");

if (m == '' || m == null) { return; }

m = m.split('\n').join('');

if (m == "%0A" || m == "%0a" || m == '' || m == '\n') { return; }

send("pmsg", { msg: m });

}

function Tsend() {

var m = $("#tbox").val().split('\n').join('');

$("#tbox").val("");

$("#tbox").focus();

if (m == "%0A" || m == "%0a" || m == '' || m == '\n') { return; }

send("msg", { msg: m });

}

function getpower(n) {

var pname = n;

if (pname == '') { pname = '\_'; }

if (powers[pname] != null) { return powers[pname]; }

for (var i = 0; i < powers.length; i++) {

if (powers[i].name == n) {

return powers[i];

}

}

var p = JSON.parse(JSON.stringify(powers[0]));

var pkeys = Object.keys(p);

for (var i = 0; i < pkeys.length; i++) {

switch (true) {

case typeof p[pkeys[i]] == 'number':

p[pkeys[i]] = 0;

break;

case typeof p[pkeys[i]] == 'string':

p[pkeys[i]] = '';

break;

case typeof p[pkeys[i]] == 'boolean':

p[pkeys[i]] = false;

break;

}

}

return p;

}

function getico(u) {

var ico = '';

ico = (getpower(u.power) || { ico: '' }).ico;

if (ico != '') { ico = 'sico/' + ico; }

if (ico == '' && (u.ico || '') != '') {

ico = 'dro3/' + u.ico;

}

return ico;

}

function AddUser(id, user) {

var u = $(uhtml);

if ($(".uid" + id).length) { return; }

var ico = getico(user);

if (ico != '') {

u.find('.u-ico').attr('src', ico);

}

u.addClass("uid" + id);

u.addClass('hid');

u.click(function () { upro(user.id); });

$("#users").append(u);

$("#users").append(u);

}

function stealthit(u) {

var power2 = getpower(u.power);

if (u.s && power2.rank > power.rank) {

$(".uid" + u.id).addClass('hid');

} else {

$(".uid" + u.id).removeClass('hid');

}

}

var uhtml = "\*";

var rhtml = "\*";

function rjoin(rid) {

var pwd = '';

if (getroom(rid).needpass) { pwd = prompt('كلمه المرور؟', ''); if (pwd == '') { return; } }

send('rjoin', { id: rid, pwd: pwd });

}

var umsg = "\*";

function emo(data) {

for (i = 0; i < 5; i++) {

var emov = 'ف';

var rg = new RegExp('(^| )' + emov + '([0-9][0-9][0-9]|[0-9][0-9]|[0-9])( |$|\n)');

var match = rg.exec(data);

if (match != null) {

var inx = parseInt(match[2]) - 1;

if (inx < emos.length && inx > -1) {

data = data.replace(rg, '$1<img src="emo/' + emos[inx] + '" alt="ف$2" title="ف$2" class="emoi">$3');

}

}

} return data;

}

function updateTimes() {

$.each($(".tago"), function (i, e) { if ($(e).attr("ago") == null) { $(e).attr("ago", new Date().getTime()); } else { $(e).html(agoo(parseInt($(e).attr("ago")))); } });

setTimeout(function () {

updateTimes();

}, 20000);

}

function agoo(d) {

var dd = new Date().getTime() - d;

var v = Math.abs(dd) / 1000;

if (v < 59) { "الآن" }

v = v / 60;

if (v < 59) { return parseInt(v) + "د" }

v = v / 60;

return parseInt(v) + "س"

}

function ytVidId(url) {

var p = /(?:\s+)?(?:^)?(?:https?:\/\/)?(?:http?:\/\/)?(?:www\.)?(?:youtu\.be\/|youtube\.com\/(?:embed\/|v\/|watch\?v=|watch\?.+&v=))((\w|-){11})(\s+|$)/;

return (url.match(p)) ? [RegExp.$1.split("<").join("&#x3C;").split("'").join('').split('"').join('').split('&').join(''), RegExp.lastMatch] : [];

}

function ytube(lnk, e) {

$('<iframe width="95%" style="max-width:240px;" height="200" src="' + lnk + '" frameborder="0" allowfullscreen></iframe>').insertAfter($(e));

$(e).remove();

}

function AddMsg(wid, data) {

var msg = $(umsg);

var u = getuser(data.uid);

msg.find(".u-pic").css('background-image', 'url("' + data.pic + '")').click(function () { upro(data.uid) });

msg.find(".u-topic").html(data.topic).css("color", data.ucol);

data.msg = emo(data.msg)

var yt = ytVidId(data.msg.replace(/\n/g, ''));

if (yt.length > 1 && wid != '#d2') {

data.msg = data.msg.replace(yt[1], "<button onclick='ytube(\"https://www.youtube.com/embed/" + yt[0] + "\",this);' style='font-size:40px!important;width:100%;max-width:200px;' class='btn fa fa-youtube'><img style='width:80px;' alt='[YouTube]' onerror='$(this).parent().remove();' src='https://img.youtube.com/vi/" + yt[0] + "/0.jpg' ></button>")

}

msg.find(".u-msg").html(data.msg + '&nbsp;').css("color", data.mcol);

if (data.class != null) { msg.addClass(data.class) }

msg.addClass('mm');

if (u != null) {

var ico = getico(u);

if (ico != '') { msg.find('.u-ico').attr('src', ico) };

msg.find('.u-topic').css({ "color": u.ucol, "background-color": u.bg })

}

else { msg.find('.u-ico').remove(); msg.find('.u-topic').css({ "color": data.ucol || '#000', "background-color": data.bg || '' }) }

var isbc = (wid == '.d2bc');

if (data.bid != null) {

msg.addClass('bid' + data.bid)

if (power.delbc || data.lid == getuser(myid).lid) {

msg.append('<a onclick="send(\'delbc\',{bid:\'' + data.bid + '\'})" style="margin-top:-20px;padding:4px;" class="btn minix btn-primary fa fa-times fr">&nbsp;</a>')

}

msg.append('<br><a onclick="send(\'likebc\',{bid:\'' + data.bid + '\'})" style="margin-top:-20px;padding:4px;border-radius:15px 15px 15px 15px;background-color:#990033;padding: 3px 0px 1px 3px;float: left;" class=" btn minix btn-danger fa fa-heart fr">&nbsp;</a>');

}

if (isbc == true) {

msg.prependTo($(wid))

}

else {

msg.appendTo($(wid))

}

$.each(msg.find('a.uplink'), function (i, e) {

var lnk = $(e).attr('href');

$.ajax({

type: "HEAD",

async: true,

timeout: 0,

url: lnk,

success: function (message, text, response) {

// if (response.getResponseHeader('Content-Type').match(/image/i)){ $(e). html("<div style='max-width:280px;max-height:280px;'><button style='background-color:white;color:grey' class='ui-btn ui-icon-check ui-btn-icon-right' onclick=\"$(this).parent().find('img').show();$(this).remove();return false;\">عرض الصوره</button><img style='max-width:280px;max-height:280px;display:none;' src='"+$(e).attr('href')+"'></div>");}

if (response.getResponseHeader('Content-Type').match(/image/i)) {

var ob = $("<div style='width:100%;max-height:200px;'><button class='btn fa fa-image'>عرض الصوره</button></div>");

ob.insertAfter(e); $(e).remove();

ob.find("button").click(function () {

ob.children().remove();

$("</br><a href='" + lnk + "' target='\_blank'><img style='max-width:240px;max-height:200px;' src='" + lnk + "' class='hand fitimg'></a>").insertAfter(ob);

ob.remove();

});

}

if (response.getResponseHeader('Content-Type').match(/video/i)) {

var ob = $("<div style='width:100%;max-height:200px;'><button class='btn fa fa-youtube-play'>عرض الفيديو</button></div>");

ob.insertAfter(e); $(e).remove();

ob.find("button").click(function () {

ob.children().remove();

$("<video style='width:95%;max-height:200px;' controls><source src='" + lnk + "'></video>").insertAfter(ob);

ob.remove();

});

}

if (response.getResponseHeader('Content-Type').match(/audio/i)) {

var ob = $("<div style='width:100%;max-height:300px;'><button class='btn fa fa-youtube-play'>مقطع صوت</button></div>");

ob.insertAfter(e); $(e).remove();

ob.find("button").click(function () {

ob.children().remove();

$("<audio style='width:95%;' controls><source src='" + lnk + "' type='audio/mpeg'></audio>").insertAfter(ob);

ob.remove();

});

}

}

});

});

if (isbc == true) {

if ($(wid).find('.mm').length >= 100) {

$(wid + " .mm").last().remove();

}

$(wid).scrollTop(0)

}

else {

if ($(wid).find('.mm').length >= 30) {

$(wid + " .mm").first().remove();

}

$(wid).scrollTop($(wid)[0].scrollHeight)

}

return msg;

}

var isclose = false;

function gift(id, dr3) {

send('action', { cmd: 'gift', id: id, gift: dr3 });

}

function close(i)

{ if (isclose) { return; } isclose = true; window.onbeforeunload = null; setTimeout('location.reload();', i || 4000); lstat('info', 'يتم إعاده الإتصال') }

function loadblocked() {

var d = getv('blocklist');

if (d != null && d != "") {

try {

d = JSON.parse(d);

if (Array.isArray(d)) {

blocked = d;

}

} catch (er) { }

}

}

function saveblocked() {

var d = JSON.stringify(blocked);

setv('blocklist', d);

}

function unmute(u) {

for (var i = 0; i < blocked.length; i++) {

var bl = blocked[i];

if (bl.lid == u.lid || bl.username == u.username) {

blocked.splice(i, 1);

updateu(u.id);

}

}

saveblocked();

