**Documentação Helper ChatBot**

**Arquivo generalFunctions.js**

**Função**

|  |
| --- |
| **searchValuesInMsg**(userMsg, objeto, keySearchEntity, onlyIfKey, onlyIfKeyValue, returnOnlyFirst) |

🡪 Finalidade: Verificar se na mensagem do usuário está contida alguma interação (saudação, cumprimento, despedida, agradecimento, hashtag ou palavra de busca).

Navega em uma determinada entidade e verifica se algum de seus valores está contido na mensagem do usuário.

Mais utilizado nas buscas de comportamento (cumprimento, agradecimento e despedida).

🡪 Parâmetros:

userMsg: Obrigatório. Mensagem do usuário.

objeto: Obrigatório. A entidade onde será buscada.

keySearchEntity: Obrigatório. Propriedade na entidade onde será comparada a mensagem do usuário.

onlyIfKey: Optativo. Indica que retornará um objeto apenas se a condição onlyIfKey for satisfeita.

onlyIfKeyValue: Optativo. Obrigatório se utilizado onlyIfKey. Valor de comparação para o parâmetro onlyIfKey.

returnOnlyFirst: Optativo. Retorna apenas o primeiro objeto, se for encontrado.

🡪 Retorno:

Sempre retorna um array com 2 elementos:

[0]: Valor booleano, indicando se encontrou resultado.

[1]: Array contendo todos os objetos (itens) encontrados.

Caso a condição não seja satisfeita, retorna o primeiro elemento como **false** e o segundo elemento um array vazio.

🡪 Exemplo:

isValidRequest = searchValuesInMsg(msgReq, botBehaviour,"userRequest", "nameBehaviour", "cumprimento", true);

Procura na entidade botBehaviour se algum elemento da propriedade userRequest pode ser encontrada na mensagem do usuário (msgReq). Porém, procura apenas nos itens de botBehaviour onde a propriedade nameBehaviour for igual a “cumprimento”. Retorna um array onde o primeiro índice será **true** ou **false** e, sendo **true**, apenas o primeiro item encontrado (parâmetro returnOnlyFirst passado como true).

**Função**

|  |
| --- |
| searchContentFromMsg(userMsg, objeto) |

🡪 Finalidade: Buscar em um determinado objeto possíveis valores contidos na mensagem digitada pelo usuário. A busca é realizada por níveis, utilizando segmentos de palavras agrupadas, contidas na mensagem do usuário. As variáveis internas **iMaxGrouper** e **iMinGrouper** define a quantidade de níveis de busca. Por padrão, a busca é feita por agrupamento de 4 palavras, decrescentemente, até segmento com 2 palavras (busca em 3 níveis).

🡪 Funcionamento: Para cada elemento do objeto, a função busca nas propriedades **Title**, **Description** e **Body**.

No primeiro nível, a mensagem do usuário é agrupada em segmentos de 4 palavras (**iMaxGrouper**), sequencialmente, e cada segmento é buscado nas propriedades **Title**, **Description** e **Body**. Se o segmento for localizado dentro de uma propriedade, é incrementado o valor de **score** para este item (variável **itemScore**).

No segundo nível de busca, repete-se o procedimento anterior, porém com segmento de 3 palavras a partir da mensagem do usuário. No terceiro nível, repete-se, porém com segmento de 2 palavras. A quantidade de níveis de busca pode ser parametrizada pelas variáveis internas **iMaxGrouper** e **iMinGrouper**.

O cálculo do score se dá pela multiplicação entre o valor do nível de busca (por padrão 4, 3 ou 2) pelo valor de *match* atribuído na constante **MATCHVALUE**. Assim, quanto maior for o nível do encontro (portanto a maior precisão), maior será o score atribuído ao item.

Ao terminar todos os níveis de busca, caso o item tenha **score** maior do que zero, é adicionado um array como buffer de retorno, na variável **arrayResult**, contendo por sua vez dois elementos: o primeiro é o elemento do objeto com todas as suas propriedades, e o segundo é o score, em formato string convertido pela função formatarItemScore, para ordenação crescente.

Assim, **arrayResult** é uma matriz, contendo arrays de dois elementos, onde o primeiro contém um objeto e o segundo o score correspondente.

🡪 Retorno: Sempre retorna **arrayResult**, porém não com o score em valor absoluto, e sim relativo em relação ao item de maior score calculado (regra da significância). A ordenação é crescente (menor valor para o maior valor, ou seja, de menor significância para maior significância).

Será retornado apenas elementos que tenha grau de significância (score proporcional) igual ou maior a 50%.

Assim, a função retorna uma matriz, onde cada elemento conterá dois outros elementos, sendo o primeiro o **score proporcional** ao item de maior pontuação e o segundo o item do objeto analisado. O promeiro elemento, portanto, terá o **score proporcional** sempre igual a 1, e será considerada a resposta mais significante.

**resultArray** = [itemFound[], itemFound[], itemFound[],...]

itemFound[0] = *Valor decimal*. Score proporcional do item, em relação à maior pontuação. Ordenação crescente.

itemFound[1] = *Objeto*. Item que teve pontuação, com todas suas propriedades.

Caso não seja encontrado nenhum elemento com pontuação, retornará um **resultArray** vazio (tamanho 0).

🡪 Parâmetros:

userMsg: Obrigatório. Mensagem do usuário.

objeto: Obrigatório. A entidade onde serão buscados valores.