Log Management API > BS2_GetFilteredLog

BS2_GetFilteredLog

[   ]   .

> BS2_GetFilteredLog
> .
>   ,
>   .
>
>   ,   .
>   ,
>   .
>   .
>
>   BS2_GetLog
>   BS2_StartMonitoringLog
>   DBMS   , DBMS
>   .

```
#include "BS_API.h"

int BS2_GetFilteredLog(void* context, uint32_t deviceId, char* uid, uint16_t
eventCode, uint32_t start, uint32_t end, uint8_t tnakey, BS2Event** logsObj,
uint32_t* numLog);
```

> BS2Event

- [In] *context* : Context
- [In] *deviceId* :
- [In] *uid* :   NULL   .
- [In] *eventCode* :   O   .
- [In] *start* :   O   .
- [In] *end* :   O   .

- [In] *tnakey* : TNA key   O   TNA key   .
- [Out] *logsObj* :
- [Out] *numLog* :

logsObj   BS2_ReleaseObject   .

BS_SDK_SUCCESS   ,

.

C#

```
Type structureType = typeof(BS2Event);
int structSize = Marshal.SizeOf(structureType);
IntPtr uid = IntPtr.Zero;
UInt16 eventCode = ;
UInt32 start = ;
UInt32 end = ;
byte tnaKey = ;
IntPtr outEventLogObjs = IntPtr.Zero;
UInt32 outNumEventLogs = ;

BS2ErrorCode result = (BS2ErrorCode)API.BS2_GetFilteredLog(sdkContext,
deviceID, uid, eventCode, start, end, tnaKey, out outEventLogObjs, out
outNumEventLogs);
if (result != BS2ErrorCode.BS_SDK_SUCCESS)
{
    Console.WriteLine("Got error({0}).", result);
}
else if (outNumEventLogs > )
{
    IntPtr curEventLogObjs = outEventLogObjs;
    for (int idx = ; idx < outNumEventLogs; idx++)
    {
        BS2Event eventLog =
(BS2Event)Marshal.PtrToStructure(curEventLogObjs, structureType);
        Console.WriteLine(Util.GetLogMsg(eventLog));
        curEventLogObjs = (IntPtr)((long)curEventLogObjs + structSize);
    }

    API.BS2_ReleaseObject(outEventLogObjs);
```

```
}
```

[BS2_GetLog](#)
[BS2_ClearLog](#)

From:
<https://kb.supremainc.com/bs2sdk./> - **BioStar 2 Device SDK**

Permanent link:
https://kb.supremainc.com/bs2sdk./doku.php?id=ko:bs2_getfilteredlog

Last update: **2024/02/27 14:08**