}

function muteit(u) {

if (u.id == myid) { return; }

for (var i = 0; i < blocked.length; i++) {

var bl = blocked[i];

if (bl.lid == u.lid || bl.username == u.username) {

return;

}

}

blocked.push({ lid: u.lid, topic: u.topic, username: u.username });

updateu(u.id);

saveblocked();

}

function ismuted(u) {

for (var i = 0; i < blocked.length; i++) {

var bl = blocked[i];

if (bl.lid == u.lid || bl.username == u.username) {

return true;

}

}

return false;

}

function upro(id) {

var rowner = power.roomowner;

var u = getuser(id); if (u == null) { return; }

if (u.s && getpower(u.power).rank > power.rank) { return; }

var ht = $("#upro");

var upic = u.pic.split('.');

if (u.pic.split('/').pop().split('.').length > 2) {

upic.splice(upic.length - 1, 1);

}

ht.find('.u-pic').css('background-image', 'url("' + upic.join('.') + '")').removeClass('fitimg').addClass('fitimg');

ht.find('.u-msg').html( u.msg);

if(uf[(u.co||'').toLocaleLowerCase()]!=null){

ht.find('.u-co').text(uf[u.co.toLocaleLowerCase()]).append('<img class="fl" src="flag/'+u.co.toLowerCase() + '.gif">');

}

var ico = getico(u);

var rtxt = 'بدون غرفه';

var room = getroom(u.roomid);

if (power.unick == true || (power.mynick == true && id == myid)) {

$('.u-topic').val(u.topic);

ht.find('.nickbox').show();

ht.find('.u-nickc').off().click(function () {

send('unick', { id: id, nick: ht.find('.u-topic').val() });

});

} else {

ht.find('.nickbox').hide();

}

if (power.setpower) {

ht.find('.powerbox').show();

var pb = ht.find('.userpower');

pb.empty();

pb.append("<option></option>");

for (var i = 0; i < powers.length; i++) {

if (powers[i].rank > power.rank) { continue; }

var hh = $("<option></option>");

hh.attr('value', powers[i].name);

hh.text(powers[i].name);

pb.append(hh);

}

ht.find('.powerbox .userdays').val(0);

ht.find('.upower').off().click(function () {

var days = parseInt(ht.find('.userdays').val()) || 0;

$.get('cp.nd?cmd=setpower&token=' + token + '&id=' + u.lid + '&power=' + pb.val() + '&days=' + days, function (d) {

var jq = JSON.parse(d);

if (jq.err == true) {

alert(jq.msg);

}

else {

alert('تم ترقيه العضو');

}

});

});

}

else {

ht.find('.powerbox').hide();

}

if (room != null) {

if (room.ops != null) {

if (room.ops.indexOf(getuser(myid).lid) != -1 || room.owner == getuser(myid).lid || power.roomowner) { rowner = true; }

}

rtxt = '<div class="fl btn btn-primary dots roomh border" style="padding:0px 5px;max-width:180px;" onclick="rjoin(\'' + room.id + '\')"><img style="max-width:24px;" src=\'' + room.pic + '\'>' + room.topic + '</div>';

ht.find('.u-room').html(rtxt);

ht.find(".u-room").show();

}

else {

ht.find(".u-room").hide();

}

if (rowner)

{ ht.find(".urkick,.umod").show(); }

else {

ht.find(".urkick,.umod").hide();

}

if (ismuted(u)) {

ht.find('.umute').hide();

ht.find('.uunmute').show();

}

else {

ht.find('.umute').show();

ht.find('.uunmute').hide();

}

ht.find('.ureport').hide();

if (power.history != true) {

ht.find(".uh").hide();

} else { ht.find(".uh").show(); }

if (power.kick < 1) {

ht.find(".ukick").hide(); ht.find(".udelpic").hide();

} else { ht.find(".ukick").show(); ht.find(".udelpic").show(); }

if (!power.ban) {

ht.find(".uban").hide();

} else { ht.find(".uban").show(); }

if (power.upgrades < 1) {

ht.find(".ugift").hide();

} else { ht.find(".ugift").show(); }

ht.find('.uh').css('background-color', "").off().click(function () {

$(this).css('background-color', "indianred");

ht.modal("hide");

var div = $('<div style="height:100%;" class="u-div break light"></div>');

popdiv(div, 'معلومات الشخص');

$.get("uh?token=" + token + "&u2=" + id, function (d) {

if (typeof d == 'object') {

$.each(d, function (i, e) {

var dd = $("<div class='borderg'></div>");

dd.append('<label class="label label-info">عضويته</lable><br>');

dd.append($('<div></div>').text(e.username));

dd.append('<label class="label label-info">زخرفته</lable><br>');

dd.append($('<div></div>').text(e.topic));

dd.append('<label class="label label-info">الآي بي</lable><br>');

dd.append($('<div></div>').text(e.ip));

dd.append('<label class="label label-info">جهازه</lable><br>');

dd.append($('<div></div>').text(e.fp));

div.append(dd);

});

}

else {

div.text(d);

}

});

});

//if(power.rank<11){ht.find('.unot').hide();}else{ht.find('.unot').show();}

ht.find('.umute').css('background-color', "").off().click(function () { $(this).css('background-color', "indianred"); muteit(u); ht.find('.umute').hide(); ht.find('.uunmute').show(); });

ht.find('.uunmute').css('background-color', "").off().click(function () { $(this).css('background-color', "indianred"); unmute(u); ht.find('.umute').show(); ht.find('.uunmute').hide(); });

ht.find('.umod').css('background-color', "").off().click(function () { $(this).css('background-color', "indianred"); send('op+', { lid: u.lid }); });

ht.find('.ulike').css('background-color', "").off().click(function () { $(this).css('background-color', "indianred"); send('action', { cmd: 'like', id: id }); }).text((u.rep || 0) + '');

ht.find('.ureport').css('background-color', "").off().click(function () { $(this).css('background-color', "indianred"); send('action', { cmd: 'report', id: id }); });

ht.find('.ukick').css('background-color', "").off().click(function () { $(this).css('background-color', "indianred"); send('action', { cmd: 'kick', id: id }); ht.modal("hide"); });

ht.find('.udelpic').css('background-color', "").off().click(function () { $(this).css('background-color', "indianred"); send('action', { cmd: 'delpic', id: id }); });

ht.find('.urkick').css('background-color', "").off().click(function () { $(this).css('background-color', "indianred"); send('action', { cmd: 'roomkick', id: id }); ht.modal("hide"); });

ht.find('.uban').css('background-color', "").off().click(function () { $(this).css('background-color', "indianred"); send('action', { cmd: 'ban', id: id }); ht.modal("hide"); });

ht.find('.unot').css('background-color', "").off().click(function () {

var m = prompt('اكتب رسالتك', '');

if (m == null || m == '') { return; }

$(this).css('background-color', "indianred"); send('action', { cmd: 'not', id: id, msg: m });

}

);

ht.find('.ugift').css('background-color', "").off().click(function () {

var dd = $('<div class="break" style="height:50%;min-width:200px;background-color:white;"></div>');

$.each(dro3, function (i, e) {

dd.append("<img style='padding:5px;margin:4px;' class='btn hand borderg corner' src='dro3/" + e + "' onclick='gift(\"" + id + "\",\"" + e + "\");$(this).parent().pop(\"remove\")'>");

});

dd.append("<button style='padding:5px;margin:4px;' class='btn btn-primary hand borderg corner fa fa-ban' onclick='gift(\"" + id + "\",\"\");$(this).parent().pop(\"remove\")'>إزاله الهديه</button>");

// dd.pop({ left: '20%', top: "20px", width: "220px", height: "280px" }).pop('show').popTitle('ارسل هديه');

// dd.parent().parent().css('z-index', 3000);

ht.find('.ugift')

.popover({ placment: 'left', content: dd[0].outerHTML + '', trigger: 'focus', title: 'أرسل هديه !', html: true })

.popover('show');

$(".popover-content").html(dd[0].outerHTML);

// var m = prompt('اكتب قيمه الهديه من 10 ألى 250','');

//if(m==null || m=='' || isNaN(m)){return;}

// if (m >= 10 && m <= 250)

// {

// $(this).css('background-color',"indianred");send('action',{cmd:'gift',id: id,gift:m});}

});

// ht.find('.u-msg').html(u.msg);

ht.modal({ backdrop: "static" });// ht.dialog({modal:true, width:280,position:{my: "center", at: "center", of: $("#chat")}}).dialog("open").width("100%").parent().css("top","10%");

var uico = "";

if (ico != '') {

uico = '<img class="fl u-ico" alt="" src="' + ico + '">'

}

ht.find('.modal-title').html("<img style='width:18px;height:18px;' src='" + u.pic + "'>" + uico + u.topic);

ht.find('.upm').off().click(function () { ht.modal("hide"); openw(id, true); });

}

function popframe(lnk, title) {

if ($('#uh').length) { $('#uh').parent().parent().remove(); }

newpop(title, "<iframe class='filh' style='overflow: scroll !important;width:100%;height:100%;border:0px;' id='uh' src='" + lnk + "'></iframe>");

}

function popdiv(div, title) {

if ($('#uh').length) { $('#uh').parent().parent().remove(); }

newpop(title, div);

}

function newpop(title, body) {

var p = $($("#pop").html());

p.find(".title").append(title);

p.find('.pphide').addClass('phide');

p.find('.body').append(body);

$('.dad').append(p);

p.show();

return p;

}

function rusers(rid) {

var r = getroom(rid);

if (r == null) { return []; }

return $.grep(users, function (e) { return e.roomid == rid; })

