**Selvy WakeUp API / C++ reference**

# class Selvy\_DNN\_Trigger 🡪 API Class

### Selvy\_DNN\_Trigger (const char **root\_path**[], const char **config\_path**[],const char \***license\_string**=NULL)

**생성자**

**생성에러 발생시 에러 코드(int) 를 throw 함 getError() 항복 참고**

| **Type** | **변수명** | **i/o** | **설명** |
| --- | --- | --- | --- |
| **Type** | **변수명** | **i/o** | **설명** |
| const char[] | root\_path | in | 앤진 resource 위치 (SDK의 assets/res 폴더의 경로) |
| const char[] | config\_path | in | 설정 ini 파일 경로, root\_path 기준으로 상대 경로로 설정 (SDK의 assets/conf/diotrg\_16k.ini 의 위치) |
| const char[] | license\_string | in | 라이선스 체크를 위한 String (추후 지원) / default NULL |

### int getError()

**DnnTrigger 엔진 생성 오류 여부 점검 (생성자에서 throw 하는 코드와 동일)**

| **Type** | **변수명** | **i/o** | **설명** |
| --- | --- | --- | --- |
| **Type** | **변수명** | **i/o** | **설명** |
| int | (리턴 값) | out | ==0 : 정상, !=0 : 오류 |

### Int detect(const int len\_sample const int16\_t\* pcm\_buf, int \*p\_trigger\_frame\_info=NULL)

**PCM 입력 받아 트리거 검출**

| **Type** | **변수명** | **i/o** | **설명** |
| --- | --- | --- | --- |
| **Type** | **변수명** | **i/o** | **설명** |
| const int | len\_sample | in | 입력 sample 개수 |
| const int16\_t[] | pcm\_buf | in | PCM 입력 |
| int \* | p\_trigger\_frame\_info | out | (option) size:2 의 int 배열의 주소를 전달  NULL 아닐 경우 Detect된 트리거의 프래임 정보를 전달한다  Index 0: detect 된 Trigger의 끝점 프래임 정보  Index 1: detect 된 Trigger의 프래임 길이  ※ 검출 알고리즘 상 검출되는 프래임 길이가 프래임 끝점의 index보다  큰 값이 나올 수 있습니다. 구현 시 주의 바랍니다. |
| int | (리턴 값) | out | <=0: 검출 안 됨,  >0: 검출됨 (검출된 frame의 index) |

### bool reset()

입력 버퍼 및 카운터 등을 초기화  
검출 중단 후 새로운 sound data를 넣기 전에 실행

# 예제 코드

|  |
| --- |
| try {      //엔진 생성 실행파일 기준으로 ../data/ 폴더에 리소스가 있을 경우      // 두번째 파라메터는 고정값      Selvy\_DNN\_Trigger dnn\_trigger("./data/", "conf/diotrg\_16k.ini");        //하기 코드는 사용되지 않음 throw로 처리 하기 catch 블록 참고 throw를 무시할경우만 사용...      /\*      if (dnn\_trigger.getError())      {          printf("error loading trigger engine\n");          return -5;      }      \*/      long feat\_size = 0;        short pcm\_buf[160];      for (int f = 0; f < samp\_len - 160; f += 160)      {          readSound(pcm\_buf, 160);  int \_finfo[2] = {};          auto ret\_dec = dnn\_trigger.detect(160, pcm\_buf, finfo);          if (0 < ret\_dec){              printf("KW detected at %d frame, Trigger Len:%d, Started at %d frame \n "           , finfo[0], finfo[1], finfo[0] - finfo[1]);          }      }  } catch (int errCode){ //생성 에러시...      printf("error loading trigger engine: %d\n", errCode);      return errCode!=0?errCode:-5;  } |