

1. 基本操作

Step 1. 登入 Asar 平台

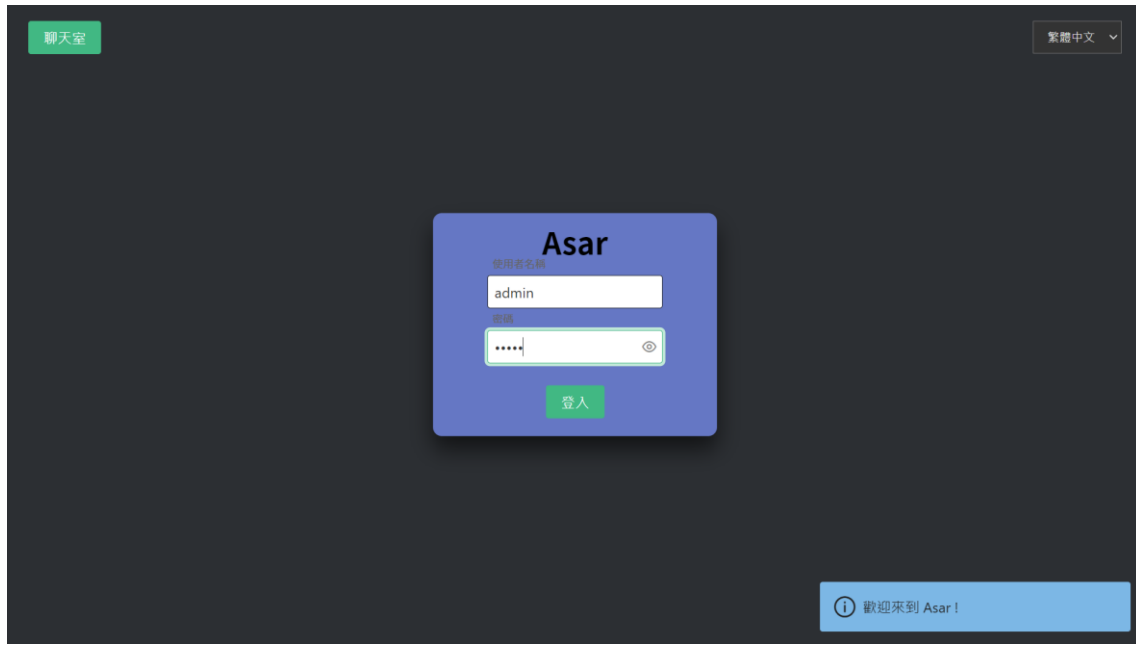


圖 1 Asar 平台登入頁面



圖 2 Asar 平台主頁面

Step 2. 創建專案

展開頁面左側選單中的「專案」，並點擊下拉選單中的「創建」來新增專案（圖 3）。創建成功後，畫面中會出現專案方塊（圖 4），點擊「編輯」進入專案設計界面（圖 5）。個人助理的範例檔案可在 Github 取得，若要使用，請將其匯入至後端 API 服務的”/data/asar_projects/”資料夾中。