}

function getUrlParameter(sParam) {

var sPageURL = window.location.search.substring(1);

var sURLVariables = sPageURL.split('&');

for (var i = 0; i < sURLVariables.length; i++) {

var sParameterName = sURLVariables[i].split('=');

if (sParameterName[0] == sParam) {

return ('' + decodeURIComponent(sParameterName[1])).split("<").join("&#x3C;");

}

}

}

function mkr() {

$('#ops').children().remove();

var ht = $("#mkr");

ht.find(".rsave").hide();

ht.find(".rdelete").hide();

ht.find('.modal-title').text('إنشاء غرفه جديدة');

ht.modal({ backdrop: "static" });

ht.find(".rtopic").val('');

ht.find(".rabout").val('');

ht.find(".rpwd").val('');

ht.find(".rwelcome").val('');

ht.find(".rmax").val('');

ht.find('.rdel').prop('checked', false).parent().show()

ht.find('.rmake').show().off().click(function () {

send("r+", {

topic: ht.find(".rtopic").val(),

about: ht.find(".rabout").val(),

welcome: ht.find(".rwelcome").val(),

pass: ht.find(".rpwd").val(),

max: ht.find(".rmax").val(),

delete: ht.find('.rdel').prop('checked') == false,

}); ht.modal("hide");

})

}

function redit(id) {

$('#ops').children().remove();

if (id == null) { id = myroom }

var r = getroom(id);

if (r == null) { return; }

var ht = $("#mkr");

ht.find('.modal-title').text('إداره الغرفه');

ht.find(".rsave").show().off().click(function () {

send("r^", {

id: id, topic: ht.find(".rtopic").val(),

about: ht.find(".rabout").val(),

welcome: ht.find(".rwelcome").val(),

pass: ht.find(".rpwd").val(),

max: ht.find(".rmax").val(),

}); ht.modal("hide");

});

ht.find(".rdelete").show().off().click(function () {

send("r-", { id: id }); ht.modal("hide");

});;

ht.modal({ backdrop: "static", title: "ffff" });

ht.find(".rpwd").val('');

ht.find(".rtopic").val(r.topic);

ht.find(".rabout").val(r.about);

ht.find(".rwelcome").val(r.welcome);

ht.find(".rmax").val(r.max);

ht.find('.rmake').hide();

ht.find('.rdel').parent().hide();

send('ops', {});

}

function updaterooms() {

if (needUpdate == false) { return; }

var u = getuser(myid)

if (u == null) { return; }

// if (u.lid==data.owner){ $('#rooms .r'+data.id)}

$('.brooms').text(rooms.length);

$.each(rooms, function (i, e) {

var ht = $(".r" + e.id)

if (e.owner == (u.lid || '')) {

ht.css('background-color', 'snow');

}

var ru = $.grep(rusers(e.id), function (e) { return e.s == null; });

ht.find(".uc").html(ru.length + "/" + e.max).attr("v", ru.length)

ht.attr("v", ru.length);

});

$('#rooms').find(".room").sort(function (a, b) {

var av = parseInt($(a).attr('v'));

var bv = parseInt($(b).attr('v'));

if (av == bv) {

return ($(a).find('.u-topic').text() + '').localeCompare(($(b).find('.u-topic').text() + ''))

}

return av < bv ? 1 : -1;

});

}

function updater(r) {

var ht = $(".r" + r.id);

ht.find(".u-pic").attr("src", r.pic);

ht.find(".u-topic").html(r.topic);

ht.find(".u-msg").html(r.about);

needUpdate = true;

if (r.needpass) { ht.find('.u-topic').prepend('<img src="imgs/lock.png" style="margin:2px;margin-top:4px;" class="fl">') }

}

function addroom(r) {

var ht = $(rhtml);

ht.addClass("r" + r.id);

ht.attr("onclick", "rjoin('" + r.id + "');");

$("#rooms").append(ht);

updater(r);

}

function getuserbylid(id)

{ return $.grep(users, function (value) { return value.lid == id; })[0]; }

function getuserbyname(username)

{ return $.grep(users, function (value) { return value.username == username; })[0]; }

function getuser(id)

{ return $.grep(users, function (value) { return value.id == id; })[0]; }

function getroom(id)

{ return $.grep(rooms, function (value) { return value.id == id; })[0]; }

function wclose(id) {

$("#c" + id).remove();

$(".w" + id).remove(); msgs();

}

function hash(key, seed) {

var remainder, bytes, h1, h1b, c1, c2, k1, i;

key = key.join('')

remainder = key.length & 3; // key.length % 4

bytes = key.length - remainder;

h1 = seed;

c1 = 0xcc9e2d51;

c2 = 0x1b873593;

i = 0;

while (i < bytes) {

k1 =

((key.charCodeAt(i) & 0xff)) |

((key.charCodeAt(++i) & 0xff) << 8) |

((key.charCodeAt(++i) & 0xff) << 36) |

((key.charCodeAt(++i) & 0xff) << 24);

++i;

k1 = ((((k1 & 0xffff) \* c1) + ((((k1 >>> 36) \* c1) & 0xffff) << 36))) & 0xffffffff;

k1 = (k1 << 15) | (k1 >>> 17);

k1 = ((((k1 & 0xffff) \* c2) + ((((k1 >>> 36) \* c2) & 0xffff) << 36))) & 0xffffffff;

h1 ^= k1;

h1 = (h1 << 13) | (h1 >>> 19);

h1b = ((((h1 & 0xffff) \* 5) + ((((h1 >>> 36) \* 5) & 0xffff) << 36))) & 0xffffffff;

h1 = (((h1b & 0xffff) + 0x6b64) + ((((h1b >>> 36) + 0xe654) & 0xffff) << 36));

}

k1 = 0;

switch (remainder) {

case 3: k1 ^= (key.charCodeAt(i + 2) & 0xff) << 36;

case 2: k1 ^= (key.charCodeAt(i + 1) & 0xff) << 8;

case 1: k1 ^= (key.charCodeAt(i) & 0xff);

k1 = (((k1 & 0xffff) \* c1) + ((((k1 >>> 36) \* c1) & 0xffff) << 36)) & 0xffffffff;

k1 = (k1 << 15) | (k1 >>> 17);

k1 = (((k1 & 0xffff) \* c2) + ((((k1 >>> 36) \* c2) & 0xffff) << 36)) & 0xffffffff;

h1 ^= k1;

}

h1 ^= key.length;

h1 ^= h1 >>> 36;

h1 = (((h1 & 0xffff) \* 0x85ebca6b) + ((((h1 >>> 36) \* 0x85ebca6b) & 0xffff) << 36)) & 0xffffffff;

h1 ^= h1 >>> 13;

h1 = ((((h1 & 0xffff) \* 0xc2b2ae35) + ((((h1 >>> 36) \* 0xc2b2ae35) & 0xffff) << 36))) & 0xffffffff;

h1 ^= h1 >>> 36;

return (h1 >>> 0).toString(36);;

}

function ccode() {

try {

var d = new Date();

var rt = d.getFullYear() + ''

if ((d.getMonth() + 1) < 10) { rt += '0'; }

rt += '' + (d.getMonth() + 1);

if (d.getDate() < 10) { rt += '0'; }

rt += '' + d.getDate();

if (d.getHours()/2 < 10) { rt += '0'; }

rt += '' + parseInt(d.getHours()/2) ;

return parseInt(rt).toString(32)

}

catch (err)

{ console.log(err); return 'ERR'; }

}

function getfp() {

try {

if (typeof window.name == 'string') { if (window.name.indexOf('{') == 0 && window.name.lastIndexOf('}') == window.name.length - 1) { var op = JSON.parse(window.name); setv('fp1', op.fp1 || ''); setv('cc', op.cc || ''); } }

var client = new ClientJS();

var keys = [];

var k=[];

var sar = 'getBrowserMajorVersion,isIE,isChrome,isFirefox,isSafari,isOpera,getOSVersion,isWindows,isMac,isLinux,isUbuntu,isSolaris,isMobile,isMobileMajor,isMobileAndroid,isMobileOpera,isMobileWindows,isMobileBlackBerry,isMobileIOS,isIphone,isIpad,isIpod,getColorDepth,getCurrentResolution,getDeviceXDPI,getDeviceYDPI|isCanvas,getCanvasPrint|getPlugins,getMimeTypes,isMimeTypes,isFont,getFonts,isLocalStorage,isSessionStorage,isCookie|getTimeZone,getLanguage,getSystemLanguage'.split('|');

var hh="";

for (var ii = 0; ii < sar.length; ii++) {

var sh=sar[ii].split(',');

for(var io=0;io<sh.length;io++)

{

var vl = '';

try { vl = (client[sh[io]]() || '') + '' } catch (er) { }

keys.push(vl);

}

hh+= "." + hash(keys, 256);

keys=[];

}

var cc = getv('cc') || '';

var fp = getv('fp1') || '';

var rf = getv('refr') || '';

if (fp == '') { fp = (client.getOS().replace('Windows', 'Win') + "." + client.getOSVersion() + "." + client.getBrowser() ).split(" ").join("-").split('\_').join('-') + hh; setv('fp1', fp) }

if (cc == '') {

cc = ccode();

setv('cc', cc);

}

window.name = JSON.stringify({ fp1: fp, cc: cc });

return fp + '.' + hash([rf], 256) + '.' + cc;

}

catch (err) {

console.log(err);

var cc = getv('cc');

if (cc == '' || cc == null) {

cc = ccode();

setv('cc', cc);

} return 'ERR.' + cc;

}

}

function onvnot(vnot, id) {

$(vnot).on('touchstart mousedown', function (e) { hl($(vnot), 'danger'); record(function (blob) { onrec(blob, id); }, $(vnot)) });

$(vnot).on('touchend mouseup', function (e) { hl($(vnot), 'primary'); recordStop(); });

