Conforme aprendizado sobre os fluxos, novos conhecimentos são adquiridos:

**NOME WHATS**

visitor.firstName

**BRANCH CHAMADA SUCESSO?**

httpStatus == 200

**EXTRAIR HISTÓRICO DA CONVERSA ATÉ X MOMENTO**

(o histórico irá corresponder ao momento que for setado o valor da expressão do Java)

#stateFunction.get(session['sessionId'])

**TAMANHO DO ARRAY**

response.size()

**PRINTAR DE MAPA(LinkedHashMap) PARA STRING**

response.toString()

**COLOCAR PONTO NO MEIO DE UMA STRING**

"valor\_entrada": "payload",

"valor1": "valor\_entrada.substring(0,3)",

"valor2": "valor\_entrada.substring(3,6)",

"valorFinal": "valor1.concat('.').concat(valor2)"

**TRANSFORMAR VALOR FLOAT EM MOEDA**

Ex:

API retorna um valor parcela1 = 2802.2367052945056

"varFloat": "new java.lang.Float(parcela1)",

"decimalInst": "new java.text.DecimalFormat('R$ #,###.00', new java.text.DecimalFormatSymbols(new java.util.Locale('pt','BR'))).format(varFloat)"

decimalInst = R$ 2.802,24

**IF TERNÁRIO SE PRECISAR INSERIR NA MENSAGEM DO USUÁRIO**

(payload == '1' ? 'Agendamento' : (payload == '2' ? 'Compras' : (payload == '3' ? 'Financeiro' : (payload == '4' ? 'Comercial' : (payload == '5' ? 'SAC' : (payload == '6' ? 'Unidade Belém' : (payload == '7' ? 'Unidade São Paulo' :  payload)))))))

**ARRAY VAZIO**

response.isEmpty()

**TAMANHO INTEGRAÇÃO WEB CHAT MENSAGEM E BOTÃO**

.message-thread .button{

  font-size: 30px !important;

}

.message-thread .message{

  font-size: 30px !important;

}

**PERMITIR X QUANTIDADES DE CARACTERES APENAS**

variavel matches '^[a-z]{2}$'

variavel matches '^[a-z]{2}|[A-Z]{2}$'

**OPERAÇÃO MATEMÁTICA A PARTIR DE STRING (TRANSFORMAR EM NÚMERO)**

"variables": {

        "calculoCamiseta": "function.extractInteger(valorUnitario) \* function.extractInteger(qtCamisetas)"

**VALIDAÇÃO DE IMAGEM COM SUAS EXTENSÕES**

(fileUrl != null AND fileMimeType == 'image/jpeg') OR (fileUrl != null AND fileMimeType == 'application/pdf') OR (fileUrl != null AND fileMimeType == 'application/jpg') OR (fileUrl != null AND fileMimeType == 'application/png')

**NOME ATENDENTE NO ROCKET - FAZER ASSOCIAÇÃO PARA APARECER NO RELATÓRIO**

livechat.agent.name

Criar uma variável que seja populada com essa expressão

**CONTEUDO DO ARRAY QUE NÃO É O ÚNICO OBJETO, É VAZIO**

response.tickets[ 0 ] != null

Exemplo:

{

    "response": {

        "status": "200 OK",

        "code": 200

    },

    "tickets": []

}

**IF TERNÁRIO**

"variables": {

        "VARIÁVEL": "out01 == 'Alergologia' ? 'CALERI' : out01 == 'Angiologia' ? 'CANGI' : out01 == 'Cardiologia' ? 'CCARDI' : out01 == 'ClinicoGeral' ? 'CCLINI' : '-'" }

**VALIDAR SE USUARIO MANDOU IMAGEM NO FLUXO**

(fileUrl != null AND fileMimeType == 'image/jpeg') OR (fileUrl != null AND fileMimeType == 'application/pdf')

**VALIDA FORMATO HORA**

VARIAVEL matches '([01]?[0-9]|2[0-3]):[0-5][0-9]'

**EXTRAI DO ARRAY O PRIMEIRO ITEM QUE ELE ACHAR COM ESSA CONDIÇÃO (^)**

#{response.^[UnidadeId == #root.Unidade].Endereco.Nome}

Unidadeid = id do botão que o usuário clicou

Unidade = variável que recebe o id

Endereco.Nome = retorno do array

**EXEMPLOS VALIDAÇÃO CONTEÚDO: Expression Language**

#string.containsAny(recebeSMS,'nao','Nao','não','Não')

#text.keyword(continuarFaq, 'Não') != null or continuarFaq

#string.containsAny(respPublicaLivro,'sim','Sim')

textosms matches 'pragmatico'

#text.keyword(resp\_modelo,'StreetBob','FatBob','SoftailSlim') != null

**VALIDA E-MAIL**

session['emailcliente'] matches '(?i)^[A-Z0-9.\_%+-]+@[A-Z0-9.-]+\.[A-Z]{2,6}$'

**TRIES / TENTATIVAS**

"variables": {

        "tries": "0"

      },

Valida:

