编写简单脚本——执行系列动作

【操作步骤】

1. 在前端编写脚本，脚本的 5 种动作：调用服务、延时、等待、条件判断、触发事件
2. 演示 1：编写与执行一个脚本，调用服务：persistent_notification.create
3. 演示 2：编写与执行一个脚本，系列动作，包括多次调用服务、延时。
4. 查看对应的文本编辑文件 scripts.yaml
5. 手工编写脚本配置，演示 3：增加条件判断，在条件满足下继续执行脚本

【参考】

- 脚本文档
  https://www.hachina.io/docs/434.html
- 服务文档
  https://www.hachina.io/docs/471.html
- 视频中演示的 scripts.yaml 完整样例

```
'1539694616968':
  alias: 演示脚本
  sequence:
  - condition: state
    entity_id: light.gateway_light_7c49eb18e3a7
    state: "off"
  - data:
      message: 运行演示脚本……
      notification_id: 123321
      title: 演示
    service: persistent_notification.create
  - data:
      entity_id: light.gateway_light_7c49eb18e3a7
    service: homeassistant.turn_on
  - delay: 00:00:05
  - data:
      entity_id: light.gateway_light_7c49eb18e3a7
    service: light.turn_off
  - data:
      notification_id: 123321
    service: persistent_notification.dismiss
```

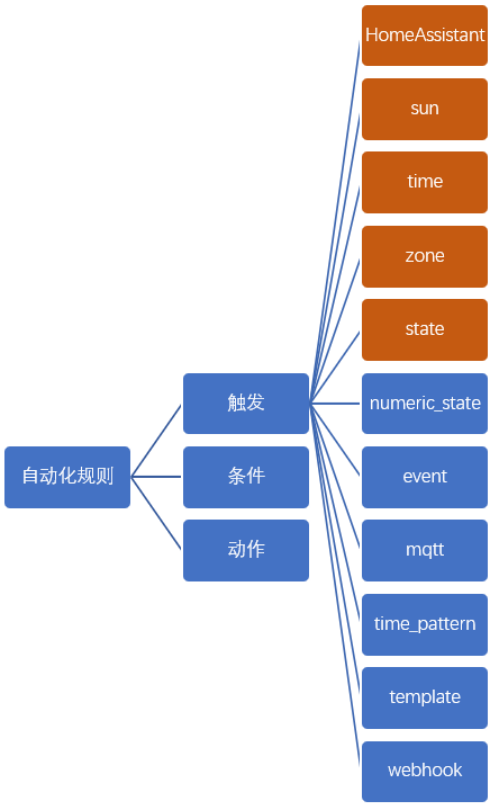
编写简单自动化规则

【操作步骤】

1. 自动化规则的三要素：触发、条件、动作
2. 前端样例编写 1：当打开灯时，语音播报开灯信息
3. 手工样例编写 2：当关闭灯时，语音播报关灯信息
4. 手工样例编写 3：每 5 秒执行开关灯操作
5. 介绍触发中的：homeassistant、sun、时间、地点

【参考】

- 自动化规则的触发



- 自动化文档
  https://www.hachina.io/docs/445.html

模板——嵌入配置文件中的程序

【操作步骤】

1. 演示模板开发者界面
2. 包含模板的脚本样例一
3. 包含模板的脚本样例二
4. 包含模板的自动化样例三

【参考】

- 模板参考文档
  https://www.hachina.io/docs/802.html
- Jinjia2 模板语法
  http://jinja.pocoo.org/docs/dev/templates/
- 模板样例 1，播报比特币行情

```
example1:
  alias: 播报比特币行情
  sequence:
  - service: tts.google_say
    entity_id: "all"
    data_template:
      message: 当前比特币行情{{ states.sensor.exchange_rate_1_btc.state }}美元
```

- 模板样例 2，如果主人在家播报比特币行情，如果主人不在家，播报主人离家距离

```
example2:
  alias: 播报比特币行情或离家距离
  sequence:
  - service: tts.google_say
    entity_id: "all"
    data_template:
      message: >
        {% if is_state('device_tracker.zhujishengiphone', 'home') %}
          当前比特币行情{{ states.sensor.exchange_rate_1_btc.state }}美元
        {% else %}
          主人离家{{ distance(states.device_tracker.zhujishengiphone)|round(2) }}公里
        {% endif %}
```

- 模板样例 3，每隔 10 秒，如果小米网关灯亮度大于 200，则运行样例 2

```
- alias: auto_broadcast
  trigger:
  - platform: time_pattern
    seconds: /10
  condition:
    condition: template
    value_template: "{{ is_state('light.gateway_light_7c49eb18e3a7', 'on') and
(states.light.gateway_light_7c49eb18e3a7.attributes.brightness > 200) }}"
  action:
  - service: script.turn_on
    entity_id: script.example2
```

事件与事件消息接收

【操作步骤】

1. 构建基于事件触发的自动化
2. 演示前端开发者界面中的事件触发
3. 传递事件中的数据
4. 编写脚本触发事件

【参考】

- 事件参考文档
  https://www.hachina.io/docs/470.html
- 视频中的自动化规则

```
- id: '1540371927161'
  alias: abc Automation
  trigger:
  - event_data: {}
    event_type: abc
    platform: event
  condition: []
  action:
  - alias: ''
    data_template:
      entity_id: all
      message: 系统中发生了事件 ABC：{{ trigger.event.data.my_message }}
    service: tts.google_say
```

