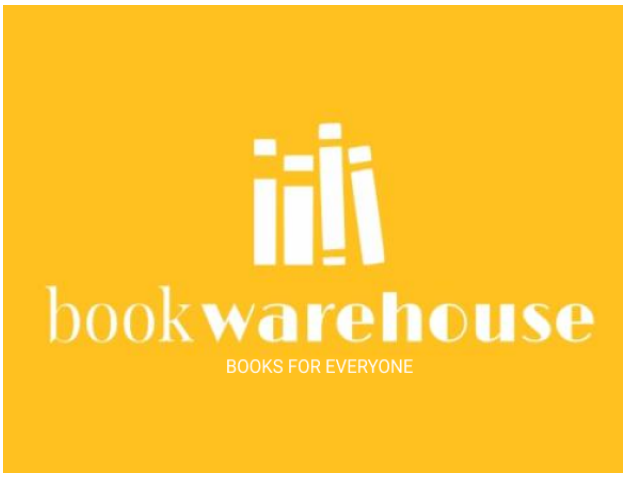
Book Warehouse

Books for everyone



Προδιαγραφές Λογισμικού και Υλοποίηση συστήματος

Del.2.1

Version 0.6 (draft)

Μωράτης Κωνσταντίνος, kmoratis@ece.auth.gr
Διαμάντης Γεώργιος-Μιχαήλ, gdiamanti@ece.auth.gr
Παρτσάνης Παναγιώτης, partsanis@ece.auth.gr
Αβραμίδης Παναγιώτης, pavramid@ece.auth.gr

31/5/2021

Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
Α. Συμεωνίδης	17/05/2007	Δημιουργία εγγράφου. Προσαρμογή των προτύπων του K. E. Wiegers[1] 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
Μωράτης Κωνσταντίνος	26	kmoratis@ece.auth.gr
Διαμάντης Γεώργιος-Μιχαήλ	26	gdiamanti@ece.auth.gr
Παρτσάνης Παναγιώτης	26	partsanis@ece.auth.gr
Αβραμίδης Παναγιώτης	26	pavramid@ece.auth.gr

[1] 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

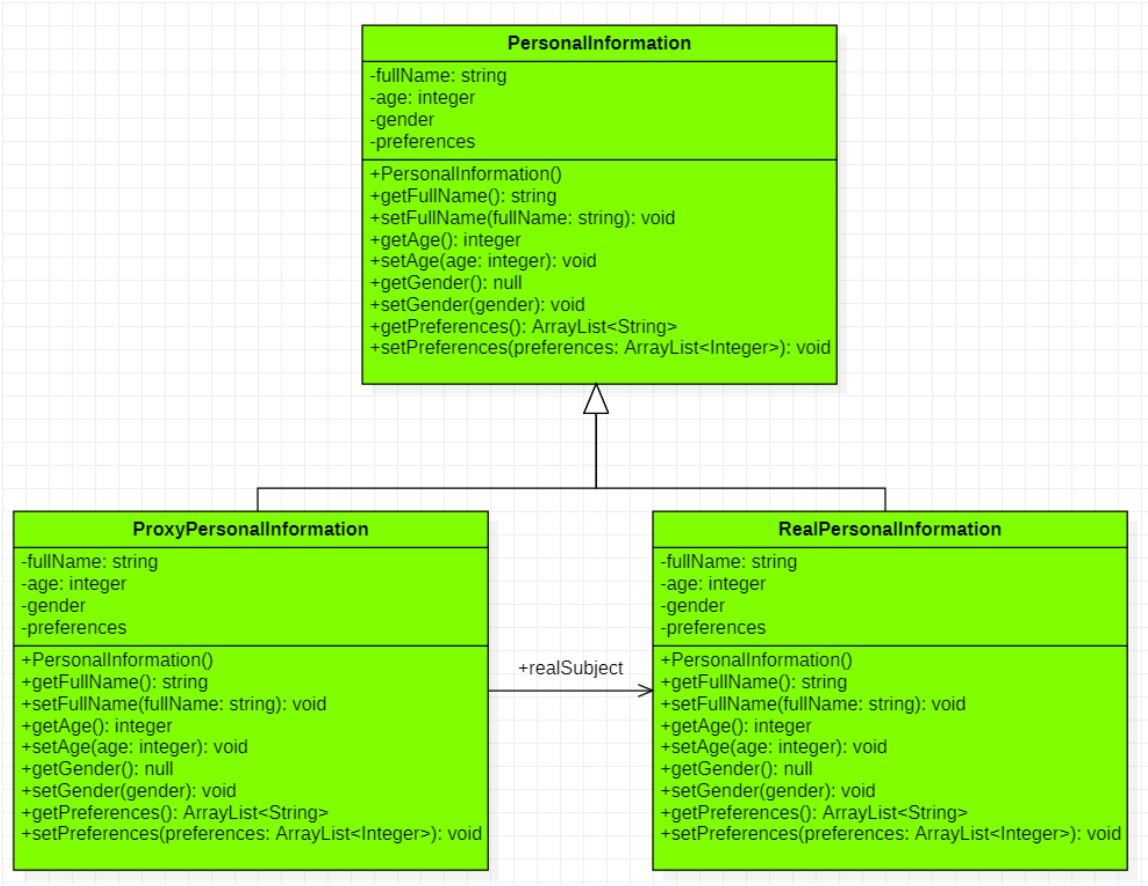
1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν

Στο σημείο αυτό ορίζουμε μία επιπλέον Μη Λειτουργική Απαίτηση την οποία πρέπει να ικανοποιεί η εφαρμογή μας.

<ΜΛΑ- 6> (Απαιτήσεις διασύνδεσης)
Το σύστημα πρέπει να μπορεί να συνδέεται σε βάσεις δεδομένων MySQL και MsSQL για να εγγράφει, να αντλεί και να αλλάζει δεδομένα για τα βιβλία και τους χρήστες του.

1.1 Πρότυπο Proxy

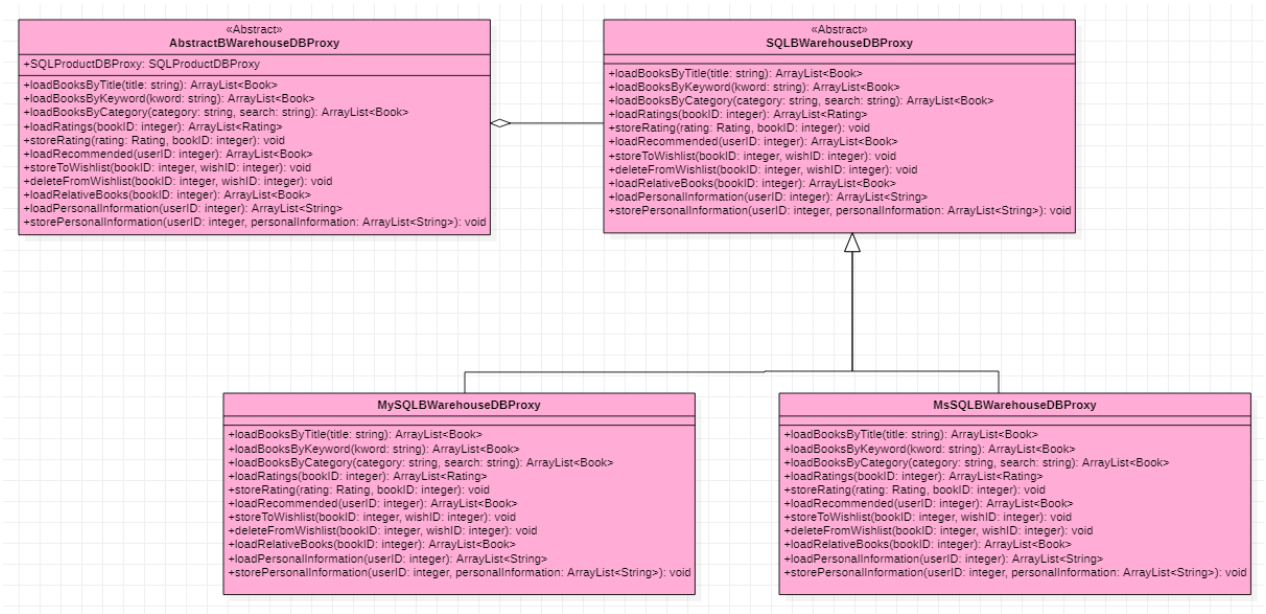
Το πρότυπο Proxy αποτελεί ένα δομικό πρότυπο. Στα πλαίσια της εφαρμογής μας το χρησιμοποιούμε ως Protection Proxy για να επιτύχουμε έλεγχο πρόσβασης στα αντικείμενα της κλάσης Personal Information. Με τον τρόπο αυτόν μπορούμε να προστατεύσουμε τα προσωπικά δεδομένα των χρηστών του συστήματος μας και επομένως ικανοποιείται η ΜΛΑ-2 που έχει οριστεί στο έγγραφο απαιτήσεων χρηστών. Ειδικότερα, θέλουμε τα προσωπικά δεδομένα των χρηστών να μπορούν να προβληθούν μόνο από τους διαχειριστές του συστήματος καθώς και από τους ίδιους τους χρήστες.



Σχήμα 1: Εφαρμογή προτύπου Proxy.

1.2 Πρότυπο Bridge

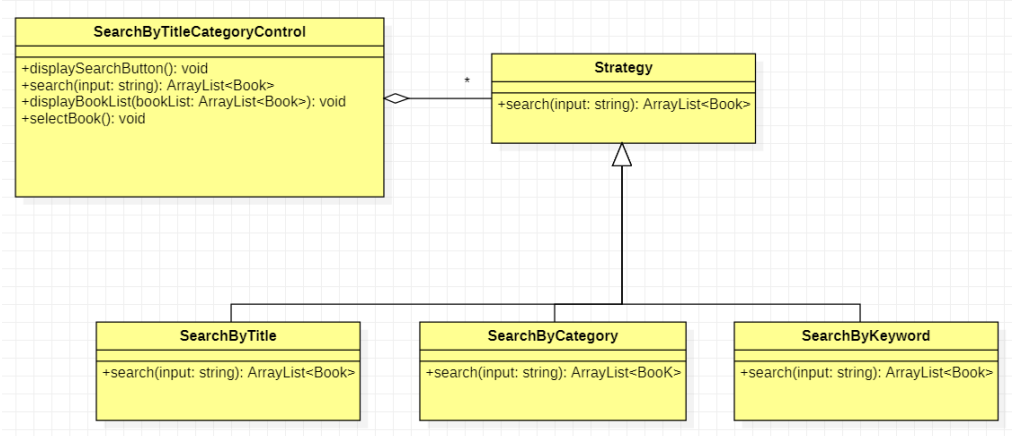
Το πρότυπο Bridge είναι ένα δομικό πρότυπο. Στο πλαίσιο της εφαρμογής μας το χρησιμοποιούμε για να επιτύχουμε ανεξαρτησία μιας συγκεκριμένης αφαίρεσης από την υλοποίησή της. Πιο συγκεκριμένα, χρησιμοποιώντας το συγκεκριμένο πρότυπο καταφέραμε να διαχωρίσουμε την αρχική διεπαφή BWarehouseDBProxy από την υλοποίησή της ώστε να μπορούν να υπάρχουν και οι δύο ως ανεξάρτητες κλάσεις. Με τον τρόπο αυτόν και χρησιμοποιώντας κληρονομικότητα μπορούμε να έχουμε δύο διαφορετικές υλοποιήσεις της διεπαφής, μία για MySQL και μία για MsSQL βάσεις. Επομένως, ικανοποιείται η ΜΛΑ-6 του συστήματος μας. Ένα ακόμη πλεονέκτημα που προκύπτει από τη χρήση του συγκεκριμένου προτύπου, είναι ότι η τροποποίηση του συστήματος ώστε να μπορεί να συνδέεται και σε άλλου τύπου βάσεις δεδομένων είναι εύκολη και δεν απαιτεί επανασχεδιασμό του υπάρχοντος συστήματος.



