

# JSON 設定檔說明文件

本專案使用多個JSON檔案來管理相機配置、事件記錄、錄影資訊和模型設定。所有JSON檔案位於 `share/` 目錄下，供前端與後端服務共享使用。

## 目錄

- 1. [cameras.json](#) - 相機配置
- 2. [events.json](#) - 事件記錄
- 3. [recordings.json](#) - 錄影資訊
- 4. [models.json](#) - AI 模型配置

## 1. cameras.json - 相機配置

檔案位置

`share/cameras.json`

用途

定義系統中所有相機的配置資訊，包括串流位置、偵測物件、虛擬圍欄等設定。

結構說明

頂層欄位

欄位名稱	類型	說明	範例
<code>webrtcServerUrl</code>	string	WebRTC 伺服器位址	<code>"http://localhost:1984"</code>
<code>availableDetectionObjects</code>	array	系統可偵測的物件/ 行為清單	<code>["book", "tv", "jumping", ...]</code>
<code>cameras</code>	array	相機配置陣列	詳見下方

### `cameras[]` 陣列物件

每個相機物件包含以下欄位：

欄位名稱	類型	必填	說明	範例
<code>id</code>	string	✓	相機唯一識別碼	<code>"cam1overlay"</code>

欄位名稱	類型	必填	說明	範例
name	string	✓	相機顯示名稱	"Front Door"
streamUrl	string	✓	WebRTC 串流 URL	"cam1overlay"
rtspUrl	string	✓	RTSP 串流完整位址	"rtsp://127.0.0.1:8556/cam1overlay"
modelId	string	✗	指定使用的 AI 模型 ID	"YOLOv8_V1"
enabled	boolean	✓	是否啟用此相機	true
location	string	✗	相機位置描述	"New Location"
resolution	string	✗	影像解析度	"1920x1080"
fps	number	✗	影格率 (FPS)	30
detectObjects	array	✗	需要偵測的物件清單	["person", "book"]
recordingEnabled	boolean	✗	是否啟用錄影	true
snapshotsEnabled	boolean	✗	是否啟用快照	true
motionDetection	boolean	✗	是否啟用動態偵測	true
minConfidence	number	✗	最小信心度門檻 (0-100)	70

欄位名稱	類型	必填	說明	範例
<code>virtualFences</code>	array	<i>X</i>	虛擬圍欄設定	詳見下方
<code>zones</code>	array	<i>X</i>	預留的區域設定	<code>[]</code>

#### `virtualFences[]` - 虛擬圍欄物件

欄位名稱	類型	必填	說明	範例
<code>name</code>	string	✓	圍欄名稱	<code>"Zone 1"</code>
<code>enabled</code>	boolean	✓	是否啟用此圍欄	<code>true</code>
<code>points</code>	array	✓	多邊形頂點座標陣列	詳見下方
<code>detectObjects</code>	array	<i>X</i>	此區域要偵測的物件	<code>["person", "walking"]</code>

#### `points[]` - 座標點物件

欄位名稱	類型	說明	範圍
<code>x</code>	number	X 座標 (正規化)	0.0 ~ 1.0
<code>y</code>	number	Y 座標 (正規化)	0.0 ~ 1.0

**注意:** 座標使用正規化值 (0~1)，相對於影像寬高的比例。

## 範例

```
{
  "webrtcServerUrl": "http://localhost:1984",
  "availableDetectionObjects": [
    "book", "tv", "jumping", "sitting_down",
    "standing_up", "bending", "falling_down", "picking_up_object"
  ],
  "cameras": [
    {
      "id": "cam1overlay",
      "name": "cam1overlay",
      "streamUrl": "cam1overlay",
      "rtspUrl": "rtsp://127.0.0.1:8556/cam1overlay",
      "modelId": "YOLOv8_V1",
      "enabled": true,
      "location": "Front Entrance",
      "resolution": "1920x1080",
      "fps": 30,
      "detectObjects": ["person"],
      "recordingEnabled": true,
      "snapshotsEnabled": true,
      "motionDetection": true,
      "minConfidence": 70,
      "virtualFences": [
        {
          "name": "Zone 1",
          "enabled": true,
          "points": [
            { "x": 0.2, "y": 0.2 },
            { "x": 0.8, "y": 0.2 },
            { "x": 0.8, "y": 0.8 },
            { "x": 0.2, "y": 0.8 }
          ]
        },
        {
          "name": "Zone 2",
          "enabled": true,
          "points": [
            { "x": 0.2, "y": 0.2 },
            { "x": 0.8, "y": 0.2 },
            { "x": 0.8, "y": 0.8 },
            { "x": 0.2, "y": 0.8 }
          ]
        }
      ],
      "detectObjects": ["person", "walking"]
    }
  ]
}
```

