



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη 2022



**MedBook**

*Searching for a doctor made easy.*

# Προδιαγραφές Λογισμικού και Υλοποίηση συστήματος

Version 0.6  
(draft)

Αντωνιάδης Πρόδρομος 9911 [piantoni@ece.auth.gr](mailto:piantoni@ece.auth.gr)

Γερογιάννης Κωνσταντίνος 9638 [konsgero@ece.auth.gr](mailto:konsgero@ece.auth.gr)

Μπαρμπουνάκης Κωνσταντίνος 9759 [kmparmpou@ece.auth.gr](mailto:kmparmpou@ece.auth.gr)

Γεροντόπουλος Αναστάσιος 9682 [ganastas@ece.auth.gr](mailto:ganastas@ece.auth.gr)

25/04/2022



## Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
Α. Συμεωνίδης	17/05/2007	Δημιουργία εγγράφου. Προσαρμογή των προτύπων του K. E. Wiegers <sup>2</sup> και του M. Smialek's.	0.1
Α. Συμεωνίδης	29/3/2014	Μικρή αναθεώρηση – τροποποίηση ενοτήτων	0.1.3
Χ. Ζολώτας	10/4/2020	Μεγάλη αναθεώρηση – αφαίρεση ενοτήτων	0.4
Χ. Ζολώτας	15/4/2020	Μεγάλη αναθεώρηση – προσθήκη ενότητας REST προδιαγραφών	0.5.3
Κ. Παναγιώτου	25/4/2020	Μεγάλη αναθεώρηση – προσθήκη ενότητας Nodered περιγραφής	0.5.7
Α. Συμεωνίδης	30/4/2020	Αναθεώρηση και τελική δομή προτύπου	0.6

## Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Αντωνιάδης Πρόδρομος	14	piantoni@ece.auth.gr
Γερογιάννης Κωνσταντίνος	14	konsgero@ece.auth.gr
Γεροντόπουλος Αναστάσιος	14	ganastas@ece.auth.gr
Μπαρμπουνάκης Κωνσταντίνος	14	kmparmpon@ece.auth.gr

<sup>2</sup> Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document. Original template is available at: <http://www.processimpact.com/>



## Πίνακας Περιεχομένων

Πίνακας Περιεχομένων .....	3
Λίστα Σχημάτων .....	5
1 Πρότυπα Σχεδιασμού που υιοθετήθηκαν.....	8
1.1 Proxy Design Pattern.....	8
1.2 Abstract Factory Design Pattern.....	9
1.3 Observer Design Pattern.....	10
2 Αρχιτεκτονική Συστήματος .....	11
2.1 Αναγνώριση Πόρων (Resources) Συστήματος .....	11
2.2 Τεκμηρίωση REST διεπαφής .....	12
2.2.1 Πόρος Patient .....	12
2.2.1.1 Μοντέλο δεδομένων Patient.....	12
2.2.1.2 Endpoint POST πόρου Patient .....	12
2.2.1.3 Endpoint POST πόρου Patient, για συγκεκριμένο Patient Id.....	14
2.2.1.4 Endpoint PUT πόρου Patient, για συγκεκριμένο Patient Id.....	15
2.2.2 Πόρος Doctor.....	17
2.2.2.1 Μοντέλο δεδομένων Doctor.....	17
2.2.2.2 Endpoint POST πόρου Doctor.....	17
2.2.2.3 Endpoint POST πόρου Doctor, για συγκεκριμένο Doctor Id.....	19
2.2.2.4 Endpoint πόρου PUT Doctor, για συγκεκριμένο Doctor Id.....	20
2.2.3 Πόρος Appointment.....	22
2.2.3.1 Μοντέλο δεδομένων Appointment.....	22
2.2.3.2 Μοντέλο δεδομένων Appointments (Λίστα με αντικείμενα τύπου Appointment).....	22
2.2.3.3 Endpoint POST πόρου appointments, για ασθενή και συγκεκριμένο patientId.....	22
2.2.3.4 Endpoint GET πόρου appointments, για ασθενή και συγκεκριμένο patientId.....	24
2.2.3.5 Endpoint DELETE πόρου appointments, για ασθενή και συγκεκριμένα patientId, appointmentId.....	25
2.2.3.6 Endpoint PUT πόρου appointments, για γιατρό και συγκεκριμένο doctorId.....	26
2.2.3.7 Endpoint GET πόρου appointments, για γιατρό και συγκεκριμένο doctorId.....	28
2.2.3.8 Endpoint PUT πόρου appointments, για γιατρό και συγκεκριμένα doctorId και appointmentId.....	29
2.2.4 Πόρος Review.....	31
2.2.4.1 Μοντέλο δεδομένων Review.....	31
2.2.4.2 Μοντέλο δεδομένων Reviews	



	(Λίστα με αντικείμενα τύπου Review).....	31
2.2.4.3	Endpoint POST πόρου reviews, για συγκεκριμένο doctorId.....	31
2.2.4.4	Endpoint GET πόρου reviews, για συγκεκριμένο doctorId.....	32
2.2.5	Πόρος Report.....	34
2.2.5.1	Μοντέλο Δεδομένων Report.....	34
2.2.5.2	Endpoint POST πόρου Report.....	34
2.2.6	Πόρος Search.....	36
2.2.6.1	Μοντέλο δεδομένων Doctors.....	36
2.2.6.2	Endpoint GET πόρου search.....	36
2.2.7	Πόρος Notification.....	38
2.2.7.1	Endpoint POST πόρου notifications, για συγκεκριμένο patientId.....	38
2.2.7.2	Endpoint GET πόρου notifications, για συγκεκριμένο patientId.....	39
2.2.7.3	Endpoint DELETE πόρου notifications, για συγκεκριμένο patientId.....	41
2.2.7.4	Endpoint POST πόρου notifications, για συγκεκριμένο doctorId.....	43
2.2.7.5	Endpoint GET πόρου notifications, για συγκεκριμένο doctorId.....	45
2.2.7.6	Endpoint DELETE πόρου notifications, για συγκεκριμένο doctorId.....	46
3	Υλοποίηση Συστήματος με Node-RED .....	48
3.1	Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed .....	48
3.1.1	Ροές Πόρου Patient.....	48
3.1.2	Ροές Πόρου Doctor.....	49
3.1.3	Ροές Πόρου Appointments.....	50
3.1.4	Ροές Πόρου Reviews.....	51
3.1.5	Ροές Πόρου Reports.....	52
3.1.6	Ροές Πόρου Notifications.....	52
3.1.7	Ροές Πόρου Search.....	54
3.2	Υλοποίηση της Εφαρμογής Πελάτη .....	55
3.2.1	Ιστορία Χρήστη <Create Patient User> .....	55
3.2.2	Ιστορία Χρήστη <Create Patient User by ID>.....	55
3.2.3	Ιστορία Χρήστη <Edit Patient Details>.....	56
3.2.4	Ιστορία Χρήστη <Create Doctor User>.....	56
3.2.5	Ιστορία Χρήστη <Create Doctor User by ID>.....	57
3.2.6	Ιστορία Χρήστη <Edit Doctor Details>.....	57
3.2.7	Ιστορία Χρήστη <Get Patient Appointment List>.....	58
3.2.8	Ιστορία Χρήστη <Book Appointment>.....	58
3.2.9	Ιστορία Χρήστη < Patient Cancel Booked Appointment >.....	59
3.2.10	Ιστορία Χρήστη < Doctor Modify Available Appointment List >.....	59
3.2.11	Ιστορία Χρήστη <Get Doctor Appointment List>.....	60
3.2.12	Ιστορία Χρήστη <Doctor Cancel Booked Appointment>.....	60
3.2.13	Ιστορία Χρήστη <Patient Submit Review>.....	61



---

3.2.14	Ιστορία Χρήστη <Get Doctor's Reviews>.....	61
3.2.15	Ιστορία Χρήστη <User Submit Report>.....	62
3.2.16	Ιστορία Χρήστη <Get Search Results Sorted>.....	62
3.2.17	Ιστορία Χρήστη <Create Patient's Notification by ID>.....	63
3.2.18	Ιστορία Χρήστη <Patient Get Notification>.....	63
3.2.19	Ιστορία Χρήστη <Delete Patient's Notification>.....	64
3.2.20	Ιστορία Χρήστη < Create Doctor's Notification by ID >.....	64
3.2.21	Ιστορία Χρήστη <Doctor Get Notification>.....	65
3.2.22	Ιστορία Χρήστη <Delete Doctor's Notification>.....	65
Παράρτημα Ι – Ανοιχτά Θέματα .....		66

## Λίστα Σχημάτων

[Σχήμα 1: Εφαρμογή προτύπου Proxy](#)

[Σχήμα 2: Εφαρμογή προτύπου Abstract Factory](#)

[Σχήμα 3: Εφαρμογή προτύπου Observer](#)

[Σχήμα 4: Μοντέλο δεδομένων Patient](#)

[Σχήμα 5: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 6: Δοκιμή του endpoint](#)

[Σχήμα 7: Αποκρίσεις του συστήματος](#)

[Σχήμα 8: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 9: Δοκιμή του endpoint](#)

[Σχήμα 10: Αποκρίσεις του συστήματος](#)

[Σχήμα 11: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 12: Δοκιμή του endpoint](#)

[Σχήμα 13: Αποκρίσεις του συστήματος](#)

[Σχήμα 14: Μοντέλο δεδομένων Doctor](#)

[Σχήμα 15: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 16: Δοκιμή του endpoint](#)

[Σχήμα 17: Αποκρίσεις του συστήματος](#)

[Σχήμα 18: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 19: Δοκιμή του endpoint](#)

[Σχήμα 20: Αποκρίσεις του συστήματος](#)

[Σχήμα 21: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 22: Δοκιμή του endpoint](#)

[Σχήμα 22: Αποκρίσεις του συστήματος](#)

[Σχήμα 23: Μοντέλο δεδομένων Appointment](#)

[Σχήμα 24: Μοντέλο δεδομένων Appointments](#)



[Σχήμα 25: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 26: Δοκιμή του endpoint](#)

[Σχήμα 27: Αποκρίσεις του συστήματος](#)

[Σχήμα 28: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 29: Δοκιμή του endpoint](#)

[Σχήμα 30: Αποκρίσεις του συστήματος](#)

[Σχήμα 31: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 32: Δοκιμή του endpoint και αποκρίσεις του συστήματος](#)

[Σχήμα 33: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 34: Δοκιμή του endpoint](#)

[Σχήμα 35: Αποκρίσεις του συστήματος](#)

[Σχήμα 36: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 37: Δοκιμή του endpoint](#)

[Σχήμα 38: Αποκρίσεις του συστήματος](#)

[Σχήμα 39: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 40: Δοκιμή του endpoint](#)

[Σχήμα 41: Αποκρίσεις του συστήματος](#)

[Σχήμα 42: Μοντέλο δεδομένων Review](#)

[Σχήμα 43: Μοντέλο δεδομένων Reviews](#)

[Σχήμα 44: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 45: Δοκιμή του endpoint](#)

[Σχήμα 46: Αποκρίσεις του συστήματος](#)

[Σχήμα 47: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 48: Δοκιμή του endpoint](#)

[Σχήμα 49: Αποκρίσεις του συστήματος](#)

[Σχήμα 50: Μοντέλο δεδομένων Report](#)

[Σχήμα 51: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 52: Δοκιμή του endpoint](#)

[Σχήμα 53: Αποκρίσεις του συστήματος](#)

[Σχήμα 54: Μοντέλο δεδομένων Doctors](#)

[Σχήμα 55: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 56: Δοκιμή του endpoint](#)

[Σχήμα 57: Αποκρίσεις του συστήματος](#)

[Σχήμα 58: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 59: Δοκιμή του endpoint](#)

[Σχήμα 60: Αποκρίσεις του συστήματος](#)

[Σχήμα 61: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 62: Δοκιμή του endpoint](#)

[Σχήμα 63: Αποκρίσεις του συστήματος](#)

[Σχήμα 64: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 65: Δοκιμή του endpoint](#)

[Σχήμα 66: Αποκρίσεις του συστήματος](#)



## Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8<sup>ο</sup> Εξάμηνο

Άνοιξη 2022

---

[Σχήμα 67: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 68: Δοκιμή του endpoint](#)

[Σχήμα 69: Αποκρίσεις του συστήματος](#)

[Σχήμα 70: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 71: Δοκιμή του endpoint](#)

[Σχήμα 72: Αποκρίσεις του συστήματος](#)

[Σχήμα 73: Εξήγηση και παράθεση των παραμέτρων του πόρου](#)

[Σχήμα 74: Δοκιμή του endpoint](#)

[Σχήμα 75: Αποκρίσεις του συστήματος](#)



# 1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν

## 1.1 Proxy design pattern

Για να ικανοποιηθεί η ΜΛΑ-2 (i.e. **ΜΛΑ-2** : Το σύστημα πρέπει να είναι σύμφωνο με τους ισχύοντες κανονισμούς προστασίας των δεδομένων που τίθενται από το GDPR.), χρησιμοποιείται το δομικό πρότυπο Proxy. Με το πρότυπο αυτό παρέχεται η απαιτούμενη ασφάλεια και αξιοπιστία στο σύστημα. Ειδικότερα, με την εφαρμογή του Protection Proxy η κλάση ProxyAppointment επιτρέπει την προσπέλαση των αντικειμένων RealAppointment μόνο σε ταυτοποιημένους χρήστες. Ένα αντικείμενο RealAppointment είναι προσπελάσιμο μόνο από τους διαχειριστές του συστήματος και τον γιατρό και ασθενή που αφορά το συγκεκριμένο Appointment (ενδεικτικά, η ταυτοποίηση για τις παραπάνω κατηγορίες χρηστών μπορεί να γίνει μέσω των ιδιοτήτων appPatientID, appDoctorID και μιας συνάρτησης isAdmin). Επιπλέον, τα δικαιώματα πρόσβασης διαφοροποιούνται ανάλογα με το είδος χρήστη που προσπελαίνει το αντικείμενο. Συμπληρωματικά, μπορεί να εφαρμοστεί και το Virtual Proxy έτσι ώστε να εμφανίζεται ένα αντικείμενο-εικόνα έως ότου φορτωθεί το RealAppointment από τη βάση δεδομένων.