Σχήμα 2: Εφαρμογή προτύπου Bridge.

1.3 Πρότυπο Strategy

Το πρότυπο Strategy είναι ένα πρότυπο Συμπεριφοράς. Με την χρήση του συγκεκριμένου προτύπου, καλύπτεται η σχεδιαστική ανάγκη της διαχείρισης των τριών διαφορετικών αλγορίθμων που υπάρχουν στην εφαρμογή για αναζήτηση βιβλίων. Έτσι έχουμε υλοποιήσει τρεις κλάσεις που κληρονομούν την κλάση Strategy, μία για αναζήτηση βιβλίων με τον τίτλο, μία για αναζήτηση ανά κατηγορία και μία για αναζήτηση με μία λέξη- κλειδί. Επιπλέον, με τον τρόπο αυτόν η σχεδίαση του συστήματός μας γίνεται τροποποιήσιμη, καθώς σε περίπτωση που χρειαστεί να προσθέσουμε έναν ακόμη αλγόριθμο για εξειδικευμένη αναζήτηση, αυτό μπορεί να γίνει εύκολα και χωρίς να επηρεάζει το ήδη υπάρχον σύστημα.



Σχήμα 3: Εφαρμογή προτύπου Strategy.

2. Αρχιτεκτονική Συστήματος

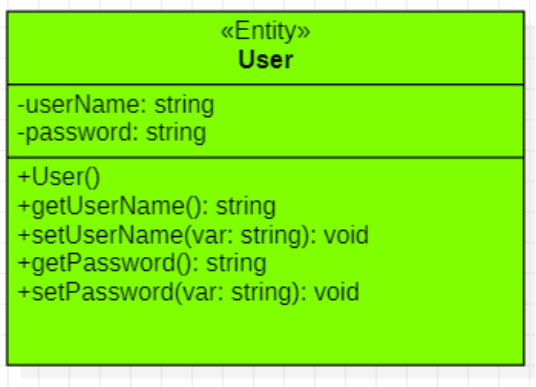
Για λόγους πληρότητας της εργασίας μας, ορίζουμε εδώ δύο ακόμη Λειτουργικές Απαιτήσεις που σχετίζονται με τη δημιουργία λογαριασμού χρήστη και τη σύνδεση σε αυτόν. Πιο συγκεκριμένα, ορίζουμε τις Λειτουργικές Απαιτήσεις:

<ΛΑ- 11>
Ο χρήστης πρέπει να μπορεί να δημιουργεί ένα λογαριασμό, εισάγοντας ένα όνομα χρήστη και έναν κωδικό πρόσβασης.

<ΛΑ- 12>
Ο χρήστης πρέπει να μπορεί να συνδέεται στο λογαριασμό του, χρησιμοποιώντας το όνομα χρήστη και τον κωδικό πρόσβασής του.

Επομένως, οι Λειτουργικές Απαιτήσεις 1-10 που έχουμε ορίσει στο Έγγραφο Απαιτήσεων Χρηστών, θα πρέπει να τροποποιηθούν και όπου αναφέρεται ο όρος <<χρήστης>>, να αντικατασταθεί από τον όρο <<συνδεδεμένος χρήστης>>.
Επιπλέον, για να προδιαγράψουμε το REST API της εφαρμογής μας, πρέπει να προσθέσουμε μία επιπλέον κλάση σε αυτές που έχουμε ορίσει στο πρώτο παραδοτέο, για την υλοποίηση της οντότητας του χρήστη (User). Η κλάση αυτή περιέχει τα χαρακτηριστικά username και password που σχετίζονται με την σύνδεση του χρήστη στην εφαρμογή.

Entity Book



Σχήμα 4: Κλάση user του συστήματος.

2.1 Αναγνώριση Πόρων (Resources) Συστήματος

Κλάση BEC	Πόρος REST	Endpoints (HTTP Verbs)
User	/user	POST
User	/user/login	PUT
Rating, (Book, User)	/user/{username}/book/{bookID}/rating	POST, GET
Book, (User)	/user/{username}/book/{bookID}	GET

Recommended, (User)	**/user/{username}/recommended**	**GET**
Rating, (Book, User)	**/user/{username}/book/{bookID}/rating/{ratingID}**	**GET**
Wishlist, (Book, User)	**/user/{username}/wishlist**	**PUT**
PersonalInfo, (User)	**/user/{username}/personalInfo**	**POST**
PersonalInfo, (User)	**/user/{username}/personalInfo/{infoID}**	**PUT**
SearchByTitleCategoryContol, (User)	**/user/{username}/searchByTitle**	**GET**
SearchByTitleCategoryControl, (User)	**/user/{username}/searchByCategory**	**GET**

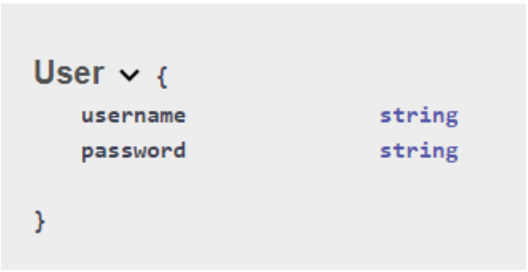
2.2 Τεκμηρίωση REST διεπαφής

Για το specification του API της εφαρμογής μας χρησιμοποιήθηκε το εργαλείο Swaggerhub, καθώς και ο Swagger Editor. Παραθέτουμε στη συνέχεια τους ενεργούς υπερσυνδέσμους για τα ζητούμενα αρχεία:

- Swagger API
- Dropbox json file
- Dropbox server

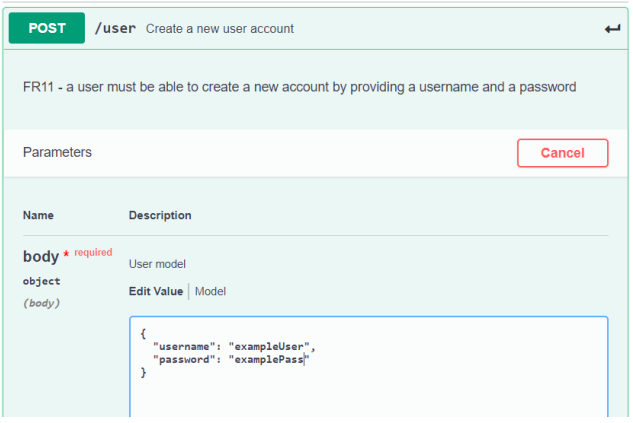
2.2.1 Πόρος User

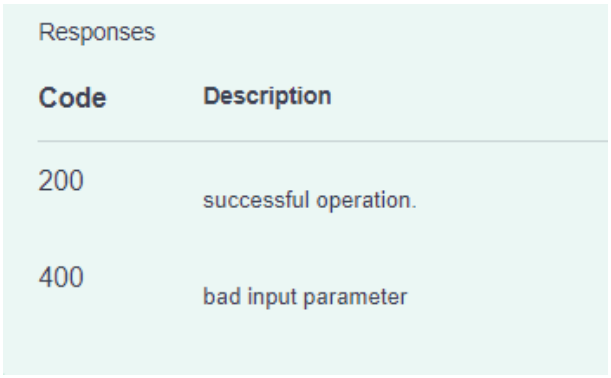
2.2.1.1 Μοντέλο δεδομένων user



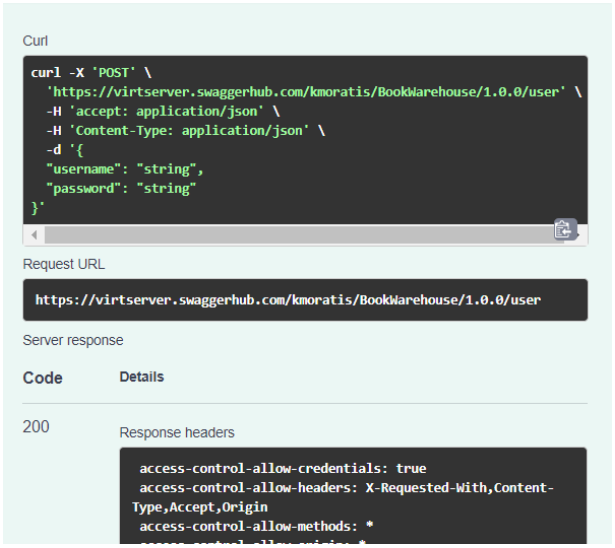
Σχήμα 5: Μοντέλο δεδομένων User.