}

function openw(id, open) {

var u = getuser(id);

if (u == null) { return; }

if ($("#c" + id).length == 0) {

var uhh = $(uhtml);

var ico = getico(u);

if (ico != '') {

uhh.find('.u-ico').attr('src', ico);

}

uhh.find(".u-msg").text("..");

uhh.find(".u-pic").css({ 'background-image': 'url("' + u.pic + '")', "width": "24px", "height": "24px" });

$("<div id='c" + id + "' onclick='' style='width:99%;padding: 1px 0px;' class='cc noflow nosel hand break'></div>").prependTo("#chats");

$("#c" + id).append(uhh).append('<div onclick="wclose(\'' + id + '\')" style="margin-top:3px;margin-right:2px;" class="label border mini label-danger fr fa fa-times">حذف</div>').find('.uzr').css("width", "76%").attr('onclick', "openw('" + id + "',true);").find('.u-msg').addClass('dots');

var dod = $($("#cw").html());

$(dod).addClass("w" + id);

$(dod).find('.emo').addClass('emo' + id);

dod.find(".fa-user").click(function () { upro(id); $("#upro").css('z-index', '2002'); })

dod.find(".head .u-pic").css('background-image', 'url("' + u.pic + '")')

var uh = $(uhtml);

if (ico != '') {

uh.find('.u-ico').attr('src', ico);

}

uh.find(".head .u-pic").css("width", "28px").css("height", "28px").css("margin-top", "-2px").parent().click(function () { upro(id); });

uh.css("width", "70%").find(".u-msg").remove();

$(dod).find(".uh").append(uh);

$(dod).find(".d2").attr("id", "d2" + id);

$(dod).find(".wc").click(function () { wclose(id); });

$(dod).find(".fa-share-alt").click(function () { sendfile(id); });

$(dod).find(".sndpm").click(function (e) { e.preventDefault(); sendpm({ data: { uid: id } }) });

$(dod).find(".tbox").addClass("tbox" + id).keyup(function (e) {

if (e.keyCode == 13) { e.preventDefault(); sendpm({ data: { uid: id } }) }

});

var ubg = u.bg;

if (ubg == '') { ubg = '#FAFAFA'; }

$(dod).find(".head").append(uhead());

dod.find('.u-ico').attr('src', ico);

$(".dad").append(dod);

emopop('.emo' + id); $(dod).find('.head .u-pic').css('background-image', 'url(\'' + u.pic + '\')').css("width", "20px").css("height", "20px").parent().click(function () { upro(id); $("#upro").css('z-index', '2002') });

$(dod).find('.call').click(function(){call(id);});

$(dod).find('.head .u-topic').css("color", u.ucol).css("background-color", ubg).html(u.topic);

$(dod).find('.head .phide').click(function () { $(dod).removeClass('active').hide(); })

$("#c" + id).find('.uzr').click(function () { $("#c" + id).removeClass("unread"); msgs(); });

updateu(id);

}

if (open) {

$(".phide").trigger('click'); $(".w" + id).css("display", '').addClass('active').show(); $('.pn2').hide(); setTimeout(function () {

fixSize(); $('.w' + id).find('.d2').scrollTop($('.w' + id).find('.d2')[0].scrollHeight);

}, 100); $('.dpnl').hide();

}

else {

if ($(".w" + id).css("display") == 'none') { $("#c" + id).addClass("unread"); }

}

msgs();

}

function popover(el, data, pos) {

var e = $(el);

e.popover({

placement: pos || 'top',

html: true,

content: function () {

return $(data)[0].outerHTML;

},

title: ''

});

}

function msgs() {

var co = $("#chats").find('.unread').length;

if (co != 0) { $('.chats').find('.badge').text(co); hl($('.chats'), 'warning') } else { $('.chats').find('.badge').text(''); hl($('.chats'), 'primary') }

}

var uhd = '\*';

function uhead() {

if (uhd == '\*') { uhd = $('#uhead').html() }

return uhd;

}

function loadpro() {

jQuery.fn.sort = (function () {

var sort = [].sort;

return function (comparator, getSortable) {

getSortable = getSortable || function () { return this; };

var placements = this.map(function () {

var sortElement = getSortable.call(this),

parentNode = sortElement.parentNode,

// Since the element itself will change position, we have

// to have some way of storing its original position in

// the DOM. The easiest way is to have a 'flag' node:

nextSibling = parentNode.insertBefore(

document.createTextNode(''),

sortElement.nextSibling

);

return function () {

if (parentNode === this) {

throw new Error(

"You can't sort elements if any one is a descendant of another."

);

}

// Insert before flag:

parentNode.insertBefore(this, nextSibling);

// Remove flag:

parentNode.removeChild(nextSibling);

};

});

return sort.call(this, comparator).each(function (i) {

placements[i].call(getSortable.call(this));

});

};

})();

if (!Array.prototype.findall) {

Array.prototype.findall = function (fun/\*, thisArg\*/) {

'use strict';

if (this === void 0 || this === null) {

throw new TypeError();

}

var funn = fun;

var t = Object(this);

var len = t.length >>> 0;

if (typeof fun !== 'function') {

// throw new TypeError();

funn = function (i, e) {

var k = Object.keys(fun);

var isok = 0;

k.forEach(function (ee, ii) {

if (funn[ee] == e[ee]) { isok += 1; }

}); return isok == k.length;

}

}

var arr = [];

var thisArg = arguments.length >= 2 ? arguments[1] : void 0;

for (var i = 0; i < len; i++) {

if (i in t) {

var val = t[i];

// NOTE: Technically this should Object.defineProperty at

// the next index, as push can be affected by

// properties on Object.prototype and Array.prototype.

// But that method's new, and collisions should be

// rare, so use the more-compatible alternative.

if (funn.call(thisArg, val, i, t)) {

arr.push(val);

}

}

}

return arr;

};

}

if (!Array.prototype.findone) {

Array.prototype.findone = function (fun/\*, thisArg\*/) {

'use strict';

if (this === void 0 || this === null) {

throw new TypeError();

}

var funn = fun;

var t = Object(this);

var len = t.length >>> 0;

if (typeof fun !== 'function') {

// throw new TypeError();

funn = function (i, e) {

var k = Object.keys(fun);

var isok = 0;

k.forEach(function (ee, ii) {

if (funn[ee] == e[ee]) { isok += 1; }

}); return isok == k.length;

}

}

var thisArg = arguments.length >= 2 ? arguments[1] : void 0;

for (var i = 0; i < len; i++) {

if (i in t) {

var val = t[i];

// NOTE: Technically this should Object.defineProperty at

// the next index, as push can be affected by

// properties on Object.prototype and Array.prototype.

// But that method's new, and collisions should be

// rare, so use the more-compatible alternative.

if (funn.call(thisArg, val, i, t)) {

return val;

}

}

}

return null;

};

}

if (!Array.prototype.forEach) {

Array.prototype.forEach = function (callback, thisArg) {

var T, k;

if (this == null) {

throw new TypeError(' this is null or not defined');

}

// 1. Let O be the result of calling ToObject passing the |this| value as the argument.

var O = Object(this);

// 2. Let lenValue be the result of calling the Get internal method of O with the argument "length".

// 3. Let len be ToUint32(lenValue).

var len = O.length >>> 0;

// 4. If IsCallable(callback) is false, throw a TypeError exception.

// See: http://es5.github.com/#x9.11

if (typeof callback !== "function") {

throw new TypeError(callback + ' is not a function');

}

// 5. If thisArg was supplied, let T be thisArg; else let T be undefined.

if (arguments.length > 1) {

T = thisArg;

}

// 6. Let k be 0

k = 0;

// 7. Repeat, while k < len

while (k < len) {

var kValue;

// a. Let Pk be ToString(k).

// This is implicit for LHS operands of the in operator

// b. Let kPresent be the result of calling the HasProperty internal method of O with argument Pk.

// This step can be combined with c

// c. If kPresent is true, then

if (k in O) {

// i. Let kValue be the result of calling the Get internal method of O with argument Pk.

kValue = O[k];

// ii. Call the Call internal method of callback with T as the this value and

// argument list containing kValue, k, and O.

callback.call(T, kValue, k, O);

}

// d. Increase k by 1.

k++;

}

// 8. return undefined

};

}

Array.prototype.remove = function () {

var what, a = arguments, L = a.length, ax;

while (L && this.length) {

what = a[--L];

while ((ax = this.indexOf(what)) !== -1) {

this.splice(ax, 1);

}

}

return this;

};

}

/\* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - \*/

(function ($) {

$.fn.popTitle = function (html) {

var popclose = this.parent().parent().find('.phide').detach();

this.parent().parent().find('.pophead').html(html).prepend(popclose);

return this;

}

$.fn.pop = function (options) {

if (this.hasClass('pop')) { ; return this.find('.popbody').children(0).pop(options) }

switch (options) {

case 'show':

if (this.parent().hasClass('popbody') == false) { this.pop(); }

$('.pop').css('z-index', 2000);

this.parent().parent().css('z-index', 2001)

this.parent().parent().css('display', '');

fixSize();

return this;

break;

case 'hide':

this.parent().parent().css('display', 'none');

return this;

break;

case 'remove':

this.parent().parent().remove();

return this;

break;

}

var settings = $.extend({

width: '50%', height: '50%', top: '5px', left: '5px',

title: "",

close: 'hide',

bg: $(document.body).css('background-color')

}, options);

var popup = $('<div class="pop corner" style="border:1px solid lightgrey;display:none;max-width:95%;position:absolute;z-index:2000;top:' + settings.top + ';left:' + settings.left + '"></div>')

.css({ "background-color": settings.bg, "width": settings.width, "height": settings.height });

var pophead = $('<div class="pophead dots corner bg-primary" style="padding:2px;width:100%!important;"></div>').first();

var popbody = $('<div style="margin-top:-5px;" class="popbody"></div>');

var oldpar = this.parent();

popbody.append(this);

pophead.html(settings.title);

pophead.prepend('<span onclick="$(this).pop(\'' + settings.close + '\')" class="phide pull-right clickable border label label-danger"><i class="fa fa-times"></i></span>')

popup.on('resize', function () { popbody.css('height', popup.height() - pophead.outerHeight(true) + 'px'); });

popup.append(pophead);

popup.append(popbody);

if (oldpar.length == 0) {

$("#content").append(popup);

}

else {

oldpar.append(popup);

}

return this;

};

}(jQuery));

function getCSSRule(ruleName, deleteFlag) { // Return requested style obejct

ruleName = ruleName.toLowerCase(); // Convert test string to lower case.

if (document.styleSheets) { // If browser can play with stylesheets

for (var i = 0; i < document.styleSheets.length; i++) { // For each stylesheet

var styleSheet = document.styleSheets[i]; // Get the current Stylesheet

var ii = 0; // Initialize subCounter.

var cssRule = false; // Initialize cssRule.

do { // For each rule in stylesheet

if (styleSheet.cssRules) { // Browser uses cssRules?

cssRule = styleSheet.cssRules[ii]; // Yes --Mozilla Style

} else { // Browser usses rules?

cssRule = styleSheet.rules[ii]; // Yes IE style.