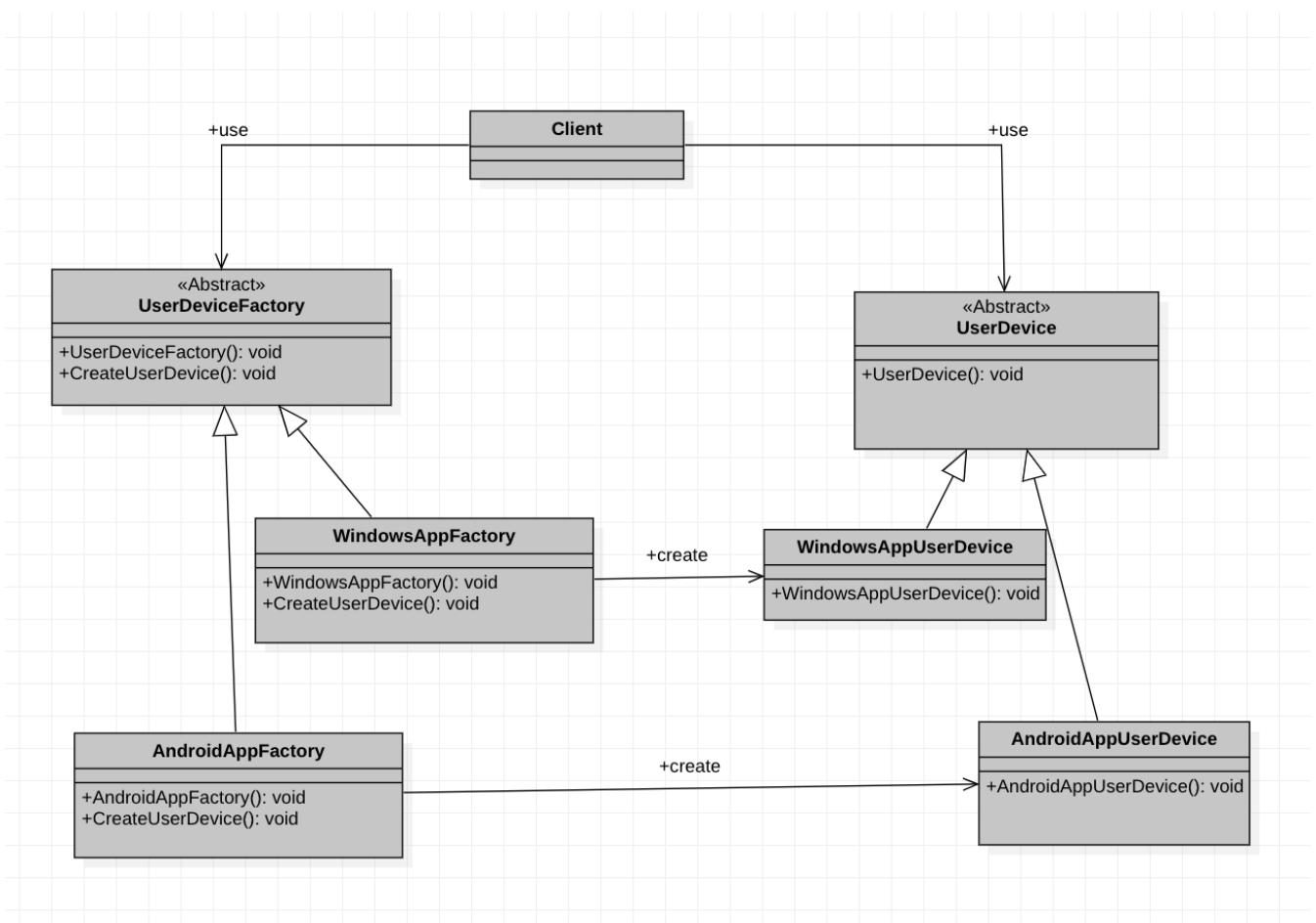
Σχήμα 1: Εφαρμογή προτύπου Proxy





## 1.2 Abstract Factory design pattern

Προκειμένου να ικανοποιηθεί η ΜΛΑ-2 (I.e. **ΜΛΑ-2** : Το σύστημα πρέπει να λειτουργεί χωρίς προβλήματα σε κάθε συσκευή που χρησιμοποιεί λειτουργικό σύστημα windows 7 ή πιο πρόσφατη έκδοση και android 8 ή πιο πρόσφατη έκδοση), χρησιμοποιείται το δημιουργικό πρότυπο Abstract Factory. Με το πρότυπο αυτό, δίνουμε τη δυνατότητα στο σύστημα μας να υποστηρίζεται από μια οικογένεια προϊόντων. Το λογισμικό του client δημιουργεί concrete υλοποιήσεις της abstract κλάσης UserDeviceFactory, και ανάλογα ποια υλοποίηση έχουμε, καλείται διαφορετική εκδοχή της createUserDevice() για να δημιουργήσει concrete υλοποίηση της abstract UserDevice. Το πρότυπο αυτό μας προσφέρει και επεκτασιμότητα σχεδίασης, έτσι ώστε αν θελήσουμε το σύστημα μας να υποστηριχθεί από κάποια νέα συσκευή, να υπάρχει αυτή η δυνατότητα μέσω της δημιουργίας των αντίστοιχων concrete κλάσεων.



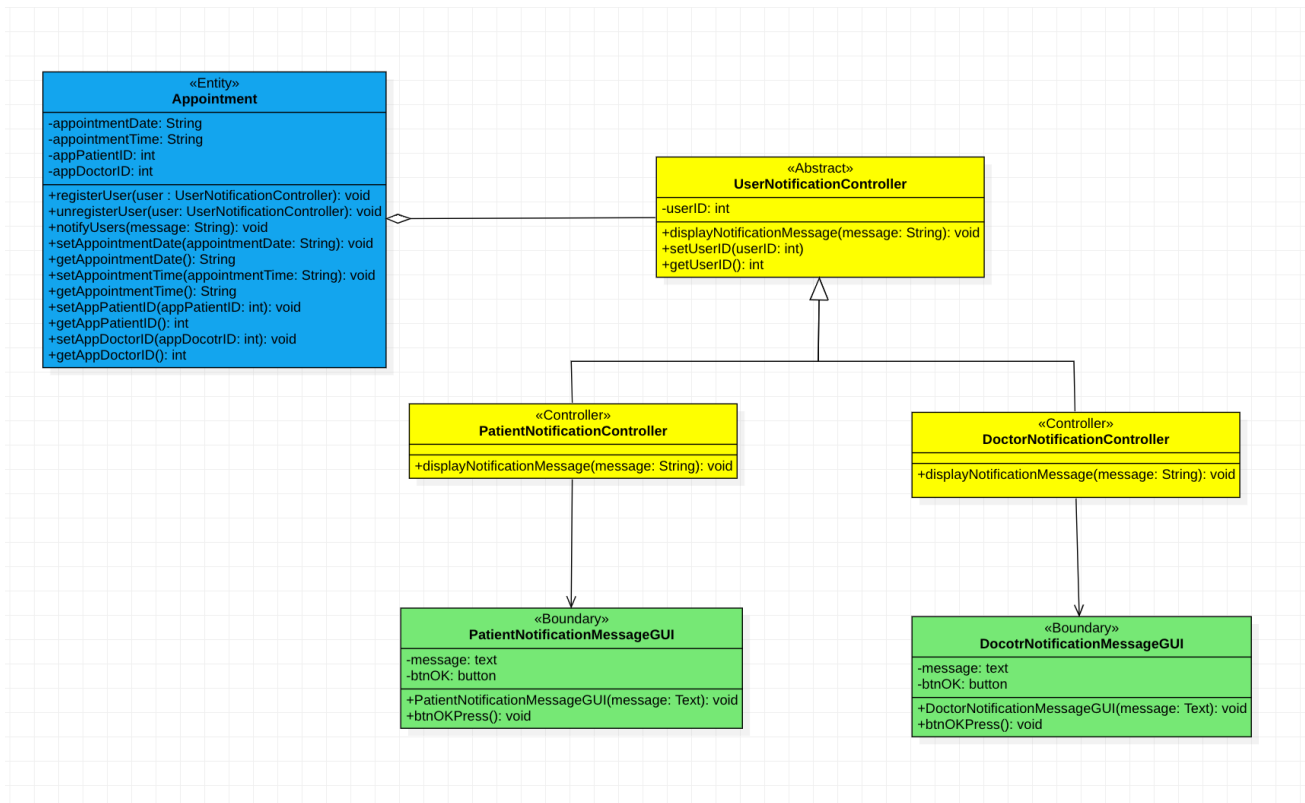
Σχήμα 2: Εφαρμογή προτύπου Abstract Factory



### 1.3 Observer design pattern

Στην προκειμένη περίπτωση χρησιμοποιείται ένα πρότυπο συμπεριφοράς για την ικανοποίηση μιας λειτουργικής απαίτησης (i.e. **ΛΑ-3** : *Το σύστημα ειδοποιήσεων πρέπει να μπορεί να στέλνει ειδοποιήσεις στο χρήστη*). Συγκεκριμένα, όποτε ανανεώνεται η κατάσταση ενός ραντεβού (e.g. Αλλαγή ώρας, αλλαγή ημέρας) ενημερώνονται οι χρήστες (Ασθενής και Γιατρός) που εμπλέκονται με το εν λόγω ραντεβού. Αυτό επιτυγχάνεται μέσω της συνάρτησης `notifyUsers`, η οποία καλεί για κάθε χρήστη τη συνάρτηση `displayNotificationMessage`, που με τη σειρά της δημιουργεί το αντίστοιχο `PatientNotificationMessageGUI` ή `DoctorNotificationMessageGUI` object και ανακατευθύνει τον εκάστοτε χρήστη σε μια οθόνη όπου εμφανίζεται σχετικό μήνυμα για την αλλαγή κατάστασης του ραντεβού (Το μήνυμα διαφέρει ανάλογα με τον τύπο χρήστη). Σε περίπτωση που κάποιος από τους εμπλεκόμενους χρήστες είναι offline, η οθόνη `NotificationMessageGUI` (και κατ' επέκταση το μήνυμα ενημέρωσης) θα εμφανιστεί στο χρήστη μόλις ξανασυνδεθεί στην εφαρμογή.

Σημείωση: Οι παρακάτω κλάσεις ενδέχεται να αποτελούνται και από άλλες ιδιότητες και μεθόδους, οι οποίες παραλείπονται καθώς επικεντρωνόμαστε στην λειτουργικότητα του pattern.



Σχήμα 3: Εφαρμογή προτύπου Observer



## 2. Αρχιτεκτονική Συστήματος

- [API στο Swaggerhub](#)
- [Αρχείο τύπου JSON με τις προδιαγραφές του API](#)
- [Αρχείο τύπου zip που περιέχει τον κώδικα για τον server της εφαρμογής](#)

### 2.1 Αναγνώριση Πόρων (Resources) Συστήματος

Με βάση τις επιταγές της αρχιτεκτονικής REST αναγνωρίστε τους Πόρους του συστήματός σας (one-off, δεδομένων και αλγοριθμικούς – όπως ορίζονται στη διάλεξη «Εξειδικευμένοι Τύποι Αρχιτεκτονικής Client Server: REST») αξιοποιώντας τις κλάσεις που αναγνωρίσατε στο προηγούμενο παραδοτέο και καταγράψτε τους σε μορφή Πίνακα όπως ο ακόλουθος (χάριν σύνοψης).

Κλάση BEC	Πόρος REST	Endpoints (HTTP Verbs)
Patient	/patient	POST
Patient	/patient/{patientId}	POST, PUT
Doctor	/doctor	POST
Doctor	/doctor/{doctorId}	POST, PUT
Appointment, (Patient)	/appointments/patient/{patientId}	POST, GET
Appointment, (Patient)	/appointments/patient/{patientId}/ /{appointmentId}	DELETE
Appointment, (Doctor)	/appointments/doctor/{doctorId}	PUT, GET
Appointment, (Doctor)	/appointments/doctor/{doctorId}/ /{appointmentId}	PUT
Review, (Doctor)	/reviews/{doctorId}	POST, GET
Report	/reports	POST
None	/search	GET
(Patient)	/notifications/{patientId}	POST, GET, DELETE
(Doctor)	/notifications/{doctorId}	POST, GET, DELETE



## 2.2 Τεκμηρίωση REST διεπαφής

### 2.2.1 Πόρος Patient

#### 2.2.1.1 Μοντέλο δεδομένων Patient

<b>Patient</b> ▾ {	
email*	string
password*	string(\$password)
firstName*	string
lastName*	string
age*	integer
	minimum: 12
	maximum: 100
sex*	string
profileImage	string(\$binary)
phoneNumber*	string
address*	string
patientId*	integer(\$int64)
}	

Σχήμα 4: Μοντέλο δεδομένων Patient

#### 2.2.1.2 Endpoint POST πόρου Patient

**POST** /patient

FR1 - A user must be able to enter his details to the system.

**Parameters** Cancel

No parameters

**Request body** required application/json ▾

The patient that will be created.

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string"
}
```

Execute Clear

Σχήμα 5: Εξήγηση και παράθεση των παραμέτρων του πόρου



Responses

Curl

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "email": "string",
    "password": "string",
    "firstName": "string",
    "lastName": "string",
    "age": 100,
    "sex": "string",
    "profileImage": "string",
    "phoneNumber": "string",
    "address": "string"
  }'
```

Request URL

https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient

Server response

Code	Details
201	<p>Response body</p> <pre>{   "email": "string",   "password": "string",   "firstName": "string",   "lastName": "string",   "age": 0,   "sex": "string",   "phoneNumber": "string",   "address": "string",   "patientId": 0 }</pre> <p>Response headers</p> <pre>content-length: 113 content-type: application/json;charset=utf-8</pre>

Σχήμα 6: Δοκιμή του endpoint

Responses

Code	Description	Links
201	Patient account created.	No links
400	Bad input parameters	No links
404	Page not found.	No links
default	unexpected error	No links

Media type: application/json

Example Value | Schema

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 0
}
```

Media type: application/json

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 7: Αποκρίσεις του συστήματος



### 2.2.1.3 Endpoint POST πόρου Patient, για συγκεκριμένο Patient Id

**POST** /patient/{patientId} Testing purposes

FR1 - A user must be able to enter his details to the system.

**Parameters** Cancel Reset

Name	Description
<b>patientId</b> * required	ID of the patient to add
integer(\$int64)	
(path)	

**Request body** required application/json

The data of the patient.

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 87654321
}
```

Execute Clear

Σχήμα 8: Εξήγηση και παράθεση των παραμέτρων του πόρου

**Responses**

**Curl**

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient/87654321' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "email": "string",
    "password": "string",
    "firstName": "string",
    "lastName": "string",
    "age": 100,
    "sex": "string",
    "profileImage": "string",
    "phoneNumber": "string",
    "address": "string",
    "patientId": 87654321
  }'
```

**Request URL**

https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient/87654321

**Server response**

**Code** **Details**

201

**Response body**

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 0,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 0
}
```

Download

**Response headers**

```
content-length: 113
content-type: application/json;charset=utf-8
```

Σχήμα 9: Δοκιμή του endpoint



Responses		
Code	Description	Links
201	Patient account created.	No links
<div>Media type: <input type="text" value="application/json"/></div> <div>Controls Accept header:</div> <div>Example Value   Schema</div> <pre>{  "email": "string",  "password": "string",  "firstName": "string",  "lastName": "string",  "age": 100,  "sex": "string",  "profileImage": "string",  "phoneNumber": "string",  "address": "string",  "patientId": 0}</pre>		
400	Bad input parameters.	No links
404	Page not found.	No links
default	unexpected error	No links
<div>Media type: <input type="text" value="application/json"/></div> <div>Example Value   Schema</div> <pre>{  "code": 0,  "message": "string"}</pre>		

Σχήμα 10: Αποκρίσεις του συστήματος

#### 2.2.1.4 Endpoint PUT πόρου Patient, για συγκεκριμένο Patient Id

PUT

/patient/{patientId}

FR2 - A user must be able to edit his details. FR13 - A user must be able to edit his profile photos.

Parameters

Cancel

Reset

Name	Description
patientId <sup>required</sup>	ID of the patient to edit
integer (\$int64)	
(path)	<input type="text" value="87654321"/>

Request body <sup>required</sup>

The new data of the patient.

```
{  "email": "string",  "password": "string",  "firstName": "string",  "lastName": "string",  "age": 100,  "sex": "string",  "profileImage": "string",  "phoneNumber": "string",  "address": "string",  "patientId": 87654321}
```

Execute

Clear

Σχήμα 11: Εξήγηση και παράθεση των παραμέτρων του πόρου



Responses

Curl

```
curl -X 'PUT' \
'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient/87654321' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 87654321
}'
```

Request URL

https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient/87654321

Server response

Code Details

200

Response headers

```
content-length: 20
content-type: application/json;charset=utf-8
```

Σχήμα 12: Δοκιμή του endpoint

Responses

Code	Description	Links
200	Patient account successfully edited.	No links
201	Patient account created.	No links
400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

application/json

Controls Accept header.

Example Value Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 13: Αποκρίσεις του συστήματος





## 2.2.2 Πόρος Doctor

### 2.2.2.1 Μοντέλο δεδομένων Doctor

Doctor ▾ {	
email*	string
password*	string(\$password)
firstName*	string
lastName*	string
age*	integer
	minimum: 12
	maximum: 100
sex*	string
profileImage	string(\$binary)
phoneNumber*	string
address*	string
costPerVisit*	integer
officeAddress*	string
specialty*	string
doctorBio	string
workingDays*	> [...]
workingHours*	> [...]
doctorId*	integer(\$int64)
officeBackgroundImg	string(\$binary)
}	

Σχήμα 14: Μοντέλο δεδομένων Doctor

### 2.2.2.2 Endpoint POST πόρου Doctor

**POST** /doctor

FR1 - A user must be able to enter his details to the system.

Parameters

No parameters

Request body required

application/json

Doctor account to be created.

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true,
    true,
    true,
    true,
    true
  ]
}
```

Execute

Clear

Σχήμα 15: Εξήγηση και παράθεση των παραμέτρων του πόρου



Server response

Code	Details
201	<p>Response body</p> <pre>{   "email": "string",   "password": "string",   "firstName": "string",   "lastName": "string",   "age": 0,   "sex": "string",   "phoneNumber": "string",   "address": "string",   "costPerVisit": 0,   "officeAddress": "string",   "specialty": "string",   "doctorBio": "string",   "workingDays": [     true   ],   "workingHours": [     true   ],   "doctorId": 0 }</pre> <p>Response headers</p> <pre>content-length: 172 content-type: application/json; charset=utf-8</pre>

Σχήμα 16: Δοκιμή του endpoint

Responses

Code	Description	Links
201	Doctor account created.	No links
400	Bad input parameters	No links
404	Page not found.	No links
default	unexpected error	No links

Media type:

Controls: Accept header.

Example Value | Schema

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true,
    true,
    true,
    true,
    true,
    true,
    true
  ],
}
```

Media type:

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 17: Αποκρίσεις του συστήματος



### 2.2.2.3 Endpoint POST πόρου Doctor, για συγκεκριμένο doctorId

**POST** /doctor/{doctorId} Testing purposes

FR1 - A user must be able to enter his details to the system.

**Parameters** Cancel

Name	Description
<b>doctorId</b> * required	ID of the doctor to add
integer(\$int64)	
(path)	<input type="text" value="12345678"/>

**Request body** required application/json

The data of the doctor.

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true,
    true,
    true,
    true,
    true
  ]
}
```

Execute Clear

Σχήμα 18: Εξήγηση και παράθεση των παραμέτρων του πόρου

**Server response**

Code	Details
201	<p><b>Response body</b></p> <pre>{   "email": "string",   "password": "string",   "firstName": "string",   "lastName": "string",   "age": 0,   "sex": "string",   "phoneNumber": "string",   "address": "string",   "costPerVisit": 0,   "officeAddress": "string",   "specialty": "string",   "doctorBio": "string",   "workingDays": [     true   ],   "workingHours": [     true   ],   "doctorId": 0 }</pre> <span>Download</span>

**Response headers**

```
content-length: 172
content-type: application/json;charset=utf-8
```

Σχήμα 19: Δοκιμή του endpoint



The screenshot displays a REST client interface with a table of responses. The first response is a 201 status code with the description 'Doctor account created.' and no links. Below this, the media type is set to 'application/json'. An example value is shown as a JSON object with fields: email, password, firstName, lastName, age, sex, profileImage, phoneNumber, address, costPerVisit, officeAddress, specialty, doctorBio, and workingDays. The second response is a 400 status code with the description 'Bad input parameters.' and no links. The third response is a 404 status code with the description 'Page not found.' and no links. The default response is an 'unexpected error' with no links. Below this, the media type is again set to 'application/json', and an example value is shown as a JSON object with fields: code and message.

Code	Description	Links
201	Doctor account created.	No links
400	Bad input parameters.	No links
404	Page not found.	No links
default	unexpected error	No links

Σχήμα 20: Αποκρίσεις του συστήματος

#### 2.2.2.4 Endpoint PUT πόρου Doctor, για συγκεκριμένο doctorId

The screenshot shows a REST client interface for a PUT endpoint. The endpoint is '/doctor/{doctorId}'. The description is 'FR2 - A user must be able to edit his details. FR13 - A user must be able to edit his profile photos.' Below this, the 'Parameters' section shows a required parameter 'doctorId' of type 'integer(\$int64)' with a value of '12345678'. The 'Request body' section is set to 'application/json' and contains a JSON object with fields: email, password, firstName, lastName, age, sex, profileImage, phoneNumber, address, costPerVisit, officeAddress, specialty, doctorBio, and workingDays. At the bottom, there are 'Execute' and 'Clear' buttons.

PUT /doctor/{doctorId}

FR2 - A user must be able to edit his details. FR13 - A user must be able to edit his profile photos.

Parameters

doctorId \* required  
integer(\$int64)  
(path)  
12345678

Request body required  
application/json

The new data of the patient.

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true,
    true,
    true,
    true,
    true,
    true
  ]
}
```

Execute Clear

Σχήμα 21: Εξήγηση και παράθεση των παραμέτρων του πόρου



Server response

Code	Details
200	<p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With,Content-Type,Accept,Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json;charset=utf-8 date: Tue,31 May 2022 10:15:11 GMT etag: W/"14-4DhJ6nhrn3soo1wXlJ6FqT6xz/E" expires: -1 status: 200 OK</pre> <p>Request duration</p> <p>758 ms</p>

Σχήμα 22: Δοκιμή του endpoint

Responses

Code	Description	Links
200	Doctor account successfully edited	No links
201	Doctor account created	No links
400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 22: Αποκρίσεις του συστήματος



## 2.2.3 Πόρος Appointment

### 2.2.3.1 Μοντέλο δεδομένων Appointment

```
Appointment {  
  appPatientId* integer($int64)  
  appDoctorId* integer($int64)  
  appointmentDate* string($date)  
  appointmentTime* string  
  appointmentId* integer($int64)  
  isBooked boolean  
}
```

Σχήμα 23: Μοντέλο δεδομένων Appointment

### 2.2.3.2 Μοντέλο δεδομένων Appointments (Λίστα με αντικείμενα τύπου Appointment)

```
Appointments {  
  Appointments {  
    appPatientId* integer($int64)  
    appDoctorId* integer($int64)  
    appointmentDate* string($date)  
    appointmentTime* string  
    appointmentId* integer($int64)  
    isBooked boolean  
  }  
}
```

Σχήμα 24: Μοντέλο δεδομένων Appointments

### 2.2.3.3 Endpoint POST πόρου appointments, για ασθενή και συγκεκριμένο patientId

**POST** /appointments/patient/{patientId}

FR4 - A patient must be able to book an appointment.

**Parameters**

Cancel

Name	Description
<b>patientId</b> * required integer(\$int64) (path)	ID of the patient

87654321

**Request body** required

application/json

The chosen appointment.

```
{  
  "appPatientId": 0,  
  "appDoctorId": 0,  
  "appointmentDate": "2022-05-31",  
  "appointmentTime": "string"  
}
```

Execute

Clear

Σχήμα 25: Εξήγηση και παράθεση των παραμέτρων του πόρου



## Responses

Curl

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "appPatientId": 0,
    "appDoctorId": 0,
    "appointmentDate": "2022-05-31",
    "appointmentTime": "string"
  }'
```

Request URL

https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321

Server response

Code Details

201

Response body

```
{
  "appPatientId": 0,
  "appDoctorId": 0,
  "appointmentDate": "2015-07-20",
  "appointmentTime": "string",
  "appointmentId": 0,
  "isBooked": true
}
```



Download

Response headers

```
content-length: 110
content-type: application/json;charset=utf-8
```

Σχήμα 26: Δοκιμή του endpoint

Responses		
Code	Description	Links
201	Doctor account created.	No links
<div>Media type application/json</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <pre>{   "appPatientId": 0,   "appDoctorId": 0,   "appointmentDate": "2022-05-31",   "appointmentTime": "string",   "appointmentId": 0,   "isBooked": true }</pre>		
400	Bad input parameters	No links
404	Page not found.	No links
default	unexpected error	No links
<div>Media type application/json</div> <div>Example Value   Schema</div> <pre>{   "code": 0,   "message": "string" }</pre>		

Σχήμα 27: Αποκρίσεις του συστήματος



### 2.2.3.4 Endpoint GET πόρου appointments, για ασθενή και συγκεκριμένο patientId

GET /appointments/patient/{patientId}

FR8 - A user must be able to view his appointments.

Parameters

Name	Description
patientId * required	ID of the patient.
integer(\$int64)	
(path)	

87654321

Execute Clear

Σχήμα 28: Εξήγηση και παράθεση των παραμέτρων του πόρου

Responses

Curl

```
curl -X 'GET' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321' \
  -H 'accept: application/json'
```

Request URL

https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321

Server response

Code	Details
200	<p>Response body</p> <pre>{   "appPatientId": 0,   "appDoctorId": 0,   "appointmentDate": "2015-07-20",   "appointmentTime": "string",   "appointmentId": 0,   "isBooked": true }</pre> <p>Response headers</p> <pre>content-length: 114 content-type: application/json; charset=utf-8</pre>

Σχήμα 29: Δοκιμή του endpoint

Responses

Code	Description	Links
200	List with patients appointments	No links
400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links

Media type: application/json

Example Value

```
{
  "appPatientId": 0,
  "appDoctorId": 0,
  "appointmentDate": "2022-05-31",
  "appointmentTime": "string",
  "appointmentId": 0,
  "isBooked": true
}
```

Media type: application/json

Example Value

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 30: Αποκρίσεις του συστήματος





### 2.2.3.5 Endpoint DELETE πόρου appointments, για ασθενή και συγκεκριμένα patientId, appointmentId.

**DELETE** /appointments/patient/{patientId}/{appointmentId}

FR5 - A user must be able to cancel an appointment.

**Parameters** Cancel

Name	Description
<b>patientId</b> * required integer(\$int64) (path)	ID of the patient
<b>appointmentId</b> * required integer(\$int64) (path)	ID of the appointment to be cancelled

Execute Clear

Σχήμα 31: Εξήγηση και παράθεση των παραμέτρων του πόρου

**Responses**

**Curl**  

```
curl -X 'DELETE' \
'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321/11223344' \
-H 'accept: application/json'
```

**Request URL**  

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321/11223344
```

**Server response**

Code	Details
200	<b>Response headers</b> <pre>content-length: 20 content-type: application/json; charset=utf-8</pre>

**Responses**

Code	Description	Links
200	Appointment successfully cancelled.	No links
400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links

**Media type**  
application/json

**Example Value** | **Schema**

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 32: Δοκίμηση του endpoint και αποκρίσεις του συστήματος



### 2.2.3.6 Endpoint PUT πόρου appointments, για γιατρό και συγκεκριμένο doctorId.

**PUT** `/appointments/doctor/{doctorId}`

FR4 - A user must be able to enter his details. FR12 - A doctor must be able to edit his available for booking appointments.

Parameters

Cancel

Name	Description
<b>doctorId</b> * required <code>integer(\$int64)</code> (path)	ID of the patient <input type="text" value="12345678"/>

Request body required

application/json

The chosen appointments.

```
[
  {
    "appPatientId": 0,
    "appDoctorId": 0,
    "appointmentDate": "2022-05-31",
    "appointmentTime": "string",
    "appointmentId": 0,
    "isBooked": true
  }
]
```

Execute

Clear

Σχήμα 33: Εξήγηση και παράθεση των παραμέτρων του πόρου



Server response

Code	Details
200	<p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With, Content-Type, Accept, Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json; charset=utf-8 date: Tue, 31 May 2022 15:23:06 GMT etag: W/"14-4DhJ6nhrn3soo1wXlJ6FqT6xz/E" expires: -1 status: 200 OK</pre> <p>Request duration</p> <p>324 ms</p>

Σχήμα 34: Δοκιμή του endpoint

Responses

Code	Description	Links
200	Doctor available appointments uploaded.	No links
201	Doctor available appointments uploaded.	No links
400	Bad input parameters	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 35: Αποκρίσεις του συστήματος



### 2.2.3.7 Endpoint GET πόρον appointments, για γιατρό και συγκεκριμένο doctorId.

GET

/appointments/doctor/{doctorId}

^

FR8 - A user must be able to view his appointments.

Parameters

Cancel

Name	Description
<b>doctorId</b> * required	ID of the doctor.
integer(\$int64)	
(path)	
	<input type="text" value="12345678"/>

Execute

Clear

Σχήμα 36: Εξήγηση και παράθεση των παραμέτρων του πόρου

Responses

Curl

```
curl -X 'GET' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/doctor/12345678' \
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/doctor/12345678
```

Server response

Code	Details
200	<div>Response body</div> <pre>[   {     "appPatientId": 0,     "appDoctorId": 0,     "appointmentDate": "2015-07-20",     "appointmentTime": "string",     "appointmentId": 0,     "isBooked": true   } ]</pre> <div><div>Download</div></div>

Response headers

Σχήμα 37: Δοκιμή του endpoint



Responses		
Code	Description	Links
200	List with doctors appointments	No links
Media type: application/json		
Controls: Accept header		
Example Value   Schema		
<pre>{   "appPatientId": 0,   "appDoctorId": 0,   "appointmentDate": "2022-05-31",   "appointmentTime": "string",   "appointmentId": 0,   "isBooked": true }</pre>		
400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links
Media type: application/json		
Example Value   Schema		
<pre>{   "code": 0,   "message": "string" }</pre>		

Σχήμα 38: Αποκρίσεις του συστήματος

### 2.2.3.9 Endpoint PUT πόρου appointments, για γιατρό και συγκεκριμένα doctorId και appointmentId.

PUT

/appointments/doctor/{doctorId}/{appointmentId}

FR5 - A user must be able to cancel an appointment. FR14 - A doctor must be able to mark an available for booking appointment as booked.

