

# Just Voice API Reference

---

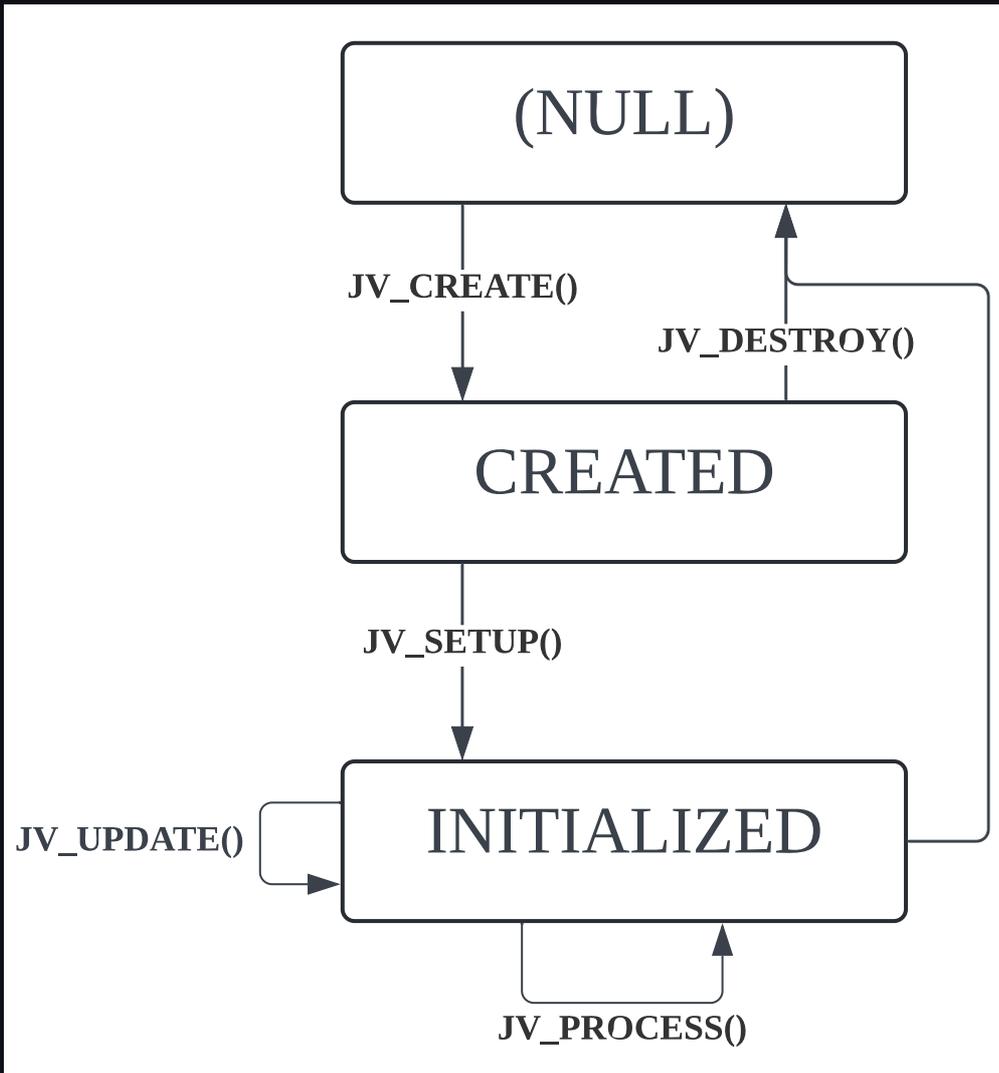
본 문서는 Just Voice C API의 전반적인 사용법을 다룹니다.

## Table of Contents

---

- [API Life Cycle](#)
- [API Description](#)
  - [Type](#)
    - `just_voice_handle_t` (Type Alias)
    - `just_voice_config_t` (Struct)
    - `just_voice_params_t` (Struct)
    - [Error Code](#) (Enum)
  - [Function](#)
    - `JV_CREATE`
    - `JV_SETUP`
    - `JV_UPDATE`
    - `JV_PROCESS`
    - `JV_DESTROY`
    - `JV_GET_LATENCY`
    - `JV_VERSION`

# API Life Cycle



## Note

- 동작에 필요한 초기 설정값은 `just_voice_config_t`와 `just_voice_params_t`에 입력한 다음, `JV_SETUP`을 통해 설정합니다.
- INITIALIZED 이후에는 `just_voice_params_t`의 항목만 `JV_UPDATE`를 통해 업데이트 할 수 있습니다.

