

BS2_EnrollUserSmall

.....

.....

.....

.....

.....

.....

.....

1

1

1

1

1

3

BS2_EnrollUserSmall

[+ 2.6.3]

BS2UserBlob

3

BS2_EnrollUser

BS2UserSmallBlob

BS2UserPhoto가

FaceStation F2

BS2_EnrollUserFaceEx

```
#include "BS_API.h"

int BS2_EnrollUserSmall(void* context, uint32_t deviceId, BS2UserSmallBlob*
userBlob, uint32_t userCount, uint8_t overwrite);
```

BS2UserSmallBlob

- [In] *context* : Context
- [In] *deviceId* :
- [In] *userBlob* :
- [In] *userCount* :
- [In] *overwrite* :

BS_SDK_SUCCESS, 가

C#

```

if (dbHandler.GetUserList(ref deviceInfo, ref userList))
{
    if (userList.Count > )
    {
        for (int idx = ; idx < userList.Count; ++idx)
        {
            Console.WriteLine("[{0:000}] ==> ", idx);
            print(userList[idx]);
        }
        Console.WriteLine("Please, choose the index of the user
which you want to enroll.");
        Console.WriteLine(">>>> ");

        Int32 selection = Util.GetInput();
        if (selection >= )
        {
            if (selection < userList.Count)
            {
                BS2User user = userList[selection];
                BS2UserSmallBlob[] userBlob =
Util.AllocateStructureArray<BS2UserSmallBlob>(1);
                if (dbHandler.GetUserBlob(ref deviceInfo, ref
user, ref userBlob[]))
                {
                    Console.WriteLine("Trying to enroll user.");
                    //BS2ErrorCode result =
(BS2ErrorCode)API.BS2_EnrollUser(sdkContext, deviceID, userBlob, 1, 1);
                    BS2ErrorCode result =
(BS2ErrorCode)API.BS2_EnrollUserSmall(sdkContext, deviceID, userBlob, 1, 1);
                    if (result != BS2ErrorCode.BS_SDK_SUCCESS)
                    {
                        Console.WriteLine("Got error({0}).",
result);
                    }

                    if (userBlob[].cardObjjs != IntPtr.Zero)
                    {
Marshal.FreeHGlobal(userBlob[].cardObjjs);
                    }

                    if (userBlob[].fingerObjjs != IntPtr.Zero)
                    {
Marshal.FreeHGlobal(userBlob[].fingerObjjs);
                    }

                    if (userBlob[].faceObjjs != IntPtr.Zero)
                    {
Marshal.FreeHGlobal(userBlob[].faceObjjs);
                    }

                    if (userBlob[].user_photo_objjs !=

```

```
IntPtr.Zero)
{
    Marshal.FreeHGlobal(userBlob[user_index].user_photo_obj);
}
}
else
{
    Console.WriteLine("Invalid selection[{0}]",
selection);
}
}
else
{
    Console.WriteLine("Invalid user index");
}
}
else
{
    Console.WriteLine("There is no user.");
}
```

[BS2_EnrollUserSmall](#)
[BS2_EnrollUserSmallEx](#)
[BS2_GetUserSmallInfos](#)
[BS2_GetUserSmallInfosEx](#)
[BS2_GetUserSmallDatas](#)
[BS2_GetUserSmallDatasEx](#)

From:

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

Permanent link:

https://kb.supremainc.com/bs2sdk/doku.php?id=ko:bs2_enrollusersmall&rev=1640741957

Last update: **2021/12/29 10:39**