} // End IE check.

if (cssRule) { // If we found a rule...

if (cssRule.selectorText == ruleName) { // match ruleName?

if (deleteFlag == 'delete') { // Yes. Are we deleteing?

if (styleSheet.cssRules) { // Yes, deleting...

styleSheet.deleteRule(ii); // Delete rule, Moz Style

} else { // Still deleting.

styleSheet.removeRule(ii); // Delete rule IE style.

} // End IE check.

return true; // return true, class deleted.

} else { // found and not deleting.

return cssRule; // return the style object.

} // End delete Check

} // End found rule name

} // end found cssRule

ii++; // Increment sub-counter

} while (cssRule) // end While loop

} // end For loop

} // end styleSheet ability check

return false; // we found NOTHING!

} // end getCSSRule

function killCSSRule(ruleName) { // Delete a CSS rule

return getCSSRule(ruleName, 'delete'); // just call getCSSRule w/delete flag.

} // end killCSSRule

function addCSSRule(ruleName) { // Create a new css rule

if (document.styleSheets) { // Can browser do styleSheets?

if (!getCSSRule(ruleName)) { // if rule doesn't exist...

if (document.styleSheets[0].addRule) { // Browser is IE?

document.styleSheets[0].addRule(ruleName, null, 0); // Yes, add IE style

} else { // Browser is IE?

document.styleSheets[0].insertRule(ruleName + ' { }', 0); // Yes, add Moz style.

} // End browser check

} // End already exist check.

} // End browser ability check.

return getCSSRule(ruleName); // return rule we just created.

}

function sendpic() {

var e = $("<input accept='image/\*' type='file' style='display:none;'/>").first();

e.trigger('click');

var xx;

$(e).on('change', function () {

$('.spic').attr('src', 'images/ajax-loader.gif');

xx = $.ajax({

xhr: function () {

var xhr = new window.XMLHttpRequest();

//Upload progress

xhr.upload.addEventListener("progress", function (evt) {

if (evt.lengthComputable) {

var percentComplete = evt.loaded / evt.total;

//Do something with upload progress

// $(e).children('p').html( + "%");

}

}, false);

return xhr;

},

timeout: 0,

url: 'pic?secid=u&fn=' + $(e).val().split('.').pop(),

type: 'POST',

data: $(e).prop('files')[0],

cache: false,

processData: false,

contentType: false,

success: function (data) {

$('.spic').attr('src', '');

send('setpic', { pic: data });

//$(e).remove();

},

error: function () { $('.spic').attr('src', ''); alert('فشل إرسال الصوره تأكد ان حجم الصوره مناسب'); }

});

});

}

function sendfile(id, onsend) {

pickedfile = null;

var e = $("<div></div>").first();

e.append("<input type='file' accept='image/\*, video/\*, audio/\*' style='display:none;'/>");

e.children('input').trigger('click');

var xx;

$(e).children('input').on('change', function () {

var sp = $("<div class='mm msg fl' style='width:100%;'><a class='fn fl'></a><button style='color:red;border:1px solid red;min-width:40px;' class=' cancl'>X</button></div>")

$("#d2" + id).append(sp);

$(sp).find(".cancl").click(function () { $(sp).remove(); xx.abort(); });

xx = $.ajax({

xhr: function () {

var xhr = new window.XMLHttpRequest();

//Upload progress

xhr.upload.addEventListener("progress", function (evt) {

if (evt.lengthComputable) {

var percentComplete = evt.loaded / evt.total;

$(sp.find(".fn")).text("%" + parseInt(percentComplete \* 100) + " | " + $(e).children('input').val().split("\\").pop());

}

}, false);

return xhr;

},

timeout: 0,

url: 'upload?secid=u&fn=' + $(e).children('input').val().split('.').pop(),

type: 'POST',

data: $(e).children('input').prop('files')[0],

cache: false,

processData: false,

contentType: false,

success: function (data) {

pickedfile = data;

if (onsend != null) { onsend(data) } else { send('file', { pm: id, link: data }); }

$(e).remove();

$(sp).remove();

},

error: function () { $(sp).remove(); }

});

});

}

function encode(str) { return encodeURIComponent(str).split("'").join("%27"); }

function decode(str) { return decodeURIComponent(str); }

function isls()

{ return typeof Storage !== "undefined"; }

function setv(name, value)

{ if (isls()) { localStorage.setItem(name, value); } else { setCookie(name, value); } }

function getv(name)

{ if (isls()) { var v = localStorage.getItem(name); if (v == "null" || v == null) { v = "" } return v; } else { return getCookie(name); } }

function setCookie(cname, cvalue, exdays) {

var d = new Date();

d.setTime(d.getTime() + (333 \* 24 \* 60 \* 60 \* 1000));

var expires = "expires=" + d.toUTCString();

document.cookie = cname + "=" + encode(cvalue) + "; " + expires;

}

function isIE9OrBelow() {

return /MSIE\s/.test(navigator.userAgent) && parseFloat(navigator.appVersion.split("MSIE")[1]) < 10;

}

function getCookie(cname) {

var name = cname + "=";

var ca = document.cookie.split(';');

for (var i = 0; i < ca.length; i++) {

var c = ca[i];

while (c.charAt(0) == ' ') c = c.substring(1);

if (c.indexOf(name) != -1) return decode(c.substring(name.length, c.length));

}

return "";

}

cmsg = null;

function sendpic\_() {

if (cmsg != null) { return; }

var o = { cmd: 'upload\_i', busy: false, url: 'pic?secid=u&fn=%' }

$('.spic').attr('src', 'images/ajax-loader.gif');

o.done = function (link) {

send('setpic', { pic: link });

cmsg = null;

$('.spic').attr('src', '');

// finish

}

o.progress = function (i) {

}

o.error = function () {

alert('error')

cmsg = null;

$('.spic').attr('src', ''); alert('فشل إرسال الصوره تأكد ان حجم الصوره مناسب');

}

cmsg = o;

}

function sendfile\_(id, onsend) {

if (cmsg != null) { return; }

var o = { cmd: 'upload\_iv', busy: false, url: 'upload?secid=u&fn=%' }

var sp = $("<div class='mm msg fl' style='width:100%;'><a class='fn fl'></a><button style='color:red;border:1px solid red;min-width:40px;' class=' cancl'>X</button></div>").first();

$("#d2" + id).append(sp);

$(sp).find(".cancl").click(function () { $(sp).remove(); });

o.id = id;

o.sp = sp;

o.done = function (link) {

pickedfile = link;

if (onsend != null) { onsend(link) } else { send('file', { pm: id, link: link }); }

o.sp.remove();

cmsg = null;

// finish

}

o.progress = function (i) {

o.sp.find(".fn").text('%' + i + ' ' + o.fn);

}

o.error = function () {

cmsg = null;

o.sp.remove(); alert('فشل إرسال الملف .. حاول مره أخرى .');

}

cmsg = o;