# API Description

## Type

### `just_voice_handle_t` (Type Alias)

```
typedef void* just_voice_handle_t;
```

`just_voice_handle_t` 생성한 Just Voice 인스턴스의 포인터를 저장합니다.

### `just_voice_config_t` (Struct)

Field	Type	Value	Description
<code>numInputChannels</code>	<code>uint32_t</code>	1 이상	입력 채널의 수
<code>numOutputChannels</code>	<code>uint32_t</code>	<code>numInputChannels</code> 와 같은 값	출력 채널의 수
<code>sampleRate</code>	<code>uint32_t</code>	8000, 16000, 24000, 32000, 48000, 64000, 96000, 192000	입출력의 샘플 레이트
<code>samplesPerBlock</code>	<code>uint32_t</code>	0, 1 ~ 32768	입출력 블록 당 샘플 수 (유동적인 경우 0으로 설정)

`just_voice_config_t`는 `JV_SETUP`을 통해 설정한 이후 업데이트 할 수 없습니다.

### `just_voice_params_t` (Struct)

Field	Type	Value	Description
<code>noiseReductionIntensity</code>	<code>float</code>	0.f ~ 1.f	잡음 제거 강도를 설정합니다. (0인 경우 입력 그대로 출력, 1인 경우 최대 제거)

`just_voice_params_t`는 `JV_SETUP`을 통해 설정한 후, `JV_UPDATE`을 통해 프로세싱 중에 업데이트 할 수 있습니다.

## Error Code (Enum)

Code Name	Value	Description
<code>JV_SUCCESS</code>	0	API가 올바르게 동작했습니다.
<code>JV_NULL_EXCEPTION</code>	1	포인터의 값이 <code>NULL</code> 이라 작업을 수행할 수 없습니다.
<code>JV_NOT_INITIALIZED</code>	2	인스턴스가 초기화되지 않아 작업을 수행할 수 없습니다.
<code>JV_ALREADY_INITIALIZED</code>	3	인스턴스가 이미 초기화되어 작업을 수행할 수 없습니다.
<code>JV_ALLOCATION_FAILED</code>	4	내부 버퍼 할당에 실패하여 작업을 수행할 수 없습니다.
<code>JV_NOT_SUPPORTED_NUM_INPUT_CHANNELS</code>	5	지원하지 않는 입력 채널 수를 입력하여 작업을 수행할 수 없습니다.
<code>JV_NOT_SUPPORTED_NUM_OUTPUT_CHANNELS</code>	6	지원하지 않는 출력 채널 수를 입력하여 작업을 수행할 수 없습니다.
<code>JV_NOT_SUPPORTED_SAMPLE_RATE</code>	7	지원하지 않는 샘플 레이트를 입력하여 작업을 수행할 수 없습니다.
<code>JV_NOT_SUPPORTED_SAMPLES_PER_BLOCK</code>	8	지원하지 않는 블록 당 샘플 수를 입력하여 작업을 수행할 수 없습니다.
<code>JV_NOT_SUPPORTED_INTENSITY</code>	9	지원하지 않는 노이즈 제거 강도를 입력하여 작업을 수행할 수 없습니다.

에러 코드는 API 함수들의 반환값으로 사용됩니다.

# 함수

## JV\_CREATE

```
int32_t JV_CREATE(just_voice_handle_t** handle);
```

### 설명

인스턴스를 생성합니다. 사용 후, 자원의 해제를 위해 `JV_DESTROY`를 호출해야 합니다.

이름	설명
<code>handle</code>	새로 인스턴스를 생성할 포인터를 지정합니다. 포인팅 주소는 <code>NULL</code> 이어야 합니다.

### 사용 예시

```
just_voice_handle_t* handle = NULL;
int32_t const create_result = JV_CREATE(&handle);

if (create_result != JV_SUCCESS) { ... }
```

## JV\_SETUP

```
int32_t JV_SETUP(just_voice_handle_t* handle, just_voice_config_t
const* config, just_voice_params_t const* params);
```

### 설명

`just_voice_config_t`와 `just_voice_params_t`를 받아, 렌더링에 필요한 정보를 인스턴스에 설정합니다.

