

Epsilon Smart e-Shop API

Περιεχόμενα

Εισαγωγή	3
1. Αυθεντικοποίηση	3
1.1. Ανανέωση του JWT token	4
1.2. Δημιουργία χρήστη API	5
2. Μέθοδοι του e-Shop API	6
2.1. Λήψη ειδών/υπηρεσιών	6
2.2. Λήψη υπολοίπων ειδών	7
2.3. Αποστολή παραστατικών πωλήσεων/υπηρεσιών	8
2.4. Λήψη εκτύπωσης παραστατικού	9
3. Περιγραφή Αντικειμένων API	10
3.1. EshopItem	10
3.2. EshopItemBalance	13
3.3. EshopSale	14
3.4. EshopItemLine	17
3.5. EshopApiDocRel	19
4. Ανάλυση εισαγωγής παραστατικών	20
4.1. Παραμετροποίηση e-Shop	20
4.2. Αναζήτηση πελάτη	21
4.3. Αναζήτηση είδους/υπηρεσίας	21
4.4. Πεδία αξιών στις γραμμές	21
5. Παραδείγματα κώδικα σε C#	22
5.1. Αυθεντικοποίηση και λήψη JWT token	22
5.2. Ανανέωση JWT token	22
5.3. Ανάγνωση μηνύματος λάθους	23

5.4.	Λήψη ειδών/υπηρεσιών	23
5.5.	Λήψη υπολοίπων ειδών.....	24
5.6.	Αποστολή παραστατικού	24
5.7.	Ανάγνωση αντιστοιχίσεων παραστατικών	26
5.8.	Λήψη εκτύπωσης παραστατικού	26

Εισαγωγή

Το e-Shop API μπορεί να χρησιμοποιηθεί για τη διασύνδεση ενός e-Shop συστήματος με την εφαρμογή Epsilon Smart. Οι δύο βασικές λειτουργίες που εξυπηρετεί είναι αρχικά να πάρει το e-Shop τα είδη/υπηρεσίες από το Epsilon Smart, και στη συνέχεια το e-Shop να στείλει στο Epsilon Smart παραστατικά.

Από τεχνικής άποψης, πρόκειται για ένα REST API που χρησιμοποιεί HTTP αιτήματα, επιστρέφει τυπικούς κωδικούς απόκρισης HTTP και αποτελέσματα σε μορφή JSON. Ο έλεγχος ταυτότητας γίνεται μέσω JWT Bearer που περνάει στην κεφαλίδα κάθε HTTP αιτήματος.

1. Αυθεντικοποίηση

Το Epsilon Smart είναι μία εφαρμογή που είναι στημένη στο cloud και σας παρέχεται ως υπηρεσία (SaaS). Για την καλύτερη και ταχύτερη εξυπηρέτηση των χρηστών, διαθέτει πολλαπλές φάρμες στο cloud, που όλες όμως χρησιμοποιούν τον ίδιο διακομιστή αυθεντικοποίησης το myAccount (<https://myaccount.epsilonnet.gr/>). Το myAccount είναι υπεύθυνο για τον έλεγχο της ταυτότητας ενός χρήστη (client) και σε περίπτωση επιτυχούς σύνδεσης, επιστρέφει ένα διακριτικό JWT (jwt token) στον client, ο οποίος στη συνέχεια το χρησιμοποιεί σε όλες τις επόμενες κλήσεις με server.

Για να μπορέσει λοιπόν ένας client να στείλει κλήσεις στο Epsilon Smart e-Shop API, θα πρέπει προηγουμένως να έχει γίνει έλεγχος της ταυτότητάς του μέσω του ίδιου ακριβώς διακομιστή αυθεντικοποίησης (myAccount), χρησιμοποιώντας το [myAccount API](#).

Επομένως, θα πρέπει αρχικά να κάνετε μία κλήση στο myAccount API και να καλέσετε τη μέθοδο **LoginToSubscription**.

(<https://myaccount.epsilonnet.gr/api/account/LoginToSubscription>)

Πρόκειται για μία POST μέθοδο, που θα πρέπει στο HTTP Body να στείλετε τις παρακάτω παραμέτρους σύνδεσης:

- **subscriptionKey**: Πρόκειται για το «Κλειδί» της συνδρομής σας, το οποίο θα ανακτήσετε από την οθόνη προβολής της συνδρομής σας. Απλά πατήστε το κουμπί στο μπλε πλαίσιο που

φαίνεται παρακάτω, για να εμφανιστεί και να μπορέσετε να το αντιγράψετε.

Προϊόν

Epsilon Smart Advanced

Λογιστής

Κλειδί

- email: Το email που χρησιμοποιείτε για να συνδεθείτε και στο Epsilon Smart.
- password: Το password που χρησιμοποιείτε για να συνδεθείτε και στο Epsilon Smart.

Σε περίπτωση που ο χρήστης περάσει τον έλεγχο ταυτότητας θα πάρετε μία επιτυχή HTTP απάντηση (HTTP Code 200) από την οποία θα μπορέσετε να αντλήσετε ένα JSON αντικείμενο με τις παρακάτω ιδιότητες:

- jwt: Πρόκειται για το JWT token που θα χρησιμοποιείτε στη συνέχεια για να κάνετε περαιτέρω κλήσεις στο Epsilon Smart e-Shop API.
- jwtExpiration: Ημερομηνία & ώρα λήξης τους παραπάνω JWT token.
- jwtRefreshToken: Πρόκειται για ένα δευτερεύον JWT token, που χρησιμοποιείται στην ανανέωση του αρχικού jwt στην περίπτωση που έχει λήξει.
- jwtRefreshTokenExpiration: Ημερομηνία & ώρα λήξης τους jwtRefreshToken.
- url2: Πρόκειται για το url της cloud φάρμας στην οποία βρίσκεται η συνδρομή σας. Το συγκεκριμένο url θα χρησιμοποιηθεί για να κάνετε τις κλήσεις στο Epsilon Smart e-Shop API.

```
curl -X POST "https://myaccount.epsilonnet.gr/api/Account/LoginToSubscription" -H "accept: text/plain" -H "Content-Type: application/json" -d '{"subscriptionKey": "\1AA21B81418434A851E7AG0CEI9B27D\","email": "\user@email.gr\","password": "\*****\"}'
```