Parameters

Cancel

Name	Description
<b>doctorId</b> * required	ID of the doctor
integer(\$int64)	
(path)	12345678
<b>appointmentId</b> * required	ID of the appointment
integer(\$int64)	
(path)	11223344

Request body required

application/json

The appointment to be changed.

```
{
  "appPatientId": 0,
  "appDoctorId": 0,
  "appointmentDate": "2022-05-31",
  "appointmentTime": "string",
  "appointmentId": 0,
  "isBooked": true
}
```

Execute

Clear

Σχήμα 39: Εξήγηση και παράθεση των παραμέτρων του πόρου



## Responses

### Curl

```
curl -X 'PUT' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/doctor/12345678/11223344' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "appPatientId": 0,
    "appDoctorId": 0,
    "appointmentDate": "2022-05-31",
    "appointmentTime": "string",
    "appointmentId": 0,
    "isBooked": true
  }'
```

### Request URL

https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/doctor/12345678/11223344

Σχήμα 40: Δοκιμή του endpoint

## Responses

Code	Description	Links
200	Appointment availability successfully changed.	No links
201	Appointment availability successfully created.	No links
400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 41: Αποκρίσεις του συστήματος



## 2.2.4 Πόρος Review

### 2.2.4.1 Μοντέλο δεδομένων Review

```
Review {  
  imgProfile      string($binary)  
  txtName*        string  
  imgHeart        string($binary)  
  txtRating*      string  
  txtDay          string  
  txtReview       string  
}
```

Σχήμα 42: Μοντέλο δεδομένων Review

### 2.2.4.2 Μοντέλο δεδομένων Reviews (Λίστα με αντικείμενα τύπου Review)

```
Reviews [Reviews {  
  imgProfile      string($binary)  
  txtName*        string  
  imgHeart        string($binary)  
  txtRating*      string  
  txtDay          string  
  txtReview       string  
}]
```

Σχήμα 43: Μοντέλο δεδομένων Reviews

### 2.2.4.3 Endpoint POST πόρου reviews, για συγκεκριμένο doctorId.

**POST** /reviews/{doctorId}

FR6 - A patient must be able to write a review about the doctor following the date of the booked appointment.

**Parameters**

Cancel

Name	Description
<b>doctorId</b> * required integer(\$int64) (path)	ID of the doctor being reviewed.

**Request body** \* required

application/json

The review to be submitted.

```
{  
  "imgProfile": "string",  
  "txtName": "string",  
  "imgHeart": "string",  
  "txtRating": "string",  
  "txtDay": "string",  
  "txtReview": "string"  
}
```

Execute

Clear

Σχήμα 44: Εξήγηση και παράθεση των παραμέτρων του πόρου



Responses

Curl

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reviews/12345678' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "imgProfile": "string",
    "txtName": "string",
    "imgHeart": "string",
    "txtRating": "string",
    "txtDay": "string",
    "txtReview": "string"
  }'
```

Request URL

https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reviews/12345678

Server response

Code	Details
201	<p>Response headers</p> <pre>content-length: 20 content-type: application/json;charset=utf-8</pre>

Σχήμα 45: Δοκιμή του endpoint

Responses

Code	Description	Links
201	The review submitted.	No links
400	Bad input parameters	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 46: Αποκρίσεις του συστήματος

#### 2.2.4.4 Endpoint GET πόρου reviews, για συγκεκριμένο doctorId.

GET /reviews/{doctorId}

FR7 - A patient must be able to view doctor's reviews

Parameters

Name	Description
doctorid * required	ID of the doctor being reviewed.
integer(\$int64)	
(path)	12345678

Execute Clear

Σχήμα 47: Εξήγηση και παράθεση των παραμέτρων του πόρου





Responses

Curl

```
curl -X 'GET' \
'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reviews/12345678' \
-H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reviews/12345678
```

Server response

Code	Details
200	<p>Response body</p> <pre>[   {     "txtName": "string",     "txtRating": "string",     "txtDay": "string",     "txtReview": "string"   } ]</pre> <p>Response headers</p> <pre>content-length: 72 content-type: application/json;charset=utf-8</pre>

Σχήμα 48: Δοκιμή του endpoint

Responses

Code	Description	Links
200	<p>Reviews are ready for reading</p> <p>Media type: <input type="text" value="application/json"/></p> <p>Controls Accept header.</p> <p>Example Value   Schema</p> <pre>[   {     "imgProfile": "string",     "txtName": "string",     "imgHeart": "string",     "txtRating": "string",     "txtDay": "string",     "txtReview": "string"   } ]</pre>	No links
400	Bad input parameter	No links
404	Page not found.	No links
default	<p>unexpected error</p> <p>Media type: <input type="text" value="application/json"/></p> <p>Example Value   Schema</p> <pre>{   "code": 0,   "message": "string" }</pre>	No links

Σχήμα 49: Αποκρίσεις του συστήματος



## 2.2.5 Πόρος Report

### 2.2.5.1 Μοντέλο δεδομένων Report

```
Report {  
  reportText*      string  
                  maxLength: 400  
  reportingUserId*  string  
  submittedByDoctor boolean  
}
```

Σχήμα 50: Μοντέλο δεδομένων Report

### 2.2.5.2 Endpoint POST πόρου reports

**POST** /reports

FR11 - A user must be able to submit a report.

**Parameters** Cancel

No parameters

**Request body** required application/json

The report to be submitted.

```
{  
  "reportText": "string",  
  "reportingUserId": "string",  
  "submittedByDoctor": true  
}
```

Execute Clear

Σχήμα 51: Εξήγηση και παράθεση των παραμέτρων του πόρου



Responses

Curl

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reports' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "reportText": "string",
    "reportingUserId": "string",
    "submittedByDoctor": true
  }'
```

Request URL

<https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reports>

Server response

Code	Details
201	<p>Response headers</p> <pre>content-length: 20 content-type: application/json;charset=utf-8</pre>

Σχήμα 52: Δοκιμή του endpoint

Responses

Code	Description	Links
201	The report submitted successfully.	No links
400	Bad input parameters	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 53: Αποκρίσεις του συστήματος



## 2.2.6 Πόρος Search

### 2.2.6.1 Μοντέλο δεδομένων Doctors

Doctors ▾ [Doctors ▾ {	
email*	string
password*	string(\$password)
firstName*	string
lastName*	string
age*	integer
	minimum: 12
	maximum: 100
sex*	string
profileImage	string(\$binary)
phoneNumber*	string
address*	string
costPerVisit*	integer
officeAddress*	string
specialty*	string
doctorBio	string
workingDays*	> [...]
workingHours*	> [...]
doctorId*	integer(\$int64)
officeBackgroundImg	string(\$binary)
}]	

Σχήμα 54: Μοντέλο δεδομένων Doctors

### 2.2.6.2 Endpoint GET πόρου search

GET /search

FR9 - A patient must be able to search for doctors.

Parameters

Name	Description
inputField string (query)	Typed in search bar <input type="text" value="keyword example"/>
visitCost array(integer) (query)	The visit cost range of the doctors. <div><input type="text" value="20"/> - <input type="text" value="40"/> <button>Add Integer Item</button></div>
rating integer (query)	Minimum doctor rating <input type="text" value="4"/>
earliestAppointment string (query)	Doctors with earliest available appointment within said time-frame <input type="text" value="earliestAppointment"/>
maxDistance string (query)	Maximum distance between patients address and doctors office address <input type="text" value="maxDistance"/>
sortingMethod string (query)	Order of doctors in the results page <div><input type="text" value="increasingCost"/></div>

Execute

Clear

Σχήμα 55: Εξήγηση και παράθεση των παραμέτρων του πόρου



Responses

Curl

```
curl -X 'GET' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/search?inputField=keyword&visitCost=20&visitCost=40&rating=4&sortingMethod=increasingCost' \
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/search?inputField=keyword&visitCost=20&visitCost=40&rating=4&sortingMethod=increasingCost
```

Server response

Code Details

200

Response body

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 0,
  "sex": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true
  ],
  "workingHours": [
    true
  ],
  "doctorId": 0
}
```

Response headers

```
content-length: 175
content-type: application/json;charset=utf-8
```

Σχήμα 56: Δοκιμή του endpoint

Responses

Code	Description	Links
200	List of doctors	No links

Media type

application/json

Example Value | Schema

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true,
    true,
    true,
    true,
    true,
    true
  ]
}
```

400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

application/json

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 57: Αποκρίσεις του συστήματος



## 2.2.7 Πόρος Notifications

### 2.2.7.1 Endpoint POST πόρου notifications, για συγκεκριμένο patientId.

POST /notifications/{patientId}

FR3 - The notification system must be able to notify the users.

Parameters

Name	Description
patientId <sup>required</sup>	ID of the patient to be notified.

integer(\$int64)  
(path)

87654321

Request body <sup>required</sup> text/plain

The notification message.

string

Execute Clear

Σχήμα 58: Εξήγηση και παράθεση των παραμέτρων του πόρου

Responses

Snippets

cURL (bash) cURL (PowerShell) cURL (CMD)

```
curl -X 'POST' \
'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/notifications/87654321' \
-H 'accept: application/json' \
-H 'Content-Type: text/plain' \
-d 'string'
```

Request URL

https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/notifications/87654321

Server response

Code	Details
201	<p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With, Content-Type, Accept, Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json; charset=utf-8 date: Tue, 31 May 2022 10:23:48 GMT etag: W/"14-4DhJ6nhn3soo1uXlJ6FqT6xz/E" expires: -1 status: 201 Created</pre> <p>Request duration</p> <p>803 ms</p>

Σχήμα 59: Δοκιμή του endpoint



Code	Description	Links
201	The notification was submitted.	No links
400	Bad input parameters	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

Controls **Accept** header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 60: Αποκρίσεις του συστήματος

### 2.2.7.2 Endpoint GET πόρου notifications, για συγκεκριμένο patientId.

**GET** /notifications/{patientId}

FR3 - The notification system must be able to notify the users.

Parameters Cancel

Name	Description
<b>patientId</b> * required	ID of the patient requesting his notification.
integer(\$int64)	
(path)	<input type="text" value="87654321"/>

Execute Clear

Σχήμα 61: Εξήγηση και παράθεση των παραμέτρων του πόρου



Server response

Code	Details
200	<p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With, Content-Type, Accept, Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json; charset=utf-8 date: Tue, 31 May 2022 10:28:07 GMT etag: W/"14-4DhJ6nhrn3soo1wXlJ6FqT6xz/E" expires: -1 status: 200 OK</pre> <p>Request duration</p> <p>279 ms</p>

Σχήμα 62: Δοκιμή του endpoint

Responses

Code	Description	Links
200	Notification Received Successfully	No links
400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 63: Αποκρίσεις του συστήματος





### 2.2.7.3 Endpoint DELETE πόρου notifications, για συγκεκριμένο patientId.

**DELETE** /notifications/{patientId}

FR3 - The notification system must be able to notify the users.

Parameters

Cancel

Name	Description
<b>patientId</b> * required	ID of the patient.
integer(\$int64)	
(path)	<input type="text" value="87654321"/>

Execute

Clear

Σχήμα 64: Εξήγηση και παράθεση των παραμέτρων του πόρου

Server response

Code	Details
200	<div>Response headers<pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With, Content-Type, Accept, Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json; charset=utf-8 date: Tue, 31 May 2022 10:30:27 GMT etag: W/"14-4DhJ6nhrn3soo1wXlJ6FqT6xz/E" expires: -1 status: 200 OK</pre></div> <div>Request duration</div> <div>1082 ms</div>

Σχήμα 65: Δοκιμή του endpoint



Responses

Code	Description	Links
200	Notification Syccesfully Deleted!	No links
400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

▼

Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 66: Αποκρίσεις του συστήματος



#### 2.2.7.4 Endpoint POST πόρου notifications, για συγκεκριμένο doctorId.

**POST** /notifications/{doctorId}

FR3 - The notification system must be able to notify the users.

Parameters

Cancel

Name	Description
<b>doctorId</b> * required	ID of the doctor to be notified.
integer(\$int64)	
(path)	
	12345678

Request body required

text/plain

The notification message.

string

Execute

Clear

Σχήμα 67: Εξήγηση και παράθεση των παραμέτρων του πόρου



Server response

Code	Details
201	<p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With, Content-Type, Accept, Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json; charset=utf-8 date: Tue, 31 May 2022 10:34:11 GMT etag: W/"14-4DhJ6nhnrn3soo1wXlJ6FqT6xz/E" expires: -1 status: 201 Created</pre> <p>Request duration</p> <p>726 ms</p>

Σχήμα 68: Δοκιμή του endpoint

Responses

Code	Description	Links
201	The notification was submitted.	No links
400	Bad input parameters	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 69: Αποκρίσεις του συστήματος



### 2.2.7.5 Endpoint GET πόρου notifications, για συγκεκριμένο doctorId.

**GET** /notifications/{doctorId}

FR3 - The notification system must be able to notify the users.

Parameters

Cancel

Name	Description
<b>doctorId</b> * required	ID of the doctor requesting his notification.
integer(\$int64)	
(path)	<input type="text" value="12345678"/>

Execute

Clear

Σχήμα 70: Εξήγηση και παράθεση των παραμέτρων του πόρου

Server response

Code	Details
200	<div>Response headers<pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With, Content-Type, Accept, Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json; charset=utf-8 date: Tue, 31 May 2022 10:35:57 GMT etag: W/"14-4DhJ6nhrn3soo1wXlJ6FqT6xz/E" expires: -1 status: 200 OK</pre></div> <div>Request duration</div> <div>798 ms</div>

Σχήμα 71: Δοκιμή του endpoint



Responses

Code	Description	Links
200	Notification Received Successfully	No links

Media type

Controls Accept header.

Example Value | Schema

```
string
```

400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

Example Value | Schema

```
{  
  "code": 0,  
  "message": "string"  
}
```

Σχήμα 72: Αποκρίσεις του συστήματος

#### 2.2.7.6 Endpoint DELETE πόρου notifications, για συγκεκριμένο doctorId.

**DELETE** /notifications/{doctorId}

FR3 - The notification system must be able to notify the users. FR10 - A patient must be able to sort the search results according to increasing/decreasing cost or doctors' ratings.

