General applications

if

if .

```
    {
        "if": {
            "expression": "${myVar} == '1'",
            "then": [],
            "else": []
        }
    }
```

switch

switch case .

```
  {
    "switch": {
      "variable": "${getIvrDigit}",
      "case": {
        "1": [],
        "2": [],
        "3": [],
        "default": []
      }
    }
  }
```

while

, . [do] . .

```
  {
    "while": {
        "condition": "${myVar} < 10",
        "maxSteps": "1000",
        "do": [
            {
                "log": "log is ${myVar}"
            },
            {
                "js": {
                    "data": "+${myVar} + 1",
                    "setVar": "myVar"
                }
            }
        ]
    }
}
```

condition - , js;

maxSteps - (- goto) 1000;

do – , .

disconnected

. .

```
{
   "trigger": {
       "disconnected": []
    }
}
```

commands

Commands recvMessage , , recvMessage .

```
{
        "trigger": {
            "commands": {
                "/queue": [
                    {
                        "sendText": "queue"
                    }
                ],
                "/show map": [
                    {
                        "sendText": "map"
                    }
                ],
                "press button 1": [
                    {
                        "sendText": "button"
                    }
                ]
            },
            "disconnected": [
                {
                    "dump": true
                }
            ]
        }
    },
    {
        "sendText": "start"
    },
    {
        "recvMessage": {
            "set": "",
            "timeout": 20
        }
    },
```

function

. .

```
{
    "function": {
        "name": "myFunction",
        "actions": []
    }
}
```

execute

.

```
{
     "execute": {
         "name": "myFunction"
     }
}
```

set

.

```
{
    "set": {
        "hangup_after_bridge": "true",
        "ivrLang": "en"
    }
}
```

export

, .

```
    {
        "export": [
            "reverted_caller_id_number"
        ]
    },
```

unSet

.

```
{
     "unSet": [
        "reverted_caller_id_number"
     ]
}
```

calendar

. .

```
{
    "calendar": {
        "extended": false,
        "name": "WorkDay",
        "setVar": "isWorkDay"
   }
}
```

cache

Redis .

```
[
    {
        "cache": {
            "action": "set",
            "set": {
                "data": {
                    "key": "value"
                },
                "ttl": "10000"
            },
            "type": "memory"
        }
    },
    {
        "cache": {
            "action": "get",
            "get": {
                "myVar": "key"
            },
            "type": "redis"
        }
    },
    {
        "cache": {
            "action": "delete",
            "delete": {
                "keys": [
                    "key",
                    "key1"
                ]
            },
            "type": "redis"
        }
    }
]
```

httpRequest

HTTP REST (get JSON)

```
{
    "httpRequest": {
            "url": "https://sales.bpmonline.com/${id}/dataservice/json
/reply/SelectQuery",
            "method": "POST",
            "timeout": 1000,
                        "insecureSkipVerify": false,
            "parser": "application/json",
            "cacheCookie": false,
            "responseCode": "http_response_code",
            "headers": {
                "Content-Type":"application/json",
                "Cookie": "${my_cookie}"
            },
            "path": {
                "id": 0
            },
            "data": {
                "Name": "Supervisor",
                "UserID": "Supervisor"
            },
            "exportVariables": {
                "effective_caller_id_name": "callerIdName",
                "owner_caller_id_number": "callerIdOwner"
            }
    }
}
```

js

JavaScript

```
{
   "js": {
     "data": "var time = LocalDate(); time.setDate(time.getDate() +
(+${dpd}*-1)); return time.getMonth() + '-' + time.getDate() + '-' + time.
getFullYear()",
     "setVar": "myVar"
   }
}
```

math

```
{
    "math": {
      "data": "${caller_id_array}",
      "setVar": "new_random_caller_id",
      "fn": "random"
    }
}
```

fn

- `random`: returns a random number from array
- `min` and `max`: can be used to find the lowest or highest value in a list of arguments
- `round`: rounds a number to the nearest integer
- `ceil`: rounds a number up to the nearest integer
- `floor`: rounds a number down to the nearest integer

