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). 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 πελάτες του καταστήματος. Επιστρέφει ένα json αντικείμενο με τις πληροφορίες του πελάτη. Τον έλεγχο για το αν είναι ενεργός ο πελάτης το κάνει το Firebase Authentication (blocking functions).

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

Refresh route

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

# 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 από το Firebase μόνο.

## [GET] /api/frames.json

Επιστρέφει όλα τα προϊόντα σε json αρχείο. Επιστρέφει μόνο την δημόσια διαθέσιμη πληροφορία (πχ δεν επιστρέφει τις τιμές των προϊόντων). Τα παρακάτω δεν χρησιμοποιούνται, διότι αντικαταστάθηκαν από τα tree routes, ώστε το front-end να λαμβάνει μόνο τα φίλτρα (tree routes) αντί για όλα τα αντικείμενα:

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

## [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 από το 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

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. Καταργήθηκε το παρακάτω, διότι τα frames (σε αντίθεση με τα lens) έρχονται όλα στο front end και φιλτράρονται εκεί.

[POST] /api/filter/frames

## [POST] /api/order

To e-shop στέλνει την παραγγελία στον server κατά την ολοκλήρωσή της από τον πελάτη. Στο body έχει το αντικείμενο order το οποίο έχει 2 βασικά properties, το order.cart και το order.customer. To back-end τη λαμβάνει, προσθέτει id και την χειρίζεται κατάλληλα στέλνει e-mail 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/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!