
ScreenRecorderSDK Documentation

Andrey Parfenov

Dec 11, 2021

CONTENTS

1	System Requirements	1
2	User API	3
2.1	Python API Reference	3
3	Code Samples	7
3.1	Python	7
4	MIT License	9
5	Search	11

SYSTEM REQUIREMENTS

- Windows \geq 10, may work on Windows 8.1 and Windows Server 2012, but we don't ensure it
- DirectX, you can install it from [Microsoft Website](#)
- Media Feature Pack, download it [here](#)
- 64 bits Java or Python, we don't provide x86 libs

2.1 Python API Reference

2.1.1 screen_recorder_sdk.screen_recorder

class screen_recorder_sdk.screen_recorder.**RecorderParams** (*desktop_num=0, pid=0*)
Bases: object

inputs parameters for `init_resources` method

Parameters

- **desktop_num** (*int*) – desktop num, counting from 0
- **pid** (*int*) – pid of process to capture

exception screen_recorder_sdk.screen_recorder.**RecorderError** (*message, exit_code*)
Bases: Exception

This exception is raised if non-zero exit code is returned from C code

Parameters

- **message** (*str*) – exception message
- **exit_code** (*int*) – exit code from low level API

screen_recorder_sdk.screen_recorder.**init_resources** (*params*)
Init resources for recording

Parameters **params** (*RecorderParams*) – params for recording

Raises **RecorderError** – if non zero exit code returned from low level API

screen_recorder_sdk.screen_recorder.**get_screenshot** (*max_attempts=1*)
Get Screenshot

Parameters **max_attempts** (*int*) – max attempts to capture frame buffer

Returns Pillow Image

Return type Pillow Image

Raises **RecorderError** – if non zero exit code returned from low level API

screen_recorder_sdk.screen_recorder.**get_pid**()
Get PID

Return type int

Returns PID

Raises RecorderError – if non zero exit code returned from low level API

`screen_recorder_sdk.screen_recorder.free_resources()`
Free Resources

Raises RecorderError – if non zero exit code returned from low level API

`screen_recorder_sdk.screen_recorder.enable_log()`
Enable Logger

Raises RecorderError – if non zero exit code returned from low level API

`screen_recorder_sdk.screen_recorder.enable_dev_log()`
Enable Dev Logger

Raises RecorderError – if non zero exit code returned from low level API

`screen_recorder_sdk.screen_recorder.disable_log()`
Disable Logger

Raises RecorderError – if non zero exit code returned from low level API

`screen_recorder_sdk.screen_recorder.start_video_recording(filename,
frame_rate=30,
bit_rate=8000000,
use_hw_transforms=True)`
Start Video Recording

Parameters

- **filename** (*str*) – filename to store video
- **frame_rate** (*int*) – FPS
- **bit_rate** (*int*) – bit rate, set higher values for better quality
- **use_hw_transforms** – if you have good GPU set this flag to True for better perf, if you see errors try to set it to false

Raises RecorderError – if non zero exit code returned from low level API

`screen_recorder_sdk.screen_recorder.stop_video_recording()`
Stop video recording

Raises RecorderError – if non zero exit code returned from low level API

`screen_recorder_sdk.screen_recorder.set_log_file(log_file)`
redirect logger from stderr to file, can be called any time :param log_file: log file name :type log_file: str :raises RecorderError: if non zero exit code returned from low level API

2.1.2 screen_recorder_sdk.exit_codes

class `screen_recorder_sdk.exit_codes.RecorderExitCodes` (*value*)

Bases: `enum.Enum`

Enum to store all possible exit codes

STATUS_OK = 0

NO_SUCH_PROCESS_ERROR = 100

RECORDING_ALREADY_RUN_ERROR = 101

RECORDING_THREAD_ERROR = 102

RECORDING_THREAD_IS_NOT_RUNNING_ERROR = 103
INVALID_ARGUMENTS_ERROR = 104
SESSION_NOT_CREATED_ERROR = 105
PREPARE_DESK_DUPL_ERROR = 106
CREATE_TEXTURE_ERROR = 107
DDA_CAPTURE_ERROR = 108
FIND_WINDOW_ERROR = 109
DDA_LOST_ACCESS_ERROR = 110
DDA_TIMEOUT_ERROR = 111
SYNC_TIMEOUT_ERROR = 112
GENERAL_ERROR = 113

CODE SAMPLES

3.1 Python

3.1.1 Python Basic Demo

```
import time
import numpy

from screen_recorder_sdk import screen_recorder

def main ():
    screen_recorder.enable_dev_log ()

    params = screen_recorder.RecorderParams ()
    # params.pid = 0 # use it to set process Id to capture
    # params.desktop_num = 0 # use it to set desktop num, counting from 0

    screen_recorder.init_resources (params)

    screen_recorder.get_screenshot (5).save ('test_before.png')

    screen_recorder.start_video_recording ('video1.mp4', 30, 8000000, True)
    time.sleep (5)
    screen_recorder.get_screenshot (5).save ('test_during_video.png')
    time.sleep (5)
    screen_recorder.stop_video_recording ()

    screen_recorder.start_video_recording ('video2.mp4', 30, 8000000, True)
    time.sleep (5)
    screen_recorder.stop_video_recording ()

    screen_recorder.free_resources ()

if __name__ == "__main__":
    main ()
```


MIT LICENSE

Copyright (c) 2018 Andrey Parfenov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SEARCH

- search