

Protocolo para el chat

Mensajes que recibe el servidor

IDENTIFY

Identifica a un usuario en el servidor:

```
{ "type": "IDENTIFY",  
  "username": "Kimberly" }
```

En caso de éxito el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "IDENTIFY",  
  "result": "SUCCESS",  
  "extra": "Kimberly" }
```

y además manda el mensaje **NEW_USER** a los demás clientes conectados:

```
{ "type": "NEW_USER",  
  "username": "Kimberly" }
```

Si el nombre de usuario ya está siendo usado el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "IDENTIFY",  
  "result": "USER_ALREADY_EXISTS",  
  "extra": "Kimberly" }
```

STATUS

Cambia el estado de un usuario:

```
{ "type": "STATUS",  
  "status": "AWAY" }
```

Si el estado cambia exitosamente, el servidor manda el mensaje **NEW_STATUS** a los demás clientes conectados:

```
{ "type": "NEW_STATUS",  
  "username": "Kimberly",  
  "status": "AWAY" }
```

USERS

Regresa la lista de usuarios en el chat:

```
{ "type": "USERS" }
```

El servidor responde un diccionario con los nombres de usuario y sus estados:

```
{ "type": "USER_LIST",  
  "users": { "Kimberly": "ACTIVE",  
             "Luis": "BUSY",  
             "Fernando": "AWAY",  
             "Antonio": "ACTIVE" } }
```

TEXT

Manda un texto privado a un usuario:

```
{ "type": "TEXT",  
  "username": "Luis",  
  "text": "Hola Luis, ¿cómo estás?" }
```

Si el usuario destinatario existe el servidor no responde nada y envía el mensaje **TEXT_FROM** al usuario:

```
{ "type": "TEXT_FROM",  
  "username": "Kimberly",  
  "text": "Hola Luis, ¿cómo estás?" }
```

Si el usuario destinatario no existe, el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "TEXT",  
  "result": "NO_SUCH_USER",  
  "extra": "Luis" }
```

PUBLIC_TEXT

Manda un texto público a todos los usuarios conectados:

```
{ "type": "PUBLIC_TEXT",  
  "text": "¡Hola a todos!" }
```

El servidor no responde nada y se envía el mensaje **PUBLIC_TEXT_FROM** a los demás usuarios en el chat:

```
{ "type": "PUBLIC_TEXT_FROM",  
  "username": "Kimberly",  
  "text": "¡Hola todos!" }
```

NEW_ROOM

Crea un cuarto en el chat:

```
{ "type": "NEW_ROOM",  
  "roomname": "Sala 1" }
```

Si el cuarto se crea exitosamente el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "NEW_ROOM",  
  "result": "SUCCESS",  
  "extra": "Sala 1" }
```

Además, el usuario que crea el cuarto es el primero y único en el mismo inmediatamente después.

Si el nombre del cuarto ya existe, el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "NEW_ROOM",  
  "result": "ROOM_ALREADY_EXISTS",  
  "extra": "Sala 1" }
```

INVITE

Invita a uno o múltiples usuarios a un cuarto; únicamente usuarios en un cuarto pueden invitar a otros usuarios a ese cuarto:

```
{ "type": "INVITE",  
  "roomname": "Sala 1",  
  "usernames": [ "Luis", "Antonio", "Fernando" ] }
```

El cuarto y todos los usuarios deben existir, en cuyo caso el servidor no responde nada y envía el mensaje **INVITATION** a cada usuario en la lista:

```
{ "type": "INVITATION",  
  "username": "Kimberly",  
  "roomname": "Sala 1" }
```

Si el cuarto no existe, el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "INVITE",  
  "result": "NO_SUCH_ROOM",  
  "extra": "Sala 1" }
```