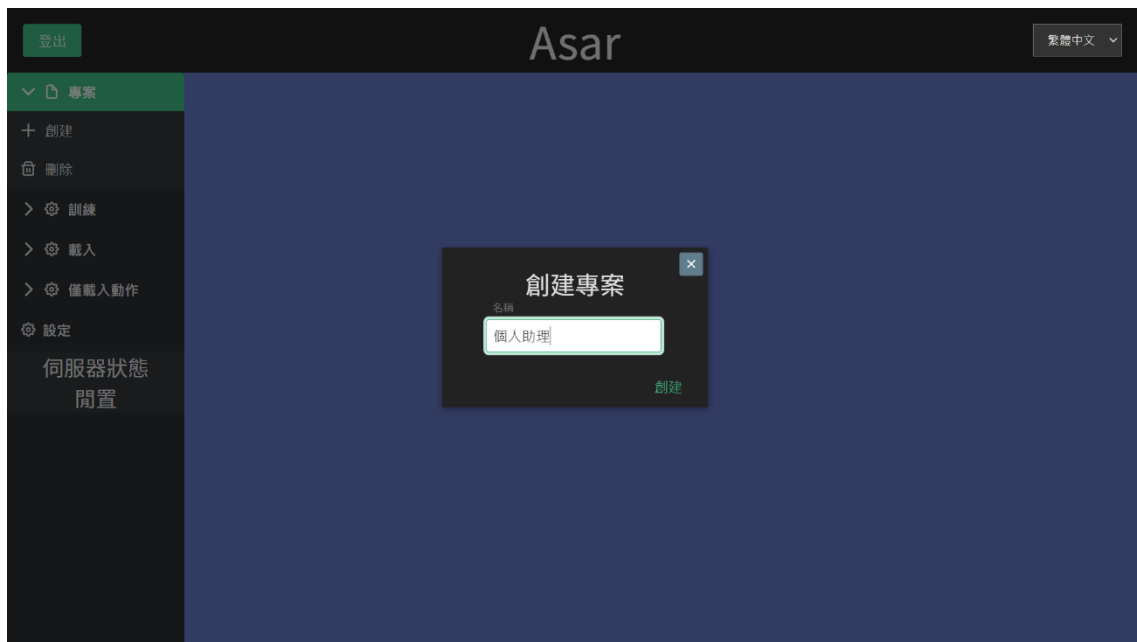


圖 3 創建專案

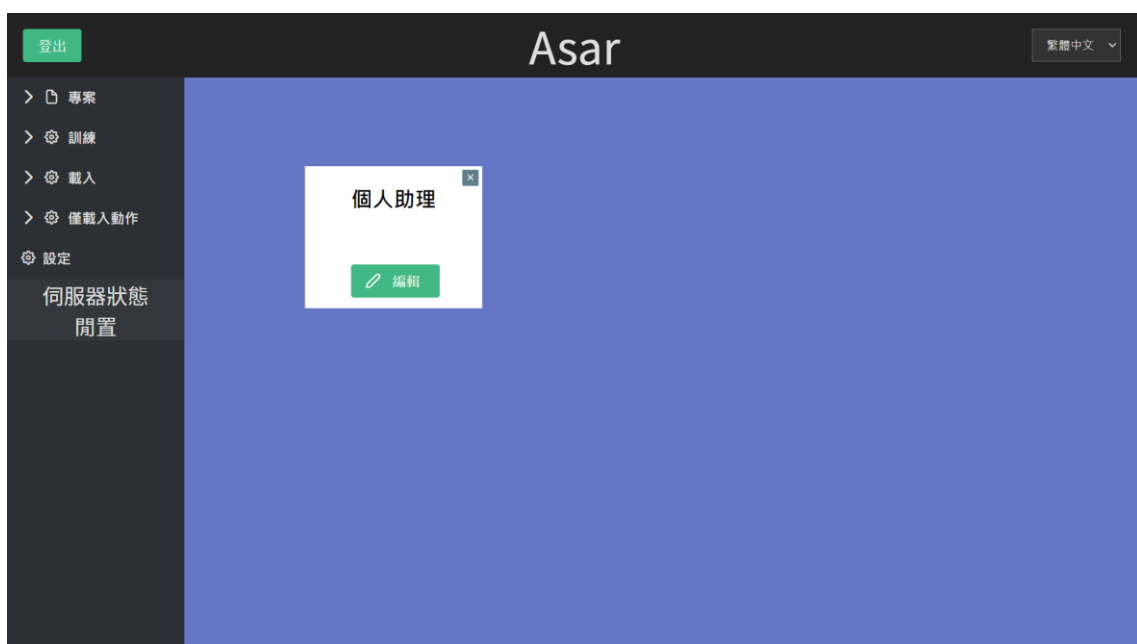


圖 4 個人助理專案



圖 5 專案設計頁面

Step 3. 定義對話劇本

當使用者向聊天機器人問候時，聊天機器人給予回覆，並且主動提問使用者當下的心情，再根據使用者的心情好壞做出不同的回應，對話劇本共有三條路線，如圖 6、圖 7、圖 8。

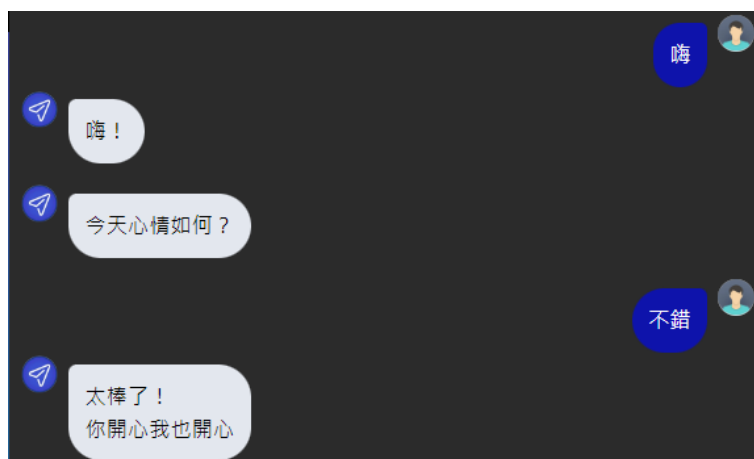


圖 6 心情好



圖 7 心情差並獲得改善



圖 8 心情差且未獲得改善

Step 4. 新增意圖

選取專案設計界面左側選單中的「意圖」，並點擊右上方的「創建」來新增意圖（圖 9）。創建成功後，點擊項目右側的綠色按鈕進入意圖設計頁面（圖 10）。在設計時，可以列舉出多種不同的文本來代表同一個意圖（圖 11）。現在根據表 1 的內容，創建出所有意圖。

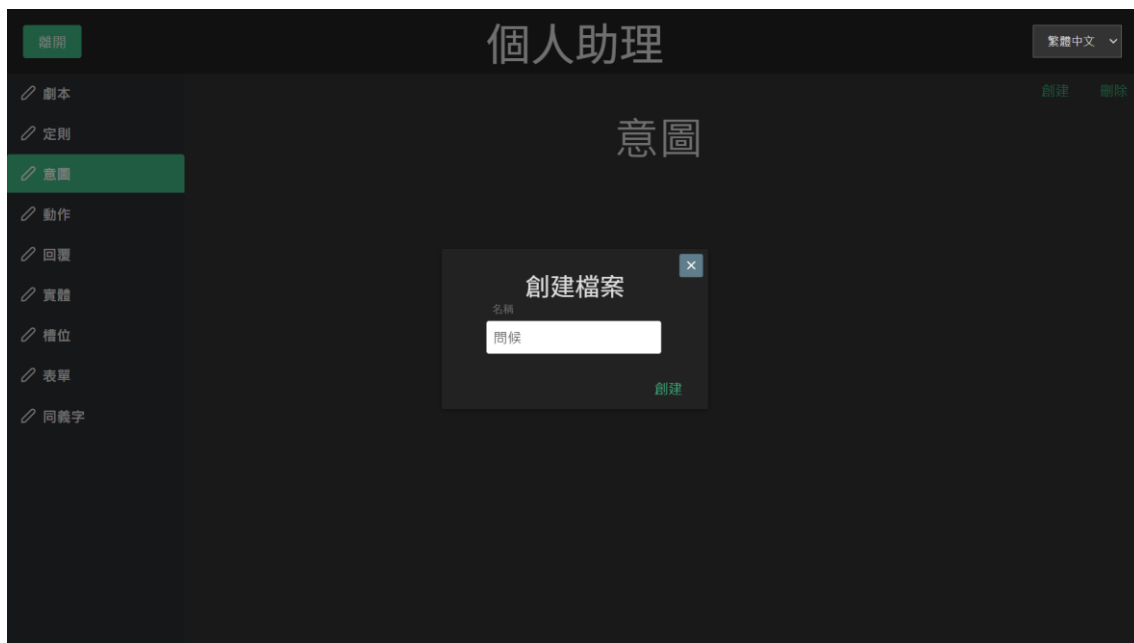


圖 9 創建立意圖



圖 10 意圖管理頁面



圖 11 意圖設計頁面（問候）

表 1 意圖與文本

意圖	文本
問候	1. 嗨 2. 哈嘍 3. 你好
心情好	1. 好 2. 非常好 3. 不錯
心情差	1. 不好 2. 有點衰 3. 很糟
同意	1. 有 2. 當然 3. 摠
拒絕	1. 沒有 2. 不

