User Management API > BS2_GetUserSmallDatas

BS2_GetUserSmallDatas

[+ 2.6.3]   .
      userMask   .
BS2UserBlob   BS2_GetUserDatas
                  3   BS2UserSmallBlob   BS2UserPhoto
          .

```
#include "BS_API.h"

int BS2_GetUserSmallDatas(void* context, uint32_t deviceId, char* uids,
uint32_t uidCount, BS2UserSmallBlob* userBlob, BS2_USER_MASK userMask);
```

> BS2UserSmallBlob

- [In] *context* : Context
- [In] *deviceId* :
- [In] *uids* :
- [In] *uidCount* :
- [Out] *userBlob* :
- [In] *userMask* :   Mask

0x0000	
0x0001	data
0x0002	
0x0004	
0x0008	
0x0010	PIN
0x0020	
0x0040	
0x0080	
0x0100	
0x0200	
0x0400	
0x0800	(FaceStation F2)

0x1000	(FaceStation F2)
0xFFFF	

BS_SDK_SUCCESS ,

.

C#

```csharp
if (numUserIds > )
{
    IntPtr curUidObjs = outUidObjs;
    BS2UserSmallBlob[] userBlobs = new BS2UserSmallBlob[USER_PAGE_SIZE];

    Console.WriteLine("Number of users : ({0}).", numUserIds);
    for (UInt32 idx = ; idx < numUserIds; )
    {
        UInt32 available = numUserIds - idx;
        if (available > USER_PAGE_SIZE)
        {
            available = USER_PAGE_SIZE;
        }

        result = (BS2ErrorCode)API.BS2_GetUserSmallDatas(sdkContext, deviceID,
curUidObjs, available, userBlobs,
        (UInt32)BS2UserMaskEnum.ALL);
        if (result == BS2ErrorCode.BS_SDK_SUCCESS)
        {
            for (UInt32 loop = ; loop < available; ++loop)
            {
                print(userBlobs[loop].user);
                // don't need to release cardObj, FingerObj, FaceObj because we
get only BS2User
                if (userBlobs[loop].cardObjs != IntPtr.Zero)
                    API.BS2_ReleaseObject(userBlobs[loop].cardObjs);
                if (userBlobs[loop].fingerObjs != IntPtr.Zero)
                    API.BS2_ReleaseObject(userBlobs[loop].fingerObjs);
                if (userBlobs[loop].faceObjs != IntPtr.Zero)
                    API.BS2_ReleaseObject(userBlobs[loop].faceObjs);
                if (userBlobs[loop].user_photo_obj != IntPtr.Zero)
                    API.BS2_ReleaseObject(userBlobs[loop].user_photo_obj);
            }

            idx += available;
            curUidObjs += (int)available * BS2Environment.BS2_USER_ID_SIZE;
```

```
    }
        else
        {
            Console.WriteLine("Got error({0}).", result);
            break;
        }
    }

    API.BS2_ReleaseObject(outUidObjs);
}
```

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