

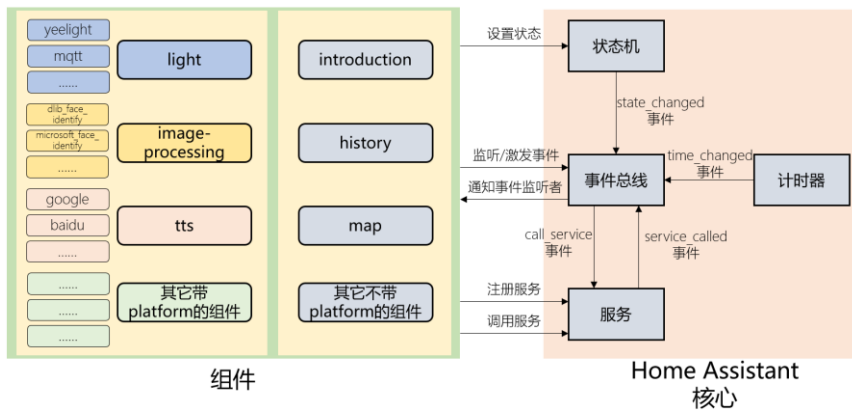
组件的工作原理

【操作步骤】

1. HomeAssistant 核心与它的 API
2. 第一个组件程序
3. 加载第一个组件程序
4. 组件的加载过程
5. 什么是 platform?

【参考】

- HomeAssistant 内部架构图



- 组件样例程序

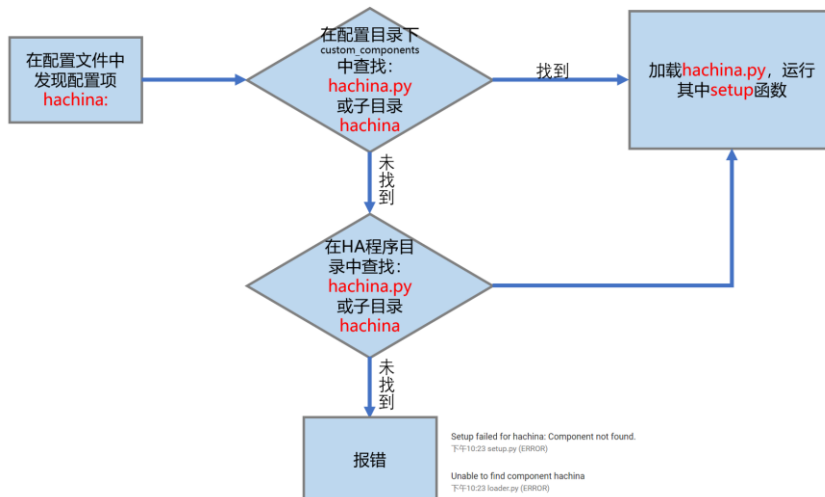
以下程序保存于 `~/.homeassistant/custom_components/hachina.py` 或 `~/.homeassistant/custom_components/hachina/__init__.py`

```
def setup(hass, config):
    # 设置实体 hachina.hello_world 的状态
    hass.states.set('hachina.hello_world', '太棒了! ')

    # 调用服务 persistent_notification.create
    service_data = {"title": "服务被调用",
                   "message": "这是调用服务后显示的信息"}
    hass.services.call('persistent_notification', 'create', service_data)

    return True
```

- 组件加载过程



- HomeAssistant 的核心机制
<https://www.hachina.io/docs/342.html>
- HomeAssistant 带有平台的组件域

alarm_control_panel	binary_sensor	calendar	camera
climate	cover	device_tracker	fan
image_processing	light	lock	mailbox
media_player	notify	remote	scene
sensor	switch	telegram_bot	tts
vacuum	weather		

程序样例：二维码识别组件

【操作步骤】

1. 使用 pillow 与 pyzbar 库
2. HomeAssistant 的 image_processing 域
3. image_processing.py 程序
4. 配置与加载 image_processing 域中的 qr 平台

【参考】

- Python 下的 pyzbar 库
<https://github.com/NaturalHistoryMuseum/pyzbar/>
- HomeAssistant 中组件程序源代码
<https://github.com/home-assistant/home-assistant/tree/dev/homeassistant/components>
- 自定义组件程序放置
将参考文档中的 `image_processing.py`、`__init__.py`、`manifest.json` 三个文件，放置在
~/.homeassistant/custom_components/qr/目录中
- 配置文件样例

```
camera:
  - platform: rpi_camera
    name: cam5
    vertical_flip: 1
image_processing:
  - platform: qr
    scan_interval: 2
  source:
    - entity_id: camera.cam5
      name: QRCode
```

Python 程序员的成长与代码规范

【操作步骤】

1. 一些教程
2. github 与 HomeAssistant 源程序
3. python 代码规范

【参考】

- 廖雪峰 Python 教程
<https://www.liaoxuefeng.com/wiki/0014316089557264a6b348958f449949df42a6d3a2e542c000>
- HomeAssistant 组件编写 StepByStep
<https://www.hachina.io/docs/1891.html>
- 一些检查程序规范的应用安装
`sudo pip3 install pylint flake8 pydocstyle`

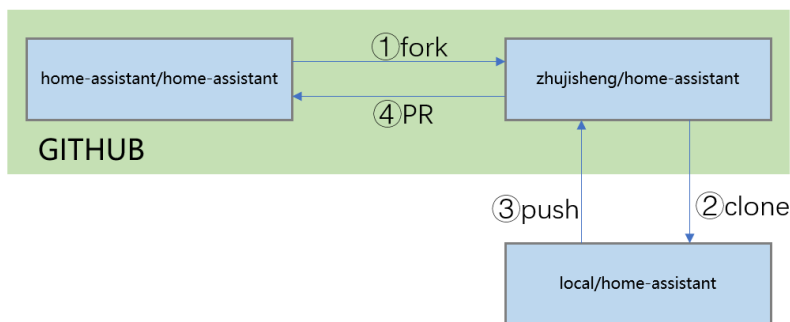
把你的代码贡献给组织

【操作步骤】

1. fork home-assistant 项目
2. 构建本地开发环境
3. 本地库修改与测试
4. 提交你的贡献
5. 提交网站文档修改

【参考】

- git 工具的介绍
<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>
- HomeAssistant 开发者指导
https://developers.home-assistant.io/docs/en/development_index.html
- 核心过程



- 构建开发环境命令
 - 安装基础软件库

```
sudo apt update
sudo apt-get install python3-pip python3-dev python3-venv
sudo apt-get install autoconf libssl-dev libxml2-dev libxslt1-dev libjpeg-dev libffi-dev libudev-dev zlib1g-dev
```
 - 创建/进入开发目录

```
mkdir my_development
cd my_development
```
 - 克隆远程库

```
git clone https://github.com/YOUR_GIT_USERNAME/home-assistant.git
cd home-assistant
git remote add upstream https://github.com/home-assistant/home-assistant.git
```
 - 构建 python 虚拟环境，从源码安装 HomeAssistant

```
python3 -m venv .
source bin/activate
pip3 install wheel
script/setup
```
- 修改与本地测试（以增加 image_processing.qrcode 为例）
 - 创建工作分支

```
git checkout -b qrcode
```
 - 修改

```
vi homeassistant/components/image_processing/qrcode.py # 编辑组件程序
script/gen_requirements_all.py      # 生成新的 requirements_all.txt
vi .coveragerc                      # 编辑测试覆盖配置文件
```

➤ 代码规范检查

```
pip3 install -r requirements_test.txt
flake8 homeassistant/components/image_processing/qrcode.py
pydocstyle homeassistant/components/image_processing/qrcode.py
pylint homeassistant/components/image_processing/qrcode.py
```

➤ 完整测试

```
tox # 完整测试命令，运行时间非常长
```

➤ 确认修改

```
git add .
git commit -m "Added some feature" --amend --author="YourName <youremail@xxx.xxx>"
```

● 提交你的贡献

➤ 获取最新官方版本内容

```
git fetch upstream dev # 获取官方的当前分支
git rebase upstream/dev # 调基
```

➤ 将修改发布到你 fork 的项目中

```
git push origin HEAD
```

➤ 创建 Pull Request (在 github 网站上进行)

本视频中创建的 PR 链接: <https://github.com/home-assistant/home-assistant/pull/20215>