Step 5. 新增回覆

新增回覆與 Step 4 新增意圖的方法類似，回覆的訊息內容支援文字與圖像。
現在根據表 2 內容，創建出所有回覆



圖 12 回覆管理頁面

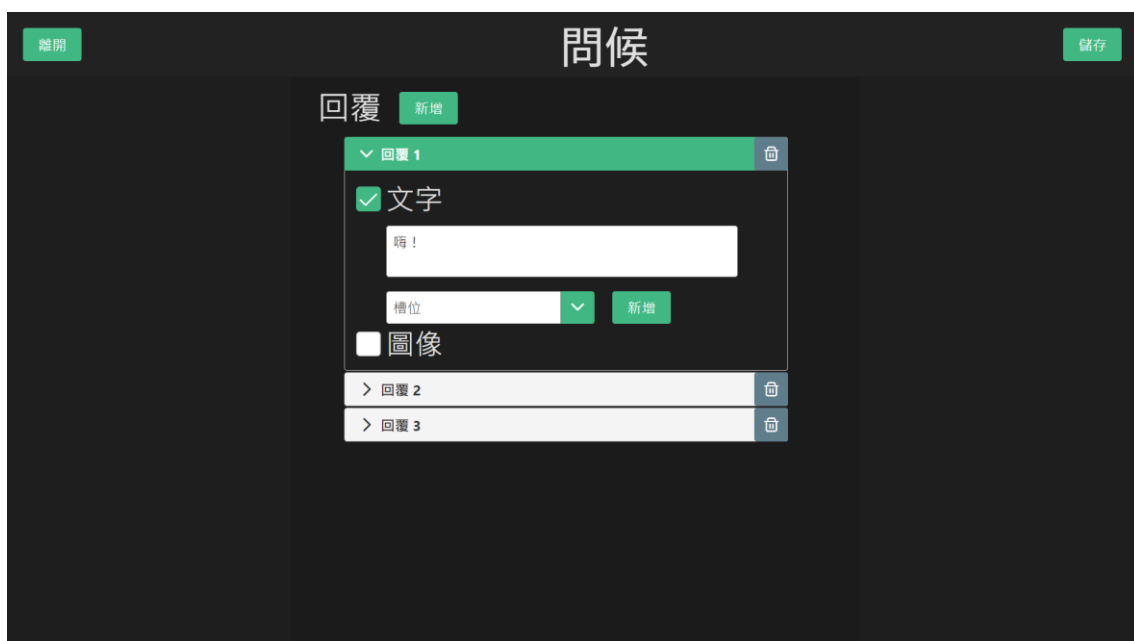


圖 13 回覆設計頁面（問候）

表 2 回覆與文本圖像

回覆	文本圖像
問候	<ol style="list-style-type: none"> 1. 嗨 2. 哈嘍 3. 嘿
問心情	<ol style="list-style-type: none"> 1. 今天心情好嗎？ 2. 今天心情如何？
心情好	<ol style="list-style-type: none"> 1. 太好了！你開心我也開心 2. 太棒了！你開心我也開心
協助改善心情	<ol style="list-style-type: none"> 1. 看看這隻可愛的動物吧！希望能讓你心情變好 http://raspberrypi/scenarios/image/animal.jpg  2. 看看這個美麗的風景吧！希望能讓你心情變好 http://raspberrypi/scenarios/image/scenery.jpg 
問心情改善	<ol style="list-style-type: none"> 1. 心情好點了嗎？ 2. 有幫助到你嗎？
心情差	<ol style="list-style-type: none"> 1. 好吧...試著做有興趣的事吧 也許能讓心情變好 2. 好吧...外出走走散散心吧