1.1. Ανανέωση του JWT token

Το JWT token έχει διάρκεια ζωής 30 λεπτά. Σε περίπτωση λοιπόν, που κάποιος client κάλεσε τη LoginToSubscription, εκτέλεσε κάποια μέθοδο του e-Shop API, και μετά πχ. από 45 λεπτά θέλει να ξανακαλέσει μία μέθοδο, δεν θα χρειάζεται να ξανακαλέσει τη LoginToSubscription. Πρέπει απλά να ελέγξει αν το JWT token έχει λήξει και να καλέσει τη **Refresh** στο myAccount API (<https://myaccount.epsilonnet.gr/api/Token/Refresh>) για να το ανανεώσει και να συνεχίσει. Πρόκειται για μία POST μέθοδο, που θα πρέπει στο HTTP Body να στείλετε τα JWT tokens που λάβατε κατά τη LoginToSubscription, όπως φαίνονται παρακάτω:

- token: Πρόκειται για το jwt που πήρατε στη LoginToSubscription.
- refreshToken: Πρόκειται για το jwtRefreshToken που πήρατε στη LoginToSubscription.

Η Refresh είναι λιγότερο χρονοβόρα σε σχέση με τη LoginToSubscription, οπότε συνιστάται να τη χρησιμοποιείτε ώστε ο κώδικάς σας να είναι πιο γρήγορος.

5

2. Μέθοδοι του e-Shop API

Το e-Shop API παρέχει τις ακόλουθες δυνατότητες:

- Λήψη ειδών/υπηρεσιών από το e-Shop σύστημα, μέσω της μεθόδου `GetItems`.
- Λήψη υπολοίπων ειδών από το e-Shop σύστημα, μέσω της μεθόδου `GetItemsBalances`.
- Αποστολή παραστατικών πωλήσεων/υπηρεσιών από το e-Shop σύστημα στο Epsilon Smart, μέσω της μεθόδου `InsertDocuments`.
- Λήψη της εκτύπωσης σε μορφή pdf ενός παραστατικού που έχει εισαχθεί στο Epsilon Smart, μέσω της μεθόδου `GetDocumentPdf`.

Για να καλέσετε τις μεθόδους του API θα χρειαστείτε:

- URL του endpoint του API: Πρόκειται για το `url2` που επιστρέφει η `LoginToSubscription` και θα αναφέρεται παρακάτω ως **smartUrl**.
- JWT token: Πρόκειται για το `jwt` που επιστρέφει `LoginToSubscription`. Χρειάζεται να το περνάτε στο HTTP Header κάθε αιτήματος προς το API, ως `Authorization Header`.

2.1. Λήψη ειδών/υπηρεσιών

Για να κάνετε λήψη των ειδών/υπηρεσιών από το Epsilon Smart σας παρέχεται η μέθοδος **GetItems** ([smartUrl/api/Eshop/GetItems](#)). Πρόκειται για μία GET μέθοδο, που θα πρέπει να προσδιορίσετε στο query string του url από ποιο `RevisionNumber` και μετά θέλετε να αντλήσετε είδη/υπηρεσίες.

Όταν σε μία συνδρομή Epsilon Smart ενεργοποιείται η επιπλέον λειτουργικότητα e-Shop, κάθε είδος/υπηρεσία αποκτά ένα μοναδικό αριθμό `RevisionNumber`. Κάθε φορά που τροποποιείται ένα είδος/υπηρεσία, αποκτά νέο μοναδικό `RevisionNumber`. Επιπλέον, στα είδη/υπηρεσίες προστέθηκε πεδίο “Αποστολή στο e-Shop” (Ναι/Όχι), το οποίο χρησιμοποιείται υπογείως ως φίλτρο κατά την άντληση των ειδών/υπηρεσιών. Έτσι, μπορείτε εύκολα να εξαιρέσετε κάποια είδη και να μην στέλνονται στο e-Shop σας.

Σε περίπτωση επιτυχίας, η μέθοδος επιστρέφει σε JSON μία λίστα από το πολύ 100 αντικείμενα τύπου `EshopItem`. Οι ιδιότητες του συγκεκριμένου αντικείμενου αναλύονται παρακάτω στην ενότητα [Περιγραφή Αντικειμένων API](#). Στο συγκεκριμένο σημείο, πρέπει μόνο να αναφερθεί ότι σε κάθε αντικείμενο επιστρέφεται και το `RevisionNumber`.

Την πρώτη φορά λοιπόν που θα θέλετε να αντλήσετε τα είδη, θα καλέσετε τη `GetItems` με `RevisionNumber=0`. Μόλις σας επιστραφεί το πρώτο πακέτο με τα 100 είδη/υπηρεσίες, θα σημειώσετε το μέγιστο `RevisionNumber` που λάβατε (`maxRevisionNumber`), και θα ξανακαλέσετε τη `GetItems` με `RevisionNumber= maxRevisionNumber +1`. Έτσι θα αντλήσετε το επόμενο πακέτο με 100 είδη/υπηρεσίες. Αυτό το βήμα θα το επαναλάβετε όσο χρειαστεί, μέχρι το πακέτο που θα

σας επιστραφεί έχει λιγότερα από 100 είδη/υπηρεσίες. Τότε θα σημαίνει ότι τα έχετε αντλήσει όλα.

Μόλις ολοκληρώσετε τη διαδικασία λήψης όλων των ειδών/υπηρεσιών, σε περίπτωση που μετά από κάποιο χρονικό διάστημα έχουν προστεθεί ή τροποποιηθεί κάποια είδη/υπηρεσίες στο Epsilon Smart και θέλετε να αντλήσετε, τότε έχετε τις παρακάτω επιλογές:

- Να αντλήσετε μόνο τις αλλαγές. Για να γίνει αυτό, θα πρέπει να βρείτε από το σύστημά σας ποιο είναι το μέγιστο RevisionNumber που έχετε αντλήσει (maxRevisionNumber) και να καλέσετε τη GetItems με RevisionNumber= maxRevisionNumber +1.
- Να αντλήσετε ξανά όλα τα είδη/υπηρεσίες, οπότε θα πρέπει να επαναληφθεί η διαδικασία που περιγράφηκε παραπάνω.

Συνιστάται η πρώτη επιλογή, δηλαδή να αντλήσετε μόνο τις αλλαγές. Γιατί με αυτό τον τρόπο θα γίνουν λιγότερα αιτήματα και θα είναι μικρότερος ο όγκος των δεδομένων που θα αντλήσετε. Επομένως, ο κώδικάς σας θα είναι πιο γρήγορος.

```
curl -X GET "https://epsilonsmart.epsilonnet.gr/api/Eshop/GetItems?RevisionNumber=0"  
-H "accept: text/plain" -H "Authorization: Bearer eyJhbGci..... a6BZdvMI"
```

2.2. Λήψη υπολοίπων ειδών

Για να κάνετε λήψη των υπολοίπων ειδών από το Epsilon Smart σας παρέχεται η μέθοδος **GetItemsBalances** ([smartUrl/api/Eshop/GetItemsBalances](https://epsilonsmart.epsilonnet.gr/api/Eshop/GetItemsBalances)). Πρόκειται για μία GET μέθοδο, η οποία δεν παίρνει κάποια παράμετρο, και επιστρέφει τα υπόλοιπα για όλα τα είδη στη συνδρομή σας που έχουν “Αποστολή στο e-Shop” Ναι. Η μέθοδος αυτή έχει νόημα μόνο σε προϊόντα τύπου Advanced, όπου παρακολουθούνται τα υπόλοιπα ειδών. Στα άλλα προϊόντα δεν επιστρέφει αποτελέσματα. Επίσης, ο σκοπός της είναι να αντλήσει το e-Shop τα υπόλοιπα των ειδών κάποια στιγμή μέσα στη μέρα και όχι να έχει live εικόνα των υπολοίπων των ειδών. Για το λόγο αυτό, έχουν προστεθεί περιορισμοί στην κλήση, ώστε να επιτρέπεται ορισμένος αριθμός κλήσεων ανά συνδρομή και να εξασφαλίζεται η ομαλή λειτουργία του server.

Σε περίπτωση επιτυχίας, η μέθοδος επιστρέφει σε JSON μία λίστα με αντικείμενα τύπου EshopItemBalance. Οι ιδιότητες του συγκεκριμένου αντικείμενου αναλύονται παρακάτω στην ενότητα [Περιγραφή Αντικειμένων API](#).

```
curl -X GET "https://epsilonsmart.epsilonnet.gr/api/Eshop/GetItemsBalances"  
-H "accept: text/plain" -H "Authorization: Bearer eyJhbGci..... a6BZdvMI"
```


Σε περίπτωση επιτυχίας (HTTP Code 200) το σώμα της απάντησης θα περιέχει μία λίστα από αντικείμενα τύπου EshopApiDocRel, που το καθένα θα περιέχει την αντιστοίχιση ανάμεσα στο παραστατικό του e-Shop (RefDocCode) και το παραστατικό που εισάχθηκε στο Epsilon Smart (SmartDocId). Οι ιδιότητες του συγκεκριμένου αντικειμένου αναλύονται παρακάτω στην ενότητα [Περιγραφή Αντικειμένων API](#). Η λίστα αυτή ενημερώνεται μόνο για τα παραστατικά e-Shop που έχουν συμπληρωμένο το πεδίο RefDocCode.

Σε περίπτωση αποτυχίας (HTTP Code 400) στο σώμα της απάντησης, που είναι τύπου ProblemDetails επιστρέφεται και το πεδίο ApiResult το οποίο πάλι θα περιέχει μία λίστα από αντικείμενα τύπου EshopApiDocRel. Πιο συγκεκριμένα στο ApiResult θα υπάρχουν οι αντιστοιχίες των παραστατικών που εισήχθησαν επιτυχώς στο Epsilon Smart και επιπλέον για το παραστατικό που απέτυχε στο πεδίο ErrorMessage θα αναγράφεται ο λόγος της αποτυχίας. Προϋπόθεση και για αυτό αποτελεί να είναι συμπληρωμένο το πεδίο RefDocCode του παραστατικού που εστάλη.

2.4. Λήψη εκτύπωσης παραστατικού

Για να μπορέσετε να πάρετε την εκτύπωση ενός παραστατικού που εισάχθηκε στο Epsilon Smart, θα πρέπει στο σύστημα του e-Shop σας να έχετε κρατήσει το ID που πήρε το παραστατικό μέσα στο Epsilon Smart και το οποίο περιέχεται στην απάντηση της InsertDocuments, όπως αναφέρθηκε παραπάνω. Για να κάνετε λοιπόν λήψη της εκτύπωσης ενός παραστατικού σε μορφή pdf, σας παρέχεται η μέθοδος **GetDocumentPdf** ([smartUrl/api/Eshop/GetDocumentPdf](#)). Πρόκειται για μία GET μέθοδο, στην οποία θα πρέπει να προσδιορίσετε στην παράμετρο docId του url το ID του παραστατικού στο Epsilon Smart και θα επιστρέψει ένα αρχείο τύπου pdf.

```
curl -X GET "https://localhost:44394/api/Eshop/GetDocumentPdf?docId= 1621fa94-3e8e-43a8-b0af-946314e8cfe7" -H "accept: text/plain"
```

3. Περιγραφή Αντικειμένων API

Σε αυτή την ενότητα θα γίνει αναλυτική καταγραφή των ιδιοτήτων κάθε αντικειμένου που χρησιμοποιεί το e-Shop API, και θα αναλυθεί όπου χρειάζεται πώς αυτές αντιστοιχίζονται μέσα στο Epsilon Smart.

3.1. EshopItem

ID `string`

Μοναδικό ID του είδους/υπηρεσίας. Είναι χρήσιμο να το φυλάξετε ώστε να χρησιμοποιηθεί στην αποστολή παραστατικών, για πιο γρήγορη ανεύρεση του είδους/υπηρεσίας.

Name `string`

Όνομα είδους/υπηρεσίας.

Code `string`

Μοναδικός κωδικός είδους/υπηρεσίας. Μπορεί να χρησιμοποιηθεί στην ανεύρεση του είδους/υπηρεσίας κατά την αποστολή παραστατικών.

ScanCode `string`

Κωδικός σάρωσης είδους/υπηρεσίας. Μπορεί να χρησιμοποιηθεί στην ανεύρεση του είδους/υπηρεσίας κατά την αποστολή παραστατικών.

EANCode `string`

Κωδικός EAN είδους. Μπορεί να χρησιμοποιηθεί στην ανεύρεση του είδους κατά την αποστολή παραστατικών.

MPNCode `string`

Κωδικός MPN είδους. Μπορεί να χρησιμοποιηθεί στην ανεύρεση του είδους κατά την αποστολή παραστατικών.

ISBNCode *string*

Κωδικός ISBN είδους. Μπορεί να χρησιμοποιηθεί στην ανεύρεση του είδους κατά την αποστολή παραστατικών.

Domain *integer* [0, 1]

Δηλώνει αν πρόκειται για είδος (0) ή υπηρεσία (1).

NormalVATPercent *decimal*

Το % Φ.Π.Α. του είδους σε εταιρίες με κανονικό καθεστώς Φ.Π.Α.

ReducedVATPercent *decimal*

Το % Φ.Π.Α. του είδους σε εταιρίες με μειωμένο καθεστώς Φ.Π.Α.

MeasurementUnit *string*

Το όνομα της μονάδας μέτρησης του είδους/υπηρεσίας.

MeasurementUnitShortcut *string*

Η συντόμευση της μονάδας μέτρησης του είδους/υπηρεσίας.

Category *string* *null*

Το όνομα της κατηγορίας του είδους/υπηρεσίας.

RetailCategory *string* *null*

Το όνομα της κατηγορίας λιανικής του είδους/υπηρεσίας. Διαθέσιμη σε προϊόντα Retail & Advanced.

RetailCategoryInnerCode *string* *null*

Πρόκειται για ένα εσωτερικό κωδικό της κατηγορίας λιανικής του είδους/υπηρεσίας, που θα σας βοηθήσει να φτιάξετε τη δενδρική μορφή των κατηγοριών λιανικής. Διαθέσιμο σε προϊόντα Retail & Advanced. (Προς κατάργηση στην επόμενη έκδοση)

RetailCategoryFullName string null

Περιλαμβάνει την πλήρη διαδρομή της δένδρικής κατηγορίας λιανικής του είδους/υπηρεσίας.

EshopCategory string null

Το όνομα της e-shop κατηγορίας του είδους/υπηρεσίας.

EshopCategoryFullName string null

Περιλαμβάνει την πλήρη διαδρομή της δένδρικής e-shop κατηγορίας του είδους/υπηρεσίας.

Manufacturer string

Ο κατασκευαστής του είδους.

EshopDescription string

Περιγραφή e-Shop του είδους/υπηρεσίας.

WholesalePrice decimal

Η τιμή χονδρικής του είδους/υπηρεσίας.

RetailPrice decimal

Η τιμή λιανικής του είδους/υπηρεσίας.

EshopPrice decimal

Η τιμή e-Shop του είδους/υπηρεσίας.

Remarks string null

Παρατηρήσεις του είδους/υπηρεσίας.

UserDefinedText01-05 string null

Τα 5 αλφαριθμητικά πεδία χρήστη του είδους/υπηρεσίας.

UserDefinedDate01-05 `datetime null`

Τα 5 ημερομηνιακά πεδία χρήστη του είδους/υπηρεσίας.

UserDefinedVNum01-03 `decimal null`

UserDefinedQNum01-02 `decimal null`

Τα 5 αριθμητικά πεδία χρήστη του είδους/υπηρεσίας.

PrimaryImage `string null`

Το URL της κύριας εικόνας του είδους/υπηρεσίας.

AdditionalImages `array of string null`

Τα URL των υπόλοιπων εικόνων του είδους/υπηρεσίας.

RevisionNumber `integer (long)`

Revision number του είδους/υπηρεσίας. Ενημερώνεται κατά την εισαγωγή και τροποποίηση είδους/υπηρεσίας και είναι μοναδικό.

3.2. EshopItemBalance

ItemID `string`

Μοναδικό ID του είδους.

Balance `decimal`

Το υπόλοιπο του είδους.

3.3. EshopSale

DocDateTime *datetime*

Πρόκειται για την ημερομηνία του παραστατικού. Αν θέλετε μπορείτε να ορίσετε και ώρα, ώστε να ενημερωθεί το αντίστοιχο πεδίο ώρα δημιουργίας. Αν δηλώσετε ώρα, θα πρέπει να προσέξετε πώς θα στην στείλετε γιατί το Epsilon Smart αποθηκεύει την ώρα δημιουργίας του client, και δεν είναι UTC. Επομένως, κατά το serialization του αντικειμένου θα πρέπει να ορίσετε για τις ημερομηνίες να γίνονται serialize με Unspecified time-zone. (βλέπε [Microsoft](#) και [Newtonsoft](#) για βοήθεια)

IsRetail *integer* [0, 1]

Δηλώνει αν πρόκειται για παραστατικό χονδρική (0) ή λιανικής (1).

VatStatus *integer* [0, 1, 2]

Δηλώνει αν καθεστώς Φ.Π.Α. του παραστατικού είναι κανονικό (0), μειωμένο (1), απαλλαγή (2).

VatExemptionCode *string* null

Δηλώνει τον κωδικό της αιτίας απαλλαγής Φ.Π.Α. που έχει το παραστατικό. Σε περίπτωση που το καθεστώς Φ.Π.Α. του παραστατικού είναι απαλλαγή, θα πρέπει να συμπληρώσετε το συγκεκριμένο πεδίο με έναν κωδικό από τον πίνακα Απαλλαγές Φ.Π.Α. μέσα στο Epsilon Smart.

RefDocCode *string* null

Κωδικός αναφοράς παραστατικού. Σε αυτό το πεδίο μπορείτε να συμπληρώσετε τον κωδικό που είχε το παραστατικό σας στο e-Shop, ώστε να αποθηκευτεί η αντιστοίχιση στο Epsilon Smart.

DocNum *long* null

Αριθμός παραστατικού. Σε αυτό το πεδίο μπορείτε να συμπληρώσετε τον αριθμό που είχε το παραστατικό σας στο e-Shop, ώστε να αποθηκευτεί η αντιστοίχιση στο Epsilon Smart.

PaymentMethod *integer* [0, 1, 2, 4, 5, 7]

Δηλώνει τον τρόπο πληρωμής του παραστατικού. Μπορεί να πάρει τις ακόλουθες τιμές:

0: Μετρητά, 1: Επί Πιστώσει, 2: Πιστωτική Κάρτα, 4: Paypal, 5: Μέσω Τραπέζης, 7: Μέσω e-Shop

Σε περίπτωση πληρωμής με πιστωτική κάρτα ή μέσω τραπεζής, θα πρέπει να συμπληρώσετε και το αντίστοιχο πεδίο με τα στοιχεία πληρωμής, δηλαδή να ορίσετε μία κάρτα ή έναν τραπεζικό λογαριασμό αντίστοιχα.

PaymentCreditCardName string null

Όνομα πιστωτικής κάρτας που υπάρχει στο Epsilon Smart. Είναι υποχρεωτικό να συμπληρωθεί σε περίπτωση που ο τρόπος πληρωμής είναι πιστωτική κάρτα.

PaymentBankAccName string null

Όνομα τραπεζικού λογαριασμού που υπάρχει στο Epsilon Smart. Είναι υποχρεωτικό να συμπληρωθεί σε περίπτωση που ο τρόπος πληρωμής είναι τραπεζική κατάθεση.

CustName string null

Όνομα πελάτη. Αναλόγως την παραμετροποίηση e-Shop, μπορεί να είναι απαιτητό εφόσον έχει οριστεί η αναζήτηση πελάτη να γίνεται μόνο με το όνομα. Διαφορετικά, είναι απαραίτητο να συμπληρωθεί ή το όνομα ή το Α.Φ.Μ. για να είναι εφικτή η αναζήτηση πελάτη.

CustTin string null

Α.Φ.Μ. πελάτη. Αναλόγως την παραμετροποίηση e-Shop, μπορεί να είναι απαιτητό εφόσον έχει οριστεί η αναζήτηση πελάτη να γίνεται μόνο με το Α.Φ.Μ.

CustStreet string null

Η οδός της διεύθυνσης πελάτη.

CustStreetNo string null

Η αριθμός της διεύθυνσης πελάτη.

CustCity string null

Η πόλη του πελάτη.

CustPostalCode string null

Ο Τ.Κ. της διεύθυνσης του πελάτη.

CustTaxOffice string null

Η Δ.Ο.Υ. του πελάτη.

CustCountry string null

Ο κωδικός ISO 2 της χώρας του πελάτη. Θα χρησιμοποιηθεί σε περίπτωση που δημιουργηθεί νέος πελάτης.

CustEmail string null

Το email του πελάτη. Θα χρησιμοποιηθεί σε περίπτωση που δημιουργηθεί νέος πελάτης.

CustPhone1 string null

Το τηλέφωνο του πελάτη. Θα χρησιμοποιηθεί σε περίπτωση που δημιουργηθεί νέος πελάτης.

ExtraCharges decimal null

Αξία έξτρα χρεώσεων, όπως μεταφορικά που θα εφαρμοστεί στο σύνολο του παραστατικού. Το πεδίο αυτό θα χρησιμοποιηθεί σε όλα τα προϊόντα, εκτός του Entry που δεν υποστηρίζει χρεώσεις. Να σημειωθεί ότι η αξία αυτή αν δηλωθεί θα πρέπει να εμπεριέχει και το Φ.Π.Α.

Justification string null

Η αιτιολογία του παραστατικού. Μέγιστο μήκος 255 χαρακτήρες.

Comments string null

Τα σχόλια του παραστατικού, χωρίς περιορισμό στο μήκος αντίστοιχου του πεδίου.

UserDefText01-05 string null

Τα 5 αλφαριθμητικά πεδία χρήστη του παραστατικού. Μέγιστο μήκος 150 χαρακτήρες το καθένα.

Lines array of EshopItemLine not null

Λίστα με τις γραμμές ειδών/υπηρεσιών του παραστατικού. Το αντικείμενο EshopItemLine περιγράφεται παρακάτω.

3.4. EshopItemLine

ItemID string null

Μοναδικό ID του είδους/υπηρεσίας που αν προσδιοριστεί χρησιμοποιείται για πιο γρήγορη αναζήτηση του είδους/υπηρεσίας.

ItemCode string null

Ο κωδικός του είδους/υπηρεσίας στο Epsilon Smart. Χρησιμοποιείται για αναζήτηση του είδους/υπηρεσίας σε περίπτωση που δεν έχει δοθεί το ID.

ScanCode string null

Ο κωδικός σάρωσης του είδους/υπηρεσίας στο Epsilon Smart. Χρησιμοποιείται για αναζήτηση του είδους/υπηρεσίας σε περίπτωση που δεν έχει δοθεί το ID.

EANCode string null

Ο κωδικός EAN του είδους στο Epsilon Smart. Χρησιμοποιείται για αναζήτηση του είδους σε περίπτωση που δεν έχει δοθεί το ID.

ISBNCode string null

Ο κωδικός ISBN του είδους στο Epsilon Smart. Χρησιμοποιείται για αναζήτηση του είδους σε περίπτωση που δεν έχει δοθεί το ID.

MPNCode string null

Ο κωδικός κατασκευαστή του είδους στο Epsilon Smart. Χρησιμοποιείται για αναζήτηση του είδους σε περίπτωση που δεν έχει δοθεί το ID.

ItemPrintDescr string null

Η περιγραφή για εκτύπωση του είδους/υπηρεσίας.

VATPercent decimal

Το ποσοστό Φ.Π.Α. του είδους/υπηρεσίας.

Qty decimal

Η ποσότητα του είδους/υπηρεσίας.

Price decimal

Η τιμή του είδους/υπηρεσίας. Αν το παραστατικό είναι λιανικής, θα πρέπει να είναι τιμή λιανικής.

