Node-red front document (persian)

(node-red/packages/node\_modules/@node-red/editor-client/src/js/comms.js):

## در ابتدا کل این فایل داخل RED.comms که خود یک فانکشن را صدا میزند میباشد.

## در داخل آن، ابتدا متغیر هایی که قراره استفاده بشه، تعریف شده اند و سپس سه فانکشن کلی وجود دارد که در آخر هر سا تایه آنها return میشوند.

## در فانکشن های subscribe و unsubscribe همانطور که از اسمش پیداست بحث چک کردن میباشد؛ در کل اگر قرار باشد فانکشن subscribe اعمال شود، به منظور ادامه فرایند از قسمتی که مانده است میباشد، و به عکس.

## کار اصلی در فانکشن connectWS اتفاق میوفتد که باز کردن یک سوکت و اعمال تنظیمات آن پورت وظیفه این فانکشن میباشد؛ نام این پورت ws است.

## در حالت کلی وظیفه این فانکشن چک کردن توکن میباشد؛ ابتدا wspath را به عنوان پچ باز کننده سوکت استفاده میکند و سوکت باز شده را در متغیر ws ذخیره میکند.

## این سوکت سه حالت onopen و onmessage و onclose دارد:

## در onopen ارور ها پاک شده و توکن بصورت فایل جیسون در سوکت ارسال میشود، که دارا بودن توکن شرط آن است و دائما در حال چک آن است.

## در onmessage دیتایی در قالب متغیر فانکشنی استخراج میشود که دو مقدار ok و fail آن بررسی میشود؛ اگر که آن مقدار برابر ok باشد، متغیری که مربوط به چک کردن توکن بود با مقدار false مقدار دهی شده و این فاکشن کلی دوباره اجرا میشود؛ در غیر اینصورت یعنی اگر مقدار مورد نظر برابر fail باشد، مقدار جیسونی updateMenu با مقدار true فرستاده میشود. اگر متغیر ورودی فانکشن مقدار دهی نشود، قانون regex جدیدی روی ۀن اعمال میشود تا خطای مربوط به آن را اصلاح کند.

## در onclose ابتدا چک میشود که مقداری که در فانکشن قبلی به عنوان ok با fail بود و در متغیر جدیدی با مقادیر بولین هست چه وضغیتی دارد؛ اگر ok بود مقدار false ست شده و به عکس؛ این کار برای چک کردن صحت و وجودیت توکن میباشد. اگر توکن موجود نباشد، ابتدا ده بار هر کدام به مدت یک ثانیه و سپس ده بار هر کدام به مدت دو ثانیه و در کل به مدت 30 ثانیه دوباره توکن را چک میکند؛ اگر مشکل توکن حل نشد، به مدت 60 ثانیه اروری را نمایش میدهد که خطای دریافتی را نمایش میدهد؛ مگر اینکه روی دکمه اینجاد شده برای بستن ارور کلیک شود.

## (node-red/packages/node\_modules/@node-red/editor-client/src/js/events.js):

## در ابتدا کل این فایل داخل RED.events که خود یک فانکشن را صدا میزند میباشد.

## این فایل شامل سه فانکشن کلی on و of و emit میباشد:

## در فانکشن on متغیر آرایه ای مقدار دهی میشود.

## در فانکشن off متغیر های مقدار دهی شده، پاک میشوند.

## فانکشن emit که RED.events.DEBUG استفاده شده است، بنظر میرسد برای debug بکار میرود و در واقع مدیریت ارور های کنسول برای راهنمایی debug میباشد.

## (node-red/packages/node\_modules/@node-red/editor-client/src/js/font-awesome.js):

## در این فایل که RED.nodes.fontAwesome بکار رفته است، مدیریت فونت ها اتفاق میوفتد؛ یعنی فونت ها در این فایل قرار دارند؛ این فایل دارای دو لیست کلی برای متغیرها میباشد که هر کدام در آخر return میشوند. در return دو مقدار به قالب جیسون return میشوند که یکی مقادیر متغیر های موجود در لیست ها ریتورن میکند (valus) و دیگری نام متغیر ها را (keys)؛ فانکشنی که نام متغیر هارا ریتورن میکند، فقط لیست اول را در نظر میگیرد اما فانکشنی که مقادیر را ریتورن میکند، هر دو لیست را در نظر میگیرد.

## node-red/packages/node\_modules/@node-red/editor-client/src/js/history.js:

## این فایل دستورات کلی undo و redu را کنترل میکند؛ شامل دو فانکشن کلی است و ریتورن آن به شکل داده های جیسونی میباشد. در این فایل RED.history استفاده شده است که همه اتفاقات درون آن اتفاق می افتد. همچنین در داخل فانکشن های آن از RED.nodes وRED.group و RED.workspaces و RED.events و RED.editor و RED.view و RED.sidebar.config و RED.subflow و RED.menu استفاده شده است.

## node-red/packages/node\_modules/@node-red/editor-client/src/js/hooks.js: