

# Egyedi fejlesztésű webshop összekötése a MuFis rendszerrel - API készítés útmutató

Az útmutatóval kapcsolatos kérdéseit és javaslatait ide küldheti: [it1@mufis.sk](mailto:it1@mufis.sk)

## Előszó

A MuFis információs rendszer számlázás, könyvelés, illetve fulfillment szolgáltatás céljából is összekapcsolható különböző webshop motorokat használó webshopokkal.

Pár konkrét ilyen webshop motor (a teljesség igénye nélkül):

- *unas*
- *shoprenter*
- *woocommerce (wordpress)*
- *shopify*

Az összekötés lényege, hogy a webshopba beérkező megrendelések bekerüljenek a MuFis rendszerbe, ahol aztán számla készülhet a vásárlónak, fulfillment szolgáltatás esetén pedig megkezdődhessen a megrendelt áru összegyűjtése, csomagolása és útnak indítása. Szintén a fulfillment szolgáltatás kapcsán lényeges, hogy a csomagolás és szállítás egyes szakaszainak megfelelően a rendszer frissíteni tudja a webshopba a megrendelés állapotát, valamint hogy új áru, illetve visszáru érkezése esetén aktualizálni tudja a webshopba a raktárkészletet. Ezeket a funkciókat a fent említett webshop motorok saját API-n keresztül biztosítják.

## Az útmutató célja

Ezen útmutató célja, hogy olyan webshopok is igénybe vehessék a MuFis szolgáltatásait, melyek nem a fentebb említett webshop motorokra épülnek, hanem saját, egyedi fejlesztések. Ehhez nincs másra szükség, mint hogy a webshop képes legyen az útmutatóban foglalt specifikációk szerint értelmezni és kiszolgálni az alábbi MuFis-ből érkező request-eket:

- **megrendelés(ek) lekérése**
- **megrendelés(ek) státusz módosítása** (*csak fulfillment szolgáltatás esetén szükséges*)
- **termék(ek) lekérése** (*csak fulfillment szolgáltatás esetén szükséges*)
- **termék(ek) raktárkészletének módosítása** (*csak fulfillment szolgáltatás esetén szükséges*)

Előzetes egyeztetést követően beállítjuk, hogy a MuFis a domain-en belül milyen útvonalra, és azon belül pontosan milyen endpoint-ra küldje az egyes request-eket a webshopnak. Az egyszerűség kedvéért az útmutatóban az alábbi endpoint megnevezéseket használjuk:

- *getOrder*
- *setOrder*
- *getProduct*
- *setProduct*

Ezek értelmében ha a domain neve *domain.hu*, az API elérési útvonal *api/mufis*, és a rendelések lekérésére a *getOrder* nevű endpoint lesz használva akkor a **https://domain.hu/api/mufis/getOrder** címre küldünk lekérést, ahol arra válaszként kell megkapnunk a kívánt adatokat.

## Autentikáció

Minden a MuFis által küldött request tartalmaz egy **API-KEY header**-t, ennek alapján a webshop ellenőrizheti, hogy valóban tőlünk érkezett a request. A használni kívánt értéket a két fél előzetesen egyezteti. Továbbá a webshop IP-cím alapján is ellenőrizheti, hogy valóban a MuFis szerverétől érkezett-e az adott request.

Érvénytelen autentikáció esetén a webshop adjon **401-es HTTP státuszt**.

## Pagination, limit, találatok sorrendje

Nem kötelező, de *getOrder* és *getProduct* request-ek kiszolgálásához javasunk beépíteni valamilyen felső határt (a továbbiakban **limit, limitáció**) arra vonatkozóan, hogy a webshop egy response-ban maximum hány a megadott szűrési feltételeknek megfelelő resource-t (megrendelést vagy terméket) adjon vissza. Ezen érték pontos meghatározása a webshopra van bízva.

Amennyiben a webshop implementálja ezt a limitációt, akkor a *getOrder* és *getProduct* request-ekre adott response-oknak mindig tartalmazniuk kell az alábbi értékeket:

érték megnevezése	érték leírása
<b>total_pages</b>	int típus, a request-ben megfogalmazott feltételekre adott válasz összesen hány "oldal"
<b>page</b>	int típus, az éppen aktuálisan küldött adat hányadik "oldal" az összesből, 1-től számozva

Amennyiben ez a két érték nem szerepel a válaszban, a MuFis úgy veszi, hogy amit kapott adat, az a feltételekben megfogalmazottaknak megfelelő összes találat.