IsGift integer [0, 1]

Δηλώνει αν πρόκειται για γραμμή δώρου. Αν λοιπόν η τιμή του πεδίου είναι 1, η γραμμή θα δηλωθεί ως δώρο και θα εισαχθεί στο Epsilon Smart με 100% έκπτωση.

LineDiscount decimal

Η αξία έκπτωσης στη γραμμή του είδους/υπηρεσίας.

NetVal decimal

Η αξία μετά την έκπτωση στη γραμμή του είδους/υπηρεσίας.

VatVal decimal

Η αξία Φ.Π.Α. στη γραμμή του είδους/υπηρεσίας.

TotalVal decimal

Η συνολική αξία της γραμμής του είδους/υπηρεσίας.

Justification string null

Η αιτιολογία της γραμμής του είδους/υπηρεσίας. Μέγιστο μήκος 100 χαρακτήρες.

3.5. EshopApiDocRel

RefDocCode `string`

Ο κωδικός αναφοράς του παραστατικού στο e-Shop.

SmartDocId `string null`

Το ID του παραστατικού στο Smart.

ErrorMessage `string null`

Το μήνυμα λάθους σε περίπτωση μη δημιουργίας του παραστατικού στο Smart.

4. Ανάλυση εισαγωγής παραστατικών

Για την καλύτερη κατανόηση της εισαγωγής παραστατικών, κρίνεται απαραίτητο να αναλυθούν περαιτέρω κάποια σημαντικά σημεία της διαδικασίας. Έτσι αρχικά θα εξετάσουμε την παραμετροποίηση του e-Shop στο Epsilon Smart, και στη συνέχεια θα αναλύσουμε πώς γίνεται η αναζήτηση του πελάτη και των ειδών/υπηρεσιών.

4.1. Παραμετροποίηση e-Shop

Όταν σε μία συνδρομή ενεργοποιηθεί η έξτρα λειτουργικότητα e-Shop, αυτομάτως εισάγεται στη συνδρομή μία προκαθορισμένη παραμετροποίηση που αφορά το e-Shop. Στην αντίστοιχη φόρμα υπάρχει αναλυτικό help για όλα τα πεδία. Θα επισημάνουμε απλά πώς συνδέονται τα παραπάνω πεδία με το αντικείμενο του [EshopSale](#).

Αναλόγως την ιδιότητα EshopSale .IsRetail, επιλέγεται από την παραμετροποίηση η σειρά του παραστατικού που θα χρησιμοποιηθεί (χονδρική ή λιανική).

Επίσης, αν το παραστατικό είναι λιανικής, προτείνουμε στην παραμετροποίηση τη “Χρήση Προεπιλεγμένου Πελάτη Λιανικής”. Σε αυτή την περίπτωση, τα παραστατικά λιανικής θα χρησιμοποιούν τον πελάτη λιανικής (χωρίς να γίνεται αναζήτηση πελάτη) και τα στοιχεία που αφορούν τον πελάτη θα ενημερώσουν τα στοιχεία διεύθυνσης του παραστατικού. (EshopSale .CustName, CustStreet, CustStreetNo, CustPostalCode). Διαφορετικά, αν έχει επιλεγεί “Δημιουργία Πελάτη Λιανικής”, θα γίνεται αναζήτηση του πελάτη και αν δεν υπάρχει θα εισάγεται σαν πελάτης λιανικής στο Epsilon Smart.

Αν το παραστατικό είναι χονδρικής, πάντα γίνεται αναζήτηση πελάτη, και αν δεν υπάρχει εισάγεται σαν πελάτης. Η λογιστική του κατηγορία καθορίζεται αναλόγως της χώρας που έχει προσδιοριστεί. Διαφορετικά, παίρνει την προεπιλεγμένη τιμή από την παραμετροποίηση εφαρμογής.

Σε περίπτωση που γίνει δημιουργία πελάτη, και στις δύο παραπάνω περιπτώσεις ενημερώνονται όλα τα πεδία που αφορούν τον πελάτη στο αντικείμενο EshopSale.

Σε περίπτωση που η συνδρομή έχει την η έξτρα λειτουργικότητα Υποκαταστήματα, όλα τα παραστατικά θα εισαχθούν στο υποκατάστημα που επιλέξετε στην παραμετροποίηση.

Σε περίπτωση που η συνδρομή περιλαμβάνει Χρεώσεις (δεν είναι προϊόν Entry), και έχει προσδιοριστεί αξία χρεώσεων (EshopSale.ExtraCharges), θα χρησιμοποιηθεί η επιλεγμένη χρέωση στην παραμετροποίηση για να ενημερώσει το παραστατικό.

4.2. Αναζήτηση πελάτη

Ο τρόπος αναζήτησης του πελάτη διαφοροποιείται ανάλογα με το εάν το παραστατικό είναι χονδρικής ή λιανικής.

Στην περίπτωση που πρόκειται για πελάτη χονδρικής το σύστημα κάνει αναζήτηση μόνο με το Α.Φ.Μ, και μόνο εάν βρεθούν παραπάνω από (1) ένας πελάτες με το ίδιο Α.Φ.Μ τότε και μόνο τότε θα αναζητήσει και με το Όνομα. Επισημαίνεται ότι σε περίπτωση που έχετε στη συνδρομή την λειτουργικότητα των υποκαταστημάτων, θα ήταν καλό εκτός του Α.Φ.Μ. να είναι συμπληρωμένο και το Όνομα ώστε να βρεθεί η σωστή εγγραφή.

Στα παραστατικά λιανικής το σύστημα κάνει αναζήτηση μόνο με το Όνομα.

4.3. Αναζήτηση είδους/υπηρεσίας

Η αναζήτηση του είδους/υπηρεσίας γίνεται αναλόγως τι πεδίο έχει καθοριστεί στο αντικείμενο [EshopItemLine](#).

Αρχικά, ελέγχεται αν έχει προσδιοριστεί το ItemID, οπότε το θα γίνει αναζήτηση με το ID του είδους/υπηρεσίας, και θα είναι και ταχύτερη σε σχέση με τις άλλες αναζητήσεις.

