**AGON CHAT PLUGIN**   
este plugin fue desarrollado haciendo uso de la librería **jQuery UI Chatbox Plugin** , parte de la documentación se encuentra en <http://magma.cs.uiuc.edu/wenpu1/chatbox.html> o <http://github.com/dexterpu/jquery.ui.chatbox>   
  
sin embargo, la configuración necesaria para el funcionamiento del mismo, con firebase se encuentra en la carpeta JS, en el archivo AgonChat.js   
donde actualmente está configurado al servidor personal en firebase, por lo que es necesario cambiar la variable config con los datos del servidor que se desea implementar ,   
  
así mismo como también está declarada una variable llamada “nodo”, que es el valor del nodo al cual hará referencia al nodo que se indique de la base de datos Firebase

**FUNCIONES DE CONEXIÓN CON FIREBASE.**

**Función PushToFirebase.**Function PushToFirebase(id,msg) {

firebase.database().ref(nodo).push({name:id,message:msg});

}

Esta función es llamada más adelante y mandara al firebase los datos que se indiquen en los parámetros

**Función startListening.**

var startListening = function() {

firebase.database().ref(nodo).on('child\_added', function(snapshot) {

var msg = snapshot.val();

msgUsername = msg.name;

msgText = msg.message;

$("#chat\_div").chatbox("option", "boxManager").addMsg( msgUsername, msgText);

});

}

**Esta función es la encargada de traer los datos de firebase, línea importante,** $("#chat\_div").chatbox("option", "boxManager").addMsg( msgUsername, msgText);

Esta línea es uso de la librería para pintar en la caja de texto del chat, para poder añadir más datos al cajón del chat es necesario sobrescribir la función **addMsg** en la librería que es ChatBoxManager.js  
  
**VARIABLES PARA EL MANEJO DE SESIONES**

var seccionLogin = document.getElementById("social");

**Esta Variable Obtiene todo el contenedor del html donde se encuentran los botones por los que se pasara a iniciar sesión**

var authService = firebase.auth();

**Obtiene la referencia de todo el módulo auth() de firebase**

var VisibilidadChat = true;

**Esta Variable es la que determinara la visibilidad del chat, por default el chat estará oculto, cuando la persona inicie sesión es que tendrá acceso a poder ver el chat, cuando uno hace una instancia nueva del chat hay un atributo hidden que podemos modificar en este caso inicialmente estará en true, por lo que no se vera.**

var username = null;

**username null, hasta que no reciba nada del sistema de login**

// para login con twiter.

var loginTwiterButton = document.getElementById("LoginTwitter");

var providerTwiter = new firebase.auth.TwitterAuthProvider();

**Necesarios para la conexión con twitter, mas info, documentación de firebase**

// para login con Facebook.

var loginFacebookButton = document.getElementById("LoginFacebook");

var providerFacebok = new firebase.auth.FacebookAuthProvider();

**Necesarios para la conexión con Facebook, mas info, documentación de firebase**

// para login con Google.

var loginGoogleButton = document.getElementById("LoginGoogle");

var providerGoogle = new firebase.auth.GoogleAuthProvider();

**Necesarios para la conexión con Google, mas info, documentación de firebase**

**Event listener en los botones sociales.**lo siguiente es añadir eventos que manejen el disparo del inicio de sesión cuando se le de click a cada uno de ellos , se explicara uno, pero aplica para los 3   
  
//login con twitter

loginTwiterButton.addEventListener("click", function () {

authService.signInWithPopup(providerTwiter)

.then(function (result) {

seccionLogin.style.display = "none";

console.log(result.user);

username = result.user.displayName;

VisibilidadChat = false;

Chat(username, VisibilidadChat);

})

.catch(function (result) {

console.log('Detectado un error:', error);

},

{remember: "none"});

});

**Luego de haber añadido el evento click sobre el botón de twitter,   
authService.signInWithPopup(providerTwiter)**

.then(function (result) {

seccionLogin.style.display = "none";

console.log(result.user);

username = result.user.displayName;

VisibilidadChat = false;

Chat(username, VisibilidadChat);

})

.catch(function (result) {

console.log('Detectado un error:', error);

},

{remember: "none"});

De la variable **authService** instanciada anteriormente , se llama el método signInWithPopup(providerTwiter) y se le pasa por parámetro la instancia del provider de twiter que lanzara la autenticación antes configurada en firebase, esa función tiene dos eventos   
.then() que indica que todo paso bien o .catch() que hubo un error   
en el .then() se pasa una función que tiene los resultados devueltos ,   
seccionLogin.style.display = "none", Se esconde la sección de los botones sociales   
, username = result.user.displayName; se iguala la variable username a los datos del inicio de sesión

VisibilidadChat = false; hacemos que la variable sea False para que se muestre

Chat(username, VisibilidadChat);

})

Se llama al chat pasándole el valor de la variable username, y la visibilidad para que muestre el chat.  
  
Esto aplica para los otros dos botones

**CHAT**function Chat (username, VisibilidadChat) {

startListening();

$("#chat\_div")

.chatbox({

id:username,

user:username,

hidden: VisibilidadChat,

title : "Agon Chat",

messageSent : function(id, user, msg) {PushToFirebase(id,msg);}});

}

Se llama inicialmente a la función, startListening();

Para que pinte todos los datos del nodo

$("#chat\_div")

.chatbox({

id:username,

user:username,

hidden: VisibilidadChat,

title : "Agon Chat",

messageSent : function(id, user, msg) {PushToFirebase(id,msg);}});

y en el div con id Chat\_div se crea al chat, la parte de messageSent, es lo que se ejecutara cuando se accione el evento para enviar un mensaje, en este caso se hará push de ese mismo mensaje a firebase, si desean sobre escribir esa función nuevamente hay que modificar el plugin que es ChatboxManager.js