Javasoljuk továbbá, hogy a response-ban visszaadott resource-ok mindig a webshop adatbázisának belső azonosítója szerint legyenek rendezve, csökkenő sorrendben (a legújabb rekordok legyenek a lista elején).

## Request-ek struktúrája

Minden a MuFis által küldött request az alábbiakat követi:

HTTP metódus	<b>POST</b>
Content-Type header	<b>application/x-www-form-urlencoded</b>
API-Key header	<b>a használatra kiválasztott érték</b>

Az egyes request-ek lehetséges paramétereinek részletes leírása lejjebb, az adott request specifikációjánál találhatóak.

## Response-ok várt struktúrája

Minden request **JSON** formátumban várja a webshop választ, illetve **200-as HTTP státuszú** response header-t, ami igazolja, hogy maga a kérés sikeres volt és a válasz releváns. Az egyes request-ekre várt **JSON response** részletes struktúrája lejjebb, az adott request specifikációjánál található.

## getOrder request és lehetséges paraméterei

A *getOrder* a webshopba beérkező megrendelések adatainak lekérésére szolgál.

A MuFis az alábbi paramétereket küldheti **POST**-ban:

paraméter megnevezése	paraméter leírása
<b>page</b>	<ul style="list-style-type: none"><li>• amennyiben a webshop implementálta a limitációt, ez a paraméter jelenti azt, hogy a MuFis éppen hányadik oldalt kéri</li><li>• ha nem szerepel a kérésben a <b>page</b> paraméter, a webshop vegye úgy, mintha az első oldalt kérné a MuFis</li></ul>
<b>order_number</b>	konkrét megrendelés lekérése a megrendelés száma alapján
<b>order_id</b>	konkrét megrendelés lekérése a megrendelés belső azonosítója alapján
<b>updated_at_min</b>	<ul style="list-style-type: none"><li>• időpont, CET (közép-európai) időzónában értve</li><li>• formátum: <i>Y-m-d H:i:s</i> (pl. <i>2023-01-20 12:23:48</i>)</li><li>• a paramétert tartalmazó lekérés azokat a megrendeléseket várja válaszul, melyek a megadott időpont után keletkeztek vagy módosultak a webshopban (beleértve a megadott időpontot is)</li></ul>
<b>status</b>	<ul style="list-style-type: none"><li>• csak a megadott státuszban lévő megrendeléseket kérjük</li><li>• egyszerre több státusz is kérhető legyen, vesszővel elválasztva</li><li>• a webshop által használt státuszok pontos megnevezéseit és azok jelentését a két fél előzetesen egyezteti</li></ul>
<b>date_from</b>	<ul style="list-style-type: none"><li>• dátum, CET (közép-európai) időzónában értve</li><li>• formátum: <i>Y-m-d</i> (pl. <i>2025-06-15</i>)</li><li>• a paramétert tartalmazó lekérés azokat a megrendeléseket várja válaszul, melyeket a megadott napon, vagy attól később adtak le</li></ul>
<b>date_to</b>	<ul style="list-style-type: none"><li>• dátum, CET (közép-európai) időzónában értve</li><li>• formátum: <i>Y-m-d</i> (pl. <i>2025-06-15</i>)</li><li>• a paramétert tartalmazó lekérés azokat a megrendeléseket várja válaszul, melyeket a megadott napon, vagy attól korábban adtak le</li></ul>

Az `order_number`, `order_id`, `updated_at_min`, `status`, `date_from` és `date_to` paraméterek mindegyikére érvényes, hogy csak akkor kell a szűrést alkalmazni, ha az adott paraméter szerepel a request-ben. Tehát ha pl. a `status` paraméter nem szerepel a lekérésben, akkor a webshop egyáltalán ne szűrjön a státuszra.

Továbbá érvényes, hogy amennyiben a kérés több lehetséges paramétert is tartalmaz, pl. jelen van a `status`, az `updated_at_min` és a `date_from` paraméter is, akkor válaszul azokat a rendeléseket várjuk, melyek MINDEN szűrési feltételnek megfelelnek.

