

---

**(주)우아한형제들**

서울시 송파구 위례성대로 2 장은빌딩 2층

TEL 1600-7001 FAX 050-6050-0400

# 배달의민족 주문정보 API

Client 기반 주문정보 연동 매뉴얼

## 변경 이력

버전	변경일자	변경 내용	작성자
1.0	2019.03.05	최초 작성	강인규
1.0.1	2019.05.21	IsBaeminInstalled, IsBaeminRunning 함수 추가	강인규
1.1.0	2019.08.01	TOnNewDeliveryFunc 고객 요청사항 추가, UpdateDeliveryStatus 함수 추가	강인규

## 개요

서버를 통하지 않고 동일한 PC에 설치된 배달의민족 PC접수 프로그램에서 접수한 주문정보를 중계하여 배달정보를 생성할 수 있도록 주소, 연락처와 같은 기본정보를 제공합니다. 주문정보가 중계되는 시점은 주문을 접수한 시점이며 취소한 주문은 전달하지 않습니다. 중계된 주문을 수신하기 위해서는 BMOOrderRelay.dll과 dll의 함수부를 구현한 샘플 프로그램, 그리고 우아한형제들이 배달대행 업체에 제공한 인증키를 사용하여 주문정보를 연동할 수 있습니다.

## Functions

### IsBaeminInstalled

```
bool IsBaeminInstalled(void);
```

배달의민족 PC접수가 설치되어 있는지 확인하기 위한 함수. 단순 프로그램 설치 여부만 검사합니다. 만일 설치되어 있다면 True를 아니면 False를 리턴합니다. 설치파일은 사장님사이트에서 다운로드 받을 수 있습니다.(<https://ceo.baemin.com/#/about/baro-order>)

### IsBaeminRunning

```
bool IsBaeminRunning(void);
```

배달의민족 PC접수가 실행되어 있는지 확인하는 함수. 실행되어 있다면 True를 아니면 False를 리턴합니다.

---

## RegisterNewDeliveryFunction

```
bool RegisterNewDeliveryFunction(TOnNewDeliveryFunc AEvent);
```

[TOnNewDeliveryFunc](#)형의 Callback Function을 등록하기 위한 함수. 신규 주문이 접수되면 이 Callback 함수를 통해 주문정보를 받을 수 있습니다. 클라이언트에 주문을 정상적으로 등록한 경우 True를 그렇지 않은 경우 False를 리턴하면 됩니다.

### Pseudo Code

```
TOnNewDeliveryFunc gNDProc = this.OnNewDelivery;  
bool Res = RegisterNewDeliveryFunction(gNDProc);  
if (Res == true) {  
    ShowMessage("Register New Delivery Function Succeeded!");  
}
```

## RegisterStatusChangedFunction

```
bool RegisterStatusChangedFunction(TOnStatusChangedProc AEvent);
```

[TOnStatusChangedProc](#)형의 Callback Function을 등록하기 위한 함수. 주문의 상태가 변경되면 이를 통해 변경된 주문상태를 받을 수 있습니다.

### Pseudo Code

```
TOnStatusChangedProc gSCProc = this.OnStatusChanged;  
bool Res = RegisterStatusChangedFunction(gSCProc);  
if (Res == true) {  
    ShowMessage("Register StatusChanged Function Succeeded!");  
}
```

## RegisterDisconnectedFunction

```
bool RegisterDisconnectedFunction(TOnDisconnectedProc AEvent);
```

[TOnDisconnectedProc](#)형의 Callback Function을 등록하기 위한 함수. PC접수가 종료되거나 시스템의 문제로 연결이 해제되었을 때 발생합니다.

### Pseudo Code

```
TOnDisconnectedProc gDCProc = this.OnDisconnected;
bool Res = RegisterDisconnectedFunction(gDCProc);
if (Res == true) {
    ShowMessage("Register Disconnected Function Succeeded!");
}
```

## InitializeService

```
int InitializeService(WCHAR[] ASignKey);
```

우아한형제들에서 배포한 인증키를 사용하여 인증을 시작합니다. 성공할 경우 S\_OK(0)를 리턴하며 인증과 접속이 모두 정상적으로 이루어졌음을 의미하며 그 외의 코드는 모두 오류입니다. 각각의 상황에 따른 오류메시지는 아래와 같습니다.