2.2.1.2 Endpoint POST πόρου User



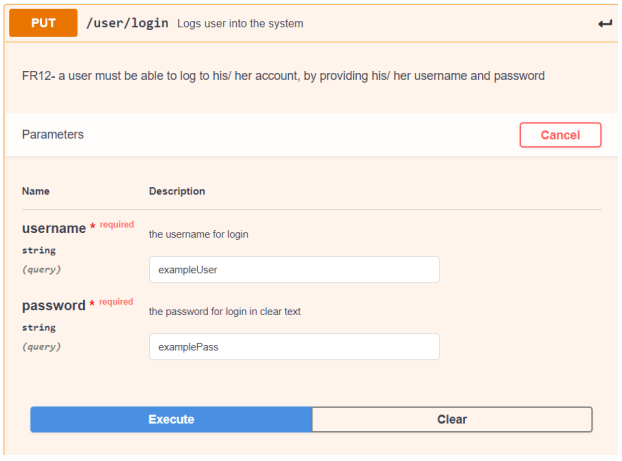


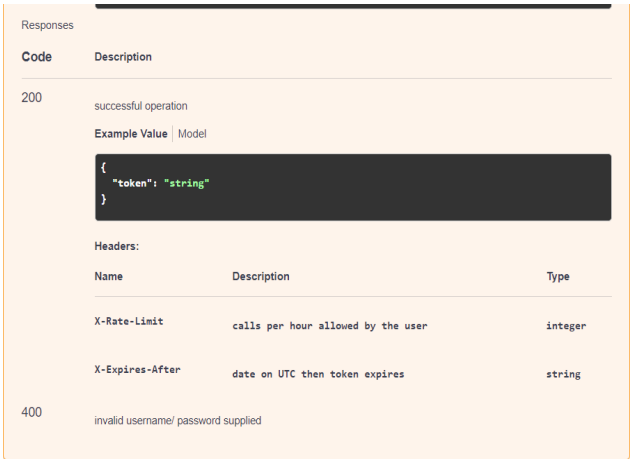
Σχήμα 6: Επεξήγηση και παράθεση των παραμέτρων του πόρου.
Σχήμα 7: Δυνατές αποκρίσεις του server.



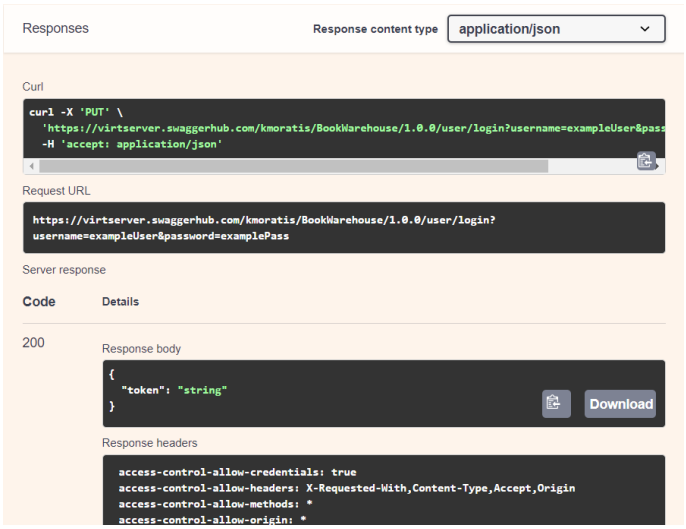
Σχήμα 8: Δοκιμή του endpoint.

2.2.1.3 Endpoint PUT πόρου User





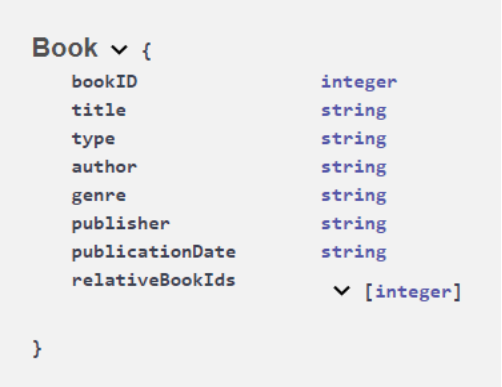
Σχήμα 9: Επεξήγηση και παράθεση των παραμέτρων του πόρου.
Σχήμα 10: Δυνατές αποκρίσεις του server.



Σχήμα 11: Δοκιμή του endpoint.

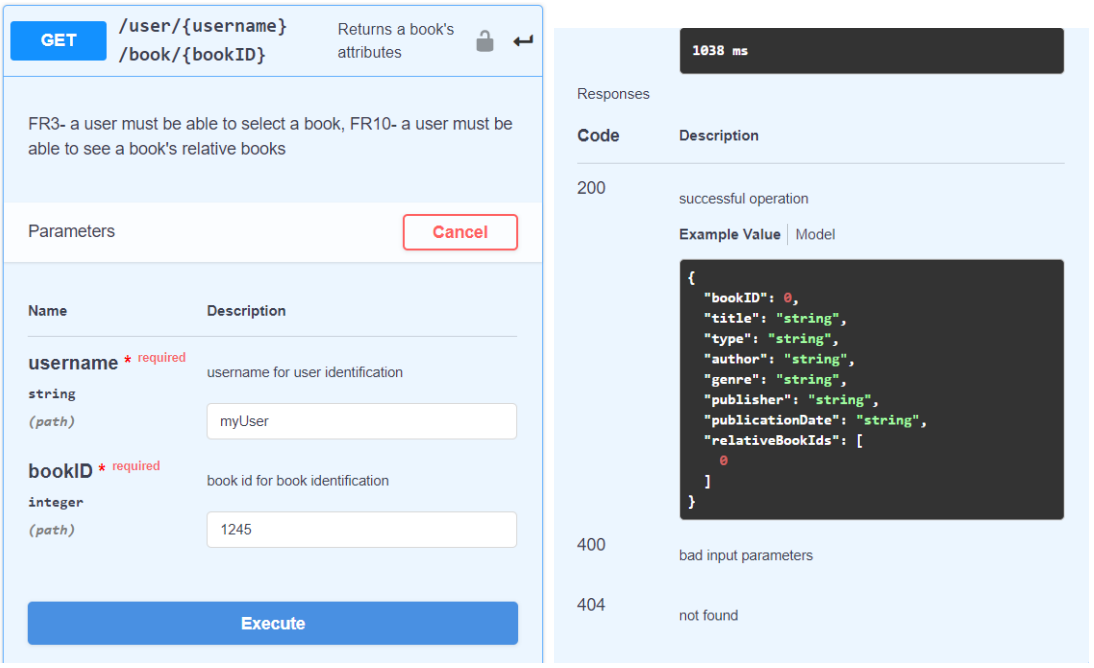
2.2.2 Πόρος Book

2.2.2.1 Μοντέλο δεδομένων Book



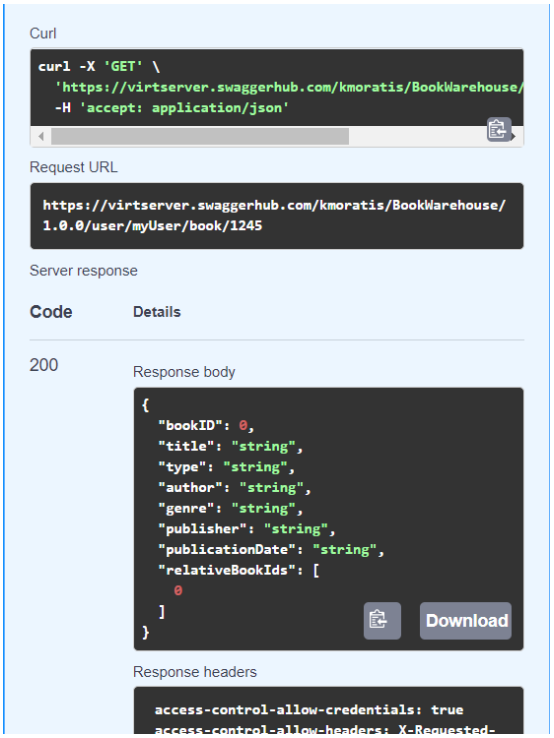
Σχήμα 12: Μοντέλο δεδομένων Book.

2.2.2.2 Endpoint GET πόρου Book, για συγκεκριμένο username και book id.



Σχήμα 13: Επεξήγηση και παράθεση των παραμέτρων του πόρου.

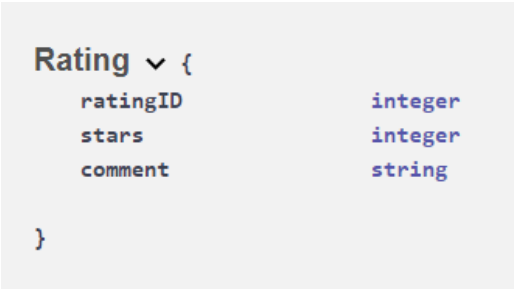
Σχήμα 14: Δυνατές αποκρίσεις του server.



Σχήμα 15: Δοκιμή του endpoint.

2.2.3 Πόρος Rating

2.2.3.1 Μοντέλο δεδομένων Rating



Σχήμα 16: Μοντέλο δεδομένων Rating.

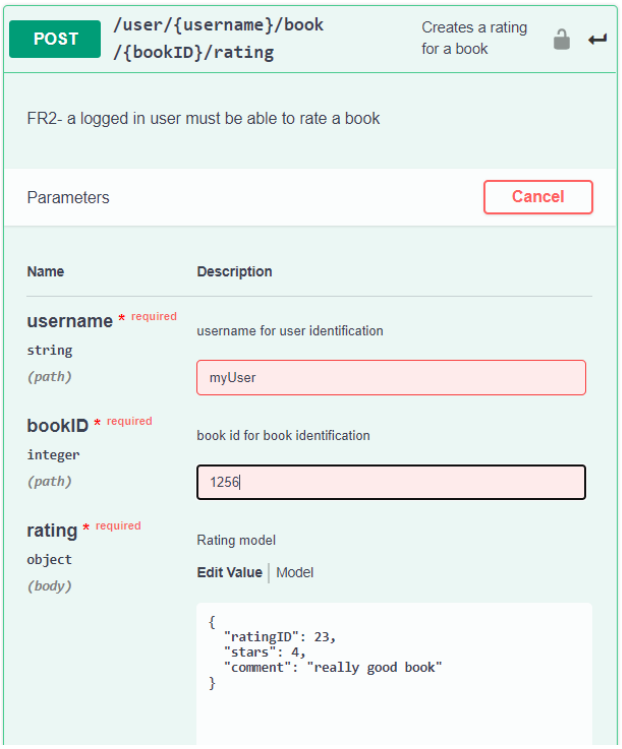
2.2.3.2 Μοντέλο δεδομένων AllRatings

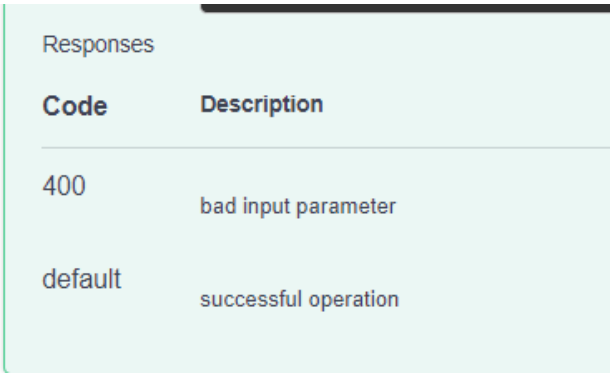
```
AllRatings [AllRatings {
    ratingID            integer
    stars               integer
    comment             string

}]
```

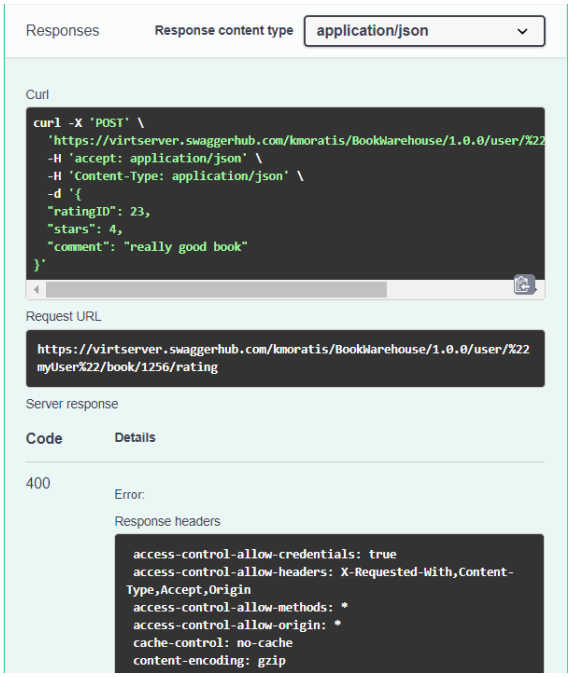
Σχήμα 17: Μοντέλο δεδομένων AllRatings.