## getOrder-re adott response várt struktúrája

Limitációt implementáló webshop esetében elvárt mezők a response-ban:

`total_pages`

`page`

Ezekről részletek a **Pagination, limit, találatok sorrendje** fejezetben.

`orders` = a request paraméterekben megadott szűrési feltételeknek megfelelő megrendelések listája (array) 0-tól indexelve. Ha nincs találat, akkor is legyen feltüntetve, üres array-ként. Pontosan 1 találat (pl. `order_number` szerinti lekérés) esetén egy 1-elemű array.

Az `orders` lista egyes elemeinek várt struktúrája:

érték megnevezése	érték leírása
<code>order_id</code>	a megrendelés belső azonosítója a webshopban erre az értékre vonatkozik a <code>getOrder</code> kérésben az <code>order_id</code> paraméter
<code>order_number</code>	a megrendelés száma erre az értékre vonatkozik a <code>getOrder</code> kérésben az <code>order_number</code> paraméter
<code>order_date</code>	a megrendelés leadásának dátuma <code>Y-m-d</code> formátumban pl. <code>2023-01-23</code> erre az értékre vonatkozik a <code>getOrder</code> kérésben a <code>date_from</code> és a <code>date_to</code> paraméter
<code>status</code>	a megrendelés állapota erre az értékre vonatkozik a <code>getOrder</code> kérésben a <code>status</code> paraméter
<code>date_mod</code>	a megrendelés utolsó módosításának időpontja <code>Y-m-d H:i:s</code> formátumban pl. <code>2023-01-08 15:02:18</code> erre az értékre vonatkozik a <code>getOrder</code> kérésben az <code>updated_at_min</code> paraméter
<code>billing_name</code>	számlázási cím - név
<code>billing_name2</code>	számlázási cím - név második sora
<code>billing_city</code>	számlázási cím - település
<code>billing_streetnum</code>	számlázási név - utca és házszám

<b>billing_zip</b>	számlázási cím - irányítószám
<b>billing_country</b>	számlázási cím - kétbetűs országkód pl. <i>HU</i> vagy <i>SK</i>
<b>shipping_name</b>	szállítási cím - név megj.: amennyiben a válasz nem tartalmazza a szállítási címre vonatkozó adatokat vagy azok mindegyike üres, a MuFis úgy veszi, hogy a szállítási cím megegyezik a számlázási címmel
<b>shipping_name2</b>	szállítási cím - név második sora
<b>shipping_city</b>	szállítási cím - település
<b>shipping_streetnum</b>	szállítási cím - utca és házszám
<b>shipping_zip</b>	szállítási cím - irányítószám
<b>shipping_country</b>	szállítási cím - kétbetűs országkód pl. <i>HU</i> vagy <i>SK</i>
<b>email</b>	vevő e-mail címe
<b>phone</b>	vevő telefonszáma, nemzetközi formátumban, szóközök nélkül pl. +36201234567 vagy +421907123456 a telefonszám elején szereplő + karakter elhagyható, vagy helyettesíthető kettő darab 0-val
<b>eu_vat_number</b>	vásárló EU-s adószáma pl. HU1234567890 vagy SK1234568790
<b>lang</b>	vásárló által használt nyelv kétbetűs kódja - számla küldése esetén a számla illetve a számlát tartalmazó e-mail szövegének nyelvét határozza meg pl. <i>hu</i> vagy <i>sk</i> vagy <i>en</i>
<b>currency</b>	megrendelés pénznemének 3-betűs kódja pl. <i>EUR</i> vagy <i>HUF</i>
<b>total_price</b>	megrendelés végösszege (beleértve mindent, tehát a szállítási díjat, ÁFA-t, kedvezményeket stb.) szóköz nélküli float típusban, tizedespontot használva, a <b>currency</b> -ben meghatározott pénznemben pl. 159.99
<b>payment_type</b>	megrendelés fizetési módja, lehetséges értékek: COD = utánvét BANK = banki átutalás CARD = kártyás fizetés PAYPAL = PayPal-os fizetés OTHER = egyéb nem utánvétes fizetési mód
<b>shipping_type</b>	a vásárló által választott szállítási mód megnevezése pl. <i>Házhozszállítás</i>
<b>delivery_point_group</b>	csomagpontra történő szállítás esetén a csomagpont típusa (szolgáltatója)