Step 6. 設計劇本

創建「問候」劇本，使用視覺化的劇本設計工具，以流程圖的概念，依照 Step 3 所定義的聊天內容來建構劇本。



圖 14 劇情管理頁面

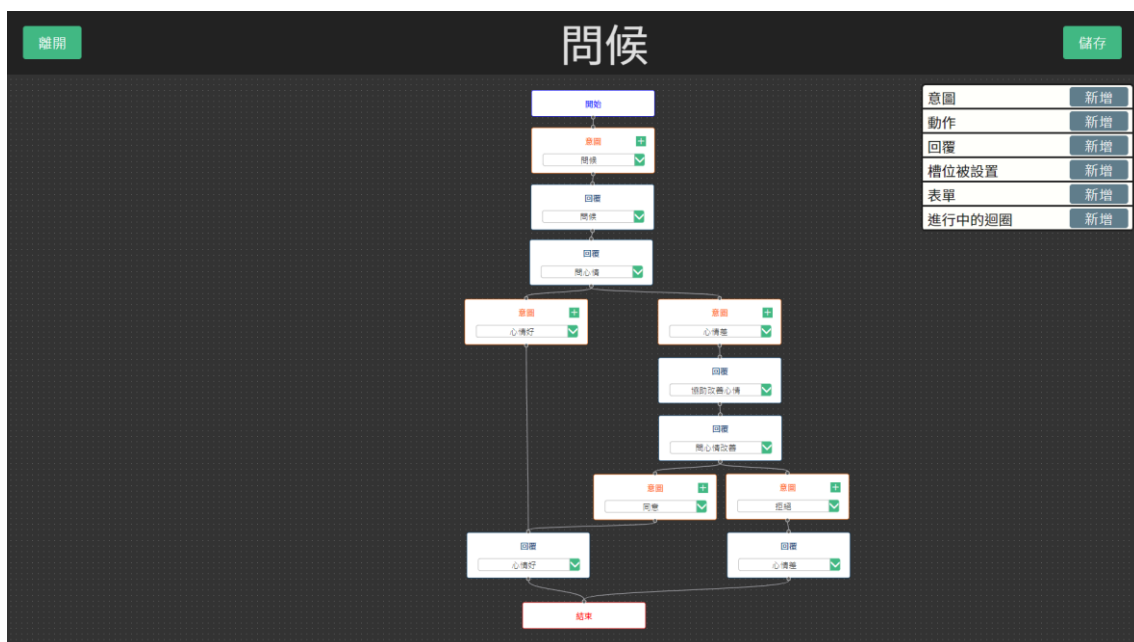


圖 15 問候劇情

Step 7. 訓練、載入對話模型

回到 Asar 平台主頁面，並展開頁面左側選單中的「訓練」及「載入」類別，點擊下拉選單中的「個人助理」專案，執行訓練及載入對話模型。



圖 16 訓練模型

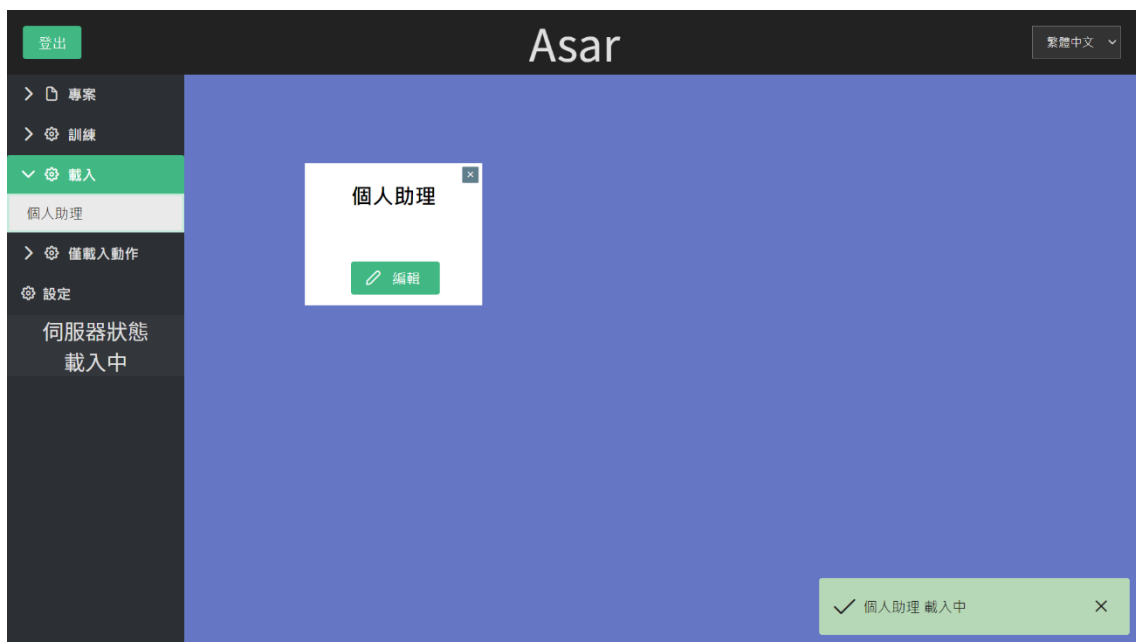


圖 17 載入模型

Step 8. 測試與調整對話模型

Asar 提供內建聊天室（圖 18）用於測試與調整對話模型，點擊登入頁面的左上方按鈕進入內建聊天室。如果發現對話內容不符合預期，可以回到主頁面，點擊頁面左側選單中「設定」（圖 19），進入模型參數編輯器來微調參數（圖 20）。