2.2.3.3 Endpoint POST πόρου Rating, για συγκεκριμένο username και book id.



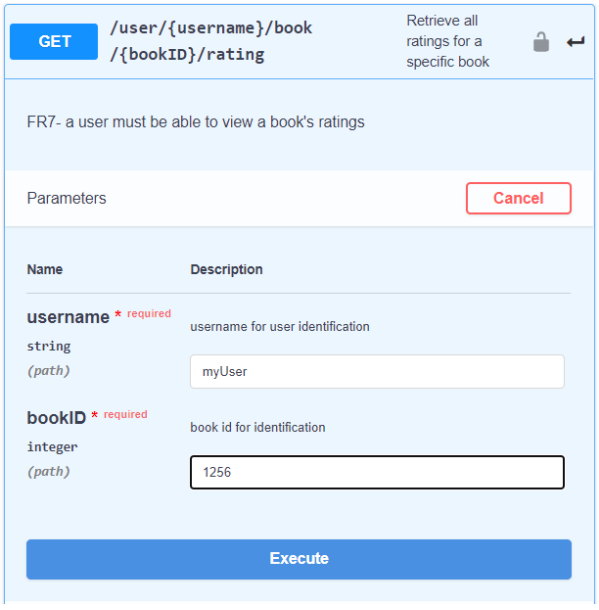


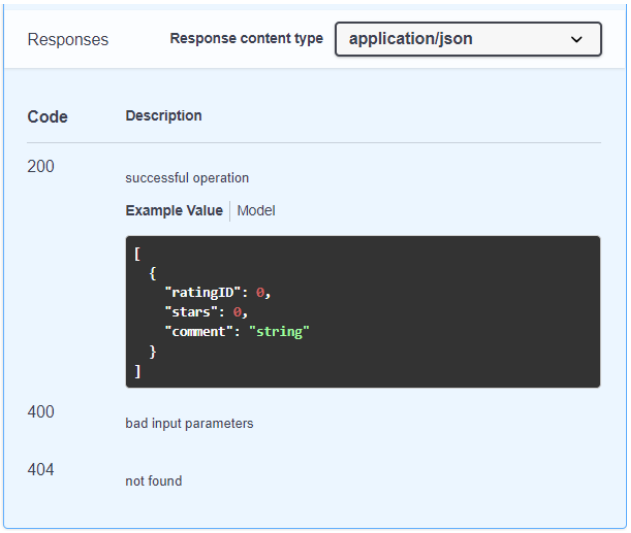
Σχήμα 18: Επεξήγηση και παράθεση των παραμέτρων του πόρου.
Σχήμα 19: Δυνατές αποκρίσεις του server.



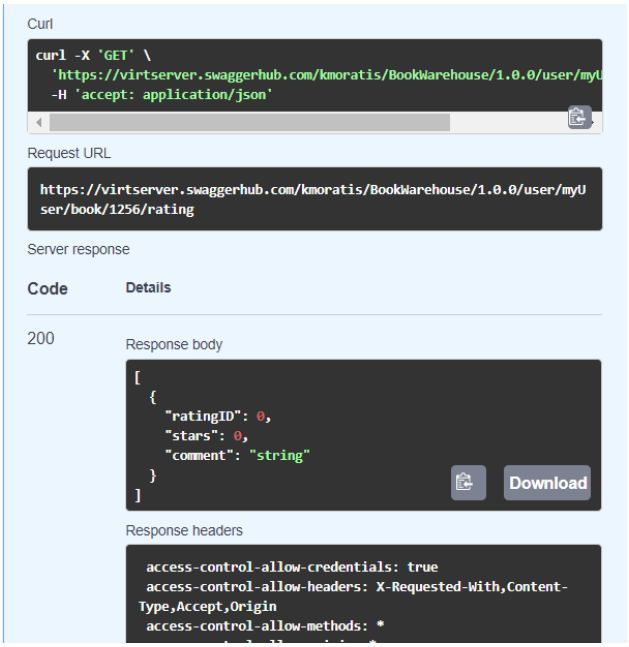
Σχήμα 20: Δοκιμή του endpoint.

2.2.3.4 Endpoint GET πόρου Rating, για συγκεκριμένο username και book id.



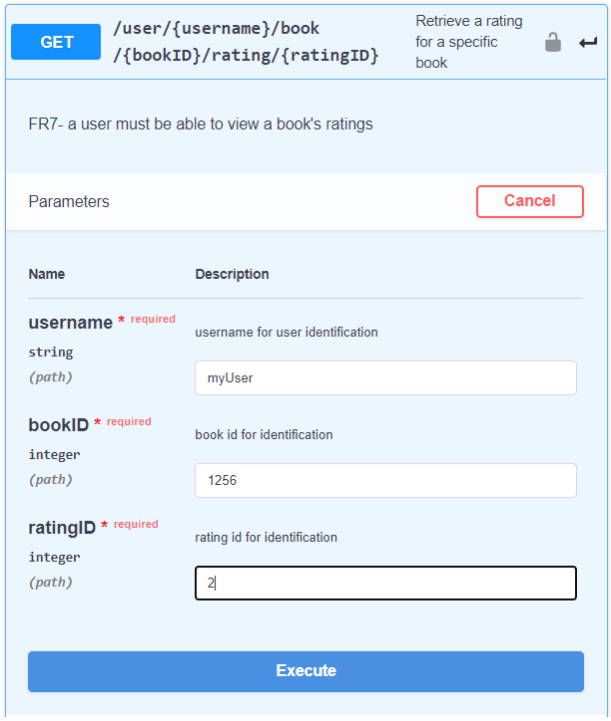


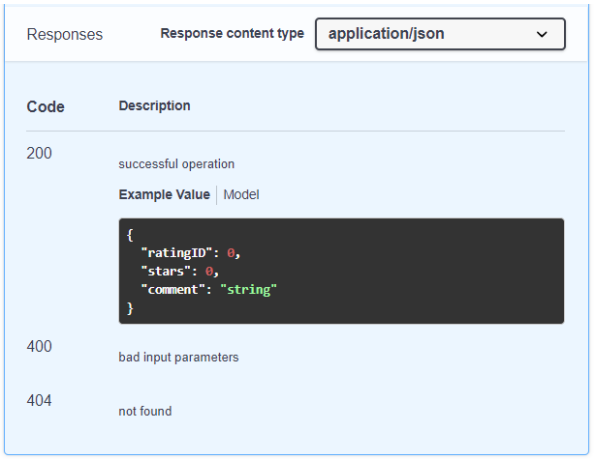
Σχήμα 21: Επεξήγηση και παράθεση των παραμέτρων του πόρου.
Σχήμα 22: Δυνατές αποκρίσεις του server.



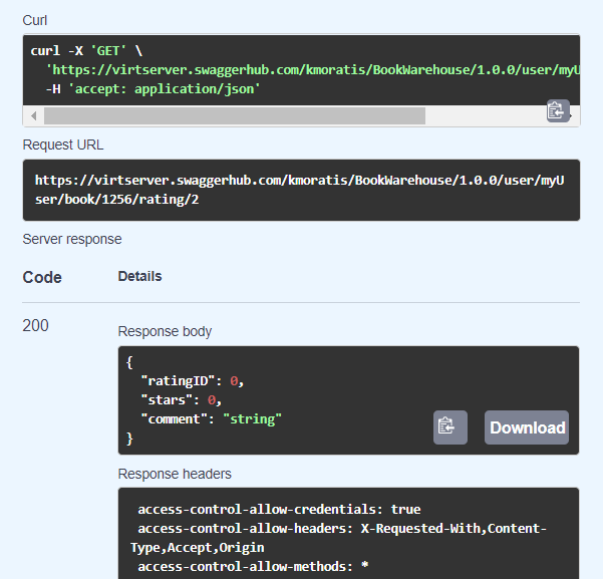
Σχήμα 23: Δοκιμή του endpoint.

2.2.3.5 Endpoint GET πόρου Rating, για συγκεκριμένο username, συγκεκριμένο book id και rating id.





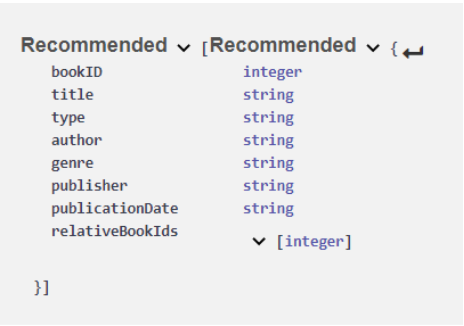
Σχήμα 24: Επεξήγηση και παράθεση των παραμέτρων του πόρου.
Σχήμα 25: Δυνατές αποκρίσεις του server.



Σχήμα 26: Δοκιμή του endpoint.

