

**BS2\_EnrollUserSmall** ..... 1

..... 1

..... 1

..... 1

..... 1

..... 3



```

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

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

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

                                        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.");
}
```

[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\\_enrollusersmall&rev=1640741957](https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2_enrollusersmall&rev=1640741957)

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