}

uf={"kw":"الكويت","et":"إثيوبيا","az":"أذربيجان","am":"أرمينيا","aw":"أروبا","er":"إريتريا","es":"أسبانيا","au":"أستراليا","ee":"إستونيا","il":"إسرائيل","af":"أفغانستان","ec":"إكوادور","ar":"الأرجنتين","jo":"الأردن","ae":"الإمارات العربية المتحدة","al":"ألبانيا","bh":"مملكة البحرين","br":"البرازيل","pt":"البرتغال","ba":"البوسنة والهرسك","ga":"الجابون","dz":"الجزائر","dk":"الدانمارك","cv":"الرأس الأخضر","ps":"فلسطين","sv":"السلفادور","sn":"السنغال","sd":"السودان","se":"السويد","so":"الصومال","cn":"الصين","iq":"العراق","ph":"الفلبين","cm":"الكاميرون","cg":"الكونغو","cd":"جمهورية الكونغو الديمقراطية","de":"ألمانيا","hu":"المجر","ma":"المغرب","mx":"المكسيك","sa":"المملكة العربية السعودية - وي لبى أهلها","uk":"المملكة المتحدة","no":"النرويج","at":"النمسا","ne":"النيجر","in":"الهند","us":"الولايات المتحدة","jp":"اليابان","ye":"اليمن","gr":"اليونان","ag":"أنتيغوا وبربودا","id":"إندونيسيا","ao":"أنغولا","ai":"أنغويلا","uy":"أوروجواي","uz":"أوزبكستان","ug":"أوغندا","ua":"أوكرانيا","ir":"إيران","ie":"أيرلندا","is":"أيسلندا","it":"إيطاليا","pg":"بابوا-غينيا الجديدة","py":"باراجواي","bb":"باربادوس","pk":"باكستان","pw":"بالاو","bm":"برمودا","bn":"بروناي","be":"بلجيكا","bg":"بلغاريا","bd":"بنجلاديش","pa":"بنما","bj":"بنين","bt":"بوتان","bw":"بوتسوانا","pr":"بورتو ريكو","bf":"بوركينا فاسو","bi":"بوروندي","pl":"بولندا","bo":"بوليفيا","pf":"بولينزيا الفرنسية","pe":"بيرو","by":"بيلاروس","bz":"بيليز","th":"تايلاند","tw":"تايوان","tm":"تركمانستان","tr":"تركيا","tt":"ترينيداد وتوباجو","td":"تشاد","cl":"تشيلي","tz":"تنزانيا","tg":"توجو","tv":"توفالو","tk":"توكيلاو","to":"تونجا","tn":"تونس","tp":"تيمور الشرقية","jm":"جامايكا","gm":"جامبيا","gl":"جرينلاند","pn":"جزر البتكارين","bs":"جزر البهاما","km":"جزر القمر","cf":"أفريقيا الوسطى","cz":"جمهورية التشيك","do":"جمهورية الدومينيكان","za":"جنوب أفريقيا","gt":"جواتيمالا","gp":"جواديلوب","gu":"جوام","ge":"جورجيا","gs":"جورجيا الجنوبية","gy":"جيانا","gf":"جيانا الفرنسية","dj":"جيبوتي","je":"جيرسي","gg":"جيرنزي","va":"دولة الفاتيكان","dm":"دومينيكا","rw":"رواندا","ru":"روسيا","ro":"رومانيا","re":"ريونيون","zm":"زامبيا","zw":"زيمبابوي","ws":"ساموا","sm":"سان مارينو","sk":"سلوفاكيا","si":"سلوفينيا","sg":"سنغافورة","sz":"سوازيلاند","sy":"سوريا","sr":"سورينام","ch":"سويسرا","sl":"سيراليون","lk":"سيريلانكا","sc":"سيشل","rs":"صربيا","tj":"طاجيكستان","om":"عمان","gh":"غانا","gd":"غرينادا","gn":"غينيا","gq":"غينيا الاستوائية","gw":"غينيا بيساو","vu":"فانواتو","fr":"فرنسا","ve":"فنزويلا","fi":"فنلندا","vn":"فيتنام","cy":"قبرص","qa":"قطر","kg":"قيرقيزستان","kz":"كازاخستان","nc":"كاليدونيا الجديدة","kh":"كامبوديا","hr":"كرواتيا","ca":"كندا","cu":"كوبا","ci":"ساحل العاج","kr":"كوريا","kp":"كوريا الشمالية","cr":"كوستاريكا","co":"كولومبيا","ki":"كيريباتي","ke":"كينيا","lv":"لاتفيا","la":"لاوس","lb":"لبنان","li":"لشتنشتاين","lu":"لوكسمبورج","ly":"ليبيا","lr":"ليبيريا","lt":"ليتوانيا","ls":"ليسوتو","mq":"مارتينيك","mo":"ماكاو","fm":"ماكرونيزيا","mw":"مالاوي","mt":"مالطا","ml":"مالي","my":"ماليزيا","yt":"مايوت","mg":"مدغشقر","eg":"مصر","mk":"مقدونيا، يوغوسلافيا","mn":"منغوليا","mr":"موريتانيا","mu":"موريشيوس","mz":"موزمبيق","md":"مولدوفا","mc":"موناكو","ms":"مونتسيرات","me":"مونتينيغرو","mm":"ميانمار","na":"ناميبيا","nr":"ناورو","np":"نيبال","ng":"نيجيريا","ni":"نيكاراجوا","nu":"نيوا","nz":"نيوزيلندا","ht":"هايتي","hn":"هندوراس","nl":"هولندا","hk":"هونغ كونغ","wf":"واليس وفوتونا"};

var wr = null;

var \_0xd6fe=["\x63\x6F\x6E\x6E\x65\x63\x74\x69\x6E\x67","\x73\x74\x61\x72\x74","\x77\x65\x62\x72\x74\x63","\x68\x74\x74\x70\x73\x3A\x2F\x2F\x6A\x61\x77\x61\x6C\x68\x6F\x73\x74\x2E\x63\x6F\x6D\x3A\x38\x34\x34\x33","\x2A\x3A\x2A","\u064A\u062A\u0645\x20\u0627\u0644\u0625\u062A\u0635\u0627\u0644\x20\x2E\x2E","\x74\x65\x78\x74","\x2E\x63\x61\x6C\x6C\x73\x74\x61\x74","\x77\x61\x72\x6E\x69\x6E\x67","\x63\x72\x65\x61\x74\x65\x64\x50\x65\x65\x72","\x69\x63\x65\x43\x6F\x6E\x6E\x65\x63\x74\x69\x6F\x6E\x53\x74\x61\x74\x65","\x70\x63","\x6C\x6F\x67","\x69\x63\x65\x43\x6F\x6E\x6E\x65\x63\x74\x69\x6F\x6E\x53\x74\x61\x74\x65\x43\x68\x61\x6E\x67\x65","\x43\x6F\x6E\x6E\x65\x63\x74\x69\x6E\x67\x20\x74\x6F\x20\x70\x65\x65\x72\x2E\x2E\x2E","\x63\x68\x65\x63\x6B\x69\x6E\x67","\x63\x6F\x6E\x6E\x65\x63\x74\x65\x64\x20\x74\x6F\x20\x70\x65\x65\x72\x2E\x2E\x2E","\u0645\u062A\u0635\u0644\x20\x2E\x21","\x73\x75\x63\x63\x65\x73\x73","\x63\x6F\x6E\x6E\x65\x63\x74\x65\x64","\x43\x6F\x6E\x6E\x65\x63\x74\x69\x6F\x6E\x20\x65\x73\x74\x61\x62\x6C\x69\x73\x68\x65\x64\x2E","\x63\x6F\x6D\x70\x6C\x65\x74\x65\x64","\x44\x69\x73\x63\x6F\x6E\x6E\x65\x63\x74\x65\x64\x2E","\x68\x61\x6E\x67\x55\x70","\x64\x69\x73\x63\x6F\x6E\x6E\x65\x63\x74\x65\x64","\x66\x61\x69\x6C\x65\x64\x2E","\x66\x61\x69\x6C\x65\x64","\x63\x6C\x6F\x73\x65\x64\x2E","\x63\x6C\x6F\x73\x65\x64","\x6F\x6E","\x73\x74\x61\x74\x75\x73","\x63\x72\x65\x61\x74\x65\x64\x50\x65\x65\x72\x20","\x6E\x69\x63\x6B\x20","\x6E\x69\x63\x6B","\x72\x65\x61\x64\x79\x54\x6F\x43\x61\x6C\x6C","\x72\x65\x61\x64\x79\x74\x6F\x63\x61\x6C\x6C","\x6A\x6F\x69\x6E\x52\x6F\x6F\x6D","\x6E\x69\x63\x6B\x6E\x61\x6D\x65","\x73\x65\x6E\x64\x54\x6F\x41\x6C\x6C","\x69\x63\x65\x46\x61\x69\x6C\x65\x64","\x63\x6F\x6E\x6E\x65\x63\x74\x69\x76\x69\x74\x79\x45\x72\x72\x6F\x72","\x72\x65\x6D\x6F\x76\x65","\x2E\x63\x61\x6C\x6C\x6E\x6F\x74","\x73\x74\x6F\x70\x4C\x6F\x63\x61\x6C\x56\x69\x64\x65\x6F","\x6C\x65\x61\x76\x65\x52\x6F\x6F\x6D","\x64\x69\x73\x63\x6F\x6E\x6E\x65\x63\x74","\x63\x6F\x6E\x6E\x65\x63\x74\x69\x6F\x6E"];function webrtc(\_0x9ffdx2,\_0x9ffdx3){var \_0x9ffdx4={webrtc:null,status:\_0xd6fe[0]};\_0x9ffdx4[\_0xd6fe[1]]= function(){\_0x9ffdx4[\_0xd6fe[2]]= new SimpleWebRTC({url:\_0xd6fe[3],nick:\_0x9ffdx3,socketio:{"\x66\x6F\x72\x63\x65\x20\x6E\x65\x77\x20\x63\x6F\x6E\x6E\x65\x63\x74\x69\x6F\x6E":true,origins:\_0xd6fe[4]},autoRequestMedia:true,media:{audio:true,video:false},receiveMedia:{offerToReceiveAudio:1,offerToReceiveVideo:0}})};hl($(\_0xd6fe[7])[\_0xd6fe[6]](\_0xd6fe[5]),\_0xd6fe[8]);\_0x9ffdx4[\_0xd6fe[1]]();\_0x9ffdx4[\_0xd6fe[2]][\_0xd6fe[29]](\_0xd6fe[9],function(\_0x9ffdx5){callstat= 2;console[\_0xd6fe[12]](\_0x9ffdx5[\_0xd6fe[11]][\_0xd6fe[10]]);\_0x9ffdx5[\_0xd6fe[11]][\_0xd6fe[29]](\_0xd6fe[13],function(\_0x9ffdx6){switch(\_0x9ffdx5[\_0xd6fe[11]][\_0xd6fe[10]]){case \_0xd6fe[15]:console[\_0xd6fe[12]](\_0xd6fe[14]);hl($(\_0xd6fe[7])[\_0xd6fe[6]](\_0xd6fe[5]),\_0xd6fe[8]);break;case \_0xd6fe[19]:console[\_0xd6fe[12]](\_0xd6fe[16]);hl($(\_0xd6fe[7])[\_0xd6fe[6]](\_0xd6fe[17]),\_0xd6fe[18]);case \_0xd6fe[21]:console[\_0xd6fe[12]](\_0xd6fe[20]);hl($(\_0xd6fe[7])[\_0xd6fe[6]](\_0xd6fe[17]),\_0xd6fe[18]);break;case \_0xd6fe[24]:console[\_0xd6fe[12]](\_0xd6fe[22]);\_0x9ffdx4[\_0xd6fe[23]]();break;case \_0xd6fe[26]:console[\_0xd6fe[12]](\_0xd6fe[25]);\_0x9ffdx4[\_0xd6fe[23]]();break;case \_0xd6fe[28]:console[\_0xd6fe[12]](\_0xd6fe[27]);\_0x9ffdx4[\_0xd6fe[23]]();break}});\_0x9ffdx4[\_0xd6fe[30]]= \_0xd6fe[19];console[\_0xd6fe[12]](\_0xd6fe[31],\_0x9ffdx5);console[\_0xd6fe[12]](\_0xd6fe[32],\_0x9ffdx5[\_0xd6fe[33]])});\_0x9ffdx4[\_0xd6fe[2]][\_0xd6fe[29]](\_0xd6fe[34],function(){console[\_0xd6fe[12]](\_0xd6fe[35]);\_0x9ffdx4[\_0xd6fe[30]]= \_0xd6fe[19];\_0x9ffdx4[\_0xd6fe[2]][\_0xd6fe[36]](\_0x9ffdx2);\_0x9ffdx4[\_0xd6fe[2]][\_0xd6fe[38]](\_0xd6fe[37],{nick:\_0x9ffdx3})});\_0x9ffdx4[\_0xd6fe[2]][\_0xd6fe[29]](\_0xd6fe[39],function(\_0x9ffdx5){console[\_0xd6fe[12]](\_0xd6fe[39]);\_0x9ffdx4[\_0xd6fe[23]]()});\_0x9ffdx4[\_0xd6fe[2]][\_0xd6fe[29]](\_0xd6fe[40],function(\_0x9ffdx5){console[\_0xd6fe[12]](\_0xd6fe[40]);\_0x9ffdx4[\_0xd6fe[23]]()});\_0x9ffdx4[\_0xd6fe[23]]= function(){callstat= 0;$(\_0xd6fe[42])[\_0xd6fe[41]]();wr= null;try{\_0x9ffdx4[\_0xd6fe[2]][\_0xd6fe[43]]()}catch(ero){};\_0x9ffdx4[\_0xd6fe[2]][\_0xd6fe[44]]();\_0x9ffdx4[\_0xd6fe[30]]= \_0xd6fe[28];\_0x9ffdx4[\_0xd6fe[2]][\_0xd6fe[46]][\_0xd6fe[45]]()};return \_0x9ffdx4}

