

# Entegrasyon Süreci

## Entegrasyon Sistemi Nedir?

Entegrasyon Sistemi, Yemeksepeti.com sistemi ile entegre bir şekilde çalışan, Yemeksepeti.com Postacı yazılımı içerisinde yer alan bir iletişim metodudur.

Restoranlar için sipariş iletimini sağlamak amacıyla online entegrasyon yöntemi kullanılmaktadır.

## Entegrasyon Sistemi Yapısı

Entegrasyon sistemi bir bütün olarak webservisler aracılığı ile sağlanmaktadır. Yemeksepeti tarafında bulunan webservisler ile restoran otomasyon programı arasında yapılan iletişim ile sipariş süreci boyunca siparişin iletimi, siparişin restoran tarafından alındı bilgisi, siparişin durumu ( sipariş reddedildi, sipariş müşteriye ulaştı vb) yemeksepetine ulaşır.

Restoranlarda sipariş sürecinin kolay olması açısından ys gateway yazılarak kurulan entegrasyon sistemi ile sipariş geldiği anda restorana kurulmuş olan printer aracılığı ile sipariş fişi olarak çıktı alınarak ya da restorana ait bilgisayar üzerinden görüntülenmesi sağlanıyor.

Siparişin yazıcı aracılığı ile alınması için;

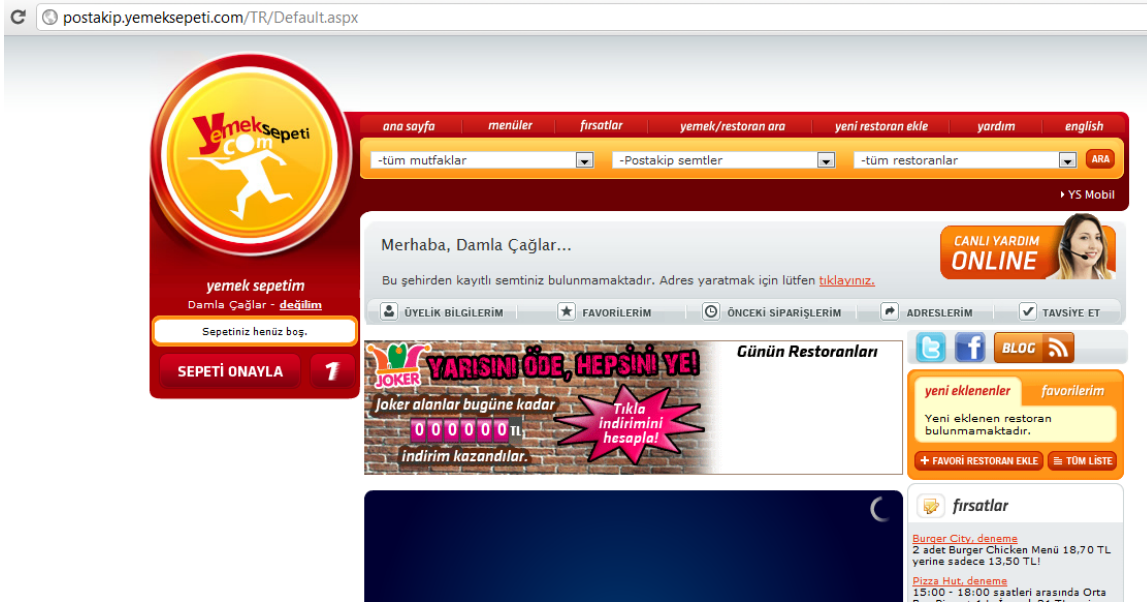
Her sipariş için bir yazıcı portu seçilir bunun amacı sipariş geldiğinde sipariş fişi olarak yazıcıya gönderilecek olmasıdır.

## Restoran otomasyon sistemi yapan firmalar tarafından yapılması gerekenler

### a. Test Aşaması

1. Otomasyon Firması için entegrasyon sistemi kullanılacağı zaman öncesinde postakip'te menü ekibi tarafından bir restoran yaratılır ve restoran menüsü hazırlanır. Restoran bu ortamda deneme siparişleri vererek testlerini yapar.

- Test yapılacak url : <http://postakip.yemeksepeti.com/default.aspx>



2. Testler tamamlandıktan sonra bu sistem canlıya alınır. Canlıya alındıktan sonra entegrasyon yapılır. Canlıda otomasyon firması, yemeksepeti tarafından verilen Kullanıcı adı/şifresini entegre modülümüze girer. Otomasyon Firması artık sistemi kullanabilir durumdadır.

