

مقدمه

در این پروژه هدف پیاده‌سازی یک چت‌سرور ساده با استفاده از چارچوب Node.js و با کمک ابزار Socket.io می‌باشد.

راه اندازی

ابتدا پروژه را دانلود کنید. پس دریافت پروژه دستورات زیر را به ترتیب اجرا کنید.

1. `npm install`
2. `npm run server`

پس از مشاهده‌ی پیام اجرای موفقیت آمیز سرور با مراجعه به آدرس `http://localhost:3000` سایت اولیه‌مان آماده شده و می‌تواند شروع به ارسال پیام کند. این چت‌سرور اکنون کارایی مناسبی ندارد و وظیفه شما در این سوال تکمیل پیاده سازی این سرور است. فرآیند ارسال و دریافت پیام به طور کلی توسط سوکت انجام می‌شود که در پروژه رخدادهای زیر تعریف شده می‌باشند و باید آنها را تکمیل کنید:

● رخداد chat message

این رخداد به کلاینت‌ها اعلام می‌کند که یک پیام جدید در سرور قرار گرفته و کلاینت‌ها پیام جدید را به انتهای لیست پیام‌های خود اضافه می‌کنند.

- کاربر برای سرور آبجکت { message, author } را ارسال می‌کند.
- سرور برای کاربر آبجکت { message, author, _id, createdAt } را ارسال می‌کند.

● رخداد request messages

کاربر در شروع فعالیت خود این رخداد را ارسال می‌کند و از سرور لیست پیام‌های بعد از یک تاریخ مشخص را درخواست می‌کند.

- کاربر یک آبجکت از نوع Date ارسال می‌کند.
- سرور اصولاً این رخداد را استفاده نمی‌کند.

● رخداد receive messages

سرور پس از دریافت رخداد قبلی و یا به طور مستقل با ارسال این رخداد از کاربر می‌خواهد که از لیست ارسال شده به عنوان لیست پیام‌ها استفاده کند.

- کاربر از این رخداد استفاده نمی‌کند.
- سرور یک لیست از پیام‌ها را با فرمتی که در بخش chat message مشخص شد برای کاربر ارسال می‌کند.

● رخداد remove chat

در این رخداد کاربر از سرور درخواست می‌کند که یک پیام مشخص را حذف کند.

- کاربر _id پیام را برای حذف به سرور ارسال می‌کند.
- سرور از این رخداد استفاده نمی‌کند.

● رخدادهای test

برای راحتی کار شما این رخداد تعریف شده که با استفاده از آن داده‌ای که برای کاربر ارسال می‌کنید در کنسول او نمایش داده می‌شود.

- کاربر از این رخداد استفاده نمی‌کند.
- سرور یک مقدار دلخواه را برای کاربر ارسال می‌کند.

در این پروژه که کدهای سمت سرور و کاربر آن در اختیار شما قرار گرفته است، شما موظف هستید توابع دو فایل `models/Message` و `models/User` را تکمیل کنید. امضای توابع به صورت کاربردی تعریف شده‌اند، به همین دلیل پیشنهاد می‌شود که آنها را تغییر ندهید. همچنین تابع `sendToUser` در فایل اصلی پروژه برای شما تعبیه شده است که به شما امکان ارسال هر پیامی به یک کاربر خاص با داشتن `userId` او را می‌دهد.

وظیفه سرور شما مدیریت پیام‌ها و ارسال آنها به کاربران می‌باشد. هنگام اتصال یک کاربر جدید نرم‌افزار شما باید به تمامی افراد اطلاع دهد که یک فرد جدید عضو سامانه شده است. همچنین هنگام خروج هر فرد نیز باید تمامی پیام‌های مربوط به او را از سیستم حذف کند. پس از ارسال یک پیام توسط یکی از کلاینت‌ها به سرور، باید آن را به تمامی کلاینت‌ها ارسال کند و همچنین در صورتی که یک فرد درخواست حذف یکی از پیام‌هایش از سیستم را صادر کند باید لیست پیام‌های همه‌ی افراد را بروزرسانی کند.