Moonbase Trigger & Action JSON Formats

Structure of a Sequence

A sequence is made up of one or more triggers and one or more actions

Creating a Basic Sequence

```
{
    "name": "Demo Remote Lock",     //The name of your moonbase sequence. This is used to give a nice name to external users
    "triggers": [                   //What is going to be used to trigger the action(s). You can have more than one trigger to a sequence,
    {                               //  but all need to evaluate as true.
        "condition_type": "time",   //Time triggers use cron expressions to trigger at a certain time.

        "cron": "0 9 * * *"         //Cron expression that is used for trigging the actions more info here https://en.wikipedia.org/wiki/Cron
      }
    ],
    "actions": [                    //What tasks to execute when the triggers are fired and they all pass validation.
      {
        "action_type": "remote_op", //Remote Operation that is sent to the vehicle.
        "pin": "0000",              //sdp pin required to execute the remote command.
        "sdp": "IGNITE",            //SDP
        "uid": "xxx",               //The uid for the sdp
        "name": "Remote Lock",      //Name of the action. Must be unique to each actions type. You can have a SMS action with the same name as
                                    //  a remote_op, but not the same name as another SMS action.

        "shared_data": [],          //Array of keys strings that should carry over to the new sequence when copied.

        "cooldown": 0,              //Time in seconds between each firing of an action. 0 means there will be no wait between firing the same action.


        "remote_op": "RDL"          //Remote command we send to the vehicle. Options currently implemented:
                                    //RDL   - Sends a remote command to Lock the vehicle
                                    //RDU   - Sends a remote command to Unlock the vehicle
                                    //REON  - Sends a remote command to turn On the vehicle
                                    //REOFF - Sends a remote command to turn Off the vehicle
      }
    ],
    "association": "MMUX Demo",     //Used to group sequences to each other. Does not need to be unique
    "active": true,                 //Is this sequence active
    "shared": false                 //Is this sequence's data able to be shared outside of the owner.
  }

```

Trigger and Action Options

Triggers

Time (First in Trigger List Only)

```
{
  "condition_type": "time",
  "cron": "* * * * *" // any valid cron expression
}
```

Time Range (Cascade only)

```
{
  "condition_type": "time_range",
  "start_time": "00:59",  // valid 24 hour time (UTC)
  "end_time": "03:45"     // valid 24 hour time (UTC)
}
```

Zonemapper (First in Trigger List Only)

```
{
  "condition_type": "zonemapper",
  "breach_type": "ENTER", // either ENTER or EXIT
  "geofence_collection_name": "moonbase" // valid name created in Zonemapper prior to sequence creation
}
```

ADA Sensor

```
{
  "condition_type": "sensor",
  "values": [                     // test values, order corresponds to sensor_names
    "test",
    45
  ],
  "sensor_names": [               // name of sensors to check for
    "Custom-Sensor",
    "Speedometer"
  ],
  "value_to_actual_operators": [  // operators for logic checking
    "!=",                         // in format of value <operator> actual value
    ">="
  ]
}
```

Weather (Precipitation)

```
{
  "condition_type": "weather",
  "weather_qualifier": "precipitation", // check for precipitation
  "precipitation_types": [               // full list of valid precipitation types
    "Clear",
    "Drizzle",
    "Snow",
    "Mist",
    "Dust",
    "Fog",
    "Tornado",
    "Squall",
    "Thunderstorm",
    "Sand",
    "Smoke",
    "Clouds",
    "Rain",
    "Haze",
    "Ash"
  ]
}
```

Weather (Temperature)

```
{
  "condition_type": "weather",
  "units": "f", // valid units are Fahrenheit (f), Celsius (c), Kelvin (k)
  "value": 0,   // value to check (must be number)
  "weather_qualifier": "temperature", // check for temperature
  "value_to_actual_operator": "<" // operator in format of value <operator> actual temperature
}
```

Actions

SMS Message

```
{
  "name": "SMS from Everything", // name to be used for cooldown purposes
  "phones": [ // phone numbers in format +<country code><number>
    "+15551234567" // equates to +1 (555) 123-4567
  ],
  "message": "This is an example.", // body of text message to send
  "cooldown": 0,      // cooldown to send next message in seconds
  "shared_data": [],  // An array of key strings, carries over to new sequence when copied
  "action_type": "sms_message"
}
```

Email

```
{
  "body": "This is an email from everything.",  // body of email, can include html tags
  "name": "Email from Everything",              // name for cooldown purposes
  "subject": "Email from Everything",           // subject of email
  "email_cc": [],                               // emails to CC
  "email_to": [                                 // emails to send to
    "example@stellantis.com"
  ],
  "email_bcc": [],                              // emails to BCC
  "cooldown": 0,                                // in seconds, a cooldown timer to send next email
  "shared_data": [],                            // Array of key strings, carries over to new sequence when copied
  "action_type": "email_message"
}
```

Remote Vehicle Operation

```
{
  "action_type": "remote_op", //Remote Operation that is sent to the vehicle.
  "pin": "0000",              //sdp pin required to execute the remote command.
  "sdp": "IGNITE",            //SDP
  "uid": "xxx",               //The uid for the sdp
  "name": "Remote Lock",      //Name of the action. Must be unique to each actions type. You can have a SMS action with the same name as
                              //  a remote_op, but not the same name as another SMS action.

  "shared_data": [],          //Array of keys strings that should carry over to the new sequence when copied.

  "cooldown": 0,              //Time in seconds between each firing of an action. 0 means there will be no wait between firing the same action.


  "remote_op": "RDL"          //Remote command we send to the vehicle. Options currently implemented:
                              //RDL   - Sends a remote command to Lock the vehicle
                              //RDU   - Sends a remote command to Unlock the vehicle
                              //REON  - Sends a remote command to turn On the vehicle
                              //REOFF - Sends a remote command to turn Off the vehicle
}
```

ADA On Demand Custom Policy

```
{
    "name": "Example",
    "policy": #######,  // policy number to send
    "shared": [], // array of key strings, carries over to new sequence when copied
    "cooldown": 0,  // in seconds, a cooldown timer to send next On demand policy
    "action_type": "on_demand",
    "vehicle_ada_environment": "dev" // ADA environment in which the targeted vehicle is
}
```

ADA On Demand Live Policy

```
// Coming soon
```

HTTP Request

```
{
  "url": "url", // url for request
  "name": "Post from everything",
  "payload": "payload", // string escaped JSON payload
  "action_type": "http",
  "http_method": "GET|POST|PUT|DELETE", // http method options are GET, POST, PUT, DELETE
  "query_string_parameters": "query string parameters"
  "cooldown": 0,       // in seconds, a cooldown timer to send next HTTP request
  "shared_data": [],   // Array of key strings, carries over to new sequence when copied
}
```

WhatsApp Message

```
{
  "action_type": "whatsapp_msg",
  "phones": [ // phone numbers in format +<country code><number>
    "+15551234567" // equates to +1 (555) 123-4567
  ],
  "cooldown": 0, // in seconds, the time to send next message
  "shared_data": [], // array of key strings that carries over to new sequence when copied
  "template" : "" // a json object that includes a list of variables to be sent to WhatsApp
}
```