

## PushHub API 调用说明

### 1. 请求方式

使用 POST 请求：

POST `http://{host}:5020/pushhub/subscribe`

使用 WebSocket 接收检测结果：

`ws://{host}:5020/pushhub/ws/results`

### 2. 调用工具

1. 文件请求：可以通过 curl 或 Python requests 发送 JSON 请求文件。
2. Postman：在 Postman 中新建一个 POST 请求，地址为
3. `http://{host}:5020/pushhub/subscribe`

在 Body 中选择 raw -> JSON 格式，输入请求体即可。

### 3. 请求示例

curl 示例

```
curl -X POST "http://localhost:5020/pushhub/subscribe" \
-H "Content-Type: application/json" \
-d '{
    "streams":
    ["rtsp://192.168.130.14:8554/test5","rtsp://192.168.130.14:8554/test13","rtsp://192.1
68.130.14:8554/face_two_man","rtsp://192.168.130.14:8554/test49","rtsp://192.168.1
30.14:8554/test42","rtsp://192.168.130.14:8554/test48","rtsp://192.168.130.14:8554/f
4","rtsp://192.168.130.14:8554/test52","rtsp://192.168.130.14:8554/test45","rtsp://19
2.168.130.14:8554/f2","rtsp://192.168.130.14:8554/test10","rtsp://192.168.130.14:855
```

```
4/test36","rtsp://192.168.130.14:8554/action2","rtsp://192.168.130.14:8554/test23","rtsp://192.168.130.14:8554/test50","rtsp://192.168.130.14:8554/test20","rtsp://192.168.130.14:8554/test12","rtsp://192.168.130.14:8554/test19","rtsp://192.168.130.14:8554/test2","rtsp://192.168.130.14:8554/test21","rtsp://192.168.130.14:8554/test25","rtsp://192.168.130.14:8554/test29","rtsp://192.168.130.14:8554/test14","rtsp://192.168.130.14:8554/test22","rtsp://192.168.130.14:8554/action1","rtsp://192.168.130.14:8554/test31","rtsp://192.168.130.14:8554/test53","rtsp://192.168.130.14:8554/test3","rtsp://192.168.130.14:8554/test32","rtsp://192.168.130.14:8554/cover2","rtsp://192.168.130.14:8554/test8","rtsp://192.168.130.14:8554/test44","rtsp://192.168.130.14:8554/test15","rtsp://192.168.130.14:8554/test35","rtsp://192.168.130.14:8554/face_wwz","rtsp://192.168.130.14:8554/test34","rtsp://192.168.130.14:8554/test27","rtsp://192.168.130.14:8554/test11","rtsp://192.168.130.14:8554/test1","rtsp://192.168.130.14:8554/face_ball","rtsp://192.168.130.14:8554/test30","rtsp://192.168.130.14:8554/test37","rtsp://192.168.130.14:8554/cover1","rtsp://192.168.130.14:8554/test6","rtsp://192.168.130.14:8554/test9","rtsp://192.168.130.14:8554/test17","rtsp://192.168.130.14:8554/f3","rtsp://192.168.130.14:8554/test47","rtsp://192.168.130.14:8554/test28","rtsp://192.168.130.14:8554/test33","rtsp://192.168.130.14:8554/face_zzs","rtsp://192.168.130.14:8554/test46","rtsp://192.168.130.14:8554/test7","rtsp://192.168.130.14:8554/area1","rtsp://192.168.130.14:8554/face_fzp","rtsp://192.168.130.14:8554/f5","rtsp://192.168.130.14:8554/test43","rtsp://192.168.130.14:8554/test18","rtsp://192.168.130.14:8554/water2","rtsp://192.168.130.14:8554/test26","rtsp://192.168.130.14:8554/water1","rtsp://192.168.130.14:8554/water3","rtsp://192.168.130.14:8554/test38","rtsp://192.168.130.14:8554/test39","rtsp://192.168.130.14:8554/test40","rtsp://192.168.130.14:8554/test4","rtsp://192.168.130.14:8554/mix_zhiliu_paihuai","rtsp://192.168.130.14:8554/f1","rtsp://192.168.130.14:8554/test54","rtsp://192.168.130.14:8554/test16","rtsp://192.168.130.14:8554/test41","rtsp://192.168.130.14:8554/test24","rtsp://192.168.130.14:8554/test51"],
```

```
"current_threshold": 18,
```

```
"edge_params": {"low": 50, "high": 150, "min_ratio": 0.02},
```

```
"tamper_params": {"entropy_threshold": 4.2, "diff_threshold": 25.0}
```

```
}
```

## 示例说明

- streams : 要订阅的视频流名称数组, 例如 "cam-lobby"、"cam-parking"。

- `current_threshold` : 黑屏检测的灰度均值阈值, 低于该值则认为画面过暗。
- `edge_params` : 遮挡检测参数 :
  - `low / high` : Canny 边缘检测阈值。
  - `min_ratio` : 最小边缘像素比例, 低于此比例认为画面被遮挡。
- `tamper_params` : 篡改检测参数 :
  - `entropy_threshold` : 画面熵低于此值认为被篡改 (低信息量) 。
  - `diff_threshold` : 相邻帧差异均值超过此值认为有剧烈运动 (可能人为干扰) 。