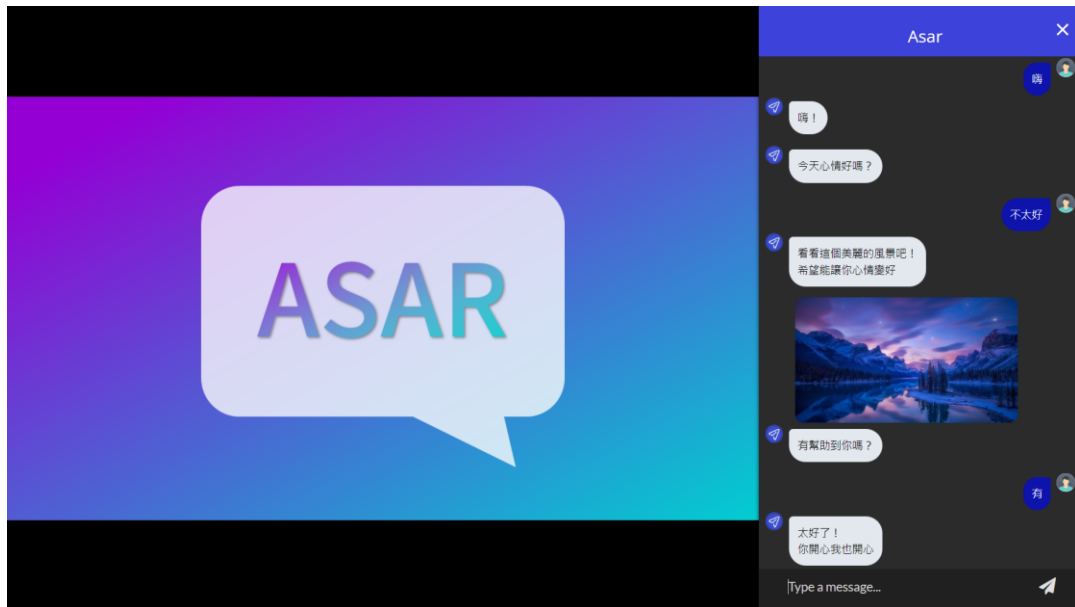


圖 18 內建聊天室

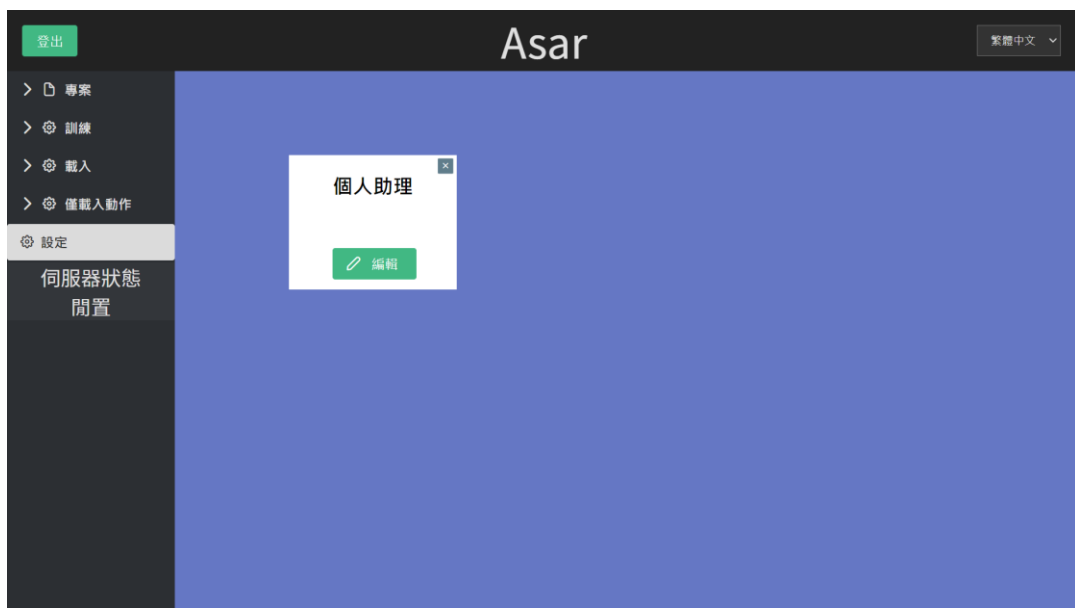


圖 19 設定模型

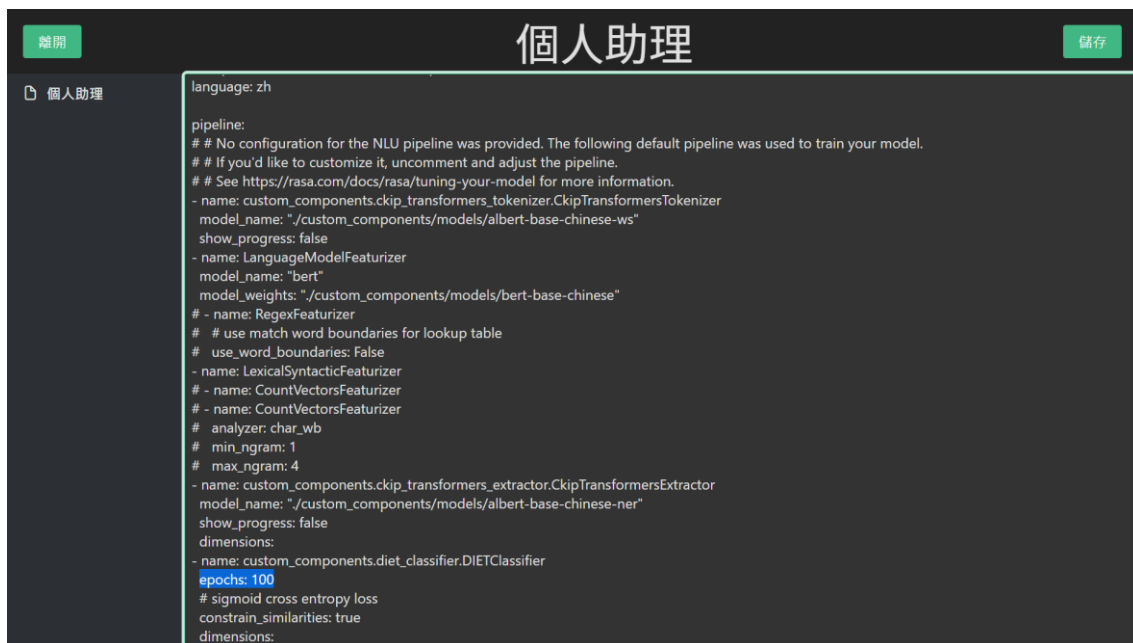


圖 20 模型參數編輯器

Step 9. 串接至通訊平台

要將 Asar 串接至通訊平台，必須把聊天機器人服務公開至網路上，本指南使用 Cloudflare Tunnel 作為範例。首先進入 Cloudflare Zero Trust 設定 Tunnel(圖 21)，並在 docker-compose.yml 新增 cloudflared 服務(圖 22)。