Parameters

**Cancel**

Name	Description
<b>doctorId</b> * required	ID of the doctor.
integer(\$int64)	
(path)	<input type="text" value="12345678"/>

**Execute** **Clear**

Σχήμα 73: Εξήγηση και παράθεση των παραμέτρων του πόρου



Server response

Code	Details
200	<p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With, Content-Type, Accept, Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json; charset=utf-8 date: Tue, 31 May 2022 10:37:15 GMT etag: W/"14-4DhJ6nhnrn3soo1wXlJ6FqT6xz/E" expires: -1 status: 200 OK</pre> <p>Request duration</p> <p>753 ms</p>

Σχήμα 74: Δοκιμή του endpoint

Responses

Code	Description	Links
200	Notification Syccesfully Deleted!	No links
400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Σχήμα 75: Αποκρίσεις του συστήματος



## 3. Υλοποίηση Συστήματος με Node-RED

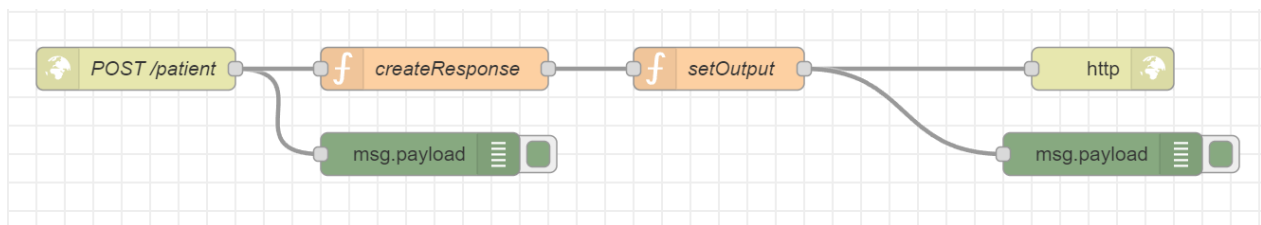
### 3.1 Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed

- [Σύνδεσμος για το αρχείο τύπου json που περιέχει τα flows.](#)

#### 3.1.1 Ροές πόρου Patient

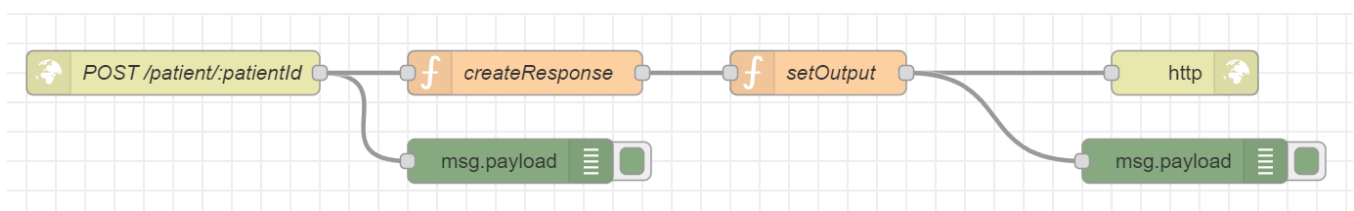
##### Ροή endpoint POST /patient

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την δημιουργία ενός πόρου ασθενή με patientId το οποίο καθορίζει ο server. Η συνάρτηση createResponse συνδυάζει τα δεδομένα εισόδου και προσθέτει το τυχαίο patientId. Η συνάρτηση setOutput θα πρέπει να αποθηκεύει την είσοδο στη βάση δεδομένων πάνω στην οποία θα υλοποιηθεί η εφαρμογή και ανάλογα με το status code που θα λαμβάνει να μεταβάλλει την έξοδο.



##### Ροή endpoint POST /patient/:patientId

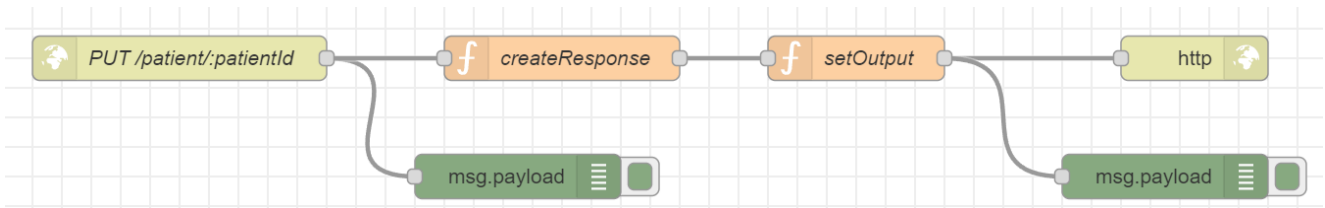
Ροή που υλοποιεί την υπηρεσία υπεύθυνη για τη δημιουργία ενός πόρου ασθενή με patientId το οποίο δίνεται από την εφαρμογή του χρήστη ( υλοποίηση για testing, όχι στην πραγματική εφαρμογή). Η συνάρτηση createResponse παίρνει τα δεδομένα εισόδου και κάνει αίτημα στη βάση δεδομένων για να αποθηκευτούν. Η συνάρτηση setOutput ανάλογα με το status code που θα λάβει θα πρέπει να μεταβάλλει αντίστοιχα την έξοδο.



##### Ροή endpoint PUT /patient/:patientId

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την επεξεργασία ενός πόρου ασθενή με patientId που δίνεται από την εφαρμογή του χρήστη. Η συνάρτηση createResponse λαμβάνει την είσοδο και στέλνει αίτημα στη βάση δεδομένων για να αποθηκευτούν στην κατάλληλη θέση. Η συνάρτηση setOutput ανάλογα με το status code που θα λάβει θα πρέπει να μεταβάλλει αντίστοιχα την έξοδο.

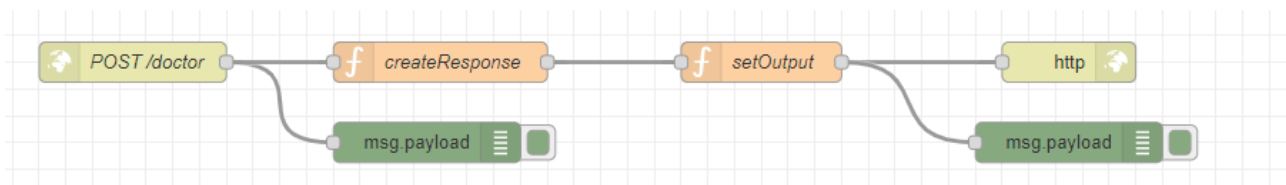




### 3.1.2 Ροές πόρου Doctor

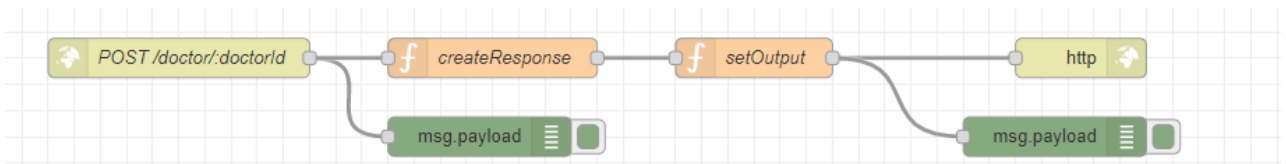
#### Ροή endpoint POST /doctor

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την δημιουργία ενός πόρου ασθενή με doctorId το οποίο καθορίζει ο server. Η συνάρτηση createResponse συνδυάζει τα δεδομένα εισόδου και προσθέτει το τυχαίο doctorId. Η συνάρτηση setOutput θα πρέπει να αποθηκεύει την είσοδο στη βάση δεδομένων πάνω στην οποία θα υλοποιηθεί η εφαρμογή και ανάλογα με το status code που θα λαμβάνει να μεταβάλλει την έξοδο.



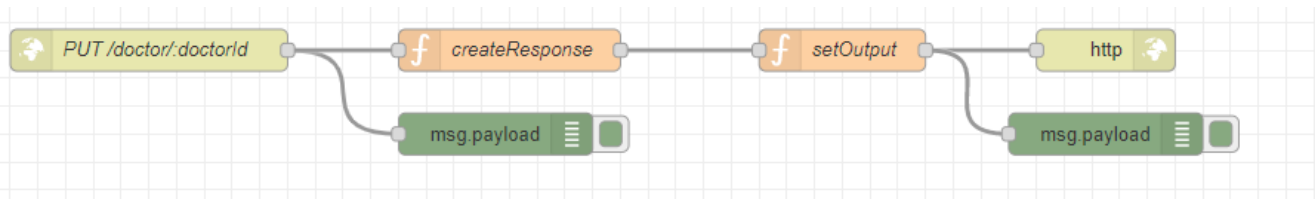
#### Ροή endpoint POST /doctor/:doctorId

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για τη δημιουργία ενός πόρου ασθενή με doctorId το οποίο δίνεται από την εφαρμογή του χρήστη ( υλοποίηση για testing, όχι στην πραγματική εφαρμογή). Η συνάρτηση createResponse παίρνει τα δεδομένα εισόδου και κάνει αίτημα στη βάση δεδομένων για να αποθηκευτούν. Η συνάρτηση setOutput ανάλογα με το status code που θα λάβει θα πρέπει να μεταβάλλει αντίστοιχα την έξοδο.



#### Ροή endpoint PUT /doctor/:doctorId

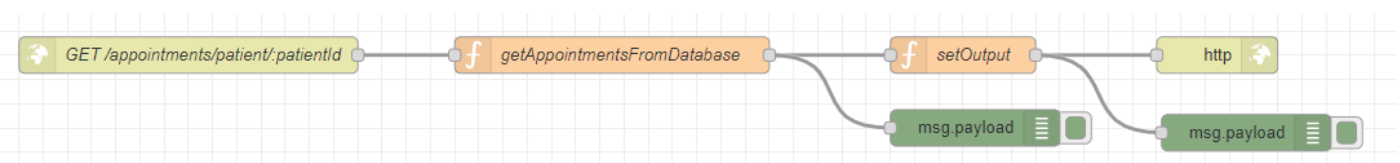
Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την επεξεργασία ενός πόρου ασθενή με doctorId που δίνεται από την εφαρμογή του χρήστη. Η συνάρτηση createResponse λαμβάνει την είσοδο και στέλνει αίτημα στη βάση δεδομένων για να αποθηκευτούν στην κατάλληλη θέση. Η συνάρτηση setOutput ανάλογα με το status code που θα λάβει θα πρέπει να μεταβάλλει αντίστοιχα την έξοδο.



### 3.1.3 Ροές πόρου Appointments

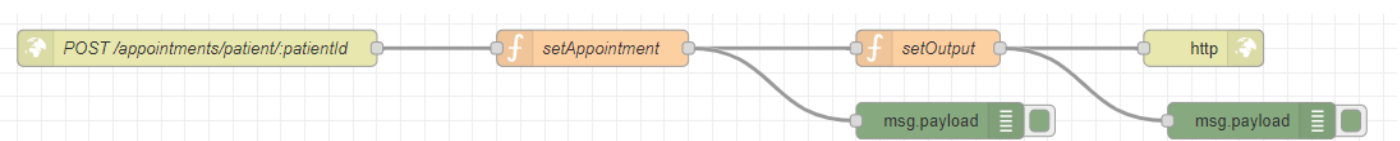
#### Ροή endpoint GET /appointments/patient/:patientId

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την επιστροφή των ραντεβού ενός χρήστη τύπου ασθενή με patientId, που είναι το μοναδικό id αυτού του ασθενή. Η συνάρτηση getAppointmentsFromDatabase αναζητεί στη βάση δεδομένων και επιστρέφει μία λίστα με τα ραντεβού του ασθενή, αν η κλήση ήταν σωστή. Η συνάρτηση setOutput βλέπει το status code που προέκυψε και ανάλογα θέτει την έξοδο της.



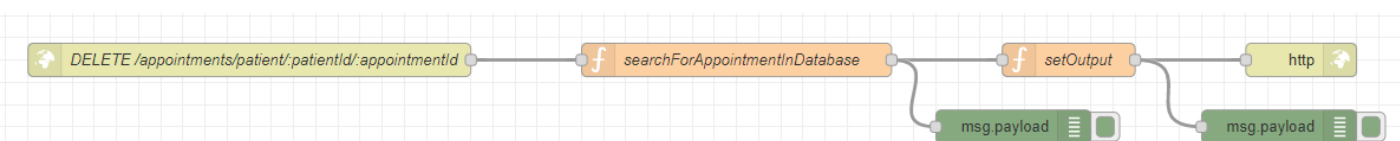
#### Ροή endpoint POST /appointments/patient/:patientId

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την καταχώρηση ενός νέου πόρου τύπου Appointment για συγκεκριμένο ασθενή με id που έχει τιμή patientId. Η συνάρτηση setAppointment διαβάζει την είσοδο και την ελέγχει. Καλεί τη βάση δεδομένων ώστε να την αποθηκεύσει και το αποτέλεσμα της πράξης έρχεται ως είσοδος στην συνάρτηση setOutput, η οποία ελέγχει το status code και επιστρέφει το αντίστοιχο μήνυμα.



#### Ροή endpoint DELETE /appointments/patient/:patientId/:appointmentId

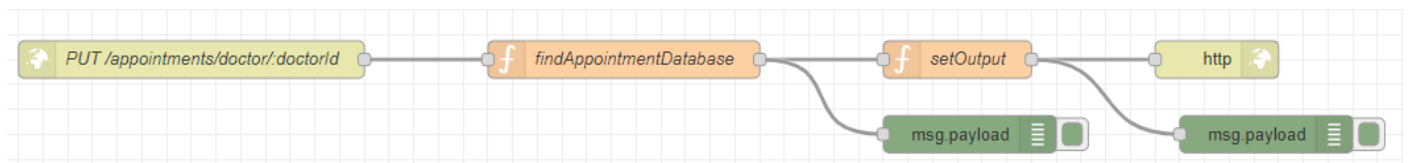
Ροή που υλοποιεί την υπηρεσία υπεύθυνη για τη διαγραφή ενός πόρου τύπου Appointment για συγκεκριμένο ασθενή με id που έχει τιμή patientId. Η συνάρτηση searchForAppointmentInDatabase ελέγχει αν υπάρχει το ραντεβού καταχωρημένο και αν ναι τότε το διαγράφει. Έπειτα η συνάρτηση setOutput βλέπει το αποτέλεσμα της προηγούμενης πράξης και αναλόγως επιστρέφει το κατάλληλο μήνυμα.



#### Ροή endpoint PUT /appointments/doctor/:doctorId

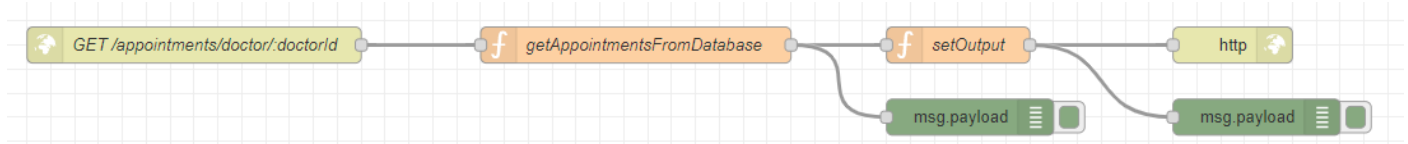


Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την επεξεργασία πόρων τύπου Appointment για συγκεκριμένο γιατρό με id που έχει την τιμή patientId. Η συνάρτηση findAppointmentDatabase ελέγχει τη βάση δεδομένων και προσπαθεί να αντικαταστήσει τα υπάρχοντα ραντεβού με τα νέα που δόθηκαν ως είσοδος. Η παραπάνω ενέργεια και ο κωδικός της που υποδεικνύει αν πραγματοποιήθηκε κανονικά, έρχονται σαν είσοδος στη συνάρτηση setOutput η οποία καθορίζει την έξοδο ανάλογα με αυτό το statusCode.



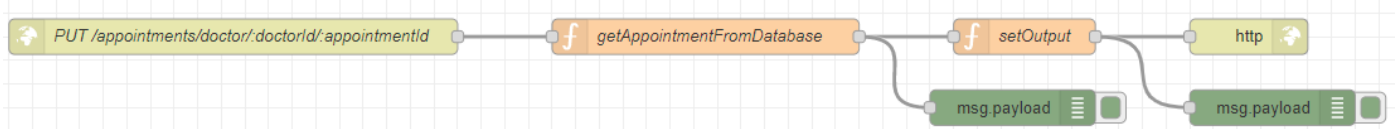
#### Ροή endpoint GET /appointments/doctor/:doctorId

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την επιστροφή των ραντεβού ενός χρήστη τύπου γιατρού με doctorId, που είναι το μοναδικό id αυτού του γιατρού. Η συνάρτηση getAppointmentsFromDatabase αναζητεί στη βάση δεδομένων και επιστρέφει μία λίστα με τα ραντεβού του ασθενή, αν η κλήση ήταν σωστή. Η συνάρτηση setOutput βλέπει το status code που προέκυψε και ανάλογα θέτει την έξοδο της.