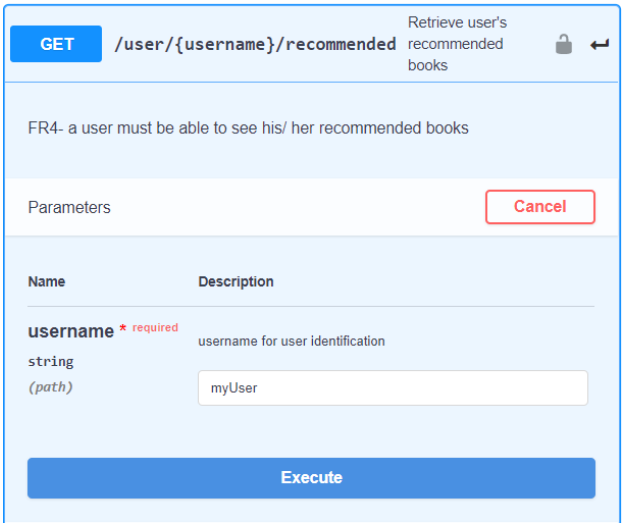
2.2.4 Πόρος Recommended

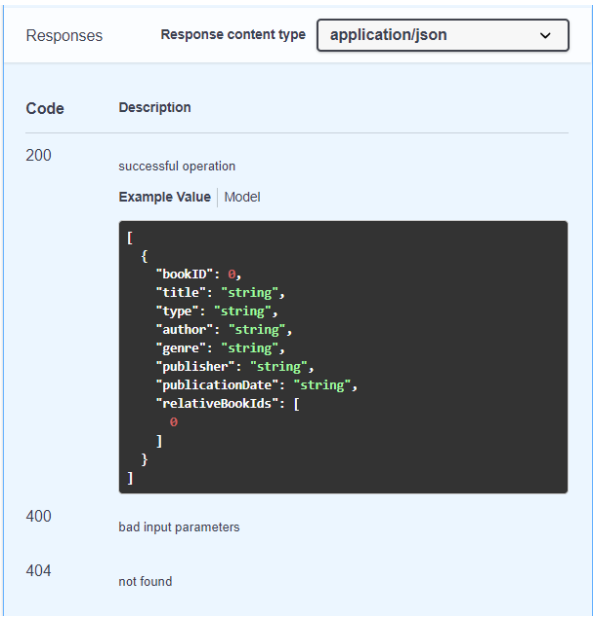
2.2.4.1 Μοντέλο δεδομένων Recommended



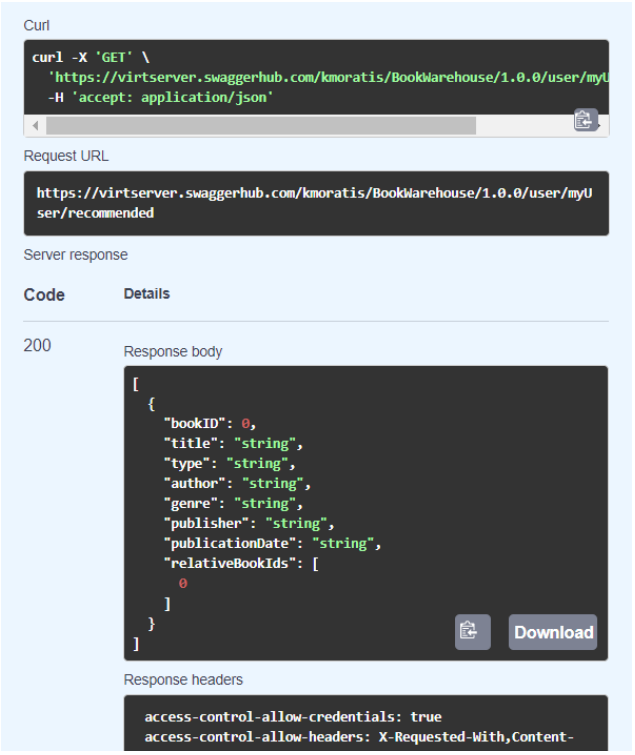
Σχήμα 27: Μοντέλο δεδομένων Recommended.

2.2.4.2 Endpoint GET πόρου Recommended, για συγκεκριμένο username.





Σχήμα 28: Επεξήγηση και παράθεση των παραμέτρων του πόρου.
Σχήμα 29: Δυνατές αποκρίσεις του server.



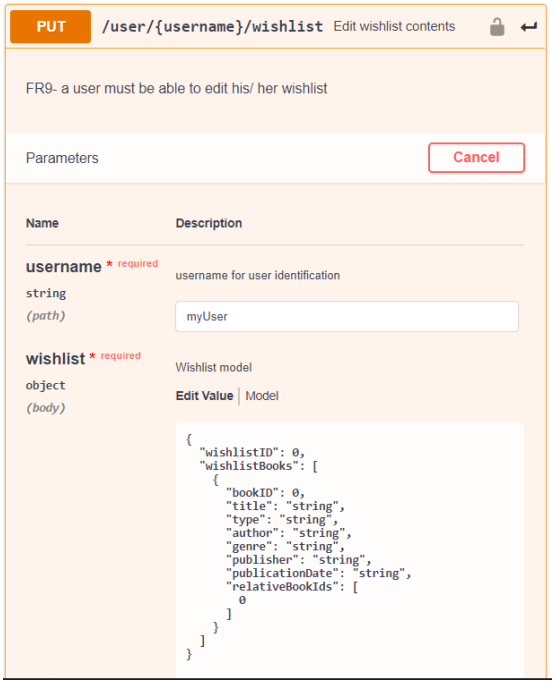
Σχήμα 30: Δοκιμή του Endpoint.

2.2.5 Πόρος Wishlist

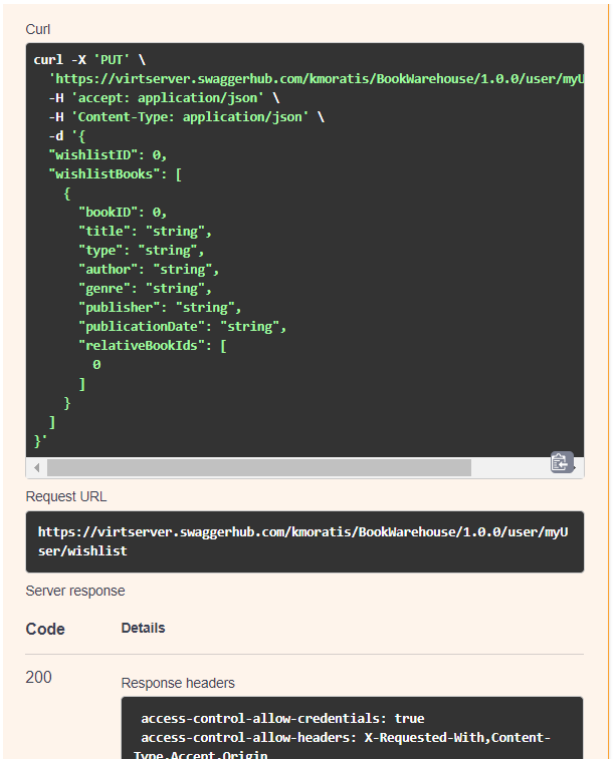
2.2.5.1 Μοντέλο δεδομένων Wishlist

Σχήμα 31: Μοντέλο δεδομένων Wishlist.

2.2.5.2 Endpoint PUT πόρου Wishlist, για συγκεκριμένο username.



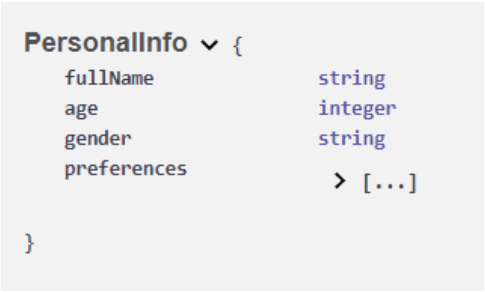
Σχήμα 32: Επεξήγηση και παράθεση των παραμέτρων του πόρου.
Σχήμα 33: Δυνατές αποκρίσεις του server.



Σχήμα 34: Δοκιμή του endpoint.

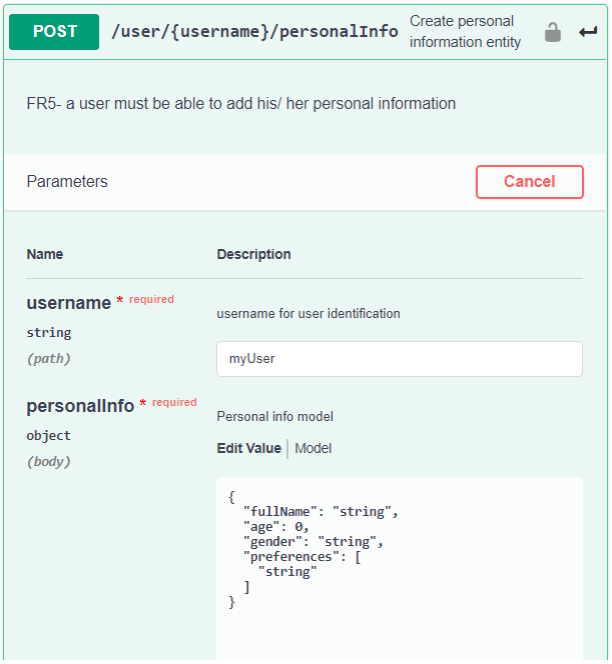
2.2.6 Πόρος PersonalInfo

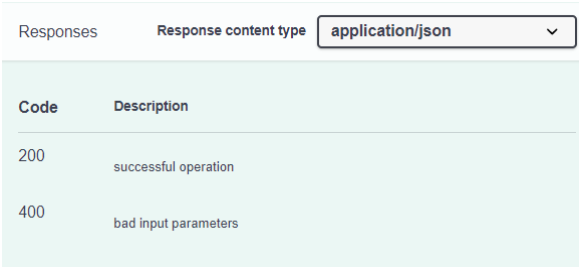
2.2.6.1 Μοντέλο δεδομένων PersonalInfo



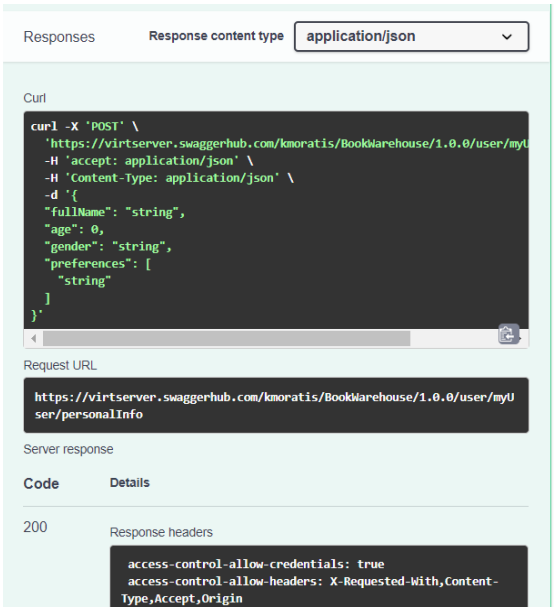
Σχήμα 35: Μοντέλο δεδομένων PersonalInfo.

2.2.6.2 Endpoint POST πόρου PersonalInfo, για συγκεκριμένο username.



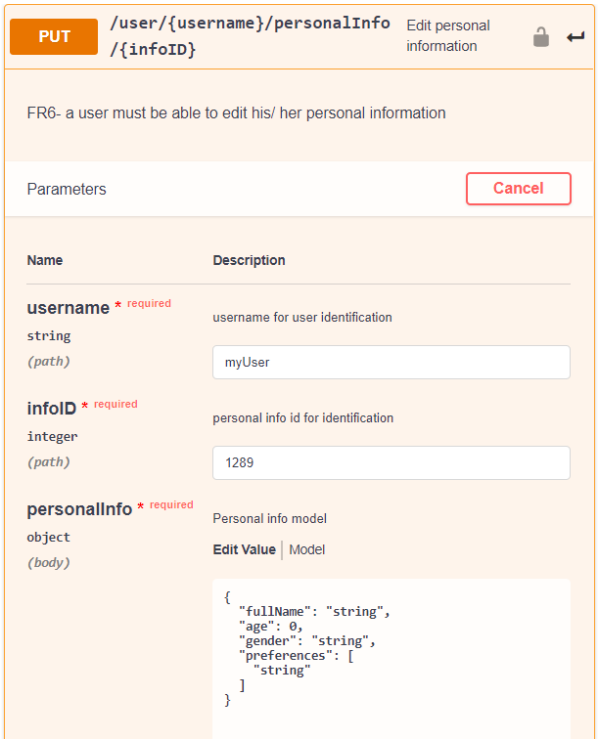


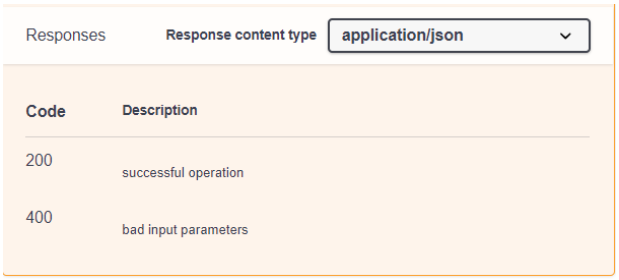
Σχήμα 36: Επεξήγηση και παράθεση των παραμέτρων του πόρου.
Σχήμα 37: Δυνατές αποκρίσεις του server.



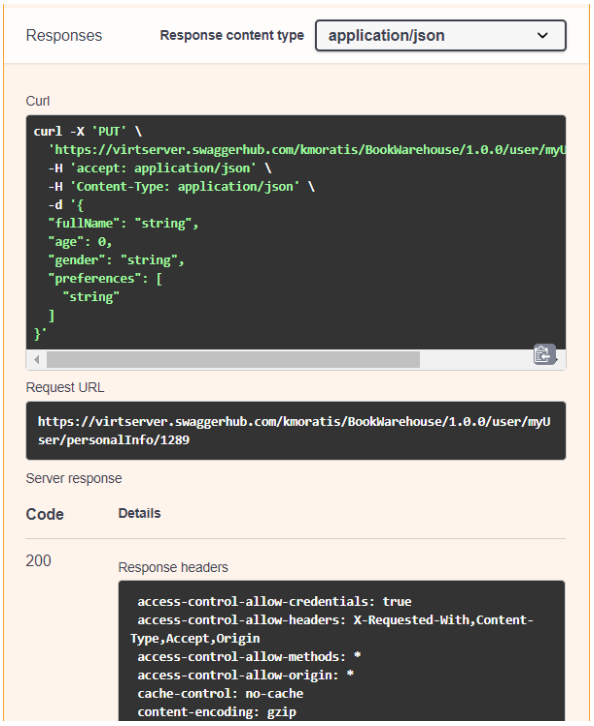
Σχήμα 38: Δοκιμή του endpoint.

2.2.6.3 Endpoint PUT πόρου PersonalInfo, για συγκεκριμένο username και info id.





Σχήμα 39: Επεξήγηση και παράθεση των παραμέτρων του πόρου.
Σχήμα 40: Δυνατές αποκρίσεις του server.



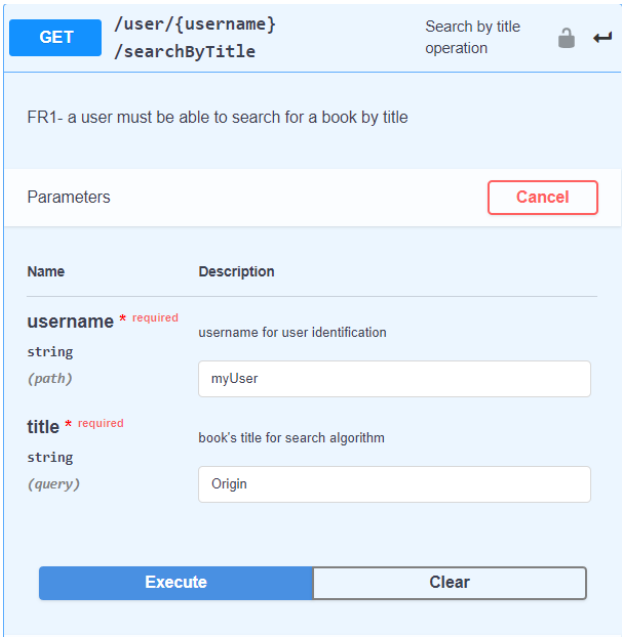
Σχήμα 41: Δοκιμή του endpoint.

2.2.7 Αλγοριθμικός πόρος SearchByTitle

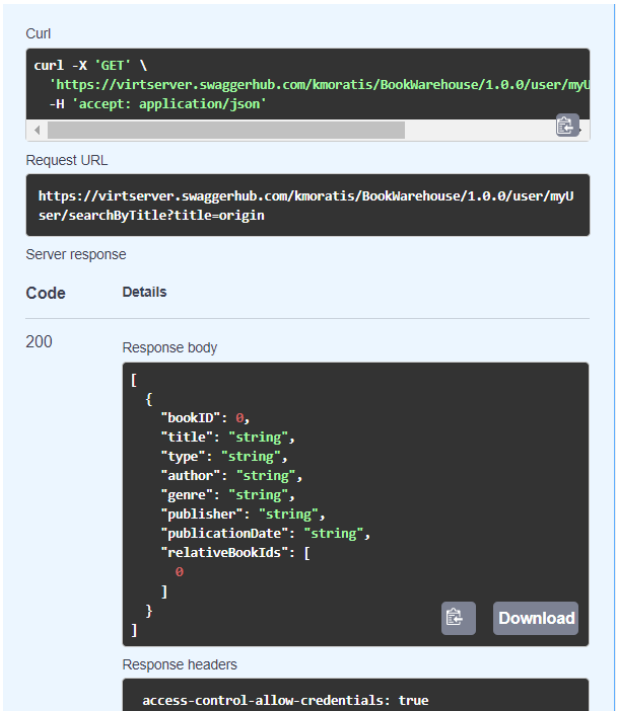
2.2.7.1 Μοντέλο δεδομένων SearchResults

Σχήμα 42: Μοντέλο δεδομένων SearchResults.

2.2.7.2 Endpoint GET πόρου SearchByTitle, για συγκεκριμένο username.



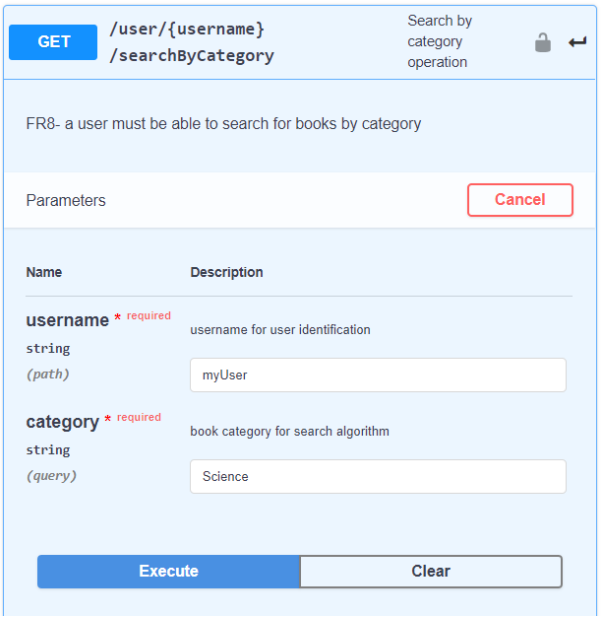
Σχήμα 43: Επεξήγηση και παράθεση των παραμέτρων του πόρου.
Σχήμα 44: Δυνατές αποκρίσεις του server.

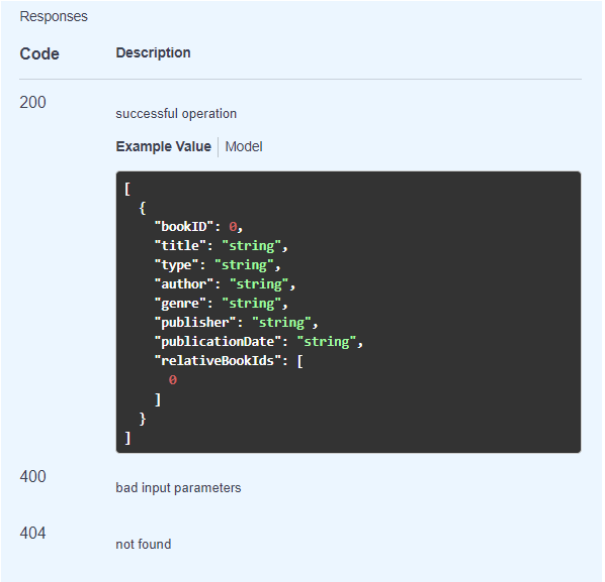


Σχήμα 45: Δοκιμή του endpoint.

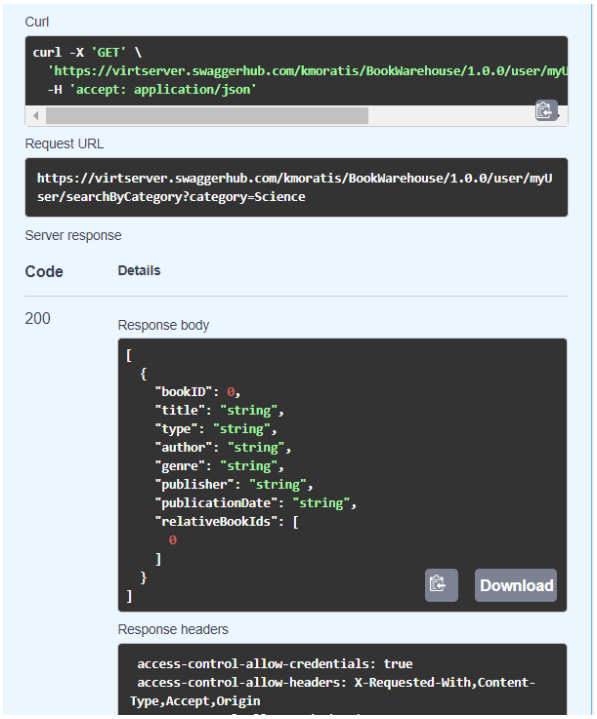
2.2.8 Αλγοριθμικός πόρος SearchByCategory

2.2.8.1 Endpoint GET πόρου SearchByCategory, για συγκεκριμένο username.





Σχήμα 46: Επεξήγηση και παράθεση των παραμέτρων του πόρου.
Σχήμα 47: Δυνατές αποκρίσεις του server.