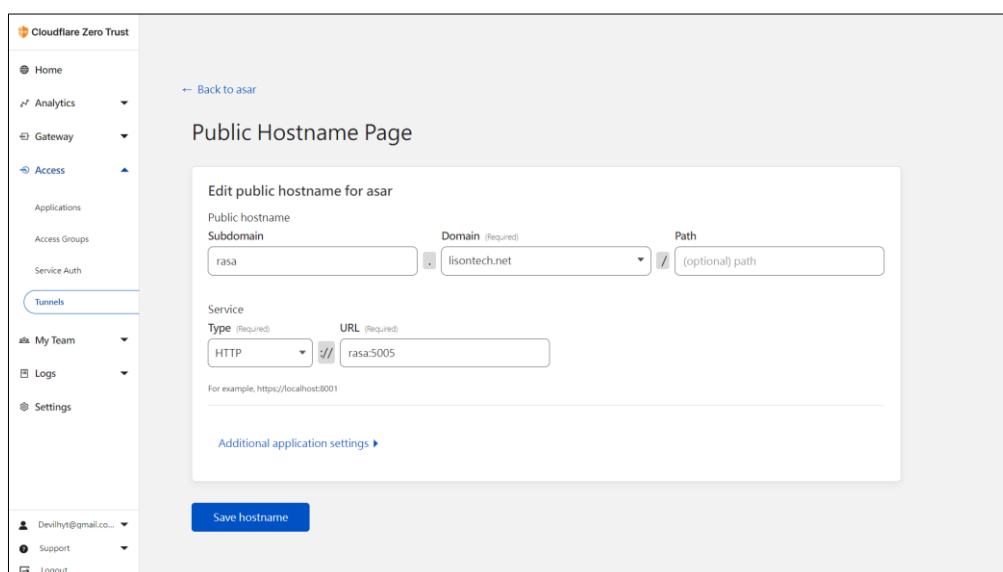


圖 21 設定 Cloudflare Tunnel

```

53   tunnel:
54     image: cloudflare/cloudflared:latest
55     command: "tunnel --no-autoupdate run"
56     environment:
57       TUNNEL_TOKEN:

```

圖 22 cloudflared 服務

打開聊天機器人服務中的”/app/credentials.yml”檔案，填寫通訊平台提供的 token 以及所有必要的驗證資訊，即完成串接，圖 23 中包含了 Telegram、Line、Facebook 的設定範例。

```

credentials.yml
asar-data > app > credentials.yml
1  # This file contains the credentials for the voice & chat platforms
2  # which your bot is using.
3  # https://rasa.com/docs/rasa/messaging-and-voice-channels
4
5  rest:
6    # # you don't need to provide anything here - this channel doesn't
7    # # require any credentials
8
9  rasa:
10   url: "http://localhost:5002/api"
11
12 telegram:
13   access_token: 571100
14   verify: Asa
15   webhook_url: https://rasa.lisontech.net/webhooks/telegram/webhook
16
17 custom_connectors.line_connector.LineConnectorInput:
18   line_channel_secret: "1ddbad033e"
19   line_channel_access_token: "8e0CmNS4exgM++ipXnmr0cmi/anJ6+DvfedmFB
20   +7VG1SYoVWyLExFAGdCGd6Q202hZI1dgOXOEUY9Y"
21
22 facebook:
23   verify: "rasa"
24   secret: "ed1c9f6c0db4c79"
25   page-access-token:
26     "EAAwpsqd6clMBACabjCzUXaZArFremg40tZBZA19howgZCo09nkyptu47x4FLvZCT
27     iqDol"