Si uno o más de los usuarios no existe, al detectar el primero el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "INVITE",  
  "result": "NO_SUCH_USER",  
  "extra": "Fernando" }
```

Si un usuario ya está en el cuarto o ya se le había invitado, ese usuario se ignora y no se le envía el mensaje **INVITATION**.

JOIN_ROOM

Se une a un cuarto; el usuario debió previamente ser invitado al mismo para poder unirse:

```
{ "type": "JOIN_ROOM",  
  "roomname": "Sala 1" }
```

Si el cuarto existe y el usuario fue invitado previamente al mismo, el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "JOIN_ROOM",  
  "result": "SUCCESS",  
  "extra": "Sala 1" }
```

Además el usuario se une al cuarto y el servidor envía el mensaje **JOINED_ROOM** a todos los usuarios en el cuarto:

```
{ "type": "JOINED_ROOM",
```

```
"roomname": "Sala 1",  
"username": "Fernando" }
```

Si el cuarto no existe el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "JOIN_ROOM",  
  "result": "NO_SUCH_ROOM",  
  "extra": "Sala 1" }
```

Si el usuario no fue invitado previamente al cuarto, el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "JOIN_ROOM",  
  "result": "NOT_INVITED",  
  "extra": "Sala 1" }
```

ROOM_USERS

```
{ "type": "ROOM_USERS",  
  "roomname": "Sala 1" }
```

Si el cuarto existe y el usuario se ha unido al mismo, el servidor responde con un diccionario con los usuarios y su estado:

```
{ "type": "ROOM_USER_LIST",  
  "roomname": "Sala 1",  
  "users": { "Kimberly": "ACTIVE",  
             "Luis": "AWAY",  
             "Antonio": "BUSY",  
             "Fernando": "ACTIVE" } }
```

Si el cuarto no existe el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "ROOM_USERS",  
  "result": "NO_SUCH_ROOM",  
  "extra": "Sala 1" }
```

Si el cuarto existe pero el usuario no ha sido invitado, o ha sido invitado pero no se ha unido, el servidor responde:

```
{ "type": "RESPONSE",
```

```
"operation": "ROOM_USERS",  
"result": "NOT_JOINED",  
"extra": "Sala 1" }
```

ROOM_TEXT

Manda un text a un cuarto.

```
{ "type": "ROOM_TEXT",  
  "roomname": "Sala 1",  
  "text": "¡Hola sala 1!" }
```

Si el cuarto existe y el usuario se ha unido al mismo, el servidor no responde nada y envía el mensaje **ROOM_TEXT_FROM** a todos los demás usuarios en el cuarto:

```
{ "type": "ROOM_TEXT_FROM",  
  "roomname": "Sala 1",  
  "username": "Kimberly",  
  "text": "¡Hola sala 1!" }
```

Si el cuarto no existe el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "ROOM_TEXT",  
  "result": "NO_SUCH_ROOM",  
  "extra": "Sala 1" }
```

Si el cuarto existe pero el usuario no ha sido invitado, o ha sido invitado pero no se ha unido, el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "ROOM_TEXT",  
  "result": "NOT_JOINED",  
  "extra": "Sala 1" }
```

LEAVE_ROOM

El usuario abandona un cuarto:

```
{ "type": "LEAVE_ROOM",  
  "roomname": "Sala 1" }
```

Si el cuarto existe y el usuario se ha unido al mismo, el servidor no responde nada y envía el mensaje **LEFT_ROOM** a los demás usuarios en el cuarto:

```
{ "type": "LEFT_ROOM",  
  "roomname": "Sala 1",  
  "username": "Fernando" }
```

Si el cuarto no existe el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "LEAVE_ROOM",  
  "result": "NO_SUCH_ROOM",  
  "extra": "Sala 1" }
```

Si el cuarto existe pero el usuario no ha sido invitado, o ha sido invitado pero no se ha unido, el servidor responde:

```
{ "type": "RESPONSE",  
  "operation": "LEAVE_ROOM",  
  "result": "NOT_JOINED",  
  "extra": "Sala 1" }
```