Const Name	Value	Description
S_OK	0	정상
ERR_CONNECTION_FAILED	-1	서버에 접속을 실패했습니다
ERR_UNKNOWN_HOST	500	잘못된 요청입니다
ERR_UNAUTHENTICATED	401	인증에 실패했습니다. 잘못된 인증 키 입니다.

## Pseudo Code

```
WCHAR[] SignKey = "@DeliveryRelationTestKey+";
int Res = InitializeService(SignKey);
if (Res == S_OK) {
    ShowMessage("Initialize Service Function Succeeded!");
}
```

## FinalizeService

```
int FinalizeService();
```

서비스를 더이상 사용하고 싶지 않을 때 사용합니다. 또는 갑자기 접속이 단절된 경우 다시 접속을 시도하기 전 안전을 위해 호출합니다. 연결했던 Callback함수의 연결을 모두 해제하고 서버와 접속을 종료합니다. 해당 함수를 호출할 때는 모든 Callback함수를 해제한 다음 접속을 종료하기 때문에 접속이 종료될 때 TOnDisconnected Callback 함수가 호출되지 않습니다.

Const Name	Value	Description
S_OK	0	정상

ERR_CONNECTION_FAILED	-1	연결을 해제하는 동안 에러가 발생했습니다.
-----------------------	----	-------------------------

#### Pseudo Code

```
int Res = FinalizeService();
if (Res == S_OK) {
    ShowMessage("Finalize Service Function Succeeded!");
}
```

### SetDeliveryCompleted

```
bool SetDeliveryCompleted(WCHAR[] AOrderNo);
```

해당 함수의 호출을 PC접수에서 정상적으로 수신한 경우 PC접수는 주문상태를 완료로 수정하며 이 결과는 [RegisterStatusChangedFunction](#)에서 등록한 [TOnStatusChangedProc](#)으로 주문완료 주문을 수신할 수 있습니다.

#### Pseudo Code

```
WCHAR[] LOrderNo = listView1.Items[listView1.SelectedIndices[0]].Text;
bool Res = SetDeliveryCompleted(LOrderNo);
if (Res == true) {
    ShowMessage("Set Delivery Completed Function Succeeded!");
}
```

### UpdateDeliveryStatus

```
bool UpdateDeliveryStatus(WCHAR[] AOrderNo, int ADeliveryStatus, WCHAR[]
ARiderKey, WCHAR[] ARiderName, ETA: Integer);
```

해당 함수를 호출하여 라이더의 정보를 업데이트 합니다. 업데이트가 가능한 상태는 세가지이며 ADeliveryStatusdp 아래의 세가지 상태와 가장 비슷한 상황에 정보를 업데이트 하면 됩니다. ARiderKey, ARiderName은 업데이트가 필요하지 않은 경우 빈문자열로 대체할 수 있습니다.

ARiderKey는 라이더를 특정할 수 있는 중복되지 않는 키값을 사용하면 됩니다. 예를 들어 배달대행 업체 자체에서 부여하는 일련번호나 라이더의 전화번호 혹은 전화번호의 해시값 등 라이더를 특정할 수 있는 업체기준 중복되지 않는 값이면 됩니다. ARiderName은 라이더의 이름 혹은 별칭을 사용하면 됩니다. 의무사항은 아니며 사용하지 않을 경우 빈문자열을 사용하면 됩니다.

ETA(Estimated time of arrival)의 단위는 분이며 고객에게 도착하기까지 소요될 예상시간을 사용하면 됩니다. 만일 예상시간을 사용하지 않는다면 0 혹은 0보다 작은 수를 입력해주시기 바랍니다.

- 해당함수는 2018.08.13 배포 버전에서 기능적으로 동작하지 않습니다. 배포가 완료되면 이 공지는 문서에서 삭제됩니다.

Const Name	Value	Description
DS_RIDER_ASSIGNED	0	라이더가 배정됨
DS_PICKUP_COMPLETED	1	음식을 매장에서 픽업함
DS_DELIVERY_COMPLETED	2	전달이 완료됨(SetDeliveryCompleted 대체)

RS\_DELIVERY\_COMPLETED를 전송할 경우 **SetDeliveryCompleted**함수는 사용하지 않아도 완료처리됩니다.

