```
varsocket = io();

var groupList = [];
var userList = [];
var tabcount = groupList.length;
var oc = 0 //temp old length of group list
var nc = groupList.length; //temp new length of group listl
var auth = false;
var thisUser = {id: 0, nick: "Bismarck", pass: "password", status: "online"};

socket.emit('getGroups', function(groups){
// puts initial groups down
groupList = groups;
tabcount = groupList.length;
getCode(groupList);
});

socket.emit('getUsers', function(users){
// grabs initial users
userList = users;
});

socket.on('updateUserList', function(newUserList){
userList = newUserList;
updateUserDisplay();
});

socket.on('groupMessage', function(user, group, message){
// put down group message
console.log(message);
displayGroupMessage(user, group, message);
});

socket.on('newGroup', function(newGroupList){
console.log('New Group');
groupList = newGroupList;
nc = groupList.length;
updateGroups(groupList);
});

```

```
//sends message to server
function sendMessage(group, message){
socket.emit('groupMessage', thisUser, group, '<b style="color:
'+hashColor(thisUser.nick)+';"> '+ thisUser.nick + '</b>
'+Date().substring(16,Date().indexOf('G'))+'</i><br>' + message);
}

// creates and returns a new user
function createNewUser(name, pass){
socket.emit('newUser', name, pass, false, function(user){
thisUser = user;
});
}

function getLandingCode(grp){
var code = "404";
socket.emit('getLandingCode', function(callback){
code = callback;
loadLandingPage(code,grp);
});
}

function createNewGroup(user, groupName){
console.log('creation of new group');
socket.emit('newGroup', user, groupName);
}

// Sets a user to the given status
function setOnline(user){
socket.emit('setStatus', "online");
}

//Sets nickname to user
function changeNickname(name){
thisUser.nick=name;
socket.emit('setNickname',thisUser, name);
$("#NicknameField").val(name); // sets the change nickname input to
requested nickname
}

// Grabs the users from the server and updates AngularJS with them
```

```
function getUsers(){
socket.emit('getUsers', function(newUserList){
userList = newUserList;
updateUserDisplay();
});
}

function updateUserDisplay(){
var scope = angular.element('[ng-controller=usersCtrl]').scope();
scope.setUserList(userList);
scope.$apply();
}
function getCode(grp){
getLandingCode(grp);
}
function updateGroups(grp){
if(nc-oc!=1){
for(i = 0; i<grp.length;i++){
oc++;
$("#menu").append('<a class="item" data-tab="tab-'+grp[i].id+'">'+grp[i].name+'</a>');
$("#tabbingwut").append('<div id="tab-'+grp[i].id+'" class="ui attached tab segment" data-tab="tab-'+grp[i].id+'" style="overflow-y: scroll; height: 70vh ; float:left; width:80vw">');
$('.menu .item')
.tab();
for(o=0;o<grp[i].messages.length;o++){
var s = '#tab-'+grp[i].id+'-spot';
var outline = '<div class= \"ui bottom attached purple text segment\">' + grp[i].messages[o] + "</div>"
$('#tab-'+grp[i].id).append(outline);
}
//$("#tab-"+grp[i].id).animate({ scrollTop: $('#tab-'+grp[i].id).prop("scrollHeight")}, 20);
}
}else{
$("#menu").append('<a class="item" data-tab="tab-'+grp[oc].id+'">'+grp[oc].name+'</a>');
$("#tabbingwut").append('<div id="tab-'+grp[oc].id+'" class="ui attached tab segment" data-tab="tab-'+grp[oc].id+'" style="overflow-y: scroll; height: 70vh ; float:left; width:80vw">');
```

```
$('.menu .item')
.tab();
oc++;
}
}



function displayGroupMessage(user, group, message) {
var s = '#tab-'+group+'-spot';
var outline = '<div class= \"ui bottom attached purple text segment\">' + message + "</div>"
$('#tab-'+group).append(outline);
$("#tab-"+group).animate({ scrollTop: $('#tab-'+group).prop("scrollHeight")}, 1000);
}
function addTab(){
tabcount = tabcount+1;
$("#menu").append('<a class="item" data-tab="tab-'+tabcount+'">'+$("#createTab").val()+'</a>');
$("#tabbingwut").append('<div id="tab-'+tabcount+'" class="ui attached tab segment" data-tab="tab-'+tabcount+'" style="overflow-y: scroll; height: 70vh ; float:left; width:80vw">');
$('.menu .item')
.tab();
}

function uploadfile(file, width, height){
var fileName = file.name;
console.log(file.name);
socket.emit('uploadFile', file.name, file, function(){
console.log("Sending image message");
console.log(file.name);
var maxWidth = 1000; // this block crops the image. It works. Do the math if you want.
var maxHeight = 500;
var ratio = width/height;
var newWidth = Math.min(width, maxWidth);
var newHeight = Math.round(newWidth / ratio);
newHeight = Math.min(newHeight, maxHeight);
newWidth = Math.round(newHeight * ratio);
```

```
sendMessage(parseInt($('#menu .active').attr("data-tab").substring(4)),'<img style="display:block; margin-left:auto; margin-right:auto;" src="upload/'+ file.name +'"" width="'+newWidth+'px"  height="'+newHeight+'px">');
});
}
function loadLandingPage(code,grp){
tabcount = tabcount+1;
$("#menu").append('<a class="active item" data-tab="tab-L"> Landing Page </a>');
$("#tabbingwut").append('<div id="tab-L" class="ui attached tab segment active" data-tab="tab-L" style="overflow-y: scroll; height: 70vh ; float:left; width:80vw">');
$("#tab-L").append(code);
  $('.menu .item')
    .tab();
updateGroups(grp);
$("#NicknameField").val(thisUser.nick); // sets the change nickname input to current nickname
}

var app = angular.module('projectApp', []);

// Controller for updating the Userlist on the client side
app.controller('usersCtrl', function($scope) {
window.usersCtrl_scope = $scope;

// Holds User list, including state, updated by server
    $scope.users = [];
//Sets users in list
    $scope.setUserList = function(userList) {
      $scope.users = userList;
    };
//Generates the iconography for the user based on their state
    $scope.getUserColor = function(user) {
    switch(user.status){
    case "online":
    return "green";
    case "offline":
    return "light grey";
    }
    };
});

app.controller('messageCtrl', function($scope) {
// $scope.groups = [];
$scope.messages = [];
 $scope.messages[1] = [];
 $scope.messages[2] = [];
 $scope.messages[3] = [];
// Updates the scope's array that represents all of the messages.
$scope.displayGroupMessage = function(usr, grp, msg) {
console.log('Displaying group message in the scope.' + grp);
$scope.messages[2].push({user: usr, group: grp, message: msg});
}
});


function hashColor(str){ // to set color represenation of usernames
var hash = 0;
    for (var i = 0; i < str.length; i++) {
       hash = str.charCodeAt(i) + ((hash << 5) - hash);
    }
    var c = (hash & 0x00FFFFFF).toString(16).toUpperCase();
    return "00000".substring(0, 6 - c.length) + c;
}
//authentication function td
function authenticate(username,password){
socket.emit('grabUserObjectByUserPass',username, password, function callback(user){
console.log(user);
if(user!=null){
thisUser = user;
auth = true;
hideLogin();
updateAdmin();
socket.emit('setStatus',thisUser,"online");
}
});
}
function cPassword(pass){
socket.emit('changePassword', user, pass);
```

```
}

function hideLogin(){
$('#mask').hide();
$('.window').hide();
}
```