## b. Menü Eşleştirme

3. İlk önce, Yemeksepeti tarafında oluşturulan kayıtlı ürünler otomasyon firması tarafından çekilir. Bu ürünlere ait id'ler ile aynı ürünler için restoran tarafında bulunan id'lerin eşleştirilmesi sağlanır.

- Örneğin; Yemeksepeti'nde id'si 15 olan bir ürünün, restoran tarafındaki id'sinin 220 olması ve bu iki id'nin eşleştirilmesi.

## c. Sipariş Alma

4. Sipariş Yemeksepetinde yaratıldıktan sonra, restorana gönderilir. Siparişin yaratılması bir message oluşması anlamına gelir. Entegre programı sürekli çalışır duruma getirilerek 10 sn'de 1 sipariş var mı görülebilir. Sipariş olması durumunda bu sipariş fişi olarak yazılır.

5. Eğer sipariş restoran tarafına ulaşmışsa, sipariş ulaştı bilgisi Yemeksepeti'ne ulaşır. Sipariş restorana ulaşmadığı sürece Yemeksepeti tarafında aynı mesaj gösterilecektir.

6. Otomasyon firması siparişi kendi sistemine uygun hale getirerek sisteme ekler ve bir problem olmadığı takdirde Yemeksepeti'ne onaylandığını bildirir.

- Siparis durumu için;
  - Accepted(Onaylandı),
  - Rejected(Reddedildi),
  - Cancelled,(İptal Edildi),
  - OnDelivery(Sipariş Yolda)
  - Delivered(Sipariş Teslim Edildi)

gibi statüler bulunmaktadır.

7. Rejected veya cancelled durumunda restoran tarafından bir not girilerek Yemeksepeti'ne reddetme ya da iptal etme sebepleri iletilir.

8. Sipariş restorana ulaştıktan sonra adisyon paket servis elemanına verilir ve sipariş teslim edilip paket servis elemanı döndüğünde bu adisyonu aldığı ödeme şekline göre kapatır.

9. Siparişin akıbeti ile ilgili statü bu adımda Delivered (Sipariş Teslim Edildi) olarak girilebilir.

10. Herhangi bir sorun olması durumunda, Yemeksepeti kullanıcı ile yapılan görüşmelerin ardından restorana bilgi geçer. Siparişin akıbeti belli olur.

## d. Sipariş Güncelleme

4. Sipariş ürünlerinde güncelleme olması durumunda Operasyon departmanı bu durumu restorana ya da kullanıcıya manuel olarak iletir ve Yemeksepeti üzerinde siparişi günceller.

5. Eğer restoran güncel siparişi sistem üzerinde görmek ister ise sipariş restorana tekrar gönderilir. Bu noktada entegrasyon sistemi gelen siparişin içerisindeki benzersiz sipariş numarasına (TrackingNumber) göre kontrol yapmalı ve yeni bir sipariş olarak işlemek yerine daha önce gelmiş siparişi güncellemeli.

Özetlersek;

- 1) Otomasyon firması guncel menuyu alır, kendi sistemiyle eslestirir.
- 2) Belirli aralıklara sipariş var mı yok mu kontrol eder.
- 3) Sipariş olduğu takdirde, siparis mesajini alarak kendi sistemine girer ve mesajin listeden silinmesini saglar.
- 4) Yemeksepeti'ne siparis durumuyla ilgili guncellemeyi yapar.

1. Menüde bir deęişiklik yapılacağı zaman restoran Yemeksepeti Menü Departmanı'na ulaşır.
2. Menü departmanında gerekli deęişiklikler yapıldıktan sonra menü "bunulanır".
3. Otomasyon firmasına menüde deęişiklik olduęunun bilgisi gider.
4. Böylece otomasyon firması da deęişiklikleri kendi tarafı ile eşleştirir.

Notlar :

- Restoranda uygulanacak fiyat deęişikliklerinin Yemeksepeti'ne entegre edilmesini sağlamak amaçlı xml ürün eşleştirmelerini çekmek için kullanılan yöntemin tersten de çalışacak şekilde kurgulanması gerekebilecektir.
- Restoranda birden fazla fiyat uygulanıyorsa bunlar Yemeksepeti ile paylaşılmalıdır. Restoranlar her ürüne restoran fiyatı, gel al servis fiyatı , paket servis fiyatı gibi uygulamalar yaptıęından bu listenin Yemeksepeti'ne ulaştırılması ile güncellemeler daha hızlı ve kolay olacaktır.