#### 4. 接收示例

##### 成功响应示例 (REST)

```
{
  "ok": true,
  "epoch": 7,
  "subscribed":
  ["rtsp://192.168.130.14:8554/test5","rtsp://192.168.130.14:8554/test13","rtsp://192.168.130.14:8554/face_two_man","rtsp://192.168.130.14:8554/test49","rtsp://192.168.130.14:8554/test42","rtsp://192.168.130.14:8554/test48","rtsp://192.168.130.14:8554/f4","rtsp://192.168.130.14:8554/test52","rtsp://192.168.130.14:8554/test45","rtsp://192.168.130.14:8554/f2","rtsp://192.168.130.14:8554/test10","rtsp://192.168.130.14:8554/test36","rtsp://192.168.130.14:8554/action2","rtsp://192.168.130.14:8554/test23","rtsp://192.168.130.14:8554/test50","rtsp://192.168.130.14:8554/test20","rtsp://192.168.130.14:8554/test12","rtsp://192.168.130.14:8554/test19","rtsp://192.168.130.14:8554/test2","rtsp://192.168.130.14:8554/test21","rtsp://192.168.130.14:8554/test25","rtsp://192.168.130.14:8554/test29","rtsp://192.168.130.14:8554/test14","rtsp://192.168.130.14:8554/test22","rtsp://192.168.130.14:8554/action1","rtsp://192.168.130.14:8554/test31","rtsp://192.168.130.14:8554/test53","rtsp://192.168.130.14:8554/test3","rts
```

```
p://192.168.130.14:8554/test32","rtsp://192.168.130.14:8554/cover2","rtsp://192.168.130.14:8554/test8","rtsp://192.168.130.14:8554/test44","rtsp://192.168.130.14:8554/test15","rtsp://192.168.130.14:8554/test35","rtsp://192.168.130.14:8554/face_wwz","rtsp://192.168.130.14:8554/test34","rtsp://192.168.130.14:8554/test27","rtsp://192.168.130.14:8554/test11","rtsp://192.168.130.14:8554/test1","rtsp://192.168.130.14:8554/face_ball","rtsp://192.168.130.14:8554/test30","rtsp://192.168.130.14:8554/test37","rtsp://192.168.130.14:8554/cover1","rtsp://192.168.130.14:8554/test6","rtsp://192.168.130.14:8554/test9","rtsp://192.168.130.14:8554/test17","rtsp://192.168.130.14:8554/f3","rtsp://192.168.130.14:8554/test47","rtsp://192.168.130.14:8554/test28","rtsp://192.168.130.14:8554/test33","rtsp://192.168.130.14:8554/face_zzs","rtsp://192.168.130.14:8554/test46","rtsp://192.168.130.14:8554/test7","rtsp://192.168.130.14:8554/area1","rtsp://192.168.130.14:8554/face_fzp","rtsp://192.168.130.14:8554/f5","rtsp://192.168.130.14:8554/test43","rtsp://192.168.130.14:8554/test18","rtsp://192.168.130.14:8554/water2","rtsp://192.168.130.14:8554/test26","rtsp://192.168.130.14:8554/water1","rtsp://192.168.130.14:8554/water3","rtsp://192.168.130.14:8554/test38","rtsp://192.168.130.14:8554/test39","rtsp://192.168.130.14:8554/test40","rtsp://192.168.130.14:8554/test4","rtsp://192.168.130.14:8554/mix_zhiliu_paihuai","rtsp://192.168.130.14:8554/f1","rtsp://192.168.130.14:8554/test54","rtsp://192.168.130.14:8554/test16","rtsp://192.168.130.14:8554/test41","rtsp://192.168.130.14:8554/test24","rtsp://192.168.130.14:8554/test51"]
}
```

## 响应说明

- ok : 布尔值, 是否成功。
- epoch : 当前订阅版本号, 每次调用 /subscribe 都会自增。
- subscribed : 已生效的流列表。

---

## WebSocket 接收示例

```
{
  "epoch": 7,
  "data": {
```

```
"stream": "rtsp://192.168.130.14:8554/test5",  
"ts_wall": "2025-09-12T16:20:11.123Z",  
"shape": [1080, 1920, 3],  
"result": {  
  "state": "Normal",  
  "ratio": 7.35  
}  
}  
}
```

#### 消息说明

- epoch：消息所属的订阅版本号。
- data.stream：流名称（对应订阅时传入的值）。
- data.ts\_wall：帧的时间戳（ISO8601 格式，UTC 时间）。
- data.shape：帧的尺寸信息（如 [高度, 宽度, 通道数]）。
- data.result：检测结果：
  - state：状态（可能的值：Normal、Black Screen、Occluded、Tampered (Low Entropy)、Tampered (Violent Motion)、Error）。
  - ratio：边缘像素比例（百分比），在 Normal 和 Occluded 场景下出现。
  - entropy：画面熵值，在低熵篡改时出现。
  - diff\_ratio：相邻帧差异均值，在剧烈运动时出现。
  - error：错误信息，仅在 Error 状态下返回。