#### Pseudo Code

```
WCHAR[] LOrderNo = listView1.Items[listView1.SelectedIndices[0]].Text;
WCHAR[] LRiderKey = "01012341234";
WCHAR[] LRiderName= "아무개";

bool Res = UpdateDeliveryStatus(LOrderNo, DS_RIDER_ASSIGNED, LRiderKey,
LRiderName, 0);
if (Res == true) {
    ShowMessage("Rider Assign Succeeded!");
}
```

## Callback Functions

### TOnNewDeliveryFunc

```
boolean TOnNewDeliveryFunc(WCHAR[] AOrderNo, WCHAR[] ARoadNameAddress,
WCHAR[] AAddress, WCHAR[] AAddressDetail, WCHAR[] APhoneNo, WCHAR[]
ALatitude, WCHAR[] ALongitude, WCHAR[] ATitle, int AQuantity, int AAmount,
int APaymentType, WCHAR[] AMemo);
```

접수한 새로운 주문이 있을 때 해당 함수가 호출됩니다. 해당 함수는 Thread상에서 호출되는 함수로 UI에 적용할 코드가 있을 때는 반드시 Invoke, TThread.Synchronize와 같이 호출되는 함수를 MainThread에서 동작하도록 강제하는 코드를 사용하는 것을 추천합니다. 연동한 프로그램에서 배달주문을 생성하였다면 True를 주문을 생성하지 못했다면 False를 리턴하면 됩니다. 지번주소, 주소 상세 정보, GPS 위경도는 필수 항목이며 도로명 주소는 없을 수 있는데 비율은 대략 0.0005% 정도입니다.

#### Parameter

Field Name	Type	Description
AOrderNo	WCHAR[]	배달의민족 에서 제공하는 주문번호
ARoadNameAddress	WCHAR[]	도로명 주소
AAddress	WCHAR[]	지번 주소
AAddressDetail	WCHAR[]	주소 상세 정보
APhoneNo	WCHAR[]	주문자 연락처
ALatitude	WCHAR[]	GPS 위도
ALongitude	WCHAR[]	GPS 경도
ATitle	WCHAR[]	주문메뉴의 첫번째 메뉴 명
AQuantity	int	총 주문한 메뉴의 합
AAmount	int	총 주문 금액
APaymentType	int	결제 유형 PT_PREPARED = 0 //이미 결제함(바로결제) PT_MEET_CARD = 1 //카드로 결제 예정 PT_MEET_CASH = 2 //현금으로 결제 예정
AMemo		요청사항

## TOnStatusChangedProc

```
void TOnStatusChangedProc(WCHAR[] AOrderNo, int AOrderStatus);
```

AOrderStatus의 종류는 아래와 같습니다만 실제로 받을 수 있는 데이터는 완료(OS\_COMPLETED) 뿐 입니다. 해당 함수는 Thread상에서 호출되는 함수로 UI에 적용할 코드가 있을 때는 반드시 Invoke, TThread.Synchronize와 같이 호출되는 함수를 MainThread에서 동작하도록 강제하는 코드를 사용하는 것을 추천합니다.

Const Name	Value	Description
OS_NEW	0	새로운 주문이 발생
OS_RECEIPT	1	주문을 접수함
OS_COMPLETED	2	주문이 완료됨
OS_CANCELED	3	주문이 취소됨

## TOnDisconnectedProc

---

```
void TOnDisconnectedProc();
```

해당 이벤트가 발생하면 FinalizeService를 호출하여 모든 설정을 초기화 한 다음 초기화 과정을 다시 수행하는 것을 권장합니다. 해당 함수는 Thread상에서 호출되는 함수로 UI에 적용할 코드가 있을 때는 반드시 Invoke, TThread.Synchronize와 같이 호출되는 함수를 MainThread에서 동작하도록 강제하는 코드를 사용하는 것을 추천합니다.

## 적용순서

### Register Callback Functions

세개의 Callback Function을 등록합니다. 등록하는 순서는 영향이 없습니다. Callback 함수는 Class의 멤버함수이거나 static함수로 선언된 함수를 사용해야 합니다.

### InitializeService

제공받은 인증키를 사용하여 서비스를 초기화 합니다. 주문이 정상적으로 발생하는지 주문이 완료되었을 때 함수가 호출되는지 PC접수 및 Test 프로그램이 종료될 경우 TOnDisconnected 함수가 호출되는지 확인합니다.

### FinalizeService

FinalizeService를 호출하여 PC접수 프로그램과 접속이 정상적으로 종료되는지 확인합니다.