

به نام خدا

داکیومنت پروژه مبانی برنامه‌سازی



نام استاد: استاد رضا فکوری

دانشکده مهندسی کامپیوتر

نام دانشجو: احسان موفق

شماره دانشجویی: ۹۸۱۰۶۱۰۱

ترم ۱ - ۱۳۹۸

فهرست

۳ مقدمه
۳ توابع کلاینت
۳ Login_Register تابع
۳ Register تابع
۳ Log in تابع
۳ Channel تابع
۳ Creat_Channel تابع
۳ join_Channel تابع
۳ logout تابع
۴ Chat_Room تابع
۴ Send_Message تابع
۴ Refresh تابع
۴ Members تابع
۴ find_users تابع
۴ find_messages تابع
۴ Leave تابع
۴ Clear تابع
۴ سرور
۴ توابع سرور
۴ Clientregister تابع
۵ Clientlogin تابع
۵ clientCreatechannel تابع
۵ clientjoinchannel تابع
۵ sendmessage تابع
۵ clientrefresh تابع
۶ channelmembers تابع
۶ clientlogout تابع
۶ clientleave تابع
۶ client_find_user تابع
۶ client_find_message تابع

مقدمه

همانطور که می‌دانید این پروژه از دو بخش سرور و کلاینت تشکیل شده است که در هر بخش از توابع مخصوص به آن بخش استفاده شده است. در این نوشتار قصد داریم توابع نوشته شده را شرح دهیم.

توابع کلاینت

همان طور که پیشتر اشاره شد پروژه از دو بخش سرور و کلاینت ساخته شده است. در نتیجه توابع مخصوص هر بخش نوشته شده است. در ابتدا لازم به ذکر است در هر یک از توابعی که ذکر خواهد شد تابعی صدا زده می‌شود. این تابع تابع `ConnectToServer` می‌باشد. این تابع سوکتی (سوکت کلاینت) ساخته و ما را به سرور متصل می‌کند. و باعث ارتباط ما با سرور می‌شود. برای ساده سازی توکن و سوکت کلاینت گلوبال تعریف شده است.

تابع Login_Register

فراخوانی این تابع در تابع `main` بوده است. به شکل کلی این تابع صفحه اولیه برنامه‌ی ما است. که در آن شخص می‌تواند `Log in` و `Register` کند. که هر کدام از این‌ها تابعی را فراخوانی می‌کنند.

تابع Register

در این تابع دو ورودی از کاربر گرفته می‌شود. ورودی اول: `Username` و ورودی دوم: `Password` است. سپس با استفاده از توابع رشته‌ای این دو را به یکدیگر وصل کرده و با استفاده از تابع `send` به سرور ارسال می‌کنیم. در این مرحله سرور پاسخی به ما می‌دهد. این پاسخ می‌تواند ارور باشد. به این معنا که چنین یوزرنیمی در `database` وجود داشته است. اگر پیام ارسالی سرور اینگونه نبود سرور فایلی را برای این کاربر با فرمت جیسون می‌سازد.

تابع Login

در این تابع همانند تابع رجیستر دو ورودی از کاربر گرفته می‌شود. ورودی اول: `Username` و ورودی دوم: `Password` است. همانند تابع `Register` رشته‌ای از این دو تابع با فرمتی خاص می‌سازد و برای سرور ارسال می‌کند. سپس سرور پاسخی به ما می‌دهد که می‌تواند ارور باشد یا نباشد. ارور به این صورت است که: یا `Password` ورودی اشتباه است یا همچنین `Username` وجود ندارد. اگر لاگ این موفقیت آمیز بود سرور به ما یک توکن می‌دهد که با استفاده از این توکن درخواست‌های بعدیمان را می‌توانیم به سرور ارسال کنیم و با این توکن کاربر مورد نظر شناخته می‌شود. سپس وارد تابع `Channel` می‌شویم.

تابع Channel

در این تابع کاربر می‌تواند کانالی را بسازد یا وارد کانالی شود یا لاگ اوت کند. و یا صفحه را کلیز کند. با توجه به انتخاب کاربر وارد توابعی می‌شویم که به شرح زیر می‌باشد.

تابع Creat_Channel

در این تابع ما نامی را از کاربر برای نام کانال دریافت می‌کنیم. با استفاده از توابع رشته‌ای توکن گرفته شده از سرور را به همراه این نام به سرور ارسال می‌کنیم. سپس سرور به ما پیامی می‌دهد که می‌تواند ارور باشد. امکان دارد نام ارسالی ما تکراری باشد. که اگر اینگونه باشد کانال ساخته نمی‌شود و پیام اروری به ما ارسال می‌شود. اگر موفقیت آمیز بود. کانال ساخته شده و وارد کانال شده و سپس وارد تابع `Chat_Room` می‌شویم.

تابع join_Channel

مانند تابع کربت چنل نامی را از کاربر دریافت کرده و به همراه توکن کاربر به سرور ارسال می‌کنیم. اگر کانال وجود نداشته باشد. سرور به ما ارور می‌دهد. و اگر وجود داشته باشد. با موفقیت وارد کانال می‌شویم. و سپس وارد تابع `Chat_Room` می‌شویم.

تابع logout

در این تابع ما توکن کاربر را به سرور ارسال کرده و سرور کاربر را از لیست کاربرهای آنلاین حذف می‌کند. و سپس وارد تابع `Login_Register` می‌شویم.

تابع Chat_Room

در این تابع ما توابع مختلفی را با توجه به درخواست کلاینت فرا می‌خوانیم. که در زیر به آن‌ها اشاره می‌کنیم.

تابع Send_Message

در این تابع ما پیام ارسالی کاربر را دریافت کرده و به همراه توکن کاربر به سرور ارسال می‌کنیم.

تابع Refresh

در این تابع ما توکن کاربر را به سرور ارسال کرده و سرور پیام‌های خوانده نشده توسط کاربر را به ما ارسال می‌کند و ما آن پیام‌ها را چاپ می‌کنیم.

تابع Members

در این تابع ما توکن کاربر را به سرور ارسال کرده و سرور اعضای کانال را در یک لیست جیسون به ما می‌دهد و ما آن‌ها را چاپ می‌کنیم.

تابع find_users

در این تابع کاربر نام یک user را وارد می‌کند. و ما این نام را به همراه توکن کاربر به سرور ارسال می‌کنیم. و سرور پاسخی به منظور حضور و یا عدم حضور کاربر را ما می‌دهد و ما آن را چاپ می‌کنیم.

تابع find_messages

در این تابع کاربر پیامی (می‌تواند یک کاراکتر و یا یک رشته‌ای از کاراکترها باشد) را وارد می‌کند. و ما با استفاده از توابع رشته‌ای این پیام به همراه توکن کاربر را به سرور ارسال می‌کنیم و پاسخی را از سرور دریافت می‌کنیم. این پاسخ شامل فرستنده‌ی پیام کامل و خود پیام کامل شده می‌باشد. (می‌تواند چند پیام و چند فرستنده داشته باشیم)

تابع Leave

در این تابع توکن کاربر را به سرور ارسال می‌کنیم. در صورت موفقیت آمیز بودن ترک کردن، پیام اروری دریافت نخواهیم کرد. و کاربر از لیست اعضای کانال که با فرمت جیسون ذخیره شده‌اند حذف خواهد شد.

تابع Clear

این تابع صرفاً صفحه را پاک می‌کند.

سعی بر این شد تا توابع بخش کلاینت را توضیح دهیم. حال سراغ توابع بخش سرور می‌رویم.

سرور

در سرور ابتدا سوکتی ساخته و با سوکتی که در کلاینت ساخته بودیم ارتباط می‌دهیم. تابعی داریم که هر بار در تابع مین صدا زده می‌شود. این تابع تابع ConnecttoClient می‌باشد که ما را به کلاینت متصل می‌کند. حال با توجه به درخواست‌های کلاینت توابعی صدا زده می‌شود. چگونگی چک کردن این توابع با استفاده از توابع رشته‌ای صورت می‌گیرد. بعد از فراخوانی هر تابع سوکت کلاینت را می‌بندیم و بعد از درخواست دوباره‌ی کلاینت سوکت جدیدی می‌سازیم. برای راحتی استراکتی تعریف می‌کنیم شامل نام کاربر، توکن کاربر، نام کانالی که الآن در آن قرار دارد و عددی گلوبال تعریف می‌کنیم. به منظور کاربرهایی که register کرده‌اند.

