_BS2_GetUserSmallInfosEx_

User Management API > BS2_GetUserSmallInfosEx

BS2_GetUserSmallInfosEx

[+ 2.6.3] .
BS2UserBlobEx BS2_GetUserInfosEx
3 BS2UserSmallBlobEx BS2UserPhoto
.

```
#include "BS_API.h"

int BS2_GetUserSmallInfosEx(void* context, uint32_t deviceId, char* uids,
uint32_t uidCount, BS2UserSmallBlobEx* userBlob);
```

> BS2UserSmallBlobEx

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

BS_SDK_SUCCESS ,
.

C#

```
result = (BS2ErrorCode)API.BS2_GetUserSmallInfosEx(sdkContext, deviceID,
curUidObjs, available, userBlobs);
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;
}
```

BS2_EnrollUserSmall
BS2_EnrollUserSmallEx
BS2_GetUserSmallInfos
BS2_GetUserSmallInfosEx
BS2_GetUserSmallDatas
BS2_GetUserSmallDatasEx