## 2. events.json - 事件記錄

### 檔案位置

share/events.json

### 用途

儲存所有偵測到的事件記錄，供前端查詢與展示。

### 結構說明

#### 頂層欄位

欄位名稱	類型	說明
events	array	事件記錄陣列

#### events[] 陣列物件

欄位名稱	類型	必填	說明	範例
id	string	✓	事件唯一識別碼	"evt_001"
cameraId	string	✓	觸發相機 ID	"cam1"
cameraName	string	✓	相機顯示名稱	"Front Door"
type	string	✓	偵測物件類型	"person", "car", "package"
timestamp	string	✓	事件時間戳 (ISO 8601)	"2025-01-30T14:23:15Z"
thumbnail	string	✓	縮圖檔案名稱	"evt_001.jpg"
score	number	✓	偵測信心度 (0~1)	0.95
zone	string	✗	觸發區域名稱	"entrance"

## 範例

```
{
  "events": [
    {
      "id": "evt_001",
      "cameraId": "cam1",
      "cameraName": "Front Door",
      "type": "person",
      "timestamp": "2025-01-30T14:23:15Z",
      "thumbnail": "evt_001.jpg",
      "score": 0.95,
      "zone": "entrance"
    },
    {
      "id": "evt_002",
      "cameraId": "cam2",
      "cameraName": "Driveway",
      "type": "car",
      "timestamp": "2025-01-30T13:45:30Z",
      "thumbnail": "evt_002.jpg",
      "score": 0.92,
      "zone": "driveway"
    }
  ]
}
```

## 相關檔案

- 縮圖檔案儲存於: [share/events/](#) 目錄

### 3. recordings.json - 錄影資訊

檔案位置

share/recordings.json

用途

管理相機錄影檔案的元資訊，包括錄影時間、檔案大小、事件統計等。

結構說明

頂層欄位

欄位名稱	類型	說明
recordings	array	錄影記錄陣列

recordings[] 陣列物件

欄位名稱	類型	必填	說明	範例
id	string	✓	錄影唯一識別碼	"rec_001"
cameraId	string	✓	錄影相機 ID	"cam1"
cameraName	string	✓	相機顯示名稱	"Front Door"
date	string	✓	錄影日期 (YYYY-MM-DD)	"2025-01-30"
startTime	string	✓	開始時間 (HH:MM:SS)	"00:00:00"
endTime	string	✓	結束時間 (HH:MM:SS)	"23:59:59"
duration	string	✓	錄影長度	"24h 0m"
size	string	✓	檔案大小	"4.2 GB"
events	number	✓	事件數量	12
path	string	✓	錄影檔案路徑/名稱	"rec_001.mp4"
thumbnail	string	✓	預覽縮圖檔案名稱	"rec_001.jpg"

## 範例

```
{
  "recordings": [
    {
      "id": "rec_001",
      "cameraId": "cam1",
      "cameraName": "Front Door",
      "date": "2025-01-30",
      "startTime": "00:00:00",
      "endTime": "23:59:59",
      "duration": "24h 0m",
      "size": "4.2 GB",
      "events": 12,
      "path": "rec_001.mp4",
      "thumbnail": "rec_001.jpg"
    },
    {
      "id": "rec_002",
      "cameraId": "cam2",
      "cameraName": "Driveway",
      "date": "2025-01-30",
      "startTime": "00:00:00",
      "endTime": "23:59:59",
      "duration": "24h 0m",
      "size": "3.8 GB",
      "events": 8,
      "path": "rec_002.mp4",
      "thumbnail": "rec_002.jpg"
    }
  ]
}
```

## 相關檔案

- 錄影檔案儲存於: [share/recordings/ 目錄](#)
- 縮圖檔案儲存於: [share/recordings/ 目錄](#)



## 4. models.json - AI 模型配置

檔案位置

share/models.json

用途

定義系統中可用的 AI 模型配置，包括模型路徑、輸入尺寸、類別等資訊。

結構說明

頂層欄位

欄位名稱	類型	說明
models	array	模型配置陣列

models[] 陣列物件

欄位名稱	類型	必填	說明	範例
type	string	✓	模型類型	"yolov8", "yolov7"
name	string	✓	模型名稱 (唯一識別)	"YOLOv8_V1"
runner	string	✓	推理程式路徑	"/runner/yolov8_inference.py"
weights	string	✓	權重檔案路徑	"/models/yolov8_coco.pt"
class_file	string	✓	類別定義檔路徑	"/class/yolov8_coco.txt"
inputSize	array	✓	輸入尺寸 [寬, 高]	[640, 640]
classes	array	✓	可辨識的類別清單	詳見下方
defaultConfidence	number	✓	預設信心度門檻 (0~1)	0.5
nmsThreshold	number	✓	NMS 門檻值	0.45
device	string	✗	運算裝置	"cuda:0", "cpu", "" (自動)

classes[] 陣列物件

欄位名稱	類型	必填	說明	範例
id	number	✓	類別 ID	0
name	string	✓	類別名稱	"person"

