

**BS2\_GetAuthGroup** ..... 1

..... 1

..... 1

..... 1

..... 1

..... 2

# BS2\_GetAuthGroup

가 .

```
#include "BS_API.h"

int BS2_GetAuthGroup(void* context, uint32_t deviceId, uint32_t*
authGroupIds, uint32_t authGroupIdCount, BS2AuthGroup** authGroupObj,
uint32_t* numAuthGroup);
```

BS2AuthGroup

- [In] *context* : Context
- [In] *deviceId* :
- [In] *authGroupIds* : 가 auth group id
- [In] *authGroupIdCount* : auth group id
- [Out] *authGroupObj* : auth group
- [Out] *numAuthGroup* : auth group

authGroupObj                      [BS2\\_ReleaseObject](#)

BS\_SDK\_SUCCESS , 가

[BS2\\_GetAllAuthGroup](#)  
[BS2\\_SetAuthGroup](#)

[BS2\\_RemoveAuthGroup](#)  
[BS2\\_RemoveAllAuthGroup](#)

C++

```
int UserControl::getUserBlobFaceAuthGroupID(BS2User& user)
{
    stringstream msg;
    msg << "Please enter a authentication group ID.\n";
    msg << "This is used for face authentication. [0: Not using]";
    uint32_t authGroupID = Utility::getInput<uint32_t>(msg.str());
    user.authGroupID = authGroupID;

    return BS_SDK_SUCCESS;
}

BS2_ReleaseObject(uidObj);
```

C#

```
IntPtr authGroupIDObj = Marshal.AllocHGlobal(4 * authGroupIDList.Count);
IntPtr curAuthGroupIDObj = authGroupIDObj;
foreach (UInt32 item in authGroupIDList)
{
    Marshal.WriteInt32(curAuthGroupIDObj, (Int32)item);
    curAuthGroupIDObj = (IntPtr)((long)curAuthGroupIDObj + 4);
}

Console.WriteLine("Trying to get auth groups from device.");
result = (BS2ErrorCode)API.BS2_GetAuthGroup(sdkContext, deviceID,
authGroupIDObj, (UInt32)authGroupIDList.Count, out authGroupObj, out
numAuthGroup);

Marshal.FreeHGlobal(authGroupIDObj);
BS2_ReleaseObject(uidObj);
```

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

Permanent link:  
[https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2\\_getauthgroup](https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2_getauthgroup)

Last update: **2022/05/29 14:58**