#### Ροή endpoint PUT /appointments/doctor/:doctorId/:appointmentId

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την επεξεργασία της τιμής ενός πόρου Appointment που έχει το μοναδικό id appointmentId για τον συγκεκριμένο γιατρό, ο οποίος προσδιορίζεται από την τιμή doctorId. Η συνάρτηση getAppointmentFromDatabase λαμβάνει την είσοδο και προσπαθεί να αντικαταστήσει το ραντεβού στη βάση δεδομένων. Ο status code της παραπάνω ενέργειας δίνεται σαν είσοδος στη συνάρτηση setOutput και με βάση αυτόν προσδιορίζεται η έξοδος της.



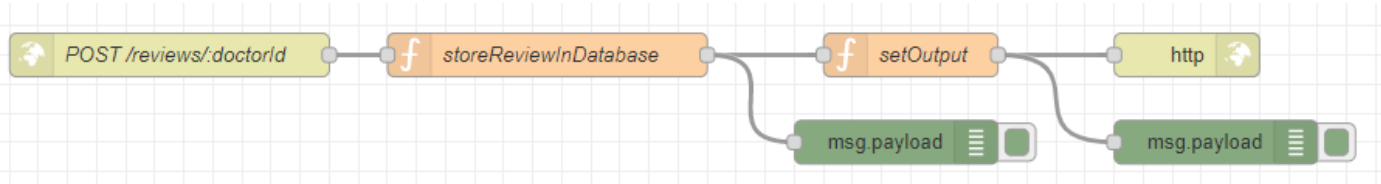
### 3.1.4 Ροές πόρου Reviews

#### Ροή endpoint POST /reviews/:doctorId

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την δημιουργία νέου πόρου Review, δίνοντας το doctorId του γιατρού στον οποίο απευθύνεται η κριτική. Η συνάρτηση storeReviewInDatabase παίρνει ως είσοδο την κριτική που έχει πραγματοποιήσει κάποιος ασθενής και προσπαθεί να την αποθηκεύσει στη βάση δεδομένων. Το status code της ενέργειας αυτής δίνεται ως είσοδος στη συνάρτηση setOutput η οποία και

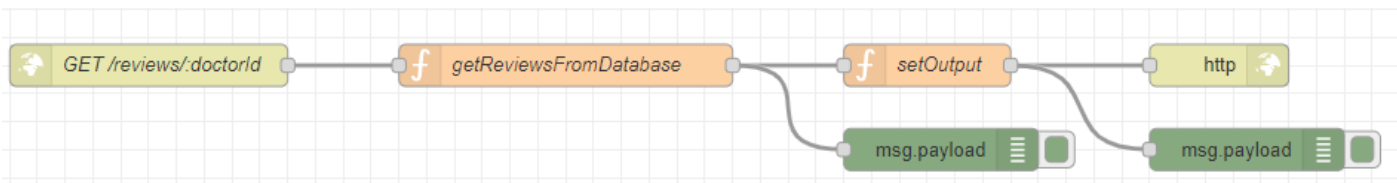


μεταβάλλει ανάλογα την έξοδο της.



#### Ροή endpoint GET /reviews/:doctorId

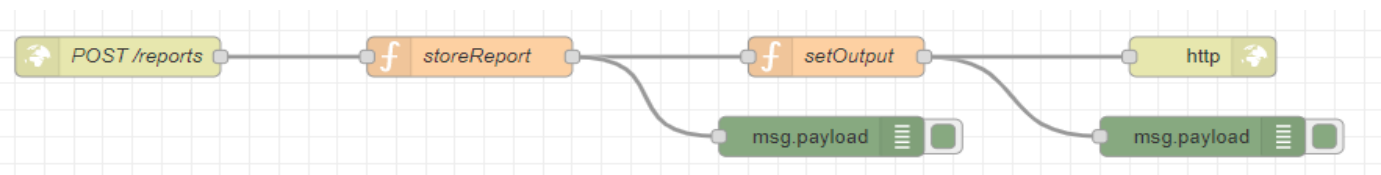
Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την επιστροφή των πόρων τύπου Review ενός συγκεκριμένου γιατρού που έχει id ίσο με την τιμή doctorId. Η συνάρτηση getReviewsFromDatabase αναζητεί τα ραντεβού στη βάση δεδομένων και τα επιστρέφει. Το status code δίνεται σαν είσοδο στη δεύτερη συνάρτηση setOutput η οποία και αλλάζει την έξοδο ανάλογα με την τιμή του κωδικού αυτού.



### 3.1.5 Ροές πόρου Reports

#### Ροή endpoint POST /reports

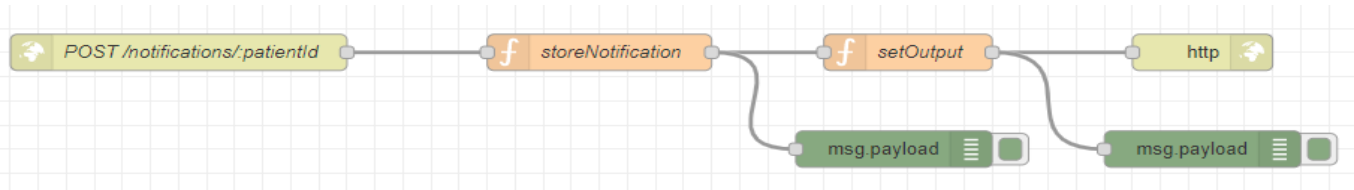
Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την καταχώρηση πόρου Report στη βάση δεδομένων. Η συνάρτηση storeReport λαμβάνει ως είσοδο το Report και κάνει κλήση στη βάση δεδομένων για να το αποθηκεύσει. Το status code της παραπάνω ενέργειας δίνεται σαν είσοδος στη συνάρτηση setOutput η οποία και αποφασίζει την έξοδο ανάλογα με τον κωδικό αυτό.



### 3.1.6 Ροές πόρου Notifications

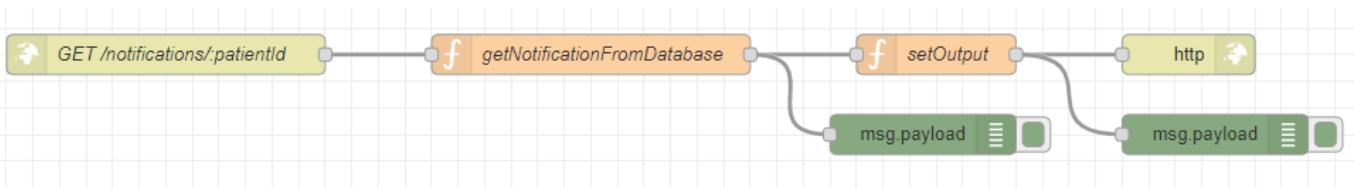
#### Ροή endpoint POST /notifications/:patientId

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την αποθήκευση πόρου Notification για συγκεκριμένο ασθενή ο οποίος έχει id ίσο με την τιμή patientId. Η συνάρτηση storeNotification παίρνει την είσοδο και την αποθηκεύει στη κατάλληλη θέση στη βάση δεδομένων και επιστρέφει τον κωδικό της ενέργειας, τον οποίο δίνουμε σαν είσοδο στη συνάρτηση setOutput ώστε να επεξεργαστεί την έξοδο της ανάλογα με αυτόν.



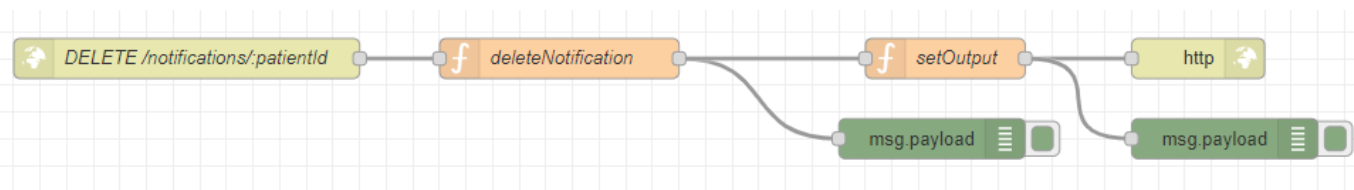
#### Ροή endpoint GET /notifications/:patientId

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την επιστροφή πόρου Notification για έναν συγκεκριμένο ασθενή με id που έχει την τιμή patientId. Η συνάρτηση getNotificationFromDatabase ζητάει από τη βάση δεδομένων τις ειδοποιήσεις που ανήκουν στον συγκεκριμένο ασθενή και η έξοδος της λαμβάνεται από την setOutput η οποία καθορίζει την τελική έξοδο.



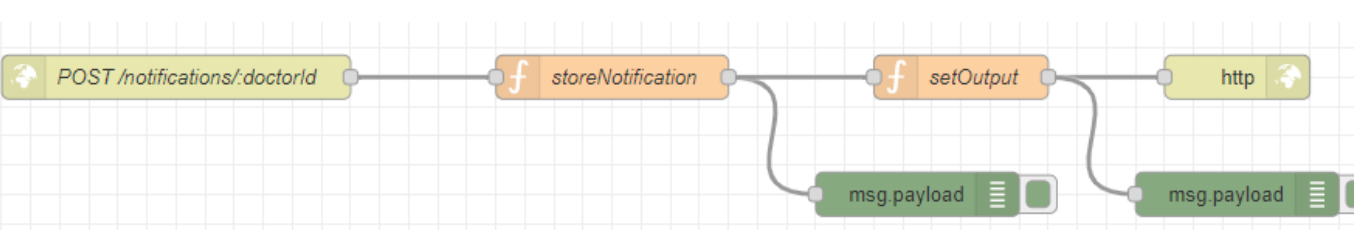
#### Ροή endpoint DELETE /notifications/:patientId

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για τη διαγραφή πόρου Notification για έναν συγκεκριμένο ασθενή με id που έχει την τιμή patientId. Πιο συγκεκριμένα, διαγράφει όλα τα notifications του ασθενή αυτού, εφόσον θεωρούμε πως οι 3 παραπάνω ενέργειες (αποθήκευση της ειδοποίησης, λήψη της ειδοποίησης και μετά διαγραφή της) γίνονται άμεσα, προτού προλάβει να έρθει δεύτερη ειδοποίηση για τον ίδιο ασθενή. Η συνάρτηση deleteNotification ζητάει την διαγραφή των ειδοποιήσεων από τη βάση δεδομένων και το αποτέλεσμα της ενέργειας δίνεται ως είσοδος στη συνάρτηση setOutput η οποία μεταβάλλει ανάλογα την έξοδο της.



#### Ροή endpoint POST /notifications/:doctorId

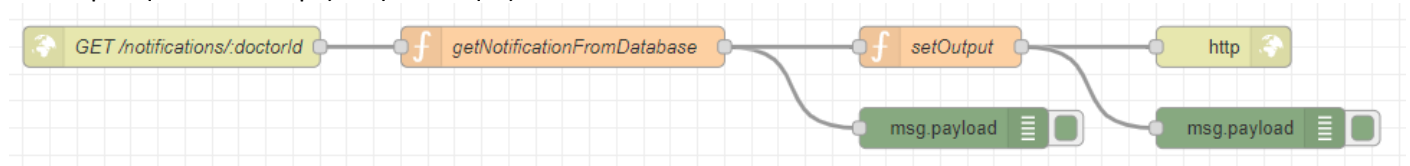
Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την αποθήκευση πόρου Notification για συγκεκριμένο γιατρό ο οποίος έχει id ίσο με την τιμή doctorId. Η συνάρτηση storeNotification παίρνει την είσοδο και την αποθηκεύει στη κατάλληλη θέση στη βάση δεδομένων και επιστρέφει τον κωδικό της ενέργειας, τον οποίο δίνουμε σαν είσοδο στη συνάρτηση setOutput ώστε να επεξεργαστεί την έξοδο της ανάλογα με αυτόν.





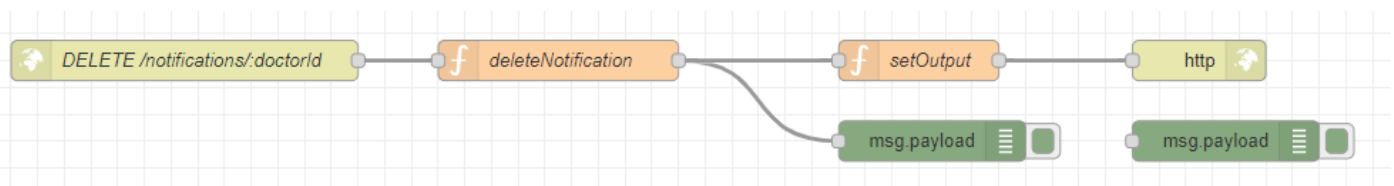
### Ροή endpoint GET /notifications/:doctorId

Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την επιστροφή πόρου Notification για έναν συγκεκριμένο γιατρό με id που έχει την τιμή doctorId. Η συνάρτηση getNotificationFromDatabase ζητάει από τη βάση δεδομένων τις ειδοποιήσεις που ανήκουν στον συγκεκριμένο γιατρό και η έξοδος της λαμβάνεται από την setOutput η οποία καθορίζει την τελική έξοδο.



### Ροή endpoint DELETE /notifications/:doctorId

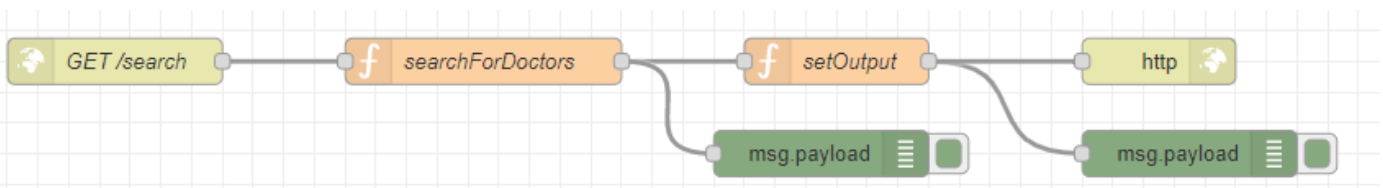
Ροή που υλοποιεί την υπηρεσία υπεύθυνη για τη διαγραφή πόρου Notification για έναν συγκεκριμένο γιατρό με id που έχει την τιμή doctorId. Πιο συγκεκριμένα, διαγράφει όλα τα notifications του γιατρού αυτού, εφόσον θεωρούμε πως οι 3 παραπάνω ενέργειες (αποθήκευση της ειδοποίησης, λήψη της ειδοποίησης και μετά διαγραφή της) γίνονται άμεσα, προτού προλάβει να έρθει δεύτερη ειδοποίηση για τον ίδιο γιατρό. Η συνάρτηση deleteNotification ζητάει την διαγραφή των ειδοποιήσεων από τη βάση δεδομένων και το αποτέλεσμα της ενέργειας δίνεται ως είσοδος στη συνάρτηση setOutput η οποία μεταβάλλει ανάλογα την έξοδο της.