JavaScript Math

string

.

```
{
    "string": {
       "data": "${caller_id_number}",
       "fn": "replace",
       "setVar": "reverted_caller_id_number",
                  "args": [
            "/^0/",
            "+84"
        ],
     }
}
```

fn

- `length`: returns the length of a string
- `indexOf` and `lastIndexOf`: returns the index of (the position of) the first or last occurrence of a specified text in a string
- `search`: searches a string for a specified value and returns the position of the match
- `slice`: extracts a part of a string and returns the extracted part in a new string
- `substring`: is similar to slice. The difference is that `substring` cannot accept negative indexes.
- `substr`: is similar to slice. The difference is that the second parameter specifies the length of the extracted part.
- `replace`: replaces a specified value with another value in a string
- `toUpperCase` or `toLowerCase`: A string is converted to upper case or to lower case
- `charAt`: returns the character at a specified index (position) in a string
- `charCodeAt`: returns the unicode of the character at a specified index in a string
- `split`: A string can be converted to an array with the `split` function
- `reverse`: Reverse the provided string
- `base64`: base64 encoder / decoder
- `MD5`: Computes a digest from a string using `MD5` algorithm
- `SHA-256`: Computes a digest from a string using `SHA-256` algorithm
- `SHA-512`: Computes a digest from a string using `SHA-512` algorithm

JavaScript String

markIVR

,

```
{
    "markIVR": {
        "name": "Log",
        "value": " "
    }
}
```

userInfo

```
{
    "userInfo": {
        "set": {
            "usrId": "id",
            "email": "email",
            "mob": "variables.fwd_mob",
            "out": "variables.out_cid"
        },
        "user": {
            "extension": "${ext}"
        }
    }
}
```

setGrantee

id

```
{
   "setGrantee": {
       "id": 10
    }
}
```

schema

```
{
    "schema": {
        "id": 1
    }
}
```

softSleep

```
{
    "softSleep": 1000
}
```

sql

Webitel,

```
[
    {
        "sql": {
            "dns": "admin:123Webitel4@tcp(database-1.cluster-cnvuzqnsxvyv.
eu-central-1.rds.amazonaws.com:3306)/mysql",
            "driver": "mysql",
            "params": [
                "db"
            ],
            "query": "select 1 as tst_number, true as tst_bool,
table_name, last_update from mysql.innodb_index_stats where table_name = ?
order by rand() limit 1"
        }
    },
    {
        "sql": {
            "dns": "user=webitel password=webitel host=10.9.8.111
dbname=webitel sslmode=disable",
            "driver": "postgres",
            "params": [
                12
            ],
            "query": "select user_id from call_center.cc_agent where id =
$1",
            "timeout": 5000
        }
    }
]
```

list

```
{
    "list": {
        "name": "StopList_Chat_bot",
        "destination": "${user}",
        "actions": [
            {
               "sendText": "BAN: ${from} ",
                "break": true
            }
        ]
    }
}
```

listAdd

```
{
     "listAdd": {
         "destination": "${caller_id_number}",
         "description": "${caller_id_name}",
                 "expireAt": "${expired_at}",
         "list": {
              "id": 16
         }
     }
}
```

log

```
{
     "log": " >> ${Hold} <<<"
}
```

goto

```
{
    "goto": "tagName"
}
```

monoPay

```
{
   "monoPay":{
      "invoice":{
         "action":"create",
         "body":{
            "amount":"1",
            "paymentType":"debit",
            "setVar":"monoInvoice",
            "validity":"3600"
         }
      },
      "token":"uLRpgdn9yE5JP11NuXVqdvWpKk6TbSiQnquzS2BxtiCw"
}
```

, .

tag

```
{
    "tag": "tagName"
}
```

async

```
 {
     "async": true
 }
```

break

```
 {
     "break": true
 }
```

limit

max. , failover.

```
 {
    "limit": {
     "max": 5,
     "failover": "anotherTag"
  }
}
```