	pl. <i>packeta</i> vagy <i>gls</i> vagy <i>foxpost</i>
<b>delivery_point_id</b>	csomagpontra történő szállítás esetén az adott szolgáltatóhoz tartozó csomagpont azonosítója
<b>order_items</b>	megrendelésben szereplő tételek listája (array), részletes struktúra lejjebb
<b>meta_data</b>	opcionális, itt bármilyen extra adat küldhető, ami a rendeléshez tartozik maga a <b>meta_data</b> lista típus (array), az egyes elemei pedig <b>key</b> és <b>value</b> értékpárban JSON-ban tartalmazzák az adott adat megnevezését és értékét pl. [ <b>{key: 'adat1', value: 'ertek1'}</b> , <b>{key: 'adat2', value: 'ertek2'}</b> ] a <b>meta_data</b> -ban küldött értékek jelentését, és azok Mufis-ban történő használatát a két fél előzetesen egyezteteti

**order\_items** lista egyes elemeinek várt struktúrája:

érték megnevezése	érték leírása
<b>sku</b>	termék cikkszám
<b>name</b>	termék neve
<b>quantity</b>	a megrendelt mennyiség
<b>unit_price</b>	egységár
<b>price_type</b>	a feltüntetett egységár típusa, lehetséges értékek: with_vat = ha a feltüntetett egységár tartalmazza az ÁFA-t without_vat = ha a feltüntetett egységár nem tartalmazza az ÁFA-t megj.: ha az ÁFA százalék 0, akkor mindegy, hogy a kettő közül melyik érték
<b>vat_rate</b>	az ÁFA százalékban kifejezett mértéke

Amennyiben a rendelés a termékeken kívül egyéb tételeket is tartalmaz (pl. szállítási díj, utánvét kezelési díj, kedvezmény stb.), ezeket a tételeket szintén az **order\_items** listában kell szerepeltetni, a megadott struktúrában.

## **setOrder request és lehetséges paraméterei**

*(csak fulfillment szolgáltatás esetén szükséges implementálni)*

A **setOrder** request egy megrendelés státuszának módosítására szolgál, továbbá ha már csomagszámmal rendelkező megrendelésről van szó, akkor ezt a csomagszámot is elküldi a webshopnak, hogy az elmenthesse azt. Más adatot a MuFis nem módosít a megrendelésekben. A request lehetséges **POST** paraméterei:

paraméter megnevezése	paraméter leírása
<b>order_number</b>	a frissíteni kívánt megrendelés száma
<b>status</b>	az adott megrendelés új státusza
<b>package_number</b>	a megrendeléshez tartozó csomagszám több csomag esetén JSON lista az összes a rendeléshez tartozó csomagszámról csak akkor szerepel paraméterként, ha már regisztrálva van a csomag a szállítást végző futárszolgálatnál
<b>tracking_link</b>	a megrendeléshez tartozó csomag nyomkövetési linkje a szállítást végző futárszolgálatnál több csomag esetén JSON lista az összes a rendeléshez tartozó csomag linkjével csak akkor szerepel paraméterként, ha már regisztrálva van a csomag a szállítást végző futárszolgálatnál
<b>multiple_packages</b>	1 vagy 0, attól függően, hogy 1 vagy több csomag készült a megrendeléshez (nagy megrendelés esetén előfordulhat, hogy nem fér be 1 csomagba) ha ennek az értéke 0, akkor a <b>package_number</b> és <b>tracking_link</b> értékek string-ként tartalmazzák az adatokat ha ennek az értéke 1, akkor a <b>package_number</b> és <b>tracking_link</b> értékek JSON listában tartalmazzák az adatokat

Az adatforgalom csökkentése érdekében a webshop implementálhatja a *setOrder* funkció olyan verzióját is, amely egyszerre több megrendeléshez tartozó adatot is képes feldolgozni. Az ilyen kötegelt *setOrder* request struktúrája csak annyiban tér el az előzőekben leírtaktól, hogy a **POST**-ban csak egy **data** paraméter szerepel, és ennek az értéke pedig egy **JSON** formátumú lista, melynek egyes elemei a fenti struktúrát követik. A két fél egyezteteti, hogy a webshop melyik verzióját implementálja a funkciónak, a MuFis pedig ez alapján fogja küldeni a request-eket.