"variables": {

        "tries": "session['tries']+1"

      }

Branch:

session['tries'] < 3

**VALIDA CPF (somente números)**

payload matches '^\d{11}$'

**VALIDA DATA DE NASCIMENTO**

#time.parseDate(resp\_data\_nasc, 'dd/MM/yyyy') != null

**VALIDA NOME (sem acentuação)**

resp\_nome\_mae matches ("[A-Za-záàâãéèêíïóôõöúçñÁÀÂÃÉÈÍÏÓÔÕÖÚÇÑ\s]+")

**VALIDAÇÃO CEP (somente números)**

payload matches '^\d{8}$'

**VALIDA SE USUÁRIO DIGITOU SÓ NÚMEROS:**

payload matches '^\d{1,9}$'

**VALIDAÇÃO DE TELEFONE (só números por extenso)**

#phone.find(session['resp\_telefone']) != null

**VALIDA ENDEREÇO SITE:**

session['site'] matches '(?i)^[A-Z0-9.\_%+/-]+\.[A-Z]{2,6}$'

**VALIDA SE CAMPO TEM X DÍGITOS (NO EXEMPLO, 4):**

session['agenciauol'] matches '^\d{4}$'

**IGUALA UMA STRING À OUTRA (RESULTADO BOOLEANO)**

confirmacpf.substring(0,3).equals(session['website']['extra']['cpf'].substring(0,3))

**COMPARAR DUAS STRINGS AO MESMO TEMPO**

Resp4\_ValorResidencial matches '\d\*' && T(Float).parseFloat(Resp4\_ValorResidencial) > 60000.0

**TRANSBORDO DEPARTAMENTO (Valida Consulta Atendente) - criar antes no bot a variável departamento**

rawCapabilities.departments.contains(departamento)

**POTENCIAÇÃO**

3.power(expoente)

**CORTAR STRING ATÉ UM CARACTERE SEPARADOR:**

#{frase.substring(0,frase.indexOf("."))}

**ISOLAR NÚMERO SEM FORMATAÇÃO**

#{dividas.replaceAll('[\\D]\*([0-9.]+).\*', '$1').replaceAll('\\.', '')}

**INCLUIR DASHBOT (final do código, abaixo do conversational channel**

  "metadata": {

    "skills": [],

    "externalReports": [

      {

        "name": "Dashbot",

        "authentication": {

          "dashbotApiKey": "fMN053wXzUH1YtmF4oI0C401BpcVz0m9wm7jnjFe"

        }

      }

    ]

  }

**TESTA CADA CARACTER DA STRING PARA VER SE ELA É COMPOSTA SÓ DE ESPAÇOS EM BRANCO:**

#string.isBlank(variável)

**TRAZER HORA ATUAL**

"horaAtual": "#time.format((#time.now()).minusHours(3), 'HH:mm')"

**>** Expressão regular **horário "bom dia"**:

T(java.time.LocalDateTime).now(T(java.time.ZoneId).of('America/Sao\_Paulo')).getHour() >= 1 AND T(java.time.LocalDateTime).now(T(java.time.ZoneId).of('America/Sao\_Paulo')).getHour() <= 11

**CARROSSEL DINÂMICO**

{

      "id": "step05",

      "type": "sendChatActivity",

      "label": "Nossas Lojas",

**"properties": {**

**"from": "#{session['chatFrom']}"**

**},**

**"contents": [**

**{**

**"type": "application/vnd.zenvia.v1.carousel+json",**

**"payload": {**

**"json": {**

**"cards": [**

**{**

**"text": "#{loja.endereco}\n#{loja.telefone}\n#{loja.texto}",**

**"label": "loja.nome",**

**"buttons": [**

**{**

**"type": "postback",**

**"text": "Quero esta Loja",**

**"payload": "loja.nome"**

**}**

**]**

**}**

**]**

**}**

**},**

**"dynamicItem": {**

**"value": "response.lojas",**

**"var": "loja",**

**"ref": "cards"**

          }

        }

      ]

    }

**TRANSFORMAR VARIÁVEL**

#text.keyword(VARIÁVEL,'1') != null

**Referência e transformação do valor do payload:** tu descreve após setar a variável qual o real valor dela ou o que ela vai conter como "payload"

     "qtClientes": "payload == '1' or payload == '1000' or payload == 'Até 1000' ? 'Até 1.000' : 'Acima de 1.000'"

qtClientes (variável do exemplo)

? (pergunta)

: (se não)

**ENVIO FLUXO VIA API REST**

Autenticação: Bearer Token (Preferências na plataforma) + Content Type

{

  "workflowId": "009e530b-09a6-4a65-8536-7a4819ed2ae0",

  "fileHeader": "name;sms",

  "scheduledStart": "2017-05-29T09:10:00.000-05:00",

  "scheduledEnd": "2017-05-29T018:00:00.000-05:00",

  "scheduledLimit": "2017-05-29T20:00:00.000-05:00",

  "campaignId": "Teste",

  "contacts": [

    "Guilherme;5551991083778"