# Yemeksepeti Entegrasyon Webservisleri

## Yemeksepeti Integration Web Service

<http://messaging.yemeksepeti.com/messagingwebservice/integration.asmx>

Integration Web Service uses Soap Headers to validate and authenticate clients. The name of the Authentication object is AuthHeader.

AuthHeader:

```
public string UserName;  
public string Password;
```

## Web Methods

GetMenu():

GetMenu is used in order to get authenticated restaurant's menu. It takes no parameters and returns DataSet. Returns null if authentication failed.

The Menu DataSet name is "Menu" and it contains two tables: Products and Options

Products Table:

Column Name	Description
Id	Product's Identifier
Name	Product's Name
Price	Unit price for the product
Title	Product's submenu title
Description	Product's description

Options Table:

Column Name	Description
Id	Option's Identifier
ProductId	Parent product's Identifier
Name	Option's Name
Price	Additional price for the option
Type	Type of the option

GetPaymentTypes()

GetPaymentTypes is used in order to get Yemeksepeti Payment Methods. Takes no parameters and returns DataSet. Returns null if authentication failed.

The PaymentTypes DataSet contains only one table: PaymentMethods

Products Table:

Column Name	Description
-------------	-------------

Id	Payment Type's identifier
Name	Payment Type's name
Description	Payment Type's description

### GetRestaurantList():

GetRestaurantList is used in order to get the restaurant list that restaurant chain contains. Takes no parameters and returns DataSet. Returns null if authentication failed.

The RestaurantList DataSet contains only one table: Restaurants

Products Table:

Column Name	Description
CatalogName	Restaurant's catalog name.
CategoryName	Restaurant's category name.
DisplayName	Restaurant's display name.

### GetMessage():

GetMessage is used in order to get authenticated restaurant's message. It takes no parameters and returns an XML String. Returns null if there is no new message or authentication failed.

Message string has three nodes. Order, products and options.

Orders:

Attribute Name	Description
Id	Yemeksepeti order identifier.
MessageId	Message's identifier.
RestaurantName	Restaurant's display name.
RestaurantCatalog	Catalog Name of the restaurant.
RestaurantCategory	Category Name of the restaurant.
CustomerName	Customer's full name.
CustomerId	Yemeksepeti customer identifier.
CustomerType	Customer's type. If 1 it is a new customer.
PaymentNote	Additional note for the payment.
OrderTotal	Order's total.
CustomerPhone	Customer's phone number.
City	Customer's city.
Region	Customer's region name.
Organization	Customer's organization.
Address	Customer's address.

AddressDescription	Customer's address' description.
AddressId	Address identifier
PaymentMethodId	Order's payment type identifier.
DeliveryTime	Order's delivery time.
ChangeInTotal	If there's a discount or some additional fees, this will be reflected in here.
Currency	Order's preferred currency.

#### Products:

Attribute Name	Description
Id	Yemeksepeti product identifier.
Name	Product's name.
Price	Product's unit price.
Quantity	Product's quantity.
Options	Product's additional options.
OptionIds	Product's additional option identifiers.
OrderIndex	Product's index on order.
ParentIndex	If product has a parent on this order, this attribute shows the index of the main product.
PromoParentIndex	If product is a promotion for another product, this attribute shows the index of the main product.
ProductOptionId	If product is an option for another product, it shows the option identifier.

#### Options:

Attribute Name	Description
Id	Yemeksepeti option identifier.
Name	Option's name.
ProductOrderId	Product's index on order.

### GetMessageV2():

GetMessageV2 is used in order to get authenticated restaurant's message. It takes no parameters and returns an XML String. Returns null if there is no new message or authentication failed.

Message string has four nodes. Order, products, options and promotions.