DISCONNECT

Desconecta al usuario del chat, incluyendo abandonar todos los cuartos donde se haya unido.

```
{ "type": "DISCONNECT" }
```

El servidor no responde nada y envía el mensaje **DISCONNECTED** a todos los usuarios conectados:

```
{ "type": "DISCONNECTED",  
  "username": "Luis" }
```

Además, si el usuario se había unido a cuartos, envía el mensaje **LEFT_ROOM** a cada cuarto:

```
{ "type": "LEFT_ROOM",  
  "roomname": "Sala 1",  
  "username": "Fernando" }
```

Mensajes que recibe el cliente

NEW_USER

Un nuevo usuario se conectó e identificó:

```
{ "type": "NEW_USER",  
  "username": "Luis" }
```

NEW_STATUS

Un usuario cambió su estado:

```
{ "type": "NEW_STATUS",  
  "username": "Kimberly",  
  "status": "AWAY" }
```

USER_LIST

En respuesta a **USERS**

```
{ "type": "USER_LIST",  
  "users": { "Kimberly": "ACTIVE",  
             "Luis": "BUSY",  
             "Fernando": "AWAY",  
             "Antonio": "ACTIVE" } }
```

TEXT_FROM

Recibe un texto privado:

```
{ "type": "TEXT_FROM",  
  "username": "Luis",  
  "text": "Hola Kim, bien ¿y tú?" }
```

PUBLIC_TEXT_FROM

Recibe un texto público:

```
{ "type": "PUBLIC_TEXT_FROM",  
  "username": "Kimberly",
```



```
"text": "¡Hola todos!" }
```

JOINED_ROOM

Un nuevo usuario se unió a un cuarto:

```
{ "type": "JOINED_ROOM",  
  "roomname": "Sala 1",  
  "username": "Fernando" }
```

ROOM_USER_LIST

En respuesta a [ROOM_USERS](#)

```
{ "type": "ROOM_USER_LIST",  
  "roomname": "Sala 1",  
  "users": { "Kimberly": "ACTIVE",  
             "Luis": "AWAY",  
             "Antonio": "BUSY",  
             "Fernando": "ACTIVE" } }
```

ROOM_TEXT_FROM

Recibe un texto en un cuarto:

```
{ "type": "ROOM_TEXT_FROM",  
  "roomname": "Sala 1",  
  "username": "Kimberly",  
  "text": "¡Bienvenidos a mi sala!" }
```

LEFT_ROOM

Un usuario abandonó un cuarto:

```
{ "type": "LEFT_ROOM",  
  "roomname": "Sala 1",  
  "username": "Fernando" }
```

DISCONNECTED

Un usuario se desconectó:

```
{ "type": "DISCONNECTED",  
  "username": "Luis" }
```

Notas

Los mensajes presentados son ejemplos; obviamente los nombres de usuario, de cuartos y textos particulares serán distintos.

Los nombres de usuario deben tener un límite de 8 caracteres y los nombres de cuartos un límite de 16 caracteres.

Cuando todos los usuarios de un cuarto lo hayan abandonado, el cuarto desaparece.

Un usuario siempre se conecta con el estado **ACTIVE**.

Si un usuario no se ha identificado no puede hacer nada hasta que se identifique; todo mensaje distinto de **IDENTIFY** se responderá con lo siguiente:

```
{ "type": "RESPONSE",  
  
  "operation": "INVALID",  
  
  "result": "NOT_IDENTIFIED" }
```

Después de responder, el servidor procederá a desconectar al cliente.

Si un mensaje es incompleto (por ejemplo, un **TEXT** que le falte la llave "**username**"); o falla con valores esperados (como un estado distinto de **ACTIVE**, **AWAY** y **BUSY**); o no se puede reconocer (en particular si no es un diccionario JSON con la llave "**type**"); el servidor responderá lo siguiente:

```
{ "type": "RESPONSE",  
  
  "operation": "INVALID",  
  
  "result": "INVALID" }
```

y se desconectará al cliente.