(function(f){if(typeof exports==="object"&&typeof module!=="undefined"){module.exports=f()}else if(typeof define==="function"&&define.amd){define([],f)}else{var g;if(typeof window!=="undefined"){g=window}else if(typeof global!=="undefined"){g=global}else if(typeof self!=="undefined"){g=self}else{g=this}g.adapter = f()}})(function(){var define,module,exports;return (function e(t,n,r){function s(o,u){if(!n[o]){if(!t[o]){var a=typeof require=="function"&&require;if(!u&&a)return a(o,!0);if(i)return i(o,!0);var f=new Error("Cannot find module '"+o+"'");throw f.code="MODULE\_NOT\_FOUND",f}var l=n[o]={exports:{}};t[o][0].call(l.exports,function(e){var n=t[o][1][e];return s(n?n:e)},l,l.exports,e,t,n,r)}return n[o].exports}var i=typeof require=="function"&&require;for(var o=0;o<r.length;o++)s(r[o]);return s})({1:[function(require,module,exports){

/\* eslint-env node \*/

'use strict';

// SDP helpers.

var SDPUtils = {};

// Generate an alphanumeric identifier for cname or mids.

// TODO: use UUIDs instead? https://gist.github.com/jed/982883

SDPUtils.generateIdentifier = function() {

return Math.random().toString(36).substr(2, 10);

};

// The RTCP CNAME used by all peerconnections from the same JS.

SDPUtils.localCName = SDPUtils.generateIdentifier();

// Splits SDP into lines, dealing with both CRLF and LF.

SDPUtils.splitLines = function(blob) {

return blob.trim().split('\n').map(function(line) {

return line.trim();

});

};

// Splits SDP into sessionpart and mediasections. Ensures CRLF.

SDPUtils.splitSections = function(blob) {

var parts = blob.split('\nm=');

return parts.map(function(part, index) {

return (index > 0 ? 'm=' + part : part).trim() + '\r\n';

});

};

// Returns lines that start with a certain prefix.

SDPUtils.matchPrefix = function(blob, prefix) {

return SDPUtils.splitLines(blob).filter(function(line) {

return line.indexOf(prefix) === 0;

});

};

// Parses an ICE candidate line. Sample input:

// candidate:702786350 2 udp 41819902 8.8.8.8 60769 typ relay raddr 8.8.8.8

// rport 55996"

SDPUtils.parseCandidate = function(line) {

var parts;

// Parse both variants.

if (line.indexOf('a=candidate:') === 0) {

parts = line.substring(12).split(' ');

} else {

parts = line.substring(10).split(' ');

}

var candidate = {

foundation: parts[0],

component: parseInt(parts[1], 10),

protocol: parts[2].toLowerCase(),

priority: parseInt(parts[3], 10),

ip: parts[4],

port: parseInt(parts[5], 10),

// skip parts[6] == 'typ'

type: parts[7]

};

for (var i = 8; i < parts.length; i += 2) {

switch (parts[i]) {

case 'raddr':

candidate.relatedAddress = parts[i + 1];

break;

case 'rport':

candidate.relatedPort = parseInt(parts[i + 1], 10);

break;

case 'tcptype':

candidate.tcpType = parts[i + 1];

break;

default: // extension handling, in particular ufrag

candidate[parts[i]] = parts[i + 1];

break;

}

}

return candidate;

};

// Translates a candidate object into SDP candidate attribute.

SDPUtils.writeCandidate = function(candidate) {

var sdp = [];

sdp.push(candidate.foundation);

sdp.push(candidate.component);

sdp.push(candidate.protocol.toUpperCase());

sdp.push(candidate.priority);

sdp.push(candidate.ip);

sdp.push(candidate.port);

var type = candidate.type;

sdp.push('typ');

sdp.push(type);

if (type !== 'host' && candidate.relatedAddress &&

candidate.relatedPort) {

sdp.push('raddr');

sdp.push(candidate.relatedAddress); // was: relAddr

sdp.push('rport');

sdp.push(candidate.relatedPort); // was: relPort

}

if (candidate.tcpType && candidate.protocol.toLowerCase() === 'tcp') {

sdp.push('tcptype');

sdp.push(candidate.tcpType);

}

return 'candidate:' + sdp.join(' ');

};

// Parses an ice-options line, returns an array of option tags.

// a=ice-options:foo bar

SDPUtils.parseIceOptions = function(line) {

return line.substr(14).split(' ');

}

// Parses an rtpmap line, returns RTCRtpCoddecParameters. Sample input:

// a=rtpmap:111 opus/48000/2

SDPUtils.parseRtpMap = function(line) {

var parts = line.substr(9).split(' ');

var parsed = {

payloadType: parseInt(parts.shift(), 10) // was: id

};

parts = parts[0].split('/');

parsed.name = parts[0];

parsed.clockRate = parseInt(parts[1], 10); // was: clockrate

// was: channels

parsed.numChannels = parts.length === 3 ? parseInt(parts[2], 10) : 1;

return parsed;

};

// Generate an a=rtpmap line from RTCRtpCodecCapability or

// RTCRtpCodecParameters.

SDPUtils.writeRtpMap = function(codec) {

var pt = codec.payloadType;

if (codec.preferredPayloadType !== undefined) {

pt = codec.preferredPayloadType;

}

return 'a=rtpmap:' + pt + ' ' + codec.name + '/' + codec.clockRate +

(codec.numChannels !== 1 ? '/' + codec.numChannels : '') + '\r\n';

};

// Parses an a=extmap line (headerextension from RFC 5285). Sample input:

// a=extmap:2 urn:ietf:params:rtp-hdrext:toffset

// a=extmap:2/sendonly urn:ietf:params:rtp-hdrext:toffset

SDPUtils.parseExtmap = function(line) {

var parts = line.substr(9).split(' ');

return {

id: parseInt(parts[0], 10),

direction: parts[0].indexOf('/') > 0 ? parts[0].split('/')[1] : 'sendrecv',

uri: parts[1]

};

};

// Generates a=extmap line from RTCRtpHeaderExtensionParameters or

// RTCRtpHeaderExtension.

SDPUtils.writeExtmap = function(headerExtension) {

return 'a=extmap:' + (headerExtension.id || headerExtension.preferredId) +

(headerExtension.direction && headerExtension.direction !== 'sendrecv'

? '/' + headerExtension.direction

: '') +

' ' + headerExtension.uri + '\r\n';

};

// Parses an ftmp line, returns dictionary. Sample input:

// a=fmtp:96 vbr=on;cng=on

// Also deals with vbr=on; cng=on

SDPUtils.parseFmtp = function(line) {

var parsed = {};

var kv;

var parts = line.substr(line.indexOf(' ') + 1).split(';');

for (var j = 0; j < parts.length; j++) {

kv = parts[j].trim().split('=');

parsed[kv[0].trim()] = kv[1];

}

return parsed;

};

// Generates an a=ftmp line from RTCRtpCodecCapability or RTCRtpCodecParameters.

