

cronapi.net.WebSocket.connect

Conectar-se a Websocket

Conecta em um servidor *WebSocket*.

Retorno

Retorna um objeto com a referência da conexão.

Parâmetros de Entrada

| Nome | Tipo | Doc |
|-------------|--------------------|---|
| Parâmetro 1 | Texto | Endereço do servidor <i>WebSocket</i> . A URL deve iniciar sempre com "wss://". |
| Parâmetro 2 | Entrada de comando | Executa ações após estabelecer uma conexão com o servidor. A variável "item" dessa entrada de comando retorna algum dado enviado pelo servidor após o sucesso da conexão. |
| Parâmetro 3 | Entrada de comando | Executa ações após receber as mensagens do servidor. A variável "item" dessa entrada de comando exibe as mensagens enviadas pelo servidor. |
| Parâmetro 4 | Entrada de comando | Executa ações quando ocorrer um erro. A variável "item" dessa entrada de comando exibe a mensagem de erro enviado pelo servidor. |

Saiba mais

O [Web Socket](#) é um protocolo de conexão que cria uma conexão contínua entre um servidor *Web Socket* e um cliente, dessa forma, a comunicação se torna mais rápida, visto que não é necessário iniciar e fechar uma conexão sempre que uma mensagem for enviada para o servidor. A tecnologia *WebSocket* é muito utilizado em aplicativos de conversas (chat) e jogos online.

Compatibilidade

 Servidor

Exemplo

No exemplo abaixo estamos utilizando o serviço do [PieSocket](#) que possui um servidor *WebSocket* de testes e uma ferramenta de envio de mensagem que dispara a mensagem enviada para todos os clientes conectados ao servidor.

A função da figura 1 conecta-se a um servidor e gera um objeto com a referência da conexão na variável "ws", o conteúdo dessa variável será usado nos blocos [Enviar Mensagem por Websocket](#) e [Fechar Websocket](#).

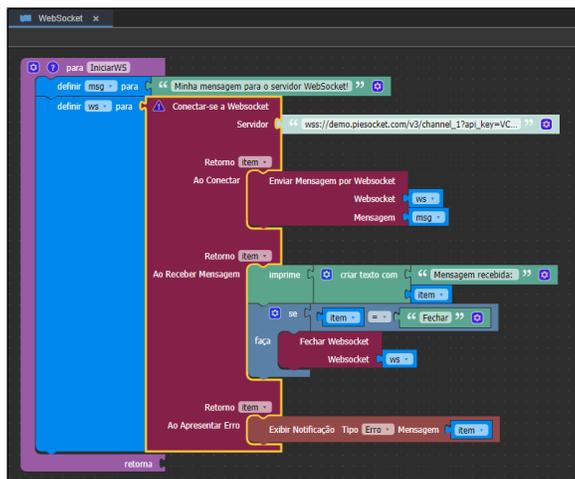


Figura 1 - Bloco que inicia a conexão e aguarda as mensagens do servidor

Feita a conexão, o bloco **Conectar-se a WebSocket** fica aguardando a resposta do servidor para executar as ações definidas nas entradas de comando do bloco:

- **Servidor:** endereço do servidor, neste exemplo estamos utilizando o [PieSocket](#).
- **Ao Conectar:** após o sucesso da conexão, o bloco [Enviar Mensagem por WebSocket](#) é chamado para enviar para o servidor o texto contido na variável "msg".
- **Ao Receber Mensagem:** toda mensagem recebida do servidor executará essa entrada de comando. Neste exemplo estamos imprimindo no console todas as mensagens recebidas até que o conteúdo da mensagem seja "Fechar", aí o bloco [Fechar WebSocket](#) será chamado e fechará a conexão com o servidor.
- **Ao Apresentar Erro:** se um erro ocorrer, o usuário será notificado do erro.

A ferramenta do [PieSocket](#) dispara a mensagem enviada para todos os clientes conectados ao servidor.

Após a conexão, a mensagem contida na variável "msg" é disparada e exibida na ferramenta do *PieSocket* (Figura 1.1). Em seguida, digitamos 3 mensagens através do chat do *PieSocket* (Figura 1.1): "Mensagem do Servidor para o cliente.", "Fechar" e "Olá".

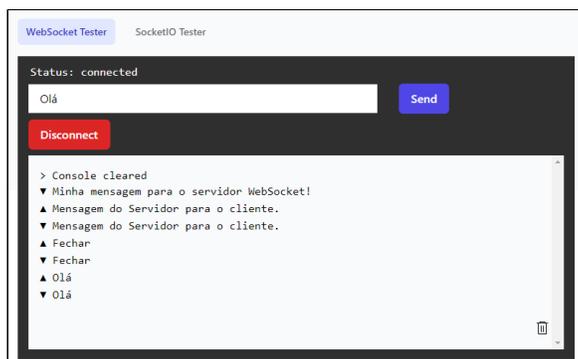


Figura 1.1 - Ferramenta de envio de mensagem do PieSocket

No console (Figura 1.2) temos 3 mensagens. A primeira é o retorno da mensagem enviado na entrada de comando **Ao Conectar** (Figura 1), o *PieSocket* sempre retorna a mensagem enviada, e as outras 2 foram digitadas usando a ferramenta do *PieSocket* (Figura 1.1). Perceba que após receber a mensagem "Fechar", o bloco [Fechar WebSocket](#) foi chamado e encerrou a conexão.

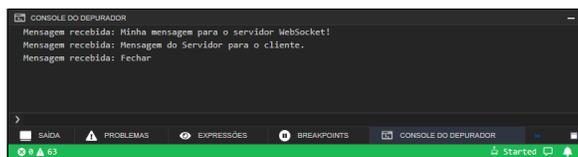


Figura 1.2 - Mensagens recebidas do servidor PieSocket