توابع سرور

تابع Clientregister

در این تابع ما نام و رمزی دریافت می‌کنیم و در database دنبال این نام می‌گردیم اگر این نام وجود داشت اروری را به کلاینت می‌فرستیم. و اگر وجود نداشت فایلی را می‌سازیم که محتوای آن نام و رمز دریافتی است. سپس پیام موفقیت آمیز بودن فرایند را به کلاینت ارسال می‌کنیم.

تابع Clientlogin

در این تابع نام و رمزی را دریافت می‌کنیم و سپس در database دنبال این نام می‌گردیم. اگر این نام وجود نداشت پیام اروری را به کلاینت ارسال می‌کنیم. و اگر این نام وجود داشت محتویات فایل را بررسی کرده و در صورت عدم تطابق اروری را به کلاینت بر می‌گردانیم. در صورت صحیح وارد کردن رمز و نام کاربری اگر کاربر آنلاین نباشد رشته‌ای می‌سازیم بدین صورت که یکسری حروف کوچک و بزرگ و اعداد در آن قرار داده و با استفاده از تابع rand() و srand() به صورت رندوم از این رشته توکنی می‌سازیم و به کلاینت ارسال می‌کنیم. در غیر این صورت پیام آنلاین بودن کاربر را به کلاینت ارسال می‌کنیم. استراکتی می‌سازیم و توکن و نام کاربر را به آن می‌دهیم و تعداد کلاینت‌ها را یکی افزایش می‌دهیم (به دلیل گلابال بودن تعداد کلاینت‌ها مشکلی نخواهیم داشت).

تابع clientCreatechannel

در این تابع نام دریافت شده و توکن دریافت شده از کلاینت را بررسی می‌کنیم. بدین شکل که اگر توکن مورد نظر در استراکت ساخته شده موجود نبود کاربر آنلاین نبوده و خطا را به کلاینت بر می‌گردانیم. حال اگر در فایل‌های ساخته شده همچین نامی برای کانالی موجود بود ارور می‌دهیم و به کاربر اعلام می‌کنیم همچین نامی از قبل رزرو شده است. در غیر این صورت فایلی با محتوایی از جنس جیسون می‌سازیم. محتوای جیسون شامل پیام‌ها کاربرها و عدد مخصوص کاربر (که در تابع رفرش توضیح داده خواهد شد) می‌باشد. حال پیامی به منظور موفقیت آمیز بودن فرایند به کاربر ارسال می‌کنیم. و تابع clientjoinchannel را فرا می‌خوانیم.

تابع clientjoinchannel

در این تابع نام کانالی را به همراه توکن دریافت کرده و بررسی می‌کنیم اگر کاربر آنلاین نباشد خطای مورد نظر را بر می‌گردانیم در غیر این صورت در database دنبال نام این کانال می‌گردیم اگر نامی پیدا نکردیم خطای مورد نظر را به کلاینت ارسال می‌کنیم. اگر نامی پیدا کردیم اطلاعات درون فایل را می‌خوانیم اگر کاربر عضو کانال بود خطای مورد نظر را باز می‌گردانیم در غیر این صورت کاربر را به اعضای کانال اضافه کرده و عدد مخصوص کاربر را صفر در نظر می‌گیریم. شایان توجه است که بعد از ساخت کانال (در تابع قبلی) این تابع فراخوانده می‌شود. به دلیل یکسان بودن بافر، تداخلی در اضافه شدن کاربر به کانال نخواهیم داشت. در این جا استراکت مورد نظر را پیدا کرده و نام کانال را برای شناسایی‌های بعدی به استراکت اضافه می‌کنیم.

تابع sendmessage

در این تابع ما یک خط دریافت می‌کنیم و از آخر توکن را جدا می‌کنیم (به دلیل این که اسپیس‌های موجود در پیام را بتوانیم دریافت کنیم) و از ابتدا نیز کاراکترهایی را در نظر نمی‌گیریم به این دلیل که مشخص کننده‌ی تابع ما بوده‌اند تا بتوانیم پیام اصلی را مشخص کنیم. توکن دریافتی را در استراکت جست و جو می‌کنیم و در صورت یافتن آن نام کانال استراکت را دریافت کرده و فایل مورد نظر را باز می‌کنیم و پیام دریافتی را در آرایه‌ی مسیج با ثبت فرستنده‌ی این پیام با استفاده از توابع جیسون اضافه می‌کنیم. و اطلاعات مورد نظر را در فایل می‌ریزیم. در نهایت پیام موفقیت آمیز بودن را به کلاینت ارسال می‌کنیم.