## 3.1.7 Ροές πόρου Search

### Ροή endpoint GET /search

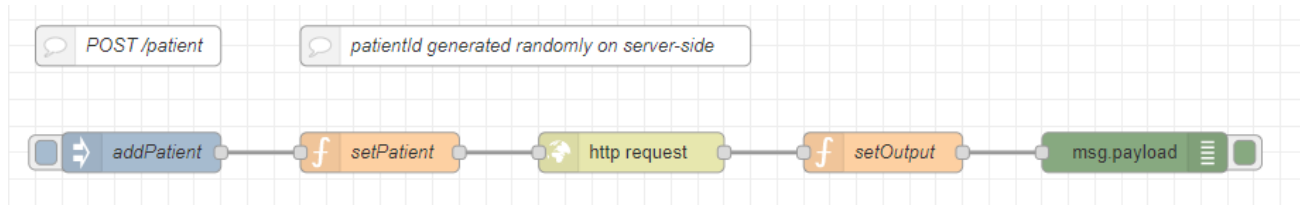
Ροή που υλοποιεί την υπηρεσία υπεύθυνη για την αναζήτηση γιατρών στη βάση δεδομένων. Η συνάρτηση searchForDoctors δέχεται ως είσοδο τα διάφορα φίλτρα αναζήτησης και τα μεταβιβάζει στη βάση δεδομένων και εν τέλει επιστρέφει τη λίστα γιατρών που πληρούν τα ζητούμενα κριτήρια. Τέλος, το αποτέλεσμα της αναζήτησης στη βάση δεδομένων περνάει από τη συνάρτηση setOutput η οποία και επιστρέφει το τελικό μήνυμα.





## 3.2 Υλοποίηση ιστοριών χρήστη

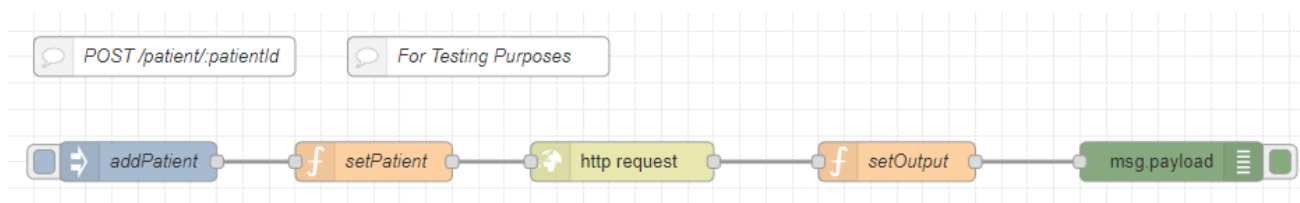
### 3.2.1 Ιστορία Χρήστη <Create Patient User>



Ροή μέσω της οποίας προστίθεται ένας νέος ασθενής στο σύστημα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
addPatient	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setPatient	function	-Επιστρέφει το νέο αντικείμενο του ασθενή σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας POST /patient η οποία δημιουργεί έναν καινούργιο χρήστη ασθενή με ID που καθορίζεται στον server.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.

### 3.2.2 Ιστορία Χρήστη <Create Patient User by ID>



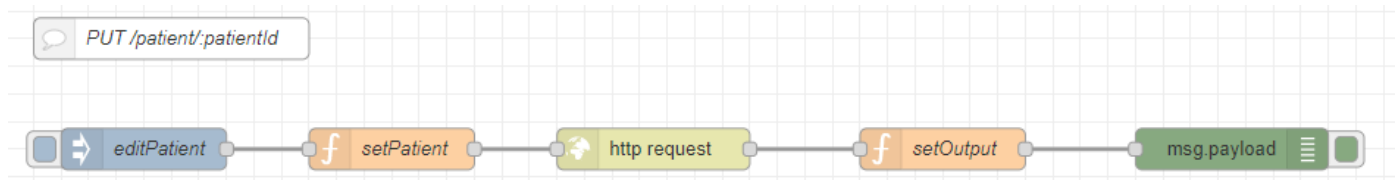
Ροή μέσω της οποίας προστίθεται ένας νέος ασθενής στο σύστημα, με προσδιορισμένο patientId.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
addPatient	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setPatient	function	-Επιστρέφει το νέο αντικείμενο του ασθενή σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας POST /patient/:patientId η οποία δημιουργεί έναν καινούργιο χρήστη ασθενή με ID το :patientId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.





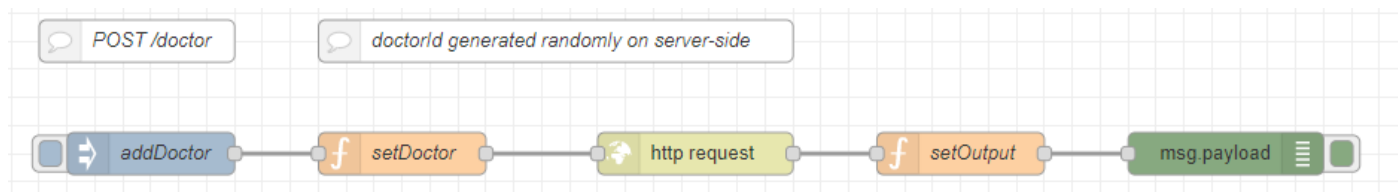
### 3.2.3 Ιστορία Χρήστη <Edit Patient Details>



Ροή μέσω της οποίας τροποποιείται ένας υπάρχων ασθενής στο σύστημα, μέσω του patientId του.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
editPatient	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setPatient	function	-Επιστρέφει το τροποποιημένο αντικείμενο του ασθενή σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας PUT /patient/:patientId η οποία τροποποιεί τα στοιχεία του χρήστη ασθενή με ID το :patientId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.

### 3.2.4 Ιστορία Χρήστη <Create Doctor User>



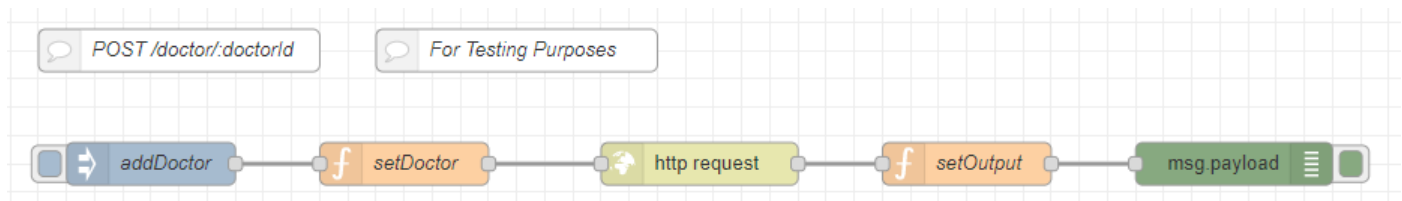
Ροή μέσω της οποίας προστίθεται ένας νέος γιατρός στο σύστημα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
addDoctor	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setDoctor	function	-Επιστρέφει το νέο αντικείμενο του γιατρού σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας POST /doctor η οποία δημιουργεί έναν καινούργιο γιατρό με ID που καθορίζεται στον server.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.





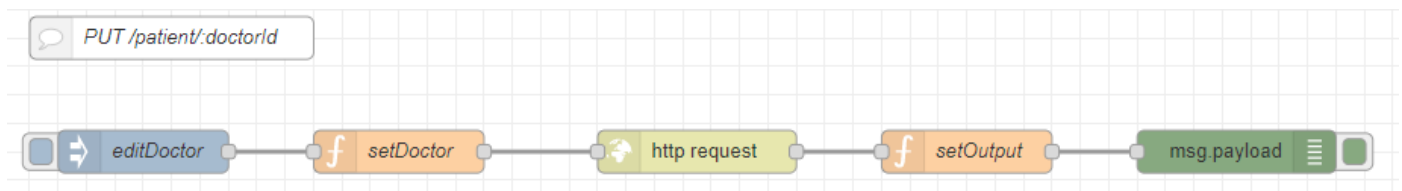
### 3.2.5 Ιστορία Χρήστη <Create Doctor User by ID>



Ροή μέσω της οποίας προστίθεται ένας νέος γιατρός στο σύστημα, με προσδιορισμένο doctorId.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
addDoctor	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setDoctor	function	-Επιστρέφει το νέο αντικείμενο του γιατρού σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας POST /doctor/:doctorId η οποία δημιουργεί έναν καινούργιο γιατρό με ID το :doctorId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.

### 3.2.6 Ιστορία Χρήστη <Edit Doctor Details>

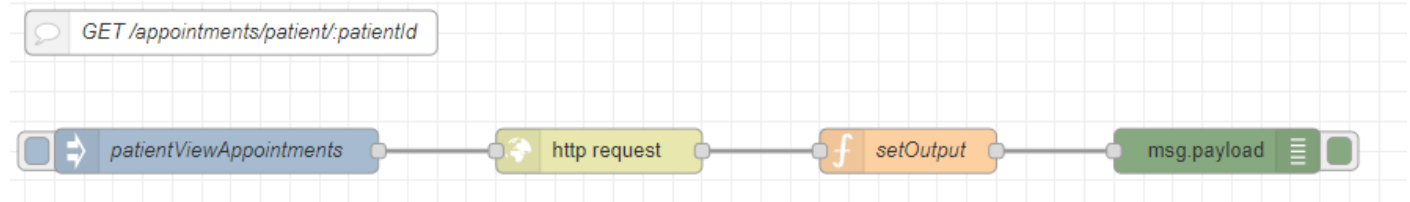


Ροή μέσω της οποίας ο τροποποιείται ένας υπάρχον γιατρός στο σύστημα, μέσω του doctorId του.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
editDoctor	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setDoctor	function	-Επιστρέφει το τροποποιημένο αντικείμενο του ασθενή σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας PUT /doctor/:doctorId η οποία τροποποιεί τα στοιχεία του γιατρού με ID το :doctorId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.



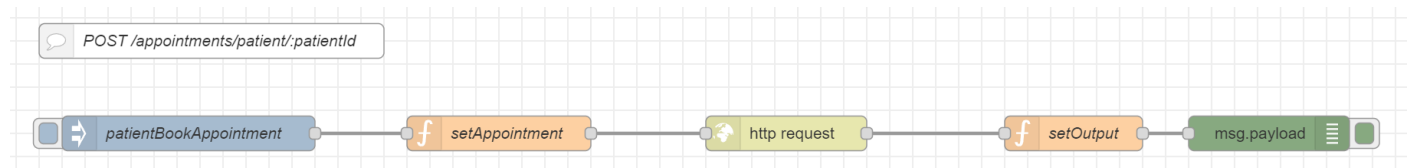
### 3.2.7 Ιστορία Χρήστη <Get Patient Appointments List>



Ροή η οποία επιστρέφει τα ραντεβού ενός ασθενή.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
patientViewAppointments	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας GET /appointments/patient/:patientId η οποία επιστρέφει τα ραντεβού του χρήστη ασθενή με ID το :patientId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.

### 3.2.8 Ιστορία Χρήστη <Book Appointment>

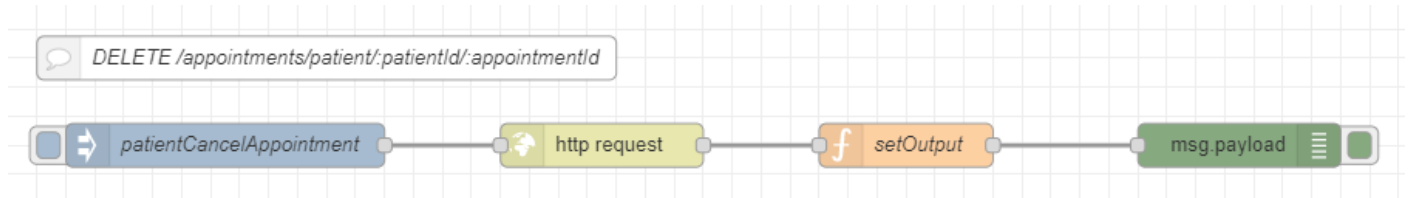


Ροή η οποία μέσω της οποίας ο ασθενής κλείνει ένα ραντεβού με κάποιον γιατρό.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
patientBookAppointment	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setAppointment	function	-Επιστρέφει το νέο αντικείμενο ραντεβού σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας POST /appointments/patient/:patientId η οποία δημιουργεί το ραντεβού του χρήστη ασθενή με ID το :patientId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.



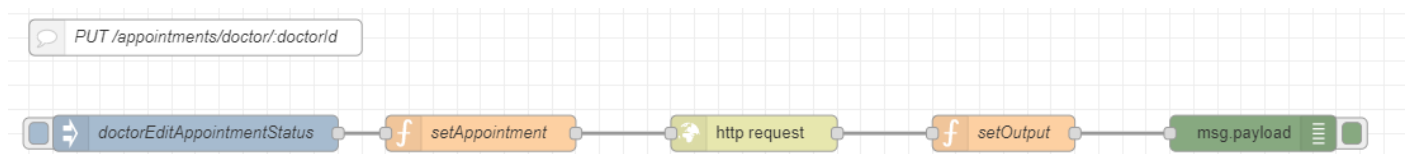
### 3.2.9 Ιστορία Χρήστη <Patient Cancel Booked Appointment>



Ροή μέσω της οποίας ο ασθενής ακυρώνει ένα κλεισμένο ραντεβού.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
patientCancelAppointment	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας DELETE /appointments/patient/:patientId/:appointmentId η οποία διαγράφει το ραντεβού με ID :appointmentId του ασθενή με ID το :patientId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.

### 3.2.10 Ιστορία Χρήστη <Doctor Modify Available Appointment List>

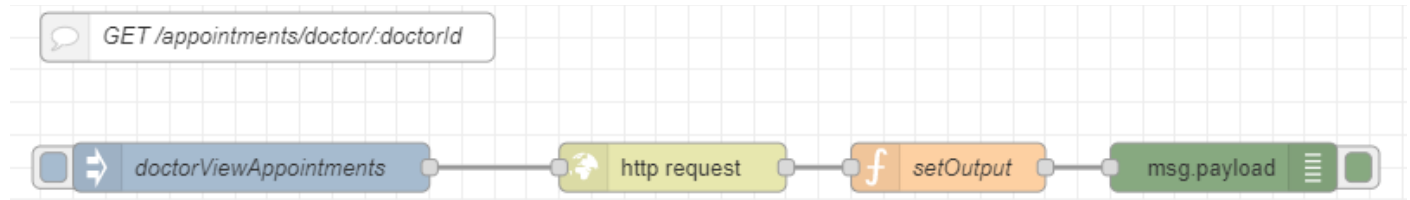


Ροή μέσω της οποίας ο γιατρός τροποποιεί την λίστα με τα διαθέσιμα ραντεβού του.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
doctorEditAppointmentStatus	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setAppointment	function	-Επιστρέφει τη τροποποιημένη λίστα ραντεβού σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας PUT /appointments/doctor/:doctorId η οποία τροποποιεί τη λίστα με τα διαθέσιμα ραντεβού του γιατρού με ID το :doctorId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.



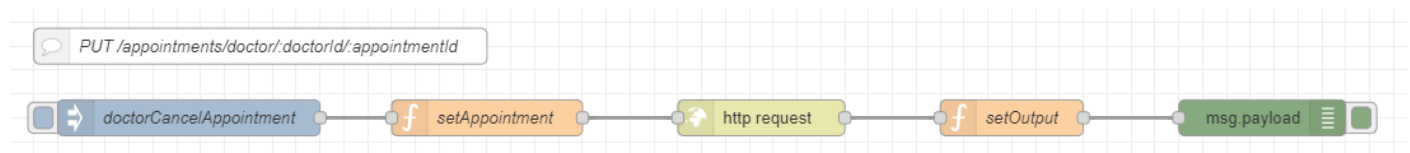
### 3.2.11 Ιστορία Χρήστη <Get Doctor Appointments List>



Ροή η οποία επιστρέφει το σύνολο των ραντεβού του γιατρού.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
doctorViewAppointments	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας GET /appointments/doctor/:doctorId η οποία επιστρέφει τα ραντεβού του γιατρού με ID το :doctorId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.