```

圖 23 credentials.yml

2. 實體提取

Asar 支援實體提取，能從使用者的聊天內容中提取出重要的文字，以提取人名為範例。

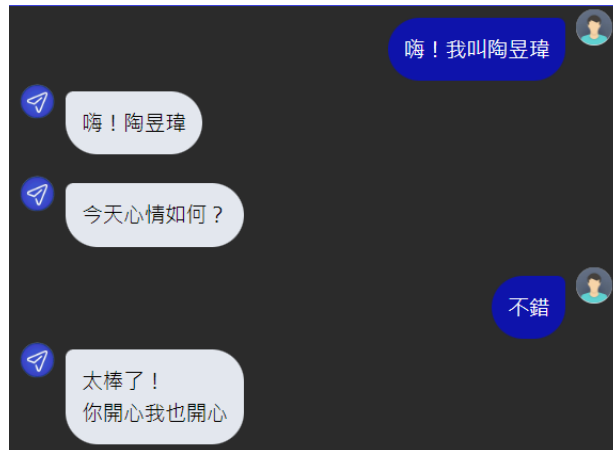


圖 1 從聊天內容中提取人名

Step 1. 創建實體類型

創建 PERSON 類型的實體，用於辨識人名。



圖 2 PERSON 類型的實體

Step 2. 創建槽位

槽位的功能是用於儲存從文本中提取出的實體內容。創建 name 槽位後，將型別定義為「文字」，並映射「問候」意圖以及「PERSON」實體。



圖 3 新增 name 槽位

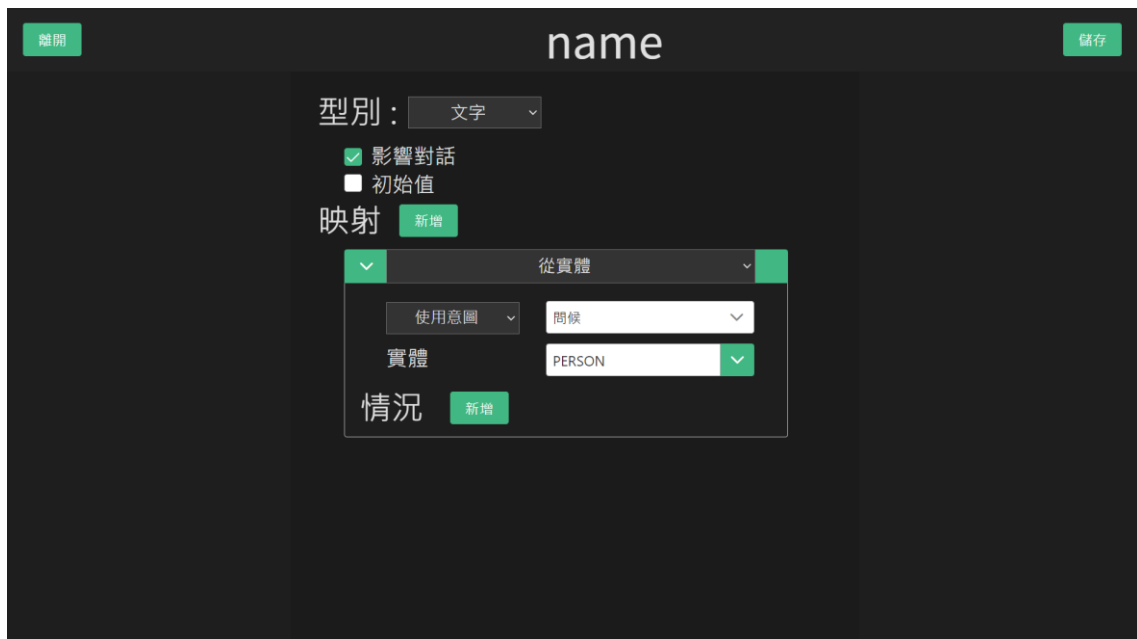


圖 4 name 槽位設定

Step 3. 實體標記

將人名加入問候意圖的文本中，並標記成 PERSON 類型的實體。



圖 5 將人名加入問候意圖的文本中



圖 6 將人名標記成 PERSON 類型的實體

Step 4. 在回覆文字中顯示提取出的實體

將 name 槽位名以花括號包住並加入至回覆文字中，可使用槽位填入工具一鍵完成。

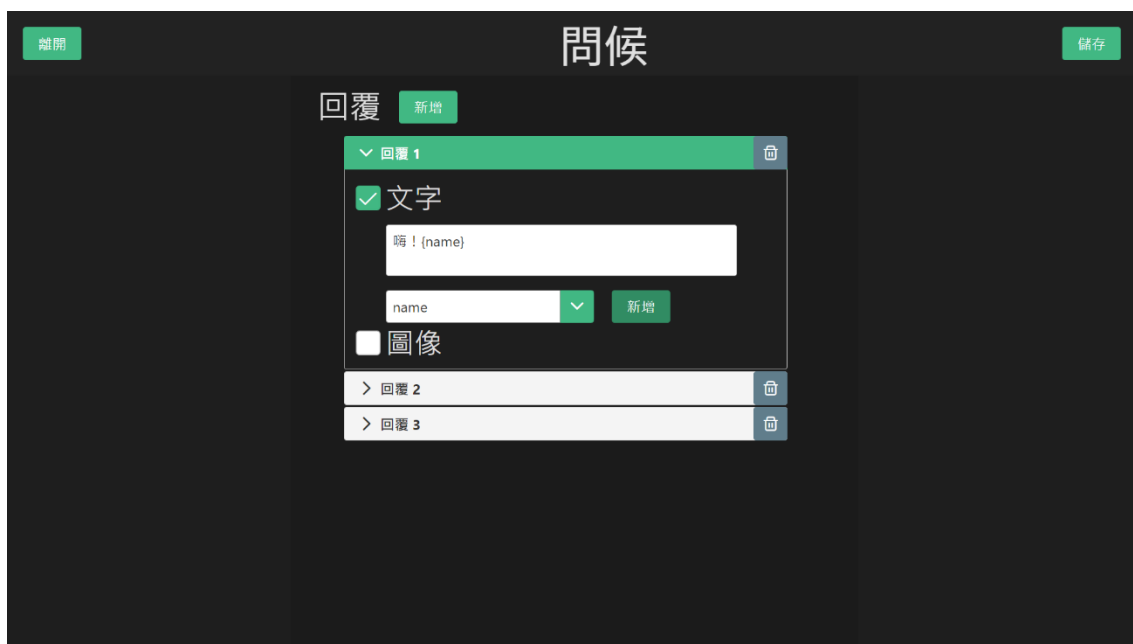


圖 7 將 name 槽位名加入至回覆文字中

3. 控制周邊設備

Asar 提供了專屬的視覺化程式設計工具，能夠輕鬆串接樹莓派的周邊設備，以控制 LED 燈為例。



圖 1 控制 LED 燈

Step 1. 建立開關燈意圖



圖 2 開燈意圖

離開 關燈 儲存

設定

預設

列舉 新增

- 把燈關掉 標記: 把燈關掉
- 關掉燈 標記: 關掉燈
- 關燈 標記: 關燈
- 熄燈 標記: 熄燈
- 有點亮 標記: 有點亮

圖 3 關燈意圖

Step 2. 建立執行完成後的回覆訊息

離開 完成 儲存

回覆 新增

- 回覆 1
 - 文字: 好
 - 槽位: [dropdown] 新增
 - 圖像: [checkbox]
- 回覆 2
 - 文字: 了解
 - 槽位: [dropdown] 新增
 - 圖像: [checkbox]