  ]

}

**MODELO HÍBRIDO COM NLP E SAÍDA PARA SOLUÇÃO ESPECÍFICA (SEBRAE/PR)**

variável matches 'exemploDeSaida'

**ADICIONAR DIRETAMENTE O NÚMERO DE WHATSAPP**

<https://wa.me/5551999999999>

**TER SUCESSO OU NÃO NA BASE DE CONHECIMENTO NLP (Qna)**

score >= 5

**DASHBOT NO JSON do BOT**

  "metadata": {

   "skills": [],

   "externalReports": [

     {

       "name": "Dashbot",

       "authentication": {

         "dashbotApiKey": "XXXXXXXXXXXXXXXXXXXXXXX"

       }

     }

   ]

 }

**SESSÃO API DE FLUXO WHATSAPP**

Entre aspas simples!

whatsappFrom - '63a6b746-b8ef-48b0-84da-a41460b06533' Ex:

      "variables": {

        "whatsappFrom": "'63a6b746-b8ef-48b0-84da-a41460b06533'"

      }

*Exemplo:*

*Tenho 4 produtos, cada um tem um payload onde todos eu posso comparar.*

A variável é "decisaoProduto". Logo, preciso fazer o match de um dos "quero" para essa variável.

Então, toda a vez que tiver o payload X dentro de decisaoProduto e ele for diferente de "null", OK!

#text.keyword(**decisaoProduto**,'queroPrepago','queroGift','queroMobox','queroPremiacao') != null

**BOTÃO URL PARA VOLTAR**

Insere a URL normal.

Vai no JSON

Altera de:

"type": "web\_url",

"text": "Saldo e Extrato",

"link": "[http://www.consultadesaldo.com.br](http://www.consultadesaldo.com.br/)"

Para:

**"type": "postback",**

"text": "Voltar",

**"payload": "voltar"**

**E não abre mais o STEP. Configura apenas no Json.**

**TAG HTML HIPERLINK**

<a href="[https://html.com/attributes/a-href/](https://l.messenger.com/l.php?u=https%3A%2F%2Fhtml.com%2Fattributes%2Fa-href%2F&h=AT3V228kBWoypGY0baoeyfX1V1RVPZhtnWhzaBIpfANsBqUjepJEed9p8CM2LM1JzFnOwrRz79KjYzBvXLOP39rm3E_NSTfQOrYdaHIYQOBZ_X74k5liTwH7f7DquoTWr_wZBw)">Learn about the a href attribute</a>

**QUIZ STRATEGY**

Escolhas...

respostaObjetivo matches 'objetivoMeta' **AND** quantidadePessoas matches 'quantidade1' **AND** tipoPremiacao matches 'pontual'

**PRINTAR INFORMAÇÕES COM STRING NO MEIO. A STRING (TEXTO) ESTÁ ENTRE ASPAS SIMPLES**

"respostaPreocupacao": "'Consultoria Seguro de Vida - ' + respostaInsatisfacaoPlanoAtual"

**VALIDADOR 3 DIGITOS DO CPF CONFORME ARQUIVO DE DISPARO ENVIADO**

confirmaCpf== attribute['cpf'].substring(0,3)

**variável**== attribute['**coluna do arquivo**'].substring(0,3)

**ARRAY**

Exemplo de array:

{

    "status": true,

    "data": [

        {

            "id": "2019-05-31",

            "value": "31/05/2019"

        },

        {

            "id": "2019-06-03",

            "value": "03/06/2019"

        },

        {

            "id": "2019-06-04",

            "value": "04/06/2019"

        },

data é o valor principal e dentro dela tem as posições de 0 a 2.

Para captar a informação:

f

**response.data[0].value**

No caso, vou printar o valor da posiçao zero.

**CORTAR UMA STRING A PARTIR DE UM DETERMINADO CARACTERE**

#{VARIALVEL.substring(VARIAVEL.indexOf(".")+1)}

**FILTRO KIBANA**

Filter

appname

is

external-api

**QUANTIDADE DE NÚMEROS**

placaVeiculo matches '^\d{7}$'

No caso, vai aceitar se placa tiver 7 caracteres.

**EXPRESSÃO QUE CONTABILIZA A QUANTIDADE DE CARACTERES:**

variável.length() == 7

OUTRA BIBLIOTECA <https://sites.google.com/a/zenvia.com/workflow/api/expression-language/library>

|  |  |  |  |
| --- | --- | --- | --- |
| **Facebook** | **Webchat** | **Whatsapp** | **SMS** |
| facebookMessengerEvent | chatEvent | whatsappEvent | moEvent |
| sendFacebookMessengerActivity | sendChatActivity | sendWhatsAppActivity | sendSmsActivity |
| facebookFrom | chatFrom | whatsappFrom | - |
| facebookTo | chatTo | whatsappTo |  |
| "defaultConversationalChannel": "FACEBOOK" | "defaultConversationalChannel": "CHAT" | "defaultConversationalChannel": "WHATSAPP" | "defaultConversationalChannel": "SMS" |