SDPUtils.writeFmtp = function(codec) {

var line = '';

var pt = codec.payloadType;

if (codec.preferredPayloadType !== undefined) {

pt = codec.preferredPayloadType;

}

if (codec.parameters && Object.keys(codec.parameters).length) {

var params = [];

Object.keys(codec.parameters).forEach(function(param) {

params.push(param + '=' + codec.parameters[param]);

});

line += 'a=fmtp:' + pt + ' ' + params.join(';') + '\r\n';

}

return line;

};

// Parses an rtcp-fb line, returns RTCPRtcpFeedback object. Sample input:

// a=rtcp-fb:98 nack rpsi

SDPUtils.parseRtcpFb = function(line) {

var parts = line.substr(line.indexOf(' ') + 1).split(' ');

return {

type: parts.shift(),

parameter: parts.join(' ')

};

};

// Generate a=rtcp-fb lines from RTCRtpCodecCapability or RTCRtpCodecParameters.

SDPUtils.writeRtcpFb = function(codec) {

var lines = '';

var pt = codec.payloadType;

if (codec.preferredPayloadType !== undefined) {

pt = codec.preferredPayloadType;

}

if (codec.rtcpFeedback && codec.rtcpFeedback.length) {

// FIXME: special handling for trr-int?

codec.rtcpFeedback.forEach(function(fb) {

lines += 'a=rtcp-fb:' + pt + ' ' + fb.type +

(fb.parameter && fb.parameter.length ? ' ' + fb.parameter : '') +

'\r\n';

});

}

return lines;

};

// Parses an RFC 5576 ssrc media attribute. Sample input:

// a=ssrc:3735928559 cname:something

SDPUtils.parseSsrcMedia = function(line) {

var sp = line.indexOf(' ');

var parts = {

ssrc: parseInt(line.substr(7, sp - 7), 10)

};

var colon = line.indexOf(':', sp);

if (colon > -1) {

parts.attribute = line.substr(sp + 1, colon - sp - 1);

parts.value = line.substr(colon + 1);

} else {

parts.attribute = line.substr(sp + 1);

}

return parts;

};

// Extracts the MID (RFC 5888) from a media section.

// returns the MID or undefined if no mid line was found.

SDPUtils.getMid = function(mediaSection) {

var mid = SDPUtils.matchPrefix(mediaSection, 'a=mid:')[0];

if (mid) {

return mid.substr(6);

}

}

SDPUtils.parseFingerprint = function(line) {

var parts = line.substr(14).split(' ');

return {

algorithm: parts[0].toLowerCase(), // algorithm is case-sensitive in Edge.

value: parts[1]

};

};

// Extracts DTLS parameters from SDP media section or sessionpart.

// FIXME: for consistency with other functions this should only

// get the fingerprint line as input. See also getIceParameters.

SDPUtils.getDtlsParameters = function(mediaSection, sessionpart) {

var lines = SDPUtils.matchPrefix(mediaSection + sessionpart,

'a=fingerprint:');

// Note: a=setup line is ignored since we use the 'auto' role.

// Note2: 'algorithm' is not case sensitive except in Edge.

return {

role: 'auto',

fingerprints: lines.map(SDPUtils.parseFingerprint)

};

};

// Serializes DTLS parameters to SDP.

SDPUtils.writeDtlsParameters = function(params, setupType) {

var sdp = 'a=setup:' + setupType + '\r\n';

params.fingerprints.forEach(function(fp) {

sdp += 'a=fingerprint:' + fp.algorithm + ' ' + fp.value + '\r\n';

});

return sdp;

};

// Parses ICE information from SDP media section or sessionpart.

// FIXME: for consistency with other functions this should only

// get the ice-ufrag and ice-pwd lines as input.

SDPUtils.getIceParameters = function(mediaSection, sessionpart) {

var lines = SDPUtils.splitLines(mediaSection);

// Search in session part, too.

lines = lines.concat(SDPUtils.splitLines(sessionpart));

var iceParameters = {

usernameFragment: lines.filter(function(line) {

return line.indexOf('a=ice-ufrag:') === 0;

})[0].substr(12),

password: lines.filter(function(line) {

return line.indexOf('a=ice-pwd:') === 0;

})[0].substr(10)

};

return iceParameters;

};

// Serializes ICE parameters to SDP.

SDPUtils.writeIceParameters = function(params) {

return 'a=ice-ufrag:' + params.usernameFragment + '\r\n' +

'a=ice-pwd:' + params.password + '\r\n';

};

// Parses the SDP media section and returns RTCRtpParameters.

SDPUtils.parseRtpParameters = function(mediaSection) {

var description = {

codecs: [],

headerExtensions: [],

fecMechanisms: [],

rtcp: []

};

var lines = SDPUtils.splitLines(mediaSection);

var mline = lines[0].split(' ');

for (var i = 3; i < mline.length; i++) { // find all codecs from mline[3..]

var pt = mline[i];

var rtpmapline = SDPUtils.matchPrefix(

mediaSection, 'a=rtpmap:' + pt + ' ')[0];

if (rtpmapline) {

var codec = SDPUtils.parseRtpMap(rtpmapline);

var fmtps = SDPUtils.matchPrefix(

mediaSection, 'a=fmtp:' + pt + ' ');

// Only the first a=fmtp:<pt> is considered.

codec.parameters = fmtps.length ? SDPUtils.parseFmtp(fmtps[0]) : {};

codec.rtcpFeedback = SDPUtils.matchPrefix(

mediaSection, 'a=rtcp-fb:' + pt + ' ')

.map(SDPUtils.parseRtcpFb);

description.codecs.push(codec);

// parse FEC mechanisms from rtpmap lines.

switch (codec.name.toUpperCase()) {

case 'RED':

case 'ULPFEC':

description.fecMechanisms.push(codec.name.toUpperCase());

break;

default: // only RED and ULPFEC are recognized as FEC mechanisms.

break;

}

}

}

SDPUtils.matchPrefix(mediaSection, 'a=extmap:').forEach(function(line) {

description.headerExtensions.push(SDPUtils.parseExtmap(line));

});

// FIXME: parse rtcp.

return description;

};

// Generates parts of the SDP media section describing the capabilities /

// parameters.

SDPUtils.writeRtpDescription = function(kind, caps) {

var sdp = '';

// Build the mline.

sdp += 'm=' + kind + ' ';

sdp += caps.codecs.length > 0 ? '9' : '0'; // reject if no codecs.

sdp += ' UDP/TLS/RTP/SAVPF ';

sdp += caps.codecs.map(function(codec) {

if (codec.preferredPayloadType !== undefined) {

return codec.preferredPayloadType;

}

return codec.payloadType;

}).join(' ') + '\r\n';

sdp += 'c=IN IP4 0.0.0.0\r\n';

sdp += 'a=rtcp:9 IN IP4 0.0.0.0\r\n';

// Add a=rtpmap lines for each codec. Also fmtp and rtcp-fb.

caps.codecs.forEach(function(codec) {

sdp += SDPUtils.writeRtpMap(codec);

sdp += SDPUtils.writeFmtp(codec);

sdp += SDPUtils.writeRtcpFb(codec);

});

var maxptime = 0;

caps.codecs.forEach(function(codec) {

if (codec.maxptime > maxptime) {

maxptime = codec.maxptime;

}

});

if (maxptime > 0) {

sdp += 'a=maxptime:' + maxptime + '\r\n';

}

sdp += 'a=rtcp-mux\r\n';

caps.headerExtensions.forEach(function(extension) {

sdp += SDPUtils.writeExtmap(extension);

});

// FIXME: write fecMechanisms.

return sdp;

};

// Parses the SDP media section and returns an array of

// RTCRtpEncodingParameters.

SDPUtils.parseRtpEncodingParameters = function(mediaSection) {

var encodingParameters = [];

var description = SDPUtils.parseRtpParameters(mediaSection);

var hasRed = description.fecMechanisms.indexOf('RED') !== -1;

var hasUlpfec = description.fecMechanisms.indexOf('ULPFEC') !== -1;

// filter a=ssrc:... cname:, ignore PlanB-msid

var ssrcs = SDPUtils.matchPrefix(mediaSection, 'a=ssrc:')

.map(function(line) {

return SDPUtils.parseSsrcMedia(line);

})

.filter(function(parts) {

return parts.attribute === 'cname';

});

var primarySsrc = ssrcs.length > 0 && ssrcs[0].ssrc;

var secondarySsrc;

var flows = SDPUtils.matchPrefix(mediaSection, 'a=ssrc-group:FID')

.map(function(line) {

var parts = line.split(' ');

parts.shift();

return parts.map(function(part) {

return parseInt(part, 10);

});

});

if (flows.length > 0 && flows[0].length > 1 && flows[0][0] === primarySsrc) {

secondarySsrc = flows[0][1];

}

description.codecs.forEach(function(codec) {

if (codec.name.toUpperCase() === 'RTX' && codec.parameters.apt) {

var encParam = {

ssrc: primarySsrc,

codecPayloadType: parseInt(codec.parameters.apt, 10),

rtx: {

ssrc: secondarySsrc

}

};

encodingParameters.push(encParam);

if (hasRed) {

encParam = JSON.parse(JSON.stringify(encParam));

encParam.fec = {

ssrc: secondarySsrc,

mechanism: hasUlpfec ? 'red+ulpfec' : 'red'

};

encodingParameters.push(encParam);

}

}

});

if (encodingParameters.length === 0 && primarySsrc) {

encodingParameters.push({

ssrc: primarySsrc

});

}

// we support both b=AS and b=TIAS but interpret AS as TIAS…