- 视频中的脚本

```
abc_script:
  sequence:
    - event: abc
      event_data:
        my_message: 欢迎观看我的视频
```

前端输入组件+packages 配置

【操作步骤】

1. 使用 packages 配置完成样例一配置文件的上传
2. 解释样例一的配置内容，并演示效果
3. 各种前端组件的演示

【参考】

- packages 配置介绍
  https://www.hachina.io/docs/4156.html
- packages 目录配置

```
homeassistant:
  ……
  packages: !include_dir_named packages
```

注：放置在 packages 目录中的文件，文件名不能出现大写字母

- 前端输入组件配置介绍
  https://www.home-assistant.io/components/#search/input
- 样例一中的配置 example_6_5_1.yaml

```
# example_6_5_1.yaml
input_number:
  turnon_duration:
    name: 亮灯时长
    initial: 0
    min: 0
    max: 8
    step: 1

automation:
  - alias: turn on light n seconds
    initial_state: True
    trigger:
      - platform: state
        entity_id: input_number.turnon_duration
    condition:
      condition: numeric_state
      entity_id: input_number.turnon_duration
      above: 0
    action:
      - service: light.turn_on
        entity_id: light.gateway_light_7c49eb18e3a7
      - delay:
          seconds: "{{ states('input_number.turnon_duration')|int }}"
      - service: light.turn_off
        entity_id: light.gateway_light_7c49eb18e3a7
      - service: input_number.set_value
        data:
          entity_id: input_number.turnon_duration
          value: 0

group:
  example1_view:
    name: 样例一
    entities: light.gateway_light_7c49eb18e3a7, input_number.turnon_duration
    view: yes
```

语音+音乐+灯光闹钟

【操作步骤】

1. 使用 samba 上传 MP3 文件
2. 配置文件介绍
3. 上传配置文件，重启 HA
4. 演示闹钟功能

【参考】

- 闹钟的配置 example_6_6_1.yaml

```yaml
# example_6_6_1.yaml
# 是否打开闹钟
input_boolean:
  alarm_clock:
    name: 闹钟
    initial: off
    icon: mdi:alarm

# 闹钟的时间
input_datetime:
  alarm_time:
    name: 时间
    has_time: true
    has_date: false
    initial: 07:00

# 播放的音乐选择
input_select:
  alarm_music:
    name: 音乐
    icon: mdi:music
    options:
      - 卡农
      - I Can Feel It Coming
      - 沧海一声笑

# 播放的文字
input_text:
  alarm_text:
    name: 语音提醒
    initial: 该起床了,该起床了!
    min: 0
    max: 30

# 亮灯时长
input_number:
  alarm_light_duration:
    name: 亮灯提醒（分钟）
    initial: 0
    min: 0
    max: 60
    step: 1

# 自动化每分钟触发
automation:
- alias: Alarm Clock
  initial_state: True
  trigger:
    - platform: time_pattern
      minutes: '/1'
      seconds: 0
  # 判断闹钟是否打开，判断当前时间是否与闹钟时间相同
  condition:
    - condition: state
```

```
      entity_id: input_boolean.alarm_clock
      state: 'on'
    - condition: template
      value_template:
'{{((as_timestamp(now())|int)|timestamp_custom("%H:%M"))==states.input_datetime.alarm_time.state|truncate(5,Fa
lse,"",0) }}'
  action:
    # 播放文字语音
    - service: tts.google_say
      entity_id: "all"
      data_template:
        message: '{{states.input_text.alarm_text.state}}'
    # 打开灯
    - service: light.turn_on
      entity_id: light.gateway_light_7c49eb18e3a7
    # 延时 5 秒钟
    - delay:
        seconds: 5
    # 等待语音播放结束，最长一分钟
    - wait_template: "{{is_state('media_player.vlc','idle')}}"
      timeout: '00:01:00'
      continue_on_timeout: 'true'
    # 延时 5 秒钟
    - delay:
        seconds: 5
    # 根据选择项，播放对应的音乐
    - service: media_player.play_media
      data_template:
        entity_id: media_player.vlc
        media_content_type: music
        media_content_id: >
          {% if states.input_select.alarm_music.state=='卡农' %}
            /home/pi/Music/Canon.mp3
          {% elif states.input_select.alarm_music.state=='沧海一声笑' %}
            /home/pi/Music/沧海一声笑.mp3
          {% else %}
            /home/pi/Music/ICanFeelitComing.mp3
          {% endif %}
    # 延时 n 分钟
    - delay:
        minutes: "{{ states('input_number.alarm_light_duration')|int }}"
    # 关灯
    - service: light.turn_off
      entity_id: light.gateway_light_7c49eb18e3a7

# 将闹钟相关的控制项纳入一个组
group:
  alarm_clock:
    name: 闹钟
    entities:
      input_boolean.alarm_clock, input_datetime.alarm_time, input_text.alarm_text, input_select.alarm_music,
input_number.alarm_light_duration

# 定义一个页面组，包含闹钟以及闹钟过程中相关的实体
  alarm_clock_view:
    name: 闹钟
    entities: light.gateway_light_7c49eb18e3a7, media_player.vlc, group.alarm_clock
    view: yes
```