

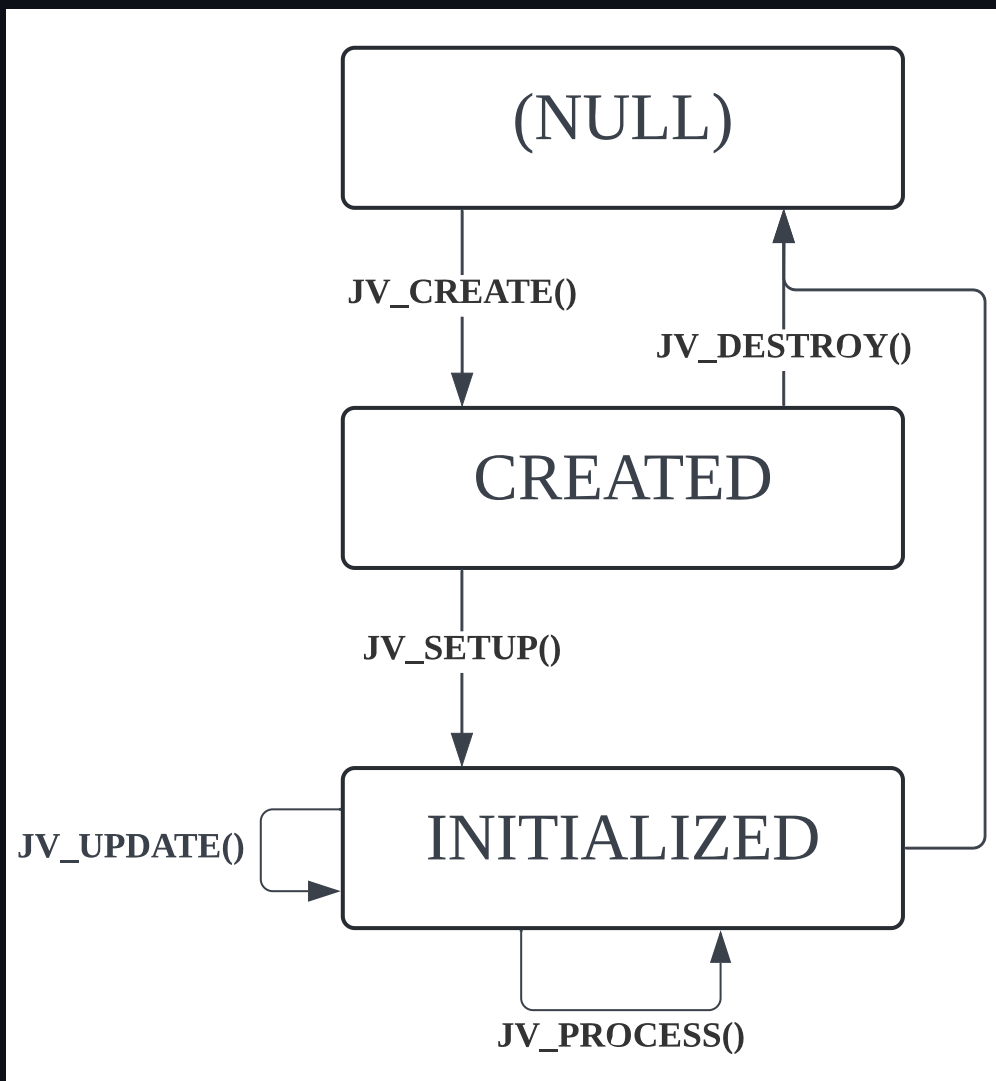
Just Voice API Reference

Introducing Just Voice's native C API.

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

- The initial settings required for processing are entered into `just_voice_config_t` & `just_voice_params_t` and then set through `JV_SETUP`.
- After `INITIALIZED`, only `just_voice_params_t` can be updated through `JV_UPDATE`.

API Description

Type

`just_voice_handle_t` (Type Alias)

```
typedef void* just_voice_handle_t;
```

`just_voice_handle_t` Stores a pointer to the JustVoice instance.