#### Orders:

Attribute Name	Description
----------------	-------------

Id	Yemeksepeti order identifier.
MessageId	Message's identifier.
RestaurantName	Restaurant's display name.
RestaurantCatalog	Catalog Name of the restaurant.
RestaurantCategory	Category Name of the restaurant.
CustomerName	Customer's full name.
CustomerId	Yemeksepeti customer identifier.
CustomerType	Customer's type. If 1 it is a new customer.
PaymentNote	Additional note for the payment.
OrderTotal	Order's total.
CustomerPhone	Customer's phone number.
City	Customer's city.
Region	Customer's region name.
Organization	Customer's organization.
Address	Customer's address.
AddressDescription	Customer's address' description.
AddressId	Address identifier
PaymentMethodId	Order's payment type identifier.
DeliveryTime	Order's delivery time.
ChangeInTotal	If there's a discount or some additional fees, this will be reflected in here.
Currency	Order's preferred currency.

#### Products:

Attribute Name	Description
Id	Yemeksepeti product identifier.
Name	Product's name.
Price	Product's unit price.
Quantity	Product's quantity.
Options	Product's additional options.
OptionIds	Product's additional option identifiers.
OrderIndex	Product's index on order.
ParentIndex	If product has a parent on this order, this attribute shows the index of the main product.
PromoParentIndex	If product is a promotion for another product, this attribute shows the index of the main product.

ProductOptionId	If product is an option for another product, it shows the option identifier.
-----------------	--

Options:

Attribute Name	Description
Id	Yemeksepeti option identifier.
Name	Option's name.
ProductOrderId	Product's index on order.

Promotions:

Attribute Name	Description
Id	Yemeksepeti promotion identifier.
DefinitionName	Promotion's definition name.
ProductOrderId	Product's index on order.

### GetTopMessages(int count):

GetTopMessages is used in order to get authenticated restaurant's top x messages in a row. It takes count as integer parameter and returns an XML String. Returns null if there is no new message or authentication failed.

TopMessages string has four nodes. Messages, Order, products and options.

This method returns order messages enclosed in <messages>.

### GetTopMessagesV2(int count):

GetTopMessagesV2 is used in order to get authenticated restaurant's top x messages in a row. It takes count as integer parameter and returns an XML String. Returns null if there is no new message or authentication failed.

TopMessages string has five nodes. Messages, Order, products, options and promotions. The difference between GetTopMessages is V2 returns promotions as well.

This method returns order messages enclosed in <messages>.

### GetAllMessages():

GetAllMessages is used in order to get authenticated restaurant's all messages in a row. It takes no parameters and returns an XML String. Returns null if there is no new message or authentication failed.

AllMessages string has four nodes. Messages, Order, products and options.

This method returns order messages enclosed in <messages>.

### GetAllMessagesV2():

GetAllMessages is used in order to get authenticated restaurant's all messages in a row. It takes no parameters and returns an XML String. Returns null if there is no new message or authentication failed.

AllMessages string has three nodes. Messages, Order, products, options and promotions. The difference between GetAllMessages is V2 returns promotions as well.



This method returns order messages enclosed in <messages>.

### GetRestaurantPointsAndComments:

GetRestaurantPointsAndComments is used in order to get authenticated restaurant's order points and comments by restaurant branch between two dates.

This method returns DataTable as result.

Parameter Name	Type	Description
catalogName	string	Restaurant's catalog name.
categoryName	string	Restaurant's category name.
startDate	datetime	Start date
endDate	datetime	End date

### GetRestaurantPromotions:

GetRestaurantPromotions is used in order to get authenticated restaurant's promotions. It takes no parameters.

This method returns DataSet as result.

### GetTopMessages:

GetTopMessages is like GetAllMessages and it is used in order to get authenticated restaurant's top desired number of messages instead of all messages in a row. It takes an integer parameter and returns an XML String. Returns null if there is no new message or authentication failed.

TopMessages string has three nodes. Messages, Order and products.

This method returns order messages enclosed in <messages>.

Parameter Name	Type	Description
count	Int	Number of returned messages

### MessageSuccessful:

When a client gets a message, it has to call MessageSuccessful Method in order to mark this message as read. It takes Message Id as a parameter and does not returns a value.

Parameter Name	Type	Description
messageId	Int	Order's message identifier.

### UpdateOrder:

When a client processed an order, it have to give feedback to the yemeksepeti.com by using UpdateOrder method. UpdateOrder method takes OrderId, OrderState and reason as parameters and returns a string. If process is completed succesfully it returns "OK". If there is an error, it returns the error description as string.

