Trian.gr API Documentation

# Γενικά routes

## [GET] /

Αρχική σελίδα. Απλώς για να ελέγχουμε ότι ο server λειτουργεί.

## [GET] /health

Χρησιμοποιείται κατά το deployment της νέας έκδοσης. Όταν αυτό το path επιστρέφει status 200, τότε τα προϊόντα έχουν φορτωθεί και το νέο deployment είναι έτοιμο για χρήση (και αντικατάσταση του προηγούμενου version). Μέχρι τότε (ή αν δεν έρθουν καθόλου πελάτες ή προϊόντα), βγάζει status 503, που σημαίνει ότι δεν είναι ακόμα έτοιμο. Αν δεν έρθουν πελάτες ή προϊόντα, αλλά αυτά υπάρχουν από προηγούμενη συλλογή τους, τότε επιστρέφεται status 200.

# Routes για τεχνική χρήση

## [GET] /api/all/:token/customers [GET] /api/all/:token/frames [GET] /api/all/:token/lens

Χρησιμοποιούνται για να κατεβάσουμε αρχείο με όλη τη διαθέσιμη πληροφορία από τα προϊόντα που έρχονται από το SoftOne. Απαιτείται token.

## [GET] /show/:token/customers [GET] /show/:token/frames [GET] /show/:token/lens

Realtime routes

Φέρνουν σε πραγματικό χρόνο τα δεδομένα από το SoftOne και επιστρέφουν τα πρώτα 100 σε beautified json μορφή. Για έλεγχο από browser σε περίπτωση αλλαγών στα δεδομένα που στέλνει το SoftOne. Απαιτείται token.

## [GET] /api/validatecustomer/:token/:email

Χρησιμοποιείται από το Firebase Authentication κατά την εγγραφή νέου χρήστη ώστε να ελεγχθεί ότι ο χρήστης ανήκει στους (ενεργούς) πελάτες του καταστήματος. Επιστρέφει ένα json αντικείμενο με τις πληροφορίες του πελάτη ή null αν ο πελάτης δεν υπάρχει. Απαιτείται token.

## [GET] /api/update/:token

Refresh route

Απαιτείται token. Όταν καλείται από τον browser, ανανεώνονται οι πελάτες και τα προϊόντα από το SoftOne χειροκίνητα. Απαιτείται token.

# Routes για χρήση από το e-shop

## [GET] /api/profile [GET] /profile

Απαιτεί Authorization Bearer Token (το οποίο αποκτάται από το Firebase Authentication) και επιστρέφει τα στοιχεία του πελάτη, ώστε να εμφανιστούν στο e-shop. Προς το παρόν, χρησιμοποιείται μόνο στη σελίδα «Παραγγελία», διότι στο «Προφίλ», η σελίδα παίρνει τα στοιχεία μόνο από το Firebase (χωρίς κλήση στο API).

## [GET] /api/tree/trian.json [GET] /api/tree/tokai.json [GET] /api/tree/trian-alt.json [GET] /api/tree/trian-alt.json

Tree routes

Φέρνουν τη δομή των πρώτων δύο φίλτρων των φακών που παρουσιάζονται στο e-shop, δηλαδή, το Σφαίρωμα και τον Κύλινδρο. Πιο συγκεκριμένα, φέρνουν τα διαθέσιμα Σφαιρώματα και, για κάθε Σφαίρωμα, τους διαθέσιμους Κυλίνδρους. Με βάση αυτά τα 2, θα δημιουργηθεί στο front-end το φίλτρο που θα σταλεί στο /api/filter/\* (filter routes) για να ληφθούν τα προϊόντα.

## [POST] /api/filter/lens/trian [POST] /api/filter/lens/tokai [POST] /api/filter/frames

Filter routes

Φιλτράρει τα προϊόντα. Δέχεται στο request body ένα object το οποίο χρησιμοποιείται ως φίλτρο για τα προϊόντα και επιστρέφει τα προϊόντα που ταιριάζουν. Για τα frames δεν εφαρμόζεται φίλτρο, διότι επιστρέφονταόλα.