تابع clientrefresh

در این تابع با دریافت توکن از کاربر به دنبال استراکت آن می‌گردیم اگر نام چنل در استراکت خالی بود یعنی کاربر در چنلی عضو نیست و ارور را به کلاینت ارسال می‌کنیم. در غیر این صورت فایل کانال مورد نظر را که نامش را از database دریافت کرده‌ایم باز می‌کنیم. و به دنبال نام کاربر در بین اعضا می‌گردیم در صورت یافتن آن مشخص می‌کنیم چندمین کاربر است که در کانال است بدین ترتیب عدد مخصوص کاربر را نیز دریافت می‌کنیم. و با استفاده از توابع جیسون آبجکتی می‌سازیم که در آن آرایه وجود دارد. در این آرایه پیام‌های ارسال شده توسط افراد مختلف (با استفاده از عدد مخصوص کاربر محل شروع پیام‌ها را مشخص می‌کنیم) را به آبجکت مورد نظر اضافه می‌کنیم. و آبجکت مورد نظر را با استفاده از توابع جیسون به کاربر ارسال می‌کنیم.

تابع channelmembers

در این تابع با دریافت توکن از کاربر به دنبال استراکت آن می‌گردیم اگر نام چنل در استراکت خالی بود یعنی کاربر در چنلی عضو نیست و ارور را به کلاینت ارسال می‌کنیم. در غیر این صورت فایل کانال مورد نظر را که نامش را از database دریافت کرده‌ایم باز می‌کنیم. و سپس اطلاعات مورد نظر را می‌خوانیم. اطلاعات مورد نظر اسامی افراد هستند و با استفاده از توابع جیسون این اسامی را به کاربر ارسال می‌کنیم.

تابع clientlogout

در این تابع با دریافت توکن کاربر استراکت مورد نظر را پیدا کرده و تمامی استرینگ‌های درون آن را خالی می‌کنیم.

تابع clientleave

در این تابع با دریافت توکن از کاربر به دنبال استراکت آن می‌گردیم اگر نام چنل در استراکت خالی بود یعنی کاربر در چنلی عضو نیست و ارور را به کلاینت ارسال می‌کنیم. در غیر این صورت فایل کانال مورد نظر را که نامش را از database دریافت کرده‌ایم باز می‌کنیم. با استفاده از توابع جیسون کاربر مورد نظر را از اعضا حذف کرده و عدد مخصوص آن را نیز حذف می‌کنیم و پیام موفقیت آمیز بودن این عمل را به کلاینت ارسال می‌کنیم.

تابع client_find_user

در این تابع با دریافت توکن از کاربر به دنبال استراکت آن می‌گردیم اگر نام چنل در استراکت خالی بود یعنی کاربر در چنلی عضو نیست و ارور را به کلاینت ارسال می‌کنیم. در غیر این صورت فایل کانال مورد نظر را که نامش را از database دریافت کرده‌ایم باز می‌کنیم. با استفاده از توابع جیسون اطلاعات مورد نظر را می‌خوانیم و می‌بینیم که آیا کاربر در کانال وجود دارد یا نه در صورت وجود پیام «وجود دارد» را ارسال می‌کنیم. در صورت عدم وجود پیام «وجود ندارد» را به کلاینت ارسال می‌کنیم.

تابع client_find_message

در این تابع با دریافت توکن از کاربر به دنبال استراکت آن می‌گردیم و فایل کانال مورد نظر را که نامش را از استراکت دریافت کرده‌ایم در database باز می‌کنیم. مانند تابع سند مسیج پیام مورد نظر را جدا می‌کنیم و اطلاعات فایل باز شده را نیز با استفاده از توابع جیسون می‌خوانیم. با استفاده از توابع رشته‌ای و جیسون پیام را با پیام‌های موجود در کانال مقایسه می‌کنیم بدین شکل که در صورت وجود این پیام در بخشی از یکی از پیام‌های موجود در کانال پیام کامل را به همراه فرستنده‌ی آن به کاربر به صورت جیسون ارسال می‌کنیم.