## setOrder-re adott response várt struktúrája

érték megnevezése	érték leírása
<b>ok</b>	a küldött adatok sikeres beírása esetén 1-es érték, ellenkező esetben 0
<b>error</b>	hiba esetén (ok = 0) ez a mező tartalmazhat egy szöveges leírást a hibáról
<b>order_number</b>	csak kötegelt request esetén, ez a mező jelzi, hogy az adott <b>ok</b> és <b>error</b> melyik megrendelésre vonatkozik

Kötegelt request esetén a response egy **orders** mező legyen, mely listaként tartalmazza a fent említett **ok**, **error** és **order\_number** mezőket - minden a requestben szereplő megrendeléshez 1 elemet.

## getProduct request és lehetséges paraméterei

(csak fulfillment szolgáltatás esetén szükséges implementálni)

A *getProduct* a webshopban árusított termékek adatainak lekérésére szolgál.

A MuFis az alábbi paramétereket küldheti **POST**-ban:

paraméter megnevezése	paraméter leírása
<b>page</b>	<ul style="list-style-type: none"><li>• amennyiben a webshop implementálta a limitációt, ez a paraméter jelenti azt, hogy a MuFis éppen hányadik oldalt kéri</li><li>• ha nem szerepel a kérésben a page paraméter, a webshop vegye úgy, mintha az első oldalt kérné a MuFis</li></ul>
<b>product_id</b>	konkrét termék lekérése a termék belső azonosítója alapján
<b>sku</b>	<ul style="list-style-type: none"><li>• csak a megadott cikkszámú terméket kérjük</li><li>• egyszerre több termék is kérhető legyen, a több cikkszámot vesszővel elválasztva küldjük</li></ul>
<b>active</b>	<ul style="list-style-type: none"><li>• értéke 1 vagy 0, azt fejezi ki, hogy csak az aktív vagy csak az inaktív termékeket kéri a MuFis<ul style="list-style-type: none"><li>○ aktív termék = jelenleg árulják a webshopban (raktárkészletől függetlenül)</li><li>○ inaktív termék = létezik a terméklistában, de jelenleg nincs árusítva a webshopban</li></ul></li><li>• ha nem szerepel a requestben ez a paraméter, akkor nem kérünk szűrést eszerint a tulajdonság szerint</li><li>• ha a webshop nem különbözteti meg ilyen módon a termékeit, hagyja figyelmen kívül ezt a paramétert</li></ul>

Ha a request-ben nem szerepel sem a **product\_id** sem a **sku** paraméter (tehát nem konkrét termék(ek) adatait kérjük le), akkor az összes termék adatait várjuk a válaszban..

## getProduct-ra adott válasz várt struktúrája

Limitációt implementáló webshop esetében elvárt mezők a response-ban:

total\_pages

page

Ezekről részletek a **Pagination, limit, találatok sorrendje** fejezetben.

**products** = a request paraméterekben megadott szűrési feltételeknek megfelelő megrendelések listája (array) 0-tól indexelve. Ha nincs találat, akkor is legyen feltüntetve, üres array-ként. Pontosan 1 találat (pl. sku szerinti lekérés) esetén egy 1-elemű array.

A **products** lista egyes elemeinek várt struktúrája:

érték megnevezése	érték leírása
product_id	a termék belső azonosítója a webshopban
sku	a termék egyedi cikkszám
name	a termék neve
active	1 vagy 0, attól függően, hogy az adott terméket jelenleg árulja-e a webshop (raktárkészletől függetlenül) ha a webshop nem különbözteti meg ilyen módon a termékeket, ebben a mezőben szerepeljen mindig 1-es
stock_quantity	a webshop adatbázisa szerint hány darab van jelenleg ebből a termékből raktáron
barcode	a terméken fellelhető vonalkód

## setProduct request és lehetséges paraméterei

(csak fulfillment szolgáltatás esetén szükséges implementálni)