`just_voice_config_t` (Struct)

Field	Type	Value	Description
<code>numInputChannels</code>	<code>uint32_t</code>	1 or more	Number of input channels
<code>numOutputChannels</code>	<code>uint32_t</code>	Same value as <code>numInputChannels</code>	Number of output channels
<code>sampleRate</code>	<code>uint32_t</code>	8000, 16000, 24000, 32000, 48000, 64000, 96000, 192000	Sample rate of input
<code>samplesPerBlock</code>	<code>uint32_t</code>	0, 1 ~ 32768	Number of samples per input block (0 if dynamic)

`just_voice_config_t` CANNOT be updated during processing.

`just_voice_params_t` (Struct)

Field	Type	Value	Description
<code>noiseReductionIntensity</code>	<code>float</code>	0.f ~ 1.f	Level of noise reduction (If 0, no noise is removed, if 1, noise is

Field	Type	Value	Description
			completely removed.)

`just_voice_params_t` CAN be updated during processing. (It takes 10ms of interpolation time.)

Error Codes (Enum)

Code Name	Value	Description
<code>JV_SUCCESS</code>	0	No error
<code>JV_NULL_EXCEPTION</code>	1	The pointer's value is <code>NULL</code> .
<code>JV_NOT_INITIALIZED</code>	2	The instance has not been initialized. (Make sure you have set <code>just_voice_config_t</code> and <code>just_voice_params_t</code> .)
<code>JV_ALREADY_INITIALIZED</code>	3	The instance has already been initialized.
<code>JV_ALLOCATION_FAILED</code>	4	Internal buffer allocation failed.
<code>JV_NOT_SUPPORTED_NUM_INPUT_CHANNELS</code>	5	The value of the set number of input channels is outside the supported range.
<code>JV_NOT_SUPPORTED_NUM_OUTPUT_CHANNELS</code>	6	The value of the set number of output channels is outside the supported range.
<code>JV_NOT_SUPPORTED_SAMPLE_RATE</code>	7	The value of the set sample rate is outside the supported range.
<code>JV_NOT_SUPPORTED_SAMPLES_PER_BLOCK</code>	8	The value of the set sample per block is outside the supported range.
<code>JV_NOT_SUPPORTED_INTENSITY</code>	9	The value of the set noise reduction intensity is outside the supported range.

Error codes are used as return values from API functions.

Function

JV_CREATE

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

Description

Create an instance. After use, `JV_DESTROY` must be called to release the resource.

Parameter	Description
<code>handle</code>	Specifies the pointer to the new instance to be created. The input pointing address must be NULL.

Example

```
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);
```

Description

Set the instance from the the values of `just_voice_config_t` and `just_voice_params_t`.

Parameter	Description
<code>handle</code>	An instance in Initialized state
<code>config</code>	A <code>just_voice_config_t</code> structure containing settings values.

Parameter	Description
<code>params</code>	A <code>just_voice_params_t</code> structure containing settings values.

Example

```

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);

```

Description

Update the instance from the the values of `just_voice_params_t` during processing.

Parameter	Description
<code>handle</code>	An instance in Initialized state
<code>params</code>	A <code>just_voice_params_t</code> structure containing settings values.

Example

```

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);
```

Description

Processes the audio from the input buffer and puts it into the output buffer.

If `samplesPerBlock` is nonzero, the values of `length` and `samplesPerBlock` must be equal. Else if is set to 0, it can be used for arbitrary lengths.

Parameter	Description
<code>handle</code>	An instance in Initialized state
<code>input</code>	Input audio buffer
<code>output</code>	Output audio buffer
<code>length</code>	The number of input and output samples.

Example

```
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);
```

Description

Remove the instance. Releases allocated resources.

Parameter	Description
<code>handle</code>	An instance to be removed. After removal, the value <code>NULL</code> is set.

Example

```
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);
```

Description

Calculates and outputs the algorithm latency in ms.

Parameter	Description
<code>handle</code>	An instance in Initialized state
<code>latency</code>	Calculated latency value

Example

```
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);
```



Description

Check the version number of the SDK.

Parameter	Description
<code>version</code>	Version number of this SDK

Example

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