圖 4 執行完成後的回覆訊息

Step 3. 設計開關燈程式碼

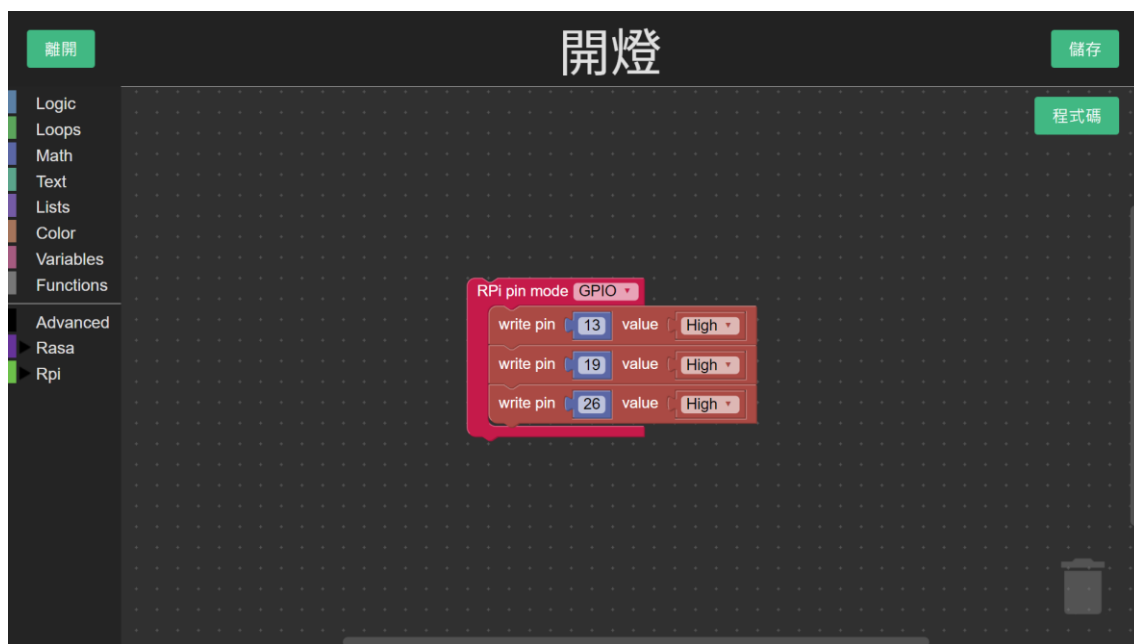


圖 5 開燈程式碼

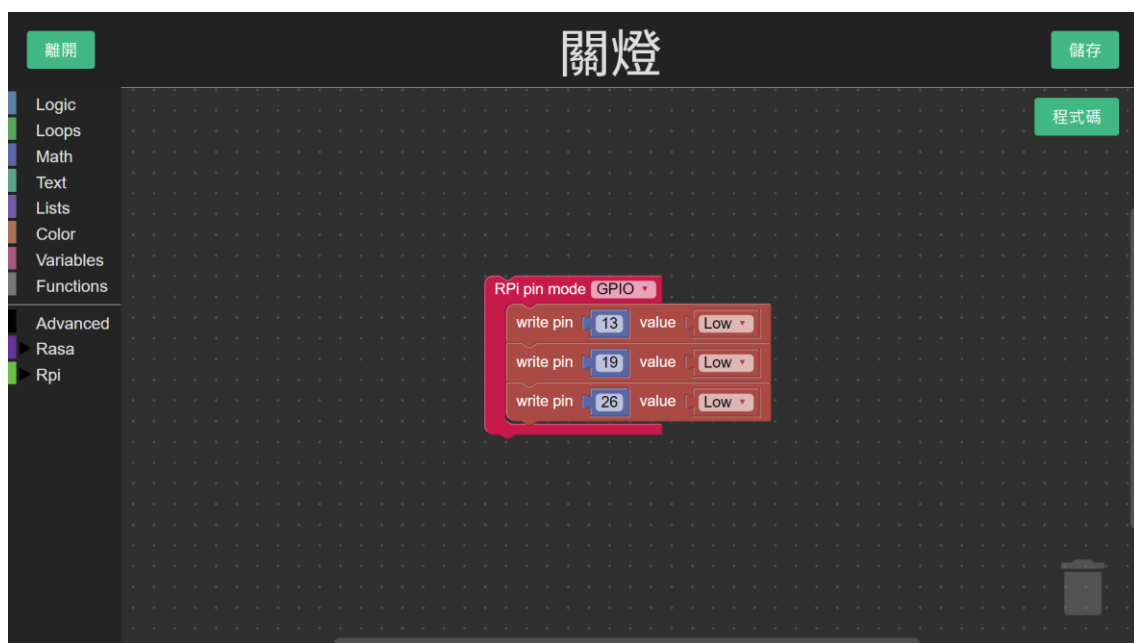


圖 6 關燈程式碼

Step 4. 新增開關燈定則（定則適用於命令型的對話指令）



圖 7 新增開關燈定則

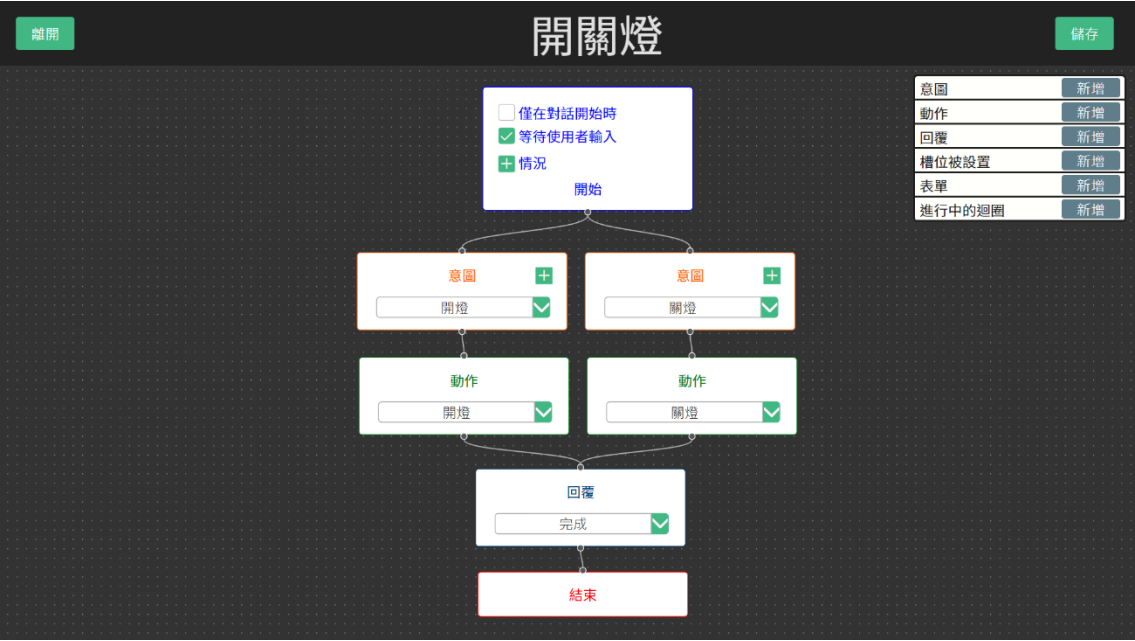


圖 8 設計開關燈定則