

APLICACIÓN CHAT

Documentación

Objetivo

Crear un chat en tiempo real entre múltiples usuarios con nombre de usuario.

Requerimientos

El programa utiliza Meteor JS como framework y MongoDB como bases de datos.

Instale Meteor desde la página oficial www.meteor.com

Diseño

Se utiliza Bootstrap como framework para el diseño de la página, así como sus dependencias JQuery, popper.js y fourseven:scss

Base de datos

Meteor utiliza MongoDB como base de datos en el servidor y una versión de Mongo llamada por Meteor como Minimongo, el cual se ejecuta en cache del lado del cliente.

Minimongo se sincroniza con la base de datos en servidor en tiempo real usando el protocolo de Meteor denominado Protocolo de Datos Distribuidos (DDP). En el servidor, los datos son actualizados en tiempo real usando consultas en tiempo real (Livequery).

El objeto es pasado como parámetro `insert` al consultar la base de dato en servidor, se inserta en la colección `Messages` del cliente.

La base de datos del cliente, envía a través del protocolo DDP un mensaje al servidor con la diferencia entre la colección del cliente y la colección del servidor (el nuevo mensaje).

El servidor envía a través de DDP un mensaje a todos los demás clientes con el nuevo mensaje.

Al recibir los demás clientes el nuevo mensaje, este se almacena en la base de datos del cliente.

Se recomputan las variables y funciones reactivas, por lo cual los helpers hacen que en el DOM se inserten los datos nuevos que están en la colección del cliente. (En el código, `messages.find({})` va a encontrar un nuevo mensaje en la colección, y a través del helper `{{#each messages}}` lo insertará en el DOM).

Ejecutando el programa en servidor local

Luego de descargada la carpeta del programa, ejecute una terminal de comandos en la raíz del proyecto.

Actualice runtime babel para el correcto funcionamiento.

```
C:\Users\Asus>meteor npm install --save babel/runtime
```

Luego se corre el programa con el comando meteor

```
C:\Users\Asus>meteor
```

Luego de ejecutado el comando, la aplicación estará disponible en la dirección por defecto

<http://localhost:3000>

Para evitar mensajes de error en la ejecución de Meteor se comentó la primera línea en el directorio `client/main.scss` en la carpeta raíz del programa. Ejecute su editor de código para descomentar la línea.

```
//@import "{}/node_modules/bootstrap/scss/bootstrap.scss";
```

```
@import "{}/node_modules/bootstrap/scss/bootstrap.scss";
```

Logueo de usuario

Se utiliza el paquete de cuentas de Meteor para el registro de usuario y contraseña.

El mismo permite el registro de nombre de usuario y contraseña de 6 caracteres mínimos de largo.

Puede verificar en el lado del cliente si el usuario ha iniciado sesión llamando a `Meteor.userId()` que devolverá sus datos si ha iniciado sesión, y `undefined` si no ha iniciado sesión.

Puede obtener parte de la información de Meteor `Meteor.user()`. Volverá indefinido si el usuario no ha iniciado sesión, y algunos datos del usuario si lo está. No le dará ninguna contraseña, por defecto mostrará el ID de usuario del usuario, el nombre del mismo y el objeto de perfil.

Si desea comprobar si un usuario ha iniciado sesión en una página, también puede utilizar el `currentUser` del `currentUser`. `Meteor.user()` los contenidos de `Meteor.user()`.