

BS2_EnrollUserSmallEx 1

..... 1

..... 1

..... 1

..... 1

..... 3

BS2_EnrollUserSmallEx

[+ 2.6.3]

[BS2UserBlobEx](#) [BS2_EnrollUserEx](#)
 3 [BS2UserSmallBlobEx](#) [BS2UserPhoto가](#)



```
#include "BS_API.h"

int BS2_EnrollUserSmallEx(void* context, uint32_t deviceId,
BS2UserSmallBlobEx* userBlob, uint32_t userCount, uint8_t overwrite);
```

[BS2UserSmallBlobEx](#)

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

BS_SDK_SUCCESS , 가

C#

```

List<BS2User> userList = new List<BS2User>();
    if (dbHandler.GetUserList(ref deviceInfo, ref userList))
    {
        if (userList.Count > )
        {
            Console.WriteLine("+-----+");
            for (int idx = ; idx < userList.Count; ++idx)
            {
                Console.Write("[{0:000}] ==> ", idx);
                print(userList[idx]);
            }
            Console.WriteLine("+-----+");
            Console.WriteLine("Please, choose the index of the user
which you want to enroll.");
            Console.Write(">>>> ");

            Int32 selection = Util.GetInput();
            if (selection >= )
            {
                if (selection < userList.Count)
                {
                    BS2User user = userList[selection];
                    BS2UserSmallBlobEx[] userBlob =
Util.AllocateStructureArray<BS2UserSmallBlobEx>(1);
                    if (dbHandler.GetUserBlobEx(ref deviceInfo, ref
user, ref userBlob[]))
                    {
                        Console.WriteLine("Trying to enroll user.");
                        //BS2ErrorCode result =
(BS2ErrorCode)API.BS2_EnrolUserEx(sdkContext, deviceID, userBlob, 1, 1);
                        BS2ErrorCode result =
(BS2ErrorCode)API.BS2_EnrollUserSmallEx(sdkContext, deviceID, userBlob, 1,
1);

                        if (result != BS2ErrorCode.BS_SDK_SUCCESS)
                        {
                            Console.WriteLine("Got error({0}).",
result);
                        }

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

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

```

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

        if (userBlob[].user_photo_obj !=
IntPtr.Zero)
        {
Marshal.FreeHGlobal(userBlob[].user_photo_obj);
        }
    }
}
else
{
    Console.WriteLine("Invalid selection[{0}]",
selection);
}
}
else
{
    Console.WriteLine("Invalid user index");
}
}
else
{
    Console.WriteLine("There is no user.");
}
}
else
{
    Console.WriteLine("An error occurred while attempting to
retrieve user list.");
}
}
```

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

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

Permanent link:
https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2_enrollusersmallex&rev=1640745041

Last update: **2021/12/29 11:30**