A *setProduct* request egy termék raktárkészletének aktualizálására szolgál. A request lehetséges **POST** paraméterei:

paraméter megnevezése	paraméter leírása
sku	az aktualizálni kívánt termék cikkszám
stock_quantity	az aktuális raktáron lévő mennyiség

Az adatforgalom csökkentése érdekében a webshop implementálhatja a *setProduct* funkció olyan verzióját is, amely egyszerre több termékhez tartozó adatot is képes feldolgozni. Az ilyen kötegelt *setProduct* request struktúrája csak annyiban tér el az előzőekben leírtaktól, hogy a **POST**-ban csak egy **data** paraméter szerepel, és ennek az értéke pedig egy **JSON** formátumú lista, melynek egyes elemei a fenti struktúrát követik. A két fél egyezteteti, hogy a webshop melyik verzióját implementálja a funkciónak, a MuFis pedig ez alapján fogja küldeni a request-eket.

## setProduct-ra adott response várt struktúrája

érték megnevezése	érték leírása
<b>ok</b>	a küldött adatok sikeres beírása esetén 1-es érték, ellenkező esetben 0
<b>error</b>	hiba esetén (ok = 0) ez a mező tartalmazhat egy szöveges leírást a hibáról
<b>sku</b>	csak kötegelt request esetén, ez a mező jelzi, hogy az adott <b>ok</b> és <b>error</b> melyik termékre vonatkozik

Kötegelt request esetén a response egy **products** mező legyen, mely listaként tartalmazza a fent említett **ok**, **error** és **sku** mezőket.

## Megrendelések automatikus importja a MuFis-ba

Amennyiben csak a számlázást veszi igénybe a webshop, elegendő a MuFis felületén beállítani, hogy a rendszer automatikusan napi 1x beolvassa a webshoptól az új megrendeléseket, és elkészítse az adatok alapján a számlát.

Fulfillment szolgáltatás esetén azonban fontos, hogy minél előbb beérkezzenek a megrendelések a mi rendszerünkbe. Ennek érdekében javasolt a webshopba ún. **webhook**-ot definiálni, ami új rendelés leadásakor, illetve létező megrendelés módosításakor a megadott linkre request-et küld a MuFis szervernek.

Ez a request csak egy egyszerű **GET** kérés, nem kell, hogy tartalmazzon semmilyen adatot az új/módosított megrendelésről. A MuFis a webhook hatására egy import folyamatot indít a webshop felé, melynek keretében a *getOrder* funkció **updated\_at\_min** paraméterét használva lekéri a webshoptól az utolsó lekérés óta módosult rendeléseket, és mindent aktualizál, amit kell.

A webhook pontos URL-jét az összekapcsolás indítását megelőzően elküldjük a webshop részére.

Amennyiben a webshop nem kívánja a webhook-ot implementálni, úgy a MuFis munkanapokon munkaidőben óránként lekéri a webshoptól az új/módosult megrendeléseket, tehát legkésőbb a megrendelés/módosítás után 1 órával eljut hozzánk az adat a webhook nélkül is.

## Megrendelés törlése

Számos okból kifolyólag megtörténhet, hogy a leadott megrendelést mégsem kell teljesíteni, tehát törölni kell. Nagyon fontos, hogy ezt a webshop **sose úgy tegye, hogy egy az egyben törli a megrendelést az adatbázisából**, hanem legyen egy **“törölt” státusz**, amibe átkerül a megrendelés. Így a változások lekérésekor a MuFis megkapja erről a megrendelésről, hogy már törölt státuszú és ezt a módosítást át tudja venni a saját adatbázisába. Ha a webshop véglegesen törli a megrendelést, akkor a változások lekérésekor a MuFis nem kap semmilyen adatot a törölt megrendeléséről, tehát a MuFis adatbázisban az előző állapot szerint marad a rendelés. Ez pedig ahhoz vezethet, hogy a törölt megrendelés mégis kiküldésre kerül az eredeti adatok szerint.

A törölt státusz pontos megnevezését a két fél előzetesen egyezteteti.

## **change log**

2023-01-24 - v1.0

2025-07-17 - v1.1

- `date_from` és `date_to` paraméterek a *getOrder* kérésben
- `meta_data` a *getOrder* válaszbán

2026-03-06 - v1.2

- `order_items` pontosítása a *getOrder* válaszbán