## 範例

```
{
  "models": [
    {
      "type": "yolov8",
      "name": "YOLOv8_v1",
      "runner": "/runner/yolov8_inference.py",
      "weights": "/models/yolov8_coco.pt",
      "class_file": "/class/yolov8_coco.txt",
      "inputSize": [640, 640],
      "classes": [
        { "id": 0, "name": "person" }
      ],
      "defaultConfidence": 0.5,
      "nmsThreshold": 0.45,
      "device": ""
    },
    {
      "type": "yolov7",
      "name": "YOLOv7_coco",
      "runner": "/runner/yolov7_inference.py",
      "weights": "/models/yolov7_coco.pt",
      "class_file": "/class/yolov7_coco.txt",
      "inputSize": [640, 640],
      "classes": [
        { "id": 0, "name": "person" }
      ],
      "defaultConfidence": 0.5,
      "nmsThreshold": 0.45,
      "device": "cuda:0"
    }
  ]
}
```



## 使用注意事項

### 1. 編碼格式

- 所有 JSON 檔案必須使用 **UTF-8** 編碼
- 確保 JSON 格式正確 (可使用線上工具驗證: [JSONLint](#))

### 2. 檔案權限

- **share/** 目錄在 docker-compose 中以 **:rw** (讀寫) 模式掛載給前端
- Python 服務通常以 **:ro** (唯讀) 模式掛載

### 3. 時間格式

- 事件時間戳使用 **ISO 8601** 格式: **YYYY-MM-DDTHH:MM:SSZ**
- 錄影日期使用: **YYYY-MM-DD**
- 錄影時間使用: **HH:MM:SS**

### 4. 座標系統

- 虛擬圍欄座標使用**正規化座標** (0.0 ~ 1.0)
- **x**: 相對於影像寬度的比例
- **y**: 相對於影像高度的比例

### 5. 信心度數值

- **score** 使用 **0~1** 的浮點數
- **minConfidence** 使用 **0~100** 的整數

### 6. 相機 ID 對應

- **cameras.json** 中的 **id** 必須對應到 **go2rtc-config.yaml** 中定義的 **stream name**
- **events.json** 和 **recordings.json** 中的 **cameraId** 應對應到實際的相機 ID

### 7. 模型配置關聯

- **cameras.json** 中的 **modelId** 應對應到 **models.json** 中的 **name** 欄位
- 若未指定 **modelId**，系統應使用預設模型



---

## 常見操作

### 新增相機

1. 在 **go2rtc-config.yaml** 中新增 **stream** 定義
2. 在 **cameras.json** 的 **cameras[]** 陣列中新增相機物件
3. 重啟相關服務: **docker compose restart go2rtc action\_recognition\_server**

### 新增偵測物件

1. 在 `cameras.json` 的 `availableDetectionObjects` 中加入新物件名稱
2. 在相機的 `detectObjects` 陣列中啟用該物件
3. 確保使用的 AI 模型支援該物件類別

## 設定虛擬圍欄

1. 透過前端介面 (html16) 繪製多邊形區域
2. 系統自動將座標轉換為正規化座標並儲存至 `cameras.json`
3. 可在 `detectObjects` 中指定該區域要偵測的物件

## 查詢事件記錄

- 前端讀取 `events.json` 並依時間戳排序
- 縮圖位於 `share/events/{thumbnail}`
- 可透過前端 Server Actions 刪除事件

## 管理錄影檔案

- 錄影檔案存放於 `share/recordings/`
- 透過前端可查看、播放、刪除錄影
- `recordings.json` 會在檔案刪除時同步更新



## 相關文件

- [README.md](#) - 專案總覽與快速開始
- [go2rtc 設定](#) - 串流伺服器配置
- [Docker Compose](#) - 容器編排設定



## 技術支援

如有 JSON 格式相關問題，請檢查:

1. JSON 語法是否正確 (逗號、引號、括號)
2. 必填欄位是否都有提供
3. 資料型態是否符合規範
4. 檔案編碼是否為 UTF-8

建議使用 JSON Schema 驗證工具確保格式正確性。