Αν δεν έχει προσδιοριστεί το ItemID, τότε θα χρησιμοποιηθούν τα ItemCode, ScanCode, EANCode, MPNCode και ISBNCode με αυτή τη σειρά. Τα τρία τελευταία πεδία δεν είναι διαθέσιμα στις υπηρεσίες αλλά μόνο στα είδη. Αναλόγως ποιο από τα τέσσερα έχει προσδιοριστεί θα γίνει αναζήτηση με βάση τον κωδικό ή τον κωδικό σάρωσης του είδους/υπηρεσίας ή τον κωδικό EAN του είδους ή τον κωδικό κατασκευαστή του είδους.

4.4. Πεδία αξιών στις γραμμές

Στο αντικείμενο EshopItemLine υπάρχουν τα ακόλουθα αξιακά πεδία διαθέσιμα:

Qty, Price, LineDiscount, NetVal, VATVal, TotalVal

Δεν χρειάζεται να τα προσδιορίσετε αναγκαστικά όλα. Αρκεί τα 2-3 απαραίτητα ώστε να προκύπτει η συνολική αξία της γραμμής, και το σύστημα θα υπολογίσει τα υπόλοιπα (όπως φαίνεται και στα [παραδείγματα κώδικα](#)). Καλό θα είναι σίγουρα να προσδιορίζεται η αξία Φ.Π.Α. (VATVal).

5. Παραδείγματα κώδικα σε C#

Στην ενότητα αυτή σας παραθέτουμε δείγματα κώδικα ανά μέθοδο του e-Shop API, προς καλύτερη κατανόηση και διευκόλυνση.

5.1. Αυθεντικοποίηση και λήψη JWT token

```
class LoginResponse {
    public string Jwt { get; set; }
    public DateTime JwtExpiration { get; set; }
    public string JwtRefreshToken { get; set; }
    public DateTime JwtRefreshTokenExpiration { get; set; }
    public string Url2 { get; set; }
}

private LoginResponse LoginToSubscription() {
    var url = "https://myaccount.epsilonnet.gr/api/Account/LoginToSubscription";
    var json = JsonSerializer.Serialize(new {
        subscriptionKey = "****",
        email = "****",
        password = "****"
    });
    var httpContent = new StringContent(json, System.Text.Encoding.UTF8, "application/json");
    using var httpClient = new HttpClient();
    var response = await httpClient.PostAsync(url, httpContent);
    if (response.IsSuccessStatusCode) {
        var responseString = await response.Content.ReadAsStringAsync();
        return JsonSerializer.Deserialize<LoginResponse>(responseString,
            new JsonSerializerOptions { PropertyNameCaseInsensitive = true });
    }
    throw new Exception();
}
```

5.2. Ανανέωση JWT token

```
class RefreshTokenResponse {
    public string Jwt { get; set; }
    public DateTime JwtExpiration { get; set; }
    public string JwtRefreshToken { get; set; }
    public DateTime JwtRefreshTokenExpiration { get; set; }
}

private RefreshTokenResponse RefreshToken(string jwt, string jwtRefreshToken) {
    var url = "https://myaccount.epsilonnet.gr/api//Token/Refresh";
    var json = JsonSerializer.Serialize(new {
        token = jwt,
        refreshToken = jwtRefreshToken
    });
}
```

```

var httpContent = new StringContent(json, System.Text.Encoding.UTF8, "application/json");
using var httpClient = new HttpClient();
var response = await httpClient.PostAsync(url, httpContent);
if (response.IsSuccessStatusCode) {
    var responseString = await response.Content.ReadAsStringAsync();
    var responseObject = JsonSerializer.Deserialize<RefreshTokenResponse>(responseString,
        new JsonSerializerOptions { PropertyNameCaseInsensitive = true });
    return responseObject;
} else {
    HandleError(response);
    return null;
}
}

```

5.3. Ανάγνωση μηνύματος λάθους

```

class ProblemDetails {
    public string Title { get; set; }
    public int? Status { get; set; }
    public string Detail { get; set; }
    public EshopApiDocRel[] ApiResult { get; set; }
}

private void HandleError(HttpResponseMessage response) {
    var responseString = await response.Content.ReadAsStringAsync();
    var responseObject = JsonSerializer.Deserialize<ProblemDetails>(responseString,
        new JsonSerializerOptions {
            PropertyNameCaseInsensitive = true
        });
    Console.WriteLine($"Call responded with error code: {response.StatusCode}. Message: {responseObject.Title}");
}

```

5.4. Λήψη ειδών/υπηρεσιών

```

private List<EshopItem> GetItems(string smartUrl, string jwt) {
    long fromRevisionNumber = 0;
    using var httpClient = new HttpClient();
    List<EshopItem> allItems = new List<EshopItem>();
    List<EshopItem> partialResult = new List<EshopItem>();
    do {
        var url = $"{smartUrl}api/Eshop/GetItems?RevisionNumber={fromRevisionNumber}";
        httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", jwt);
        var response = await httpClient.GetAsync(url);
        if (response.IsSuccessStatusCode) {
            var responseString = await response.Content.ReadAsStringAsync();
            partialResult = JsonSerializer.Deserialize<List<EshopItem>>(responseString,
                new JsonSerializerOptions { PropertyNameCaseInsensitive = true });
            if (partialResult.Count > 0) {
                fromRevisionNumber = partialResult.Max(m => m.RevisionNumber) + 1;
                allItems.AddRange(partialResult);
            }
        }
    } while (partialResult.Count > 0);
    return allItems;
}

```

```

    }
    }
    }
    while (partialResult.Count == 100);
    return allItems;
}

```

5.5. Λήψη υπολοίπων ειδών

```

private List<EshopItemBalance> GetItemsBalances(string smartUrl, string jwt) {
    using var httpClient = new HttpClient();
    List<EshopItemBalance> balances = new List<EshopItemBalance>();
    var url = $"{smartUrl}api/Eshop/GetItemsBalances";
    httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", jwt);
    var response = await httpClient.GetAsync(url);
    if (response.IsSuccessStatusCode) {
        var responseString = await response.Content.ReadAsStringAsync();
        balances = JsonSerializer.Deserialize<List<EshopItemBalance>>(responseString,
            new JsonSerializerOptions { PropertyNameCaseInsensitive = true });
    }
    return balances;
}