OrderState is an enum. It's values are Accepted or Rejected or Cancelled or OnDelivery or Delivered

Parameter Name	Type	Description
orderId	Decimal	Order's identifier.
orderState	OrderState	Order's current state.
reason	string	If order is cancelled or rejected, you may specify a reason.

### SetRestaurantForOrder:

If an order belongs to a restaurant branch other than intended one, you can inform yemeksepeti.com about this change by this method. SetRestaurantForOrder method takes orderId and restaurantCategory as parameters and returns true or false.

Parameter Name	Type	Description
orderId	Decimal	Order's identifier.
restaurantCategory	string	Redirected restaurant's category name.

### UpdateRestaurantState:

UpdateRestaurantState is used in order to change the Status for the selected restaurant.

RestaurantState is an enum. It's values are Open or Closed or Huge Demand

Parameter Name	Type	Description
catalogName	string	Restaurant's catalog name.
categoryName	string	Restaurant's category name.
restaurantState	RestaurantState	Restaurant's status.

### SetRestaurantServiceTime:

SetRestaurantServiceTime is used in order to change the Service Time for the selected restaurant.

Parameter Name	Type	Description
catalogName	string	Restaurant's catalog name.
categoryName	string	Restaurant's category name.
ServiceTime*	int	Restaurant's new service time.

\*: Service time can't be more than 75 minutes and less than 30 minutes.

### EXAMPLES:

In order to use "Yemeksepeti Integration Web Service" you should add this web service as a web reference for the project.

#### GetMessage: (C#)

```
IntegrationWebService.Integration iws = new IntegrationTest.IntegrationWebService.Integration();
```

```
iws.AuthHeaderValue = new IntegrationTest.IntegrationWebService.AuthHeader();
```

```
iws.AuthHeaderValue.UserName = "Your UserName";
```

```
iws.AuthHeaderValue.Password = "Your Password";
```

```
string message = iws.GetMessage();
```

# Sık Sorulan Sorular

## Sık Sorulan Sorular Ve Cevapları

1) Yemeksepeti tarafından otomasyon firmamıza verilmiş olan kullanıcı adı ve şifre ile sisteme giriş yapılamıyor.

Çözüm : Yemeksepeti tarafından gönderilen kullanıcı adı ve şifre otomasyon firması tarafından doğru giriliyor olmalıdır.

2) Menüü tarafımızda güncellediğimiz halde site üzerinde güncel menüyü göremiyoruz.

Çözüm : Yemeksepetinde menü ekibi güncelleme işlemini yapmadan restoran güncelleme işlemini gerçekleştirdiği için eski menüsü ile devam ediyor. Menü güncelleme işlemi, Yemeksepeti ile paralel olarak yapılmalıdır. Yemeksepeti tarafından menü güncellendikten sonra restoran da kendi tarafında güncelleme yapıyor olmalıdır. Böylece menü doğru haliyle görüntülenecektir.

3) Eşleşme tarafımızca yapıldı fakat menü yine de olması gerektiği gibi görüntülenmiyor.

Çözüm : Yemeksepeti'nde menü ekibi menüyü göndermesine rağmen ürün eşleşmediği veya yanlış eşleştiği için menü olması gerektiği gibi görüntülenmiyor olabilir. Yemeksepeti menü ekibi tarafından gönderilen ürün id'leri ile restoran tarafındaki ürün id'lerinin eşleştirmesinin doğru yapılması gerekmektedir.

4) Yemeksepeti'nde olan fakat Restoran tarafında olmayan bir ürün var. Bu da sipariş esnasında sorun yaratıyor. Ne yapmalıyız?

Çözüm : Bu eşleştirmenin eksik/yanlış yapılmasından kaynaklanıyor olabilir. Oluşturulmuş olan yedek stok kartları yardımıyla siparişi verilmiş olan ürünün ne olduğu belirlenebilir. Yapılması gereken, Yemeksepeti tarafından seçilmiş olan stok kartından satış yapılmış gibi işlenmeli, daha sonra restoran bu kartı sipariş ekranından değiştirmelidir. Açıklama olarak da yemeksepeti tarafından gönderilen stok adı yazılmalıdır.

# What is Integration System

## What is Integration System?

Integration system is a transmission method inside Yemeksepeti.com Order Delivery software which works integrated with Yemeksepeti.com system.

Online integration method is used in order to provide order transmission for restaurants.

