

**BS2\_EnrolUser** ..... 1  
..... 1  
..... 1  
..... 1  
..... 3

# BS2\_EnrolUser



```
#include "BS_API.h"

int BS2_EnrolUser(void* context, uint32_t deviceId, BS2UserBlob* userBlob,
uint32_t userCount, uint8_t overwrite);
```

BS2UserBlob

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

BS\_SDK\_SUCCESS , 가

C++

```
sdkResult = BS2_EnrolUser(context_, id, &userBlob, 1, 1);
if (BS_SDK_SUCCESS != sdkResult)
TRACE("BS2_EnrolUser call failed: %d", sdkResult);
```

C#

```
public void insertUserIntoDevice(IntPtr sdkContext, UInt32 deviceID, bool
isMasterDevice)
{
    List<BS2User> userList = new List<BS2User>();
    if (dbHandler.GetUserList(ref deviceInfo, ref userList))
    {
        if (userList.Count > 0)
        {
            for (int idx = 0; idx < userList.Count; ++idx)
            {
                Console.WriteLine("[{0:000}] ==> ", idx);
                print(userList[idx]);
            }

            Int32 selection = Util.GetInput();
            if (selection >= 0)
            {
                if (selection < userList.Count)
                {
                    BS2User user = userList[selection];
                    BS2UserSmallBlob[] userBlob =
Util.AllocateStructureArray<BS2UserSmallBlob>(1);
                    if (dbHandler.GetUserBlob(ref deviceInfo, ref user, ref
userBlob[0]))
                    {
                        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[0].cardObjs != IntPtr.Zero)
                        {
                            Marshal.FreeHGlobal(userBlob[0].cardObjs);
                        }

                        if (userBlob[0].fingerObjs != IntPtr.Zero)
                        {
                            Marshal.FreeHGlobal(userBlob[0].fingerObjs);
                        }

                        if (userBlob[0].faceObjs != IntPtr.Zero)
                        {
                            Marshal.FreeHGlobal(userBlob[0].faceObjs);
                        }
                    }
                }
            }
        }
    }
}
```

```
        if (userBlob[0].user_photo_obj != IntPtr.Zero)
        {
            Marshal.FreeHGlobal(userBlob[0].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\\_GetUserList](#)  
[BS2\\_RemoveUser](#)  
[BS2\\_RemoveAllUser](#)  
[BS2\\_GetUserInfos](#)  
[BS2\\_GetUserInfosEx](#)  
[BS2\\_EnrolUser](#)  
[BS2\\_EnrolUserEx](#)  
[BS2\\_GetUserDatas](#)  
[BS2\\_GetUserDatasEx](#)

From:

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

Permanent link:

[https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2\\_enroluser&rev=1640837545](https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2_enroluser&rev=1640837545)

Last update: **2021/12/30 13:12**