Σχήμα 48: Δοκιμή του endpoint.

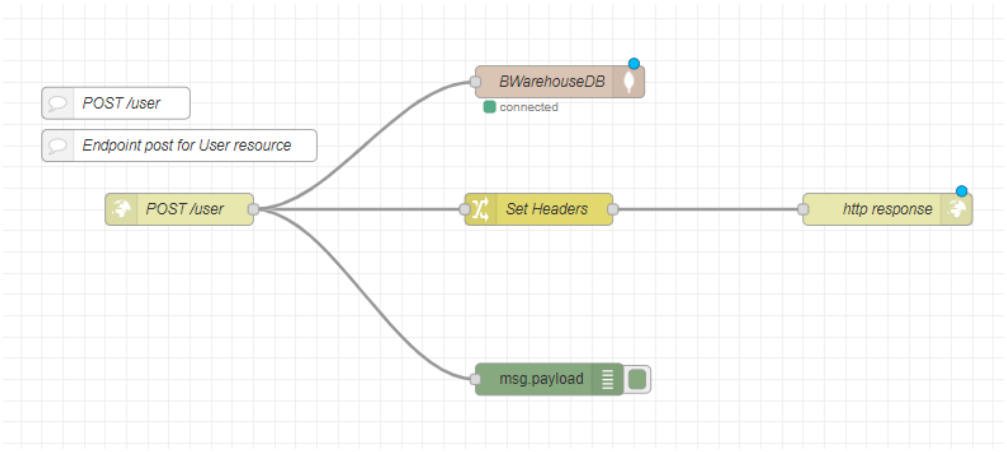
3. Υλοποίηση Συστήματος με Node-RED

- Σύνδεσμος για το αρχείο τύπου zip που περιέχει τα flows που υλοποιήθηκαν με node-red.
- Σύνδεσμος για το αρχείο τύπου zip που περιέχει την MongoDB βάση δεδομένων που χρησιμοποιήθηκε στις δοκιμές τις εφαρμογής.

3.1 Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed

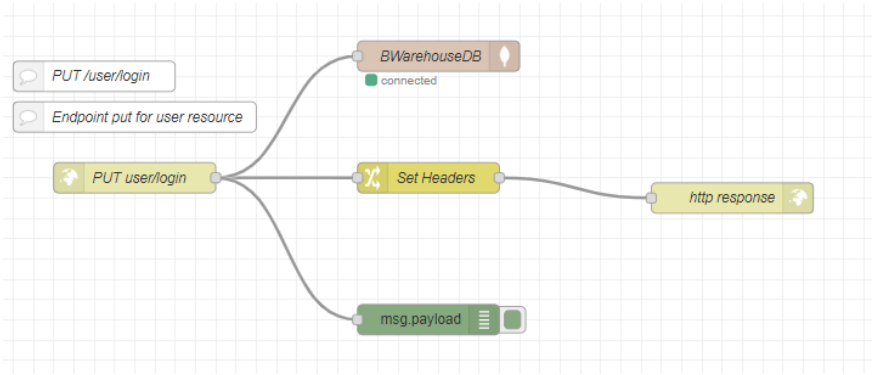
3.1.1 Ροές πόρου User

Ροή του endpoint POST /user:
Η συγκεκριμένη ροή υλοποιεί την υπηρεσία μέσω της οποίας ένας χρήστης μπορεί να δημιουργεί έναν λογαριασμό στην βάση δεδομένων.



Σχήμα 49: Ροή endpoint POST /user.

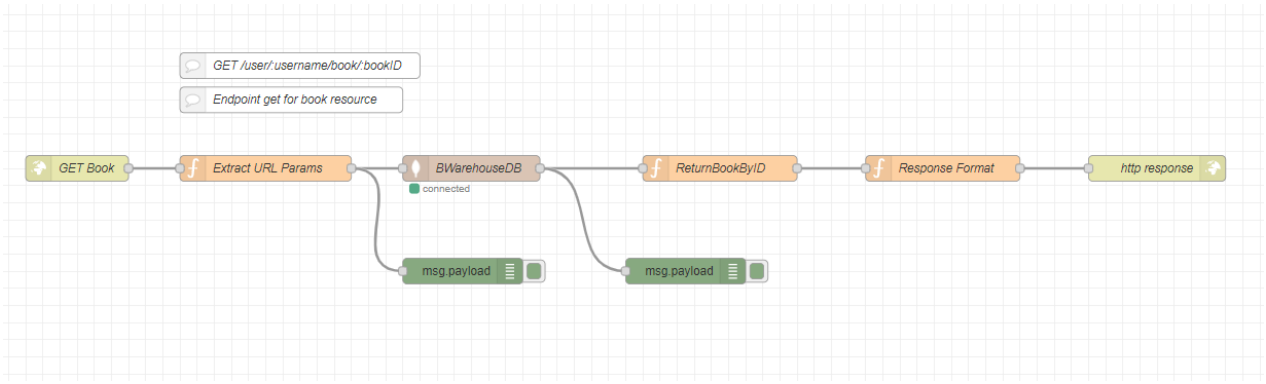
Ροή του endpoint PUT /user/login:
Με τη συγκεκριμένη ροή υλοποιείται η υπηρεσία με την οποία ένας χρήστης μπορεί να συνδεθεί στο λογαριασμό του, αλλάζοντας την κατάσταση του λογαριασμού.



Σχήμα 50: Ροή endpoint PUT /user/login.

3.1.2 Ροή πόρου Book

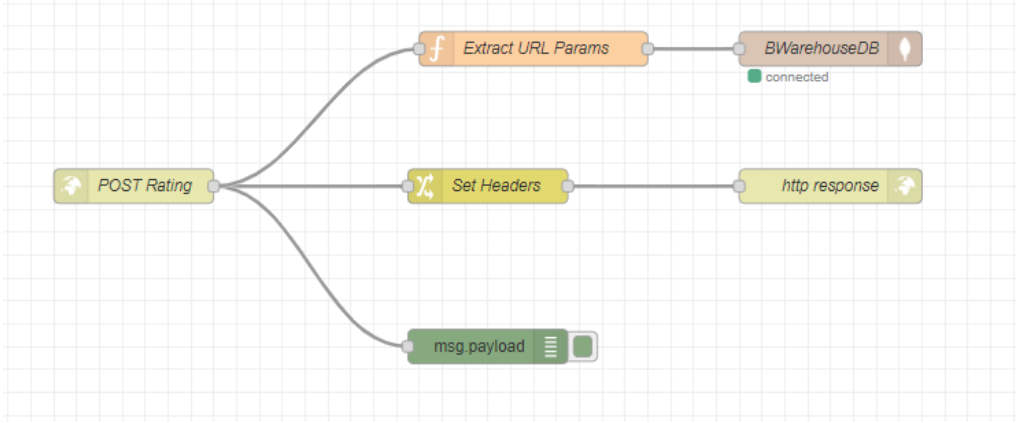
Ροή του endpoint GET /user/{username}/book/{bookID}:
Η υπηρεσία αυτή είναι υπεύθυνη για την επιστροφή των χαρακτηριστικών ενός συγκεκριμένου βιβλίου, δεδομένου το id του βιβλίου, το οποίο δίνεται σαν URL (path) parameter. Η συνάρτηση Response Format μετασχηματίζει την απάντηση του server στο επιθυμητό από το χρήστη format.



Σχήμα 51: Ροή endpoint GET /user/{username}/book/{bookID}.

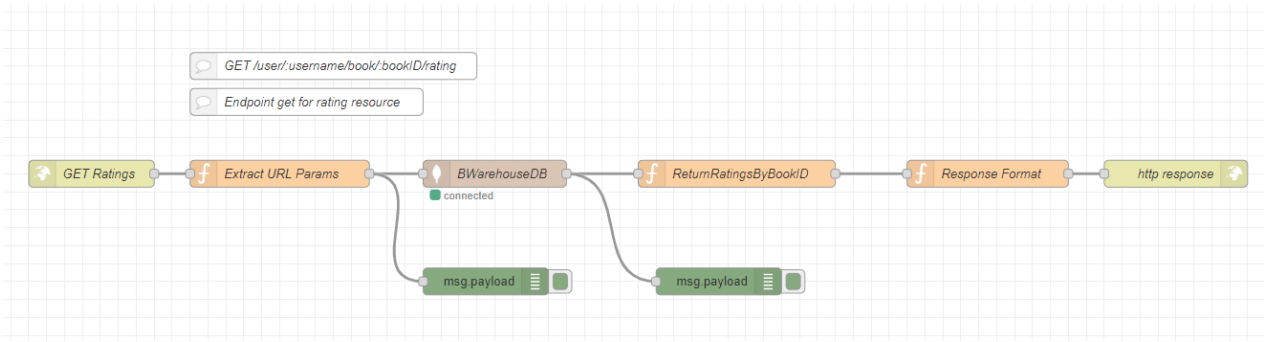
3.1.3 Ροές πόρου Rating

Ροή του endpoint POST /user/{username}/book/{bookID}/rating:
Η συγκεκριμένη ροή υλοποιεί την υπηρεσία με την οποία ο χρήστης με όνομα χρήστη {username}, μπορεί να δημιουργήσει μία καινούργια κριτική για το βιβλίο με book id το {bookID}.



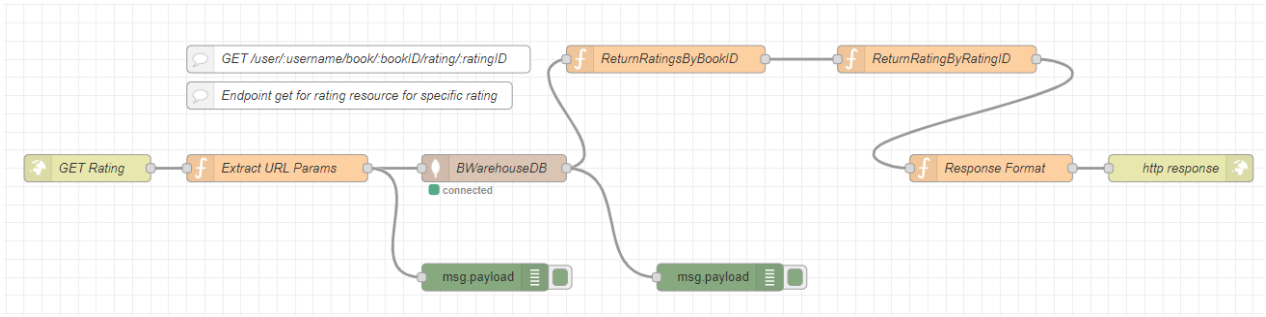
Σχήμα 52: Ροή endpoint POST /user/{username}/book/{bookID}/rating.