## Integration System Structure

Integration system, as a whole, is provided through webservises. By using webservises provided by Yemeksepeti.com, 3<sup>rd</sup> party companies are able to get restaurant's menu, get restaurants in restaurant chain, set restaurants' availability statuses, set restaurants' order delivery times, get orders and set order statuses.

## Actions to be taken by firms that create restaurant automation system

### a. Test Process

1. First of all, restaurants and menus are created on Yemeksepeti.com (the url will be provided when the process is completed) and the automation firm will be provided with a username and password pair for testing purposes.

- Test url : <http://postakip.yemeksepeti.com>



1. After the tests are completed successfully, the system goes live. Automation firm will be provided another username and password pair for the live system. Test accounts will still be intact for future tests.

For Reaching Yemeksepeti Integration Webservice Document You Can Click The Link Below.

[Yemeksepeti Entegrasyon Webservisleri](#)

### b. Menu Matching

3. First of all, the automation firm gets the menu stored on Yemeksepeti.com system. Since it is not possible for us to duplicate the exact menu provided by 3<sup>rd</sup> party companies due to our menu standards, the automation firm should create a product matching gateway for integration.

- - Example; A product with an ID of 15 on Yemeksepeti and a product on restaurant with an ID of 220 and matching of these two ID's.

### c. Receiving Orders

4. For every order placed on Yemeksepeti.com, an order message is formed for integration. The automation firm should check continuously for new messages in order to see these order messages via integration webservice.
5. After the message is received, the automation firm should call another method with the message id to inform Yemeksepeti.com that the message is received. The automation firm will see the same message if that method is not called.
6. Automation firm adds the order to the system by customizing the order according to their own system and notifies Yemeksepeti about approval if there is no problem.

- - For order status, following statuses do exist;
    - Approved,
    - Rejected,
    - Cancelled,
    - OnDelivery
    - Delivered

7. For orders rejected or cancelled, the reason for rejection / cancellation is transmitted to Yemeksepeti by restaurant by entering a note.
8. After the order reaches to the restaurant the slip is handed to the driver and after the order is delivered and driver has returned the slip is closed according to the payment method received.
9. On this step the order status can be chosen as Delivered.
10. In case of any problem Yemeksepeti informs the restaurant after communicating with user. The outcome of the order becomes clear.

#### In Summary;

- 1) Automation firm receives the updated menu and matches with their own system.
- 2) Checks periodically if there is an order not.
- 3) If there is an order they take the order message and input on their system and assure that the message is deleted from list.
- 4) Updates Yemeksepeti regarding order status.

#### Menu Update Case

1. When there will be an update on the menu the restaurant reaches Yemeksepeti Menu Department.
2. After menu department completes required updates, a simplified version of the menu is created on the integration database.
3. Information of menu change is sent to automation firm.
4. Hence automation firm matches the updates with their side.

#### Notes:

- Method to retrieve xml product matching for providing price changes that will be applied in restaurant to be integrated with Yemeksepeti might be needed to build in a way to operate reversely.
- If more than one price is applied on Restaurant, these should be shared with Yemeksepeti. Since the restaurants have applications for each product like dine in price, take away price, delivery price the updates will be faster and easier if these lists are shared with Yemeksepeti.

### Frequently Asked Questions & Answers

- 1) Cannot login to the system with user name and password provided by Yemeksepeti for our automation firm.  
Solution : The user name and password provided by Yemeksepeti should be entered correctly by automation firm.
- 2) We are unable to see the updated menu on website even if we have updated the menu on our side.  
Solution : The old menu is present since the restaurant practised the update process before menu crew at Yemeksepeti did. Menu update process should be done in accordance with Yemeksepeti. The restaurant should be updating on their side after yemeksepeti updates the menu. This way menu will be displayed in correct format.
- 3) Matching process has been done by us but the menu is still not displayed as it should.  
Solution : It might be possible that the menu is not displayed as it should even Yemeksepeti menu crew sent the menu because the product might be mis-matched or not matched. The matching of ID's sent by Yemeksepeti menu crew and ID's on restaurant side should be done correctly.

4) There is a product present at Yemeksepeti but not on restaurant site. This is causing a problem during order. What should we do?

Solution : This might be arise from matching has been done missing / wrong. It might be possible to identify the product by the help of created substitute stock cards. Course of action is to process the order like it's been sold through the stock card chosen by Yemeksepeti and after that restaurant should change this card on order screen. For explanation the stock name sent by Yemeksepeti should be written.