```

5.6. Αποστολή παραστατικού

πχ. Δημιουργία παραστατικού λιανικής

```

private EshopSale CreateRetailDoc() {
    var doc = new EshopSale() {
        DocDateTime = new DateTime(2021, 5, 2, 9, 12, 0, DateTimeKind.Unspecified),
        IsRetail = 1, VatStatus = 0, RefDocCode = "ΠΑΡ-001", PaymentMethod = 5,
        CustName = "ΓΙΩΡΓΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ", CustStreet = "Λ.ΚΗΦΙΣΙΑΣ",
        CustStreetNo = "10", CustCity = "ΑΘΗΝΑ",
        Justification = "Online Παραγγελία",
        ExtraCharges = 3, Lines = new List<EshopItemLine>()
    };
    EshopItemLine line1 = new EshopItemLine() {
        ItemCode = "ΕΙΔ0000003", VATPercent = 24, Qty = 2, Price = 15,
        VATVal = (decimal)5.81, TotalVal = (decimal)30
    };
    doc.Lines.Add(line1);
    EshopItemLine line2 = new EshopItemLine() {
        ItemCode = "ΕΙΔ0000001", VATPercent = 24, Qty = 1, Price = 20,
        VATVal = (decimal)3.87, TotalVal = (decimal)20
    };
    doc.Lines.Add(line2);
    return doc;
}

```

πχ. Δημιουργία παραστατικού χονδρικής

```
private EshopSale CreateWholesaleDoc() {
    var doc = new EshopSale() {
        DocDateTime = new DateTime(2021, 5, 2, 11, 50, 0, DateTimeKind.Unspecified),
        IsRetail = 0, VatStatus = 0, RefDocCode = "ΠΑΡ-002", PaymentMethod = 1,
        CustName = "EpsilonNet", CustTIN = "099357493", CustStreet = "Λ.Συγγρού",
        CustStreetNo = "350", CustCity = "ΑΘΗΝΑ", CustPostalCode = "17674",
        CustCountry = "GR", CustEmail = "info@epsilon.net.gr",
        CustPhone1 = "211 5007000", Justification = "Online Παραγγελία",
        ExtraCharges = 3, Lines = new List<EshopItemLine>()
    };
    EshopItemLine line1 = new EshopItemLine() {
        ItemCode = "ΕΙΔ0000003", VATPercent = 24, Qty = 2, Price = 5,
        NetVal = (decimal)10, VATVal = (decimal)2.4
    };
    doc.Lines.Add(line1);
    EshopItemLine line2 = new EshopItemLine() {
        ItemCode = "ΕΙΔ0000001", VATPercent = 24, Qty = 1, Price = 20,
        NetVal = (decimal)20, VATVal = (decimal)4.8
    };
    doc.Lines.Add(line2);
    return doc;
}
```

πχ. Αποστολή παραστατικών

```
private void InsertDocuments(string smartUrl, string jwt) {
    var docs = new List<EshopSale>();
    docs.Add(CreateRetailDoc());
    docs.Add(CreateWholesaleDoc());

    var json = JsonSerializer.Serialize(docs);

    //Alternative serialization with Newtonsoft.Json
    //var json = Newtonsoft.Json.JsonConvert.SerializeObject(docs,
    // new Newtonsoft.Json.JsonSerializerSettings() {
    //     DateFormatHandling =
    //     DateFormatHandling.IsoDateFormat,
    //     Culture = System.Globalization.CultureInfo.InvariantCulture,
    //     DateTimeZoneHandling =
    //     DateTimeZoneHandling.Unspecified,
    // });

    var url = $"{smartUrl}api/Eshop/InsertDocuments";
    var httpContent = new StringContent(json, System.Text.Encoding.UTF8, "application/json");
    using var httpClient = new HttpClient();
    httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", jwt);
    var response = await httpClient.PostAsync(url, httpContent);
    if (!response.IsSuccessStatusCode)
        HandleError(response);
    } else
        GetDocRelations(response);
}
```

5.7. Ανάγνωση αντιστοιχίσεων παραστατικών

Σε περίπτωση επιτυχημένης αποστολής παραστατικών (HTTP Code 200), στο σώμα της απάντησης της InsertDocuments μπορείτε να αντλήσετε τις αντιστοιχίσεις των παραστατικών του e-Shop με αυτά που εισήχθησαν στο Epsilon Smart.

```
private void GetDocRelations(HttpResponseMessage responseMessage) {
    var responseString = await response.Content.ReadAsStringAsync();

    var relations = JsonSerializer.Deserialize<List<EshopApiDocRel>>(
        responseString,
        new JsonSerializerOptions { PropertyNameCaseInsensitive = true });
}
```

Αντίστοιχα, σε περίπτωση αποτυχημένης αποστολής παραστατικών (HTTP Code 400), στο σώμα της απάντησης μπορείτε να αντλήσετε τις αντιστοιχίσεις των επιτυχημένων από το πεδίο ApiResult που εμπεριέχεται στο ProblemDetails. Για παράδειγμα δείτε στο παράδειγμα [Ανάγνωση μηνύματος λάθους](#).

5.8. Λήψη εκτύπωσης παραστατικού

Για τη λήψη της εκτύπωσης σε μορφή pdf ενός παραστατικού πρέπει να χρησιμοποιηθεί το ID του παραστατικού στο Epsilon Smart.

```
private void GetDocumentPdf(string smartUrl, string jwt, string docId, string filePath) {
    var url = $"{smartUrl}api/Eshop/GetDocumentPdf?{docId}";
    var fileName = Path.Combine(filePath, "document.pdf");

    using var httpClient = new HttpClient();

    httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", jwt);

    using var response = await httpClient.GetAsync(url);

    if (response.IsSuccessStatusCode) {
        using var file = File.Create(fileName);
        var contentStream = await response.Content.ReadAsStreamAsync();
        await contentStream.CopyToAsync(file);
        await file.FlushAsync();
    }
}
```