### 3.2.12 Ιστορία Χρήστη <Doctor Cancel Booked Appointment>



Ροή μέσω της οποίας ο γιατρός τροποποιεί την κατάσταση ενός συγκεκριμένου ραντεβού(για ακύρωση).

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
doctorCancelAppointment	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setAppointment	function	-Επιστρέφει το τροποποιημένο ραντεβού του γιατρού σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας PUT /appointments/doctor/:doctorId/:appointmentId η οποία τροποποιεί ραντεβού με ID :appointmentId του γιατρού με ID το :doctorId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.



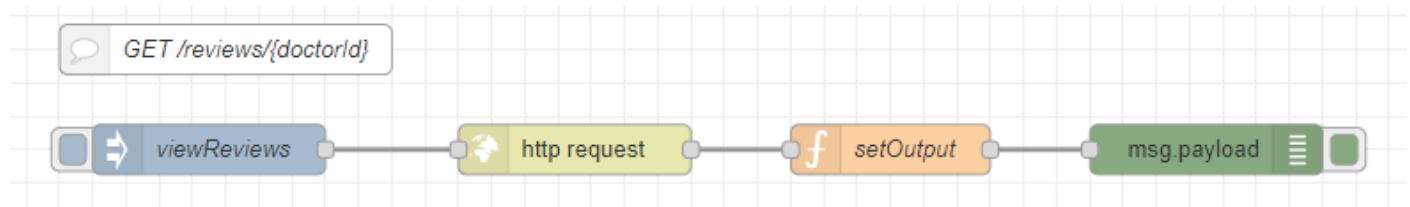
### 3.2.13 Ιστορία Χρήστη <Patient Submit Review>



Ροή μέσω της οποίας ο ασθενής υποβάλει κριτική για κάποιον γιατρό.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
<b>addReview</b>	<b>Inject</b>	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
<b>setReview</b>	<b>function</b>	-Επιστρέφει τη νέα κριτική σε μορφή json.
<b>http request</b>	<b>http-request</b>	-Εκτελεί κλήση της υπηρεσίας POST /reviews/:doctorId η οποία δημιουργεί μία νέα κριτική για τον γιατρό με ID :doctorId.
<b>setOutput</b>	<b>function</b>	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
<b>msg.payload</b>	<b>debug</b>	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.

### 3.2.14 Ιστορία Χρήστη <Get Doctor's Reviews>

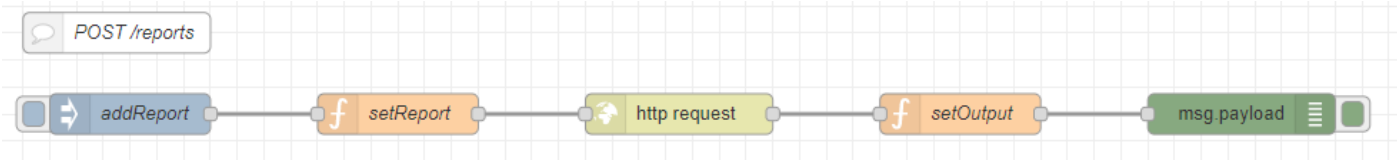


Ροή η οποία επιστρέφει το σύνολο των κριτικών που έχουν υποβληθεί για έναν γιατρό.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
<b>viewReviews</b>	<b>Inject</b>	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
<b>http request</b>	<b>http-request</b>	-Εκτελεί κλήση της υπηρεσίας GET /reviews/:doctorId η οποία επιστρέφει τις κριτικές του γιατρού με ID :doctorId.
<b>setOutput</b>	<b>function</b>	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
<b>msg.payload</b>	<b>debug</b>	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.



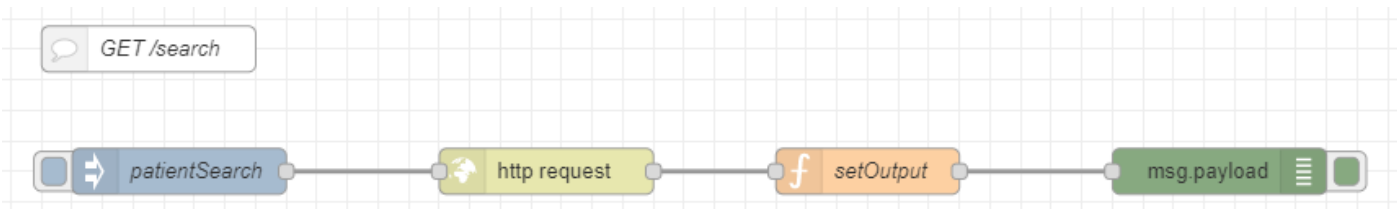
### 3.2.15 Ιστορία Χρήστη <User Submit Report>



Ροή μέσω της οποίας ένας χρήστης καταθέτει αναφορά στο σύστημα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
addReport	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setReport	function	-Επιστρέφει το νέο αντικείμενο αναφοράς του χρήστη σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας POST /reports η οποία δημιουργεί μία νέα αναφορά.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.

### 3.2.16 Ιστορία Χρήστη <Get Search Results Sorted>

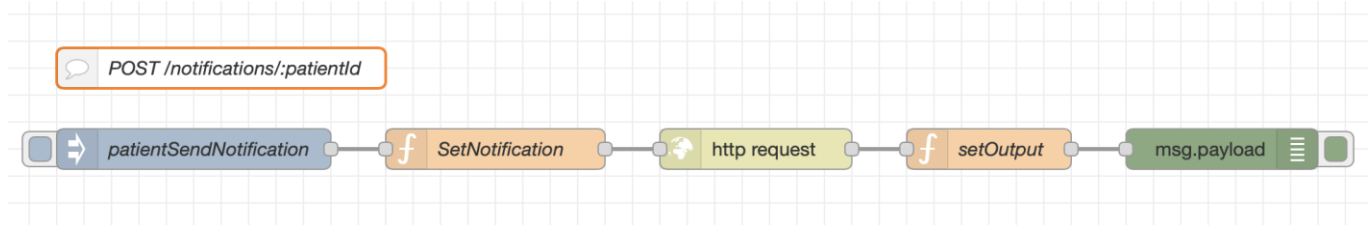


Ροή η οποία επιστρέφει το σύνολο των αποτελεσμάτων αναζήτησης με κάποιο κριτήριο ταξινόμησης.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
patientSearch	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας GET /search η οποία επιστρέφει τα αποτελέσματα αναζήτησης γιατρού από ασθενή με κατάλληλα queries.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.



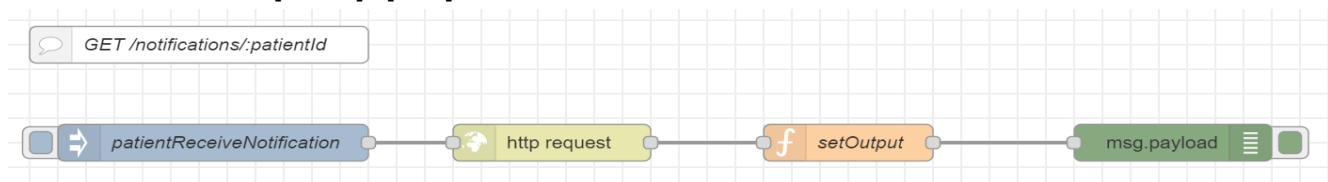
### 3.2.17 Ιστορία Χρήστη <Create Patient's Notification by ID>



Ροή η οποία δημιουργεί ένα notification message που αφορά τον ασθενή με το συγκεκριμένο patientId.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
patientSendNotification	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setNotification	function	-Επιστρέφει το νέο notification message(συμβολοσειρά) που προορίζεται για τον ασθενή σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας POST /notifications/:patientId η οποία δημιουργεί ένα νέο notification message για τον ασθενή με ID:patientId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.

### 3.2.18 Ιστορία Χρήστη <Patient Get Notification>



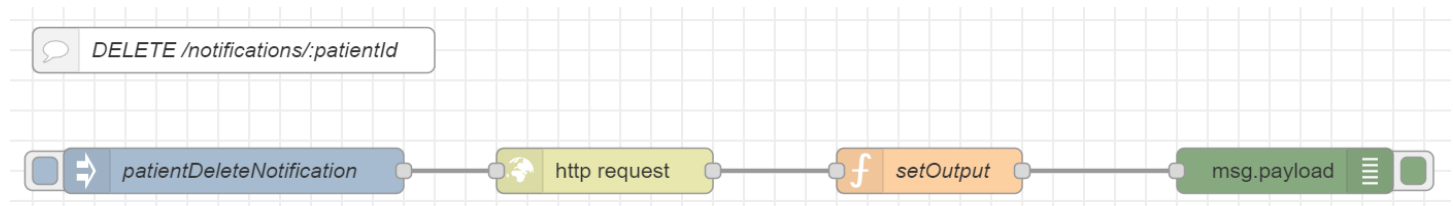
Ροή η οποία επιστρέφει ένα notification message στον ασθενή με το συγκεκριμένο patientId.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
patientReceiveNotification	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας GET /notifications/:patientId η οποία επιστρέφει το notification message που αφορά τον ασθενή με ID:patientId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.





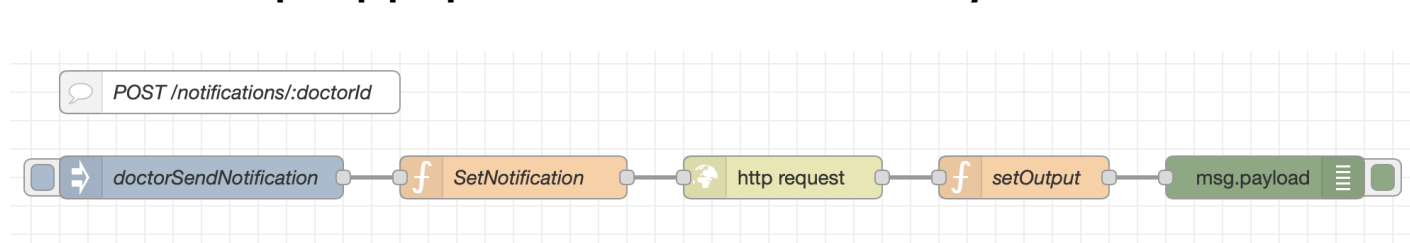
### 3.2.19 Ιστορία Χρήστη <Delete Patient's Notification>



Ροή η οποία διαγράφει ένα notification message του ασθενή με το συγκεκριμένο patientId.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
patientDeleteNotification	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας DELETE /notifications/:patientId η οποία διαγράφει το notification message του ασθενή με ID:patientId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.

### 3.2.20 Ιστορία Χρήστη <Create Doctor's Notification by ID>



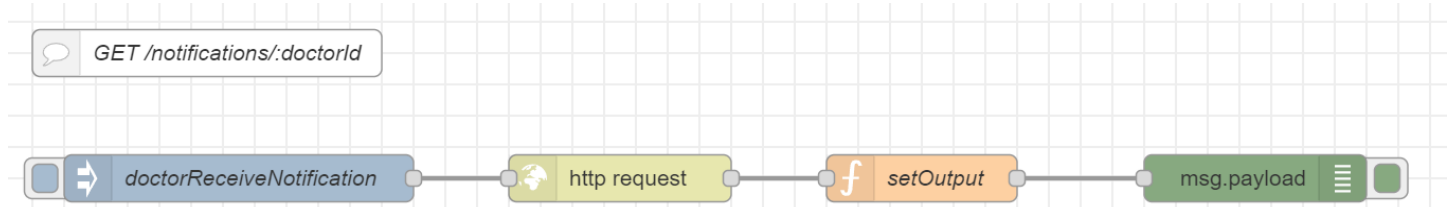
Ροή η οποία δημιουργεί ένα notification message που αφορά τον γιατρό με το συγκεκριμένο doctorId.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
doctorSendNotification	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
setNotification	function	-Επιστρέφει το νέο notification message(συμβολοσειρά) που προορίζεται για τον γιατρό σε μορφή json.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας POST /notifications/:doctorId η οποία δημιουργεί ένα νέο notification message για τον γιατρό με ID:docotrId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.





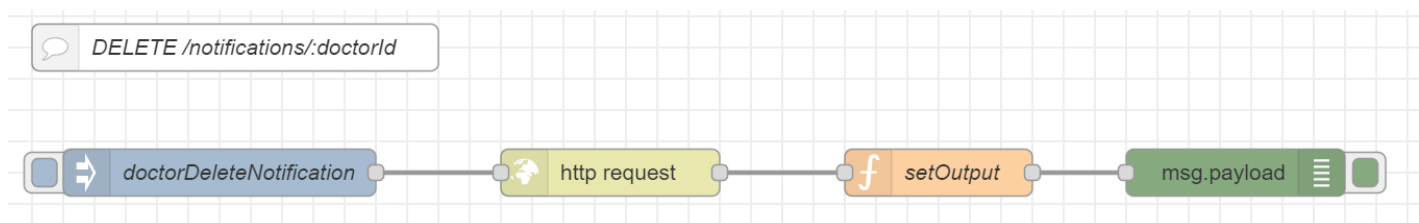
### 3.2.21 Ιστορία Χρήστη <Doctor Get Notification>



Ροή η οποία επιστρέφει ένα notification message στον γιατρό με το συγκεκριμένο doctorId.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
doctorReceiveNotification	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας GET /notifications/:doctorId η οποία επιστρέφει το notification message που αφορά τον γιατρό με ID:doctorId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.

### 3.2.22 Ιστορία Χρήστη <Delete Doctor's Notification>



Ροή η οποία διαγράφει ένα notification message του γιατρού με το συγκεκριμένο doctorId.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
doctorDeleteNotification	Inject	-Κόμβος υπεύθυνος για την ενεργοποίηση της ροής.
http request	http-request	-Εκτελεί κλήση της υπηρεσίας DELETE /notifications/: doctorId η οποία διαγράφει το notification message του γιατρού με ID:doctorId.
setOutput	function	-Αποθηκεύει στη μεταβλητή msg.payload κατάλληλη κατατοπιστική συμβολοσειρά η οποία σε συνδυασμό με το http status code θα τυπωθεί στην κονσόλα με σκοπό τον έλεγχο της ορθής λειτουργίας της ροής.
msg.payload	debug	-Τυπώνει στην κονσόλα το περιεχόμενο της μεταβλητής msg.payload.



---

## Παράρτημα Ι – Ανοιχτά Θέματα

- Χρειάζεται η δημιουργία βάσης δεδομένων για την σωστή λειτουργία των ροών που δημιουργήθηκαν.
- Υπάρχουν περιθώρια υλοποίησης επιπλέον προτύπων σχεδίασης και για άλλες μη λειτουργικές ή/και λειτουργικές απαιτήσεις. Στο συγκεκριμένο παραδοτέο ζητήθηκε να υλοποιηθούν ενδεικτικά 3 πρότυπα σχεδίασης.