Ροή του endpoint GET /user/{username}/book/{bookID}/rating:
Η ροή που υλοποιεί την υπηρεσία με την οποία ο χρήστης μπορεί να παίρνει όλες τις κριτικές για ένα συγκεκριμένο βιβλίο.



Σχήμα 53: Ροή endpoint GET /user/{username}/book/{bookID}/rating.

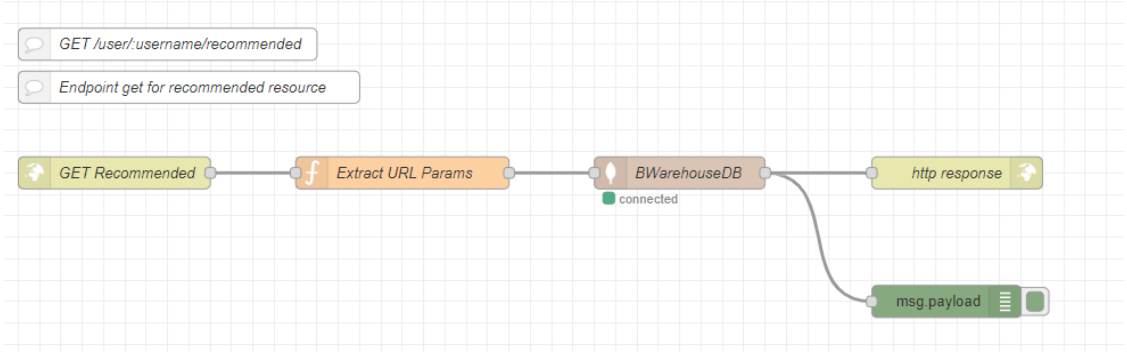
Ροή του endpoint GET /user/{username}/book/{bookID}/rating/{ratingID}:
Ροή της υπηρεσίας που επιστρέφει στο χρήστη μία συγκεκριμένη κριτική του βιβλίου με id το {bookID}. Τα rating και Book ids δίνονται από τον χρήστη και αποστέλλονται στην υπηρεσία μέσω URL parameters.



Σχήμα 54: Ροή endpoint GET /user/{username}/book/{bookID}/rating/:ratingID.

3.1.4 Ροή πόρου Recommended

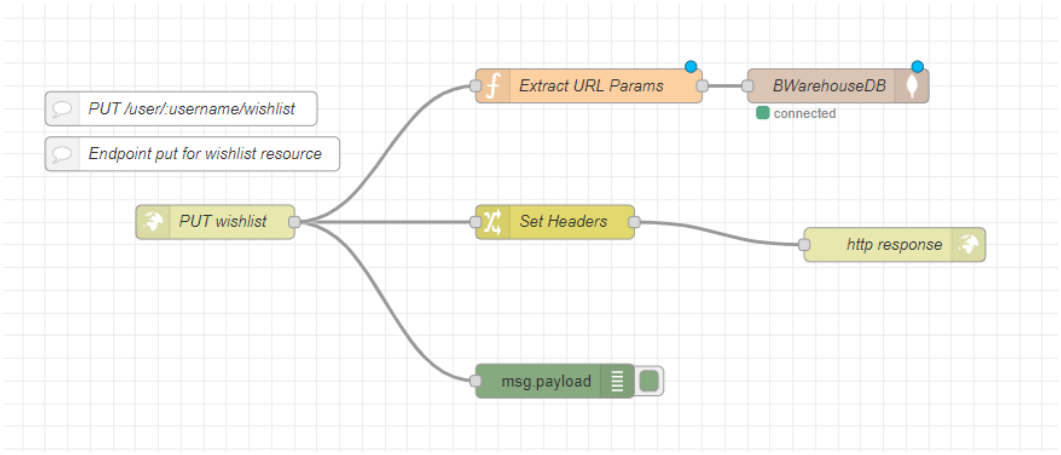
Ροή του endpoint GET /user/{username}/recommended:
Η υπηρεσία αυτή είναι υπεύθυνη για την επιστροφή των προτεινόμενων βιβλίων για ένα συγκεκριμένο χρήστη, δεδομένο το username του, το οποίο δίνεται σαν URL (path) parameter.



Σχήμα 55: Ροή endpoint GET /user/{username}/recommended.

3.1.5 Ροή πόρου Wishlist

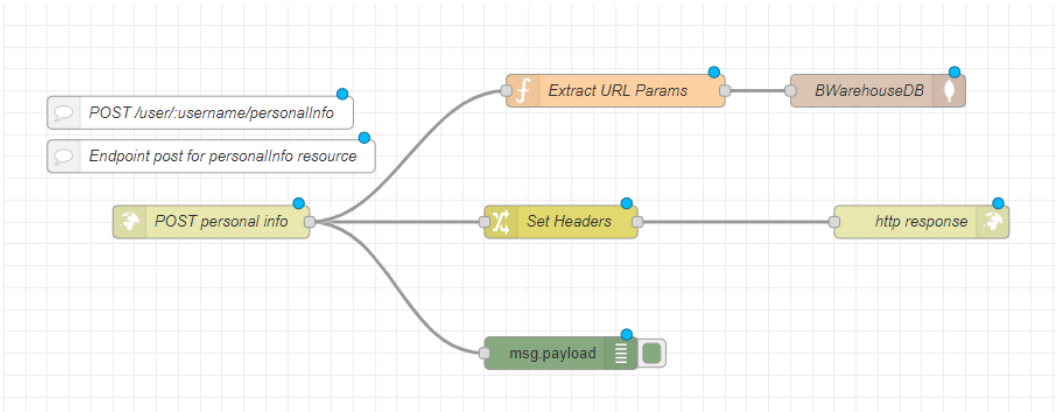
Ροή του endpoint PUT /user/{username}/wishlist:
Η συγκεκριμένη ροή υλοποιεί την υπηρεσία μέσω της οποίας ο χρήστης μπορεί να αλλάζει τα περιεχόμενα απο το wishlist του.



Σχήμα 56: Ροή endpoint PUT /user/{username}/wishlist.

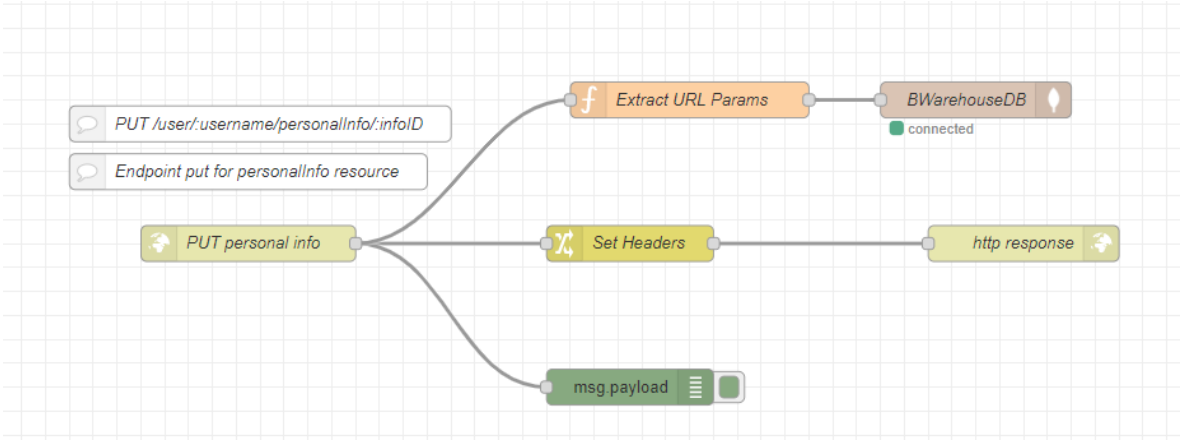
3.1.6 Ροές πόρου PersonalInfo

Ροή του endpoint POST /user/{username}/personalInfo:
Η συγκεκριμένη ροή υλοποιεί την υπηρεσία μέσω της οποίας ο χρήστης μπορεί να εισάγει τα προσωπικά του στοιχεία στο σύστημα.



Σχήμα 57: Ροή endpoint POST /user/{username}/personalInfo.

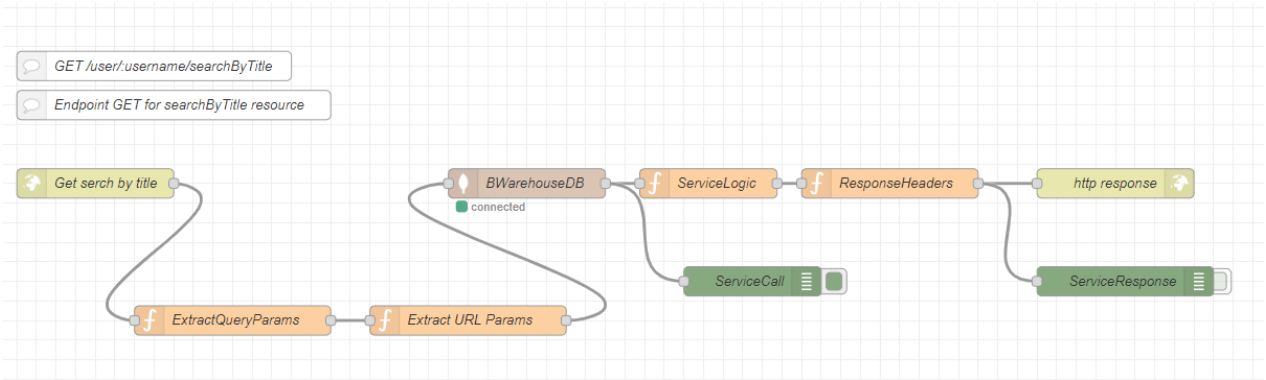
Ροή του endpoint PUT /user/{username}/personalInfo/{infoID}:
Η συγκεκριμένη ροή υλοποιεί την υπηρεσία μέσω της οποίας ο χρήστης μπορεί να τροποποιεί τα προσωπικά του στοιχεία στο σύστημα.



Σχήμα 58: Ροή endpoint PUT /user/{username}/personalInfo/{infoID}.

3.1.7 Ροή αλγοριθμικού πόρου SearchByTitle

