

مباحث ویژه در برنامه نویسی

علیرضا سلطانی نشان
دانشکده شمسی پور
مهندسی نرم افزار
asn80.asn@hotmail.com

۵ آذر ۱۳۹۹

فهرست مطالب

۲	۱ مقدمه
۲	۲ HTTP و Socket Web
۲	۳ توضیح مختصر در رابطه با پروژه
۳	۴ فراموش نشود



گزارش پروژه چت چند وجهی مبتنی بر پروتکل ساکت در وب

۱ مقدمه

تمامی موارد صورت گرفته در قسمت سمت سرور، به لطف وجود پلتفرم قدرتمند Node.js با استفاده از کتابخانه و فریمورک Express صورت گرفته است. در این پروژه علاوه بر پروتکل HTTP برای ارتباط بلادرنگ^۱ از پروتکل وب سوکت^۲ و از کتابخانه و فریمورک Socket.io مورد استفاده قرار گرفته است. لازم به ذکر است که تمامی موارد سمت کاربر از تکنولوژی خاصی استفاده نشده است، بلکه خیلی به صورت ساده از ساختار HTML استفاده شده.

۲ HTTP و Socket Web

در این پروژه برای داشتن ارتباط بلادرنگ در یک زمان و داشتن وضعیت پایدار، سعی شده که از پروتکل HTTP نمونه ای گرفته شده به سمت پروتکل وب سوکت انجام شده بطوری که بیشتر در مورد Namespace ها و Endpoint هایی که قرار است کاربران در آن به گفت و گو بپردازند، بحث شده. ارتباطات دیگر به صورت حالت HTTP که در آن در مورد درخواست ها Requests و پاسخ دهی Response نمی باشد. بلکه برای ارسال اطلاعات چه از سمت کاربر چه از سمت سرور با استفاده از فریمورک ساکت ای او، از دستور on برای گرفتن اطلاعات و از دستوری به نام emit برای ارسال اطلاعات استفاده شده است.

۳ توضیح مختصر در رابطه با پروژه

در این پروژه یکسری از فضاهایی وجود دارد که در آن کلاس ها و موضوع های مختلفی برای گفت و گو کردن وجود دارد، و کاربران به اختیار خودشان میتوانند وارد این فضاها شده و کلاس مربوط به خود را استفاده و در آن بحث کنند.

^۱ RealTime
^۲ Socket

در این پروژه سعی شده تا از فضا نام های ساده ای مانند. Wikipedia، Linux، Mozilla وجود دارد، که برای مثلا در قسمت موزیلا بحث هایی در مورد فضای اینترنتی و فضای مجازی شده، یا مثلا در قسمت لینوکس در مورد سیستم عامل Mac os Big Sur یا در مورد Redhat صحبت های مختلفی شده است.

لازم به ذکر است که تمام گفت و گو ها بین کاربران تا زمانی که سرور خاموش نشود (نمیرد) در یک آرایه خیلی ساده باقی خواهد ماند تا کاربران بتوانند از سوابق گفت و گو های خود یک پیش نمایش حدودی داشته باشند.

قسمت فضا ها و کلاس ها و ذخیره موقت پیام ها در آرایه کاملا به صورت شی گرا انجام شده.

بطوری که در فایلی ساده در مورد فضاهای نام یک نمونه از کلاس مربوط به آن گرفته شده و به وسیله این نمونه فضای نام مربوطه به آرایه مربوط به آن اضافه شده. کلاس ها هم به همین ترتیب صورت گرفته و در نهایت به عنوان آرایه ای داخل فضا ذخیره شده است.

در قسمت سرور در قسمت اصلی این اطلاعات مربوط به فضاها و کلاس ها سعی شده که یک رندر شود و در قسمت کاربر به وسیله ارتباط سوکت ارسال شود، و در قسمت کاربر این اطلاعات on شده و بعد از آن به لطف زبان JS تمام فرایندها در قسمت کاربر از صدا زدن تا ارسال و نمایش اطلاعات از سمت سرور به سمت کاربر صورت گرفته است.

نکته ای که باید توجه داشته باشیم آن است که پیام ها از نقطه ای به سمت سرور ارسال میشود و آن پیام از سمت سرور به سمت کاربر دوباره هدایت میشود تا کاربر پیام خود را در کلاس مربوطه ببیند، این فرایند برای تمامی کاربران قابل مشاهده و انجام می باشد.

نکته بعدی آن است که کاربری که از کلاس میخواهد به کلاس دیگری برود، ابتدا از کلاس مربوطه تابع خروج یا Left() فراخوانی میشود و برای رفتن به کلاس جدید تابع Join() برای ورود به آن گفت و گو استفاده میشود. دلیل این کار آن است که اگر کاربر از کلاسی لغت ندهد و وارد کلاس بعد جویین شود تمام پیام های او نه تنها برادکست خواهد شد بلکه تمام پیام هایی که در کلاس قبلی نوشته بوده در کلاس جدیدی که وارد شده اضافه می شود و به نوعی با تخریب و ریداندنسی رو به رو خواهیم شد.

نکته آخری که مربوط به بخش کاربر می باشد آن است که کاربر با کلیک بر روی هر کدام از Namespace ها در حقیقت با استفاده از تابع io.emit() متن آن فضایی که در آن قرار دارد به سمت سرور ارسال شده و سمت سرور به وسیله سوکت داخلی آن فضا را گرفته و کلا های مربوط به آن را ارسال خواهد کرد و در سمت کاربر این کلاس ها دریافت خواهد شد و با Loop Event مناسبی تمام کلاس ها برای کاربران به نمایش گذاشته می شود.

۴ فراموش نشود

در این پروژه هیچ گونه احراز هویتی خاصی با امنیت خاص ایجاد نشده، چرا که به دلیل فرآیندهای زیاد و زمان کمی که در اختیار داشته ام قسمت چت بین کاربران با دقت بالایی پیاده سازی شده و فقط قسمت احراز هویت آن صورت نگرفته، اما به لطف کتابخانه Passport.js

این احراز هویت میتواند صورت گیرد اما به دلیل فرایندهای جاری پروژه تا همین قسمت به صورت کامل انجام شده است.

منابع

You can click on my GitHub page[1] to see all source of this project as Clone Slack base on Javascript with [2] Socket.io and [3] Express framework .