Αν υπάρχει έγκυρο Authorization Bearer Token πελάτη, τότε επιστρέφονται επιπλέον και οι τιμές των προϊόντων browser.

## [POST] /api/order

To e-shop στέλνει την παραγγελία στον server κατά την ολοκλήρωσή της από τον πελάτη. Το request έχει στο body το αντικείμενο order το οποίο έχει properties το order.cart, το order.costs, καθώς και άλλα properties. To back-end τη λαμβάνει, προσθέτει id και την χειρίζεται κατάλληλα (στέλνει e-mail κλπ). Επιστρέφει το JSON object της παραγγελίας, «ενισχυμένο» (με id, λεπτομέρειες πελάτη κλπ).

# Ανενεργά Routes

## [GET] /api/validatemail/:token/:email

Χρησιμοποιείται μόνο για test σκοπούς. Επιστρέφει true ή false, ανάλογα αν το email ανήκει σε υπάρχοντα πελάτη. Δεν κάνει έλεγχο αν ο πελάτης είναι ενεργός. Καταργήθηκε!

## [GET] /api/frames.json [GET] /api/lens/tokai.json [GET] /api/lens/trian.json

Επιστρέφει όλα τα προϊόντα σε json αρχείο. Δεν χρησιμοποιούνται. Τα lens, αντικαταστάθηκαν από τα tree routes (για τα lens) ώστε το front-end να λαμβάνει μόνο τα φίλτρα (tree routes) αντί για όλα τα αντικείμενα. Τα frames αντικαταστάθηκαν από το filter route για τα frames.

## [GET] /api/unique/frames.json [GET] /api/unique/lens.json [GET] /api/unique/lens-trian.json [GET] /api/unique/lens-tokai.json

Δεν χρησιμοποιούνται! Επιστρέφουν ένα json object, τα properties του οποίου έχουν όνομα τα properties των προϊόντων και είναι arrays που έχουν όλες τις δυνατές τιμές των properties των προϊόντων. Χρησιμοποιήθηκαν για να ελέγχουν τις «ενδιάμεσες» συναρτήσεις uniqueOf. Καταργήθηκαν!

# cotrollers

## DATA

Στον controller SoftOne.js υπάρχουν 3 αντικείμενα δεδομένων:

* Data, που έχουν όλα τα δεδομένα
* PublicData, που περιέχουν την πληροφορία που είναι διαθέσιμη σε επισκέπτες
* DataForCustomers, που έχει την πληροφορία που είναι διαθέσιμη σε πιστοποιημένους χρήστες / πελάτες.

## Autorization Middleware

Στον controller firebase.js υπάρχουν τα εξής middleware:

* validateFirebaseToken  
  Λαμβάνει το authorization Bearer token από το request και το επαληθεύει. Αν βρει έγκυρο πελάτη, τότε το req.customer παίρνει ως τιμή τον πελάτη ως αντικείμενο, και το request συνεχίζει. Αν όχι (πχ χωρίς token ή ληγμένο token ή πελάτης που δεν υπάρχει πια γιατί διαγράφτηκε), τότε επιστρέφει error 401.
* firebaseUser  
  Λαμβάνει το authorization Bearer token από το request και το επαληθεύει. Αν βρει έγκυρο πελάτη, τότε το req.customer παίρνει ως τιμή το χρήστη ως αντικείμενο. Αν όχι (στις περιπτώσεις όπως παραπάνω), τότε το req.customer γίνεται null και το request συνεχίζει με δεδομένο ότι ο χρήστης είναι επισκέπτης (guest).
* validateSystemToken  
  Αυτά είναι για τα show routes και για τα system routes (πχ χρήση από Firebase Authentication). Είναι tokens για τεχνική χρήση και δεν είναι tokens που αντιστοιχούν σε κάποιο χρήστη.

## Validation

Δημιουργήθηκε το /api/validate/cart το οποίο λαμβάνει το Cart από το e-shop, ως array από items, διορθώνει τις τιμές τους, και το επιστρέφει πίσω.