이름	설명
<code>handle</code>	정보를 설정할 인스턴스를 지정합니다.
<code>config</code>	설정값을 입력합니다.(이후 업데이트 불가)
<code>params</code>	설정값을 입력합니다.(이후 업데이트 가능)

### 사용 예시

```

just_voice_handle_t* handle = NULL;
just_voice_config_t const config = { 1, 1, 48000, 480 };
just_voice_params_t const params = { 1.f };

...

int32_t const setup_result = JV_SETUP(handle, &config, &param);

if (setup_result != JV_SUCCESS) { ... }

```

## JV\_UPDATE

```

int32_t JV_UPDATE(just_voice_handle_t* handle,
just_voice_params_t const* params);

```

### 설명

`just_voice_params_t`를 받아, 렌더링에 필요한 정보를 인스턴스에 설정합니다. `JV_SETUP`와는 다르게, 여러 번 호출하여 그 때 그 때 다른 설정으로 렌더링할 수 있도록 합니다.

이름	설명
<code>handle</code>	정보를 설정할 인스턴스를 지정합니다.
<code>params</code>	업데이트할 값을 입력합니다.

### 사용 예시

```

just_voice_handle_t* handle = NULL;
just_voice_params_t const params = { 1.f };

...

int32_t const update_result = JV_UPDATE(handle, &param);

if (update_result != JV_SUCCESS) { ... }

```

## JV\_PROCESS

```
int32_t JV_PROCESS(just_voice_handle_t* handle, float const* in,
float* out, uint32_t length);
```

## 설명

입력 버퍼의 오디오를 프로세싱하여 출력 버퍼에 넣습니다.

`samplesPerBlock`을 0으로 설정한 경우, 임의의 입력 버퍼 길이를 입력할 수 있습니다.

이름	설명
<code>handle</code>	프로세싱을 수행할 인스턴스를 지정합니다.
<code>input</code>	입력 오디오 정보를 담고 있는 버퍼를 지정합니다.
<code>output</code>	출력 오디오 정보를 담은 버퍼를 지정합니다.
<code>length</code>	입출력 샘플 수를 지정합니다.

## 사용 예시

```
just_voice_handle_t* handle = NULL;
float const* input_buffer = ...;
float* output_buffer = ...;
uint32_t length = ...;

...

int32_t const process_result = JV_PROCESS(handle, input_buffer,
output_buffer, length);

if (process_result != JV_SUCCESS) { ... }
```

## JV\_DESTROY

```
int32_t JV_DESTROY(just_voice_handle_t** handle);
```

## 설명

인스턴스를 제거하고 할당한 자원을 해제합니다.

이름	설명
<code>handle</code>	제거할 인스턴스의 포인터를 지정합니다. 제거 후, <code>NULL</code> 값이 설정됩니다.

### 사용 예시

```

just_voice_handle_t* handle = NULL;

...

int32_t destroy_result = JV_DESTROY(&handle);

if (destroy_result != JV_SUCCESS) { ... }

```

### JV\_GET\_LATENCY

```

int32_t JV_GET_LATENCY(just_voice_handle_t const* handle, float*
latency);

```

### 설명

알고리즘 레이턴시를 ms 단위로 계산하여 제공합니다.

이름	설명
<code>handle</code>	렌더링 정보가 담긴 인스턴스를 지정합니다.
<code>latency</code>	레이턴시 정보를 받을 변수를 지정합니다.

### 사용 예시

```

just_voice_handle_t* handle = NULL;
float* latency = 0.f;

...

int32_t const latency_result = JV_GET_LATENCY(handle, &latency);

if (latency_result != JV_SUCCESS) { ... }

```

### JV\_VERSION

```
int32_t JV_GET_VERSION(char const** version);
```



## 설명

SDK의 버전 정보를 제공합니다.

이름	설명
<code>version</code>	버전 정보를 받을 변수를 지정합니다.

## 사용 예시

```
char const* version;  
int32_t const version_result = JV_GET_VERSION(&version);  
  
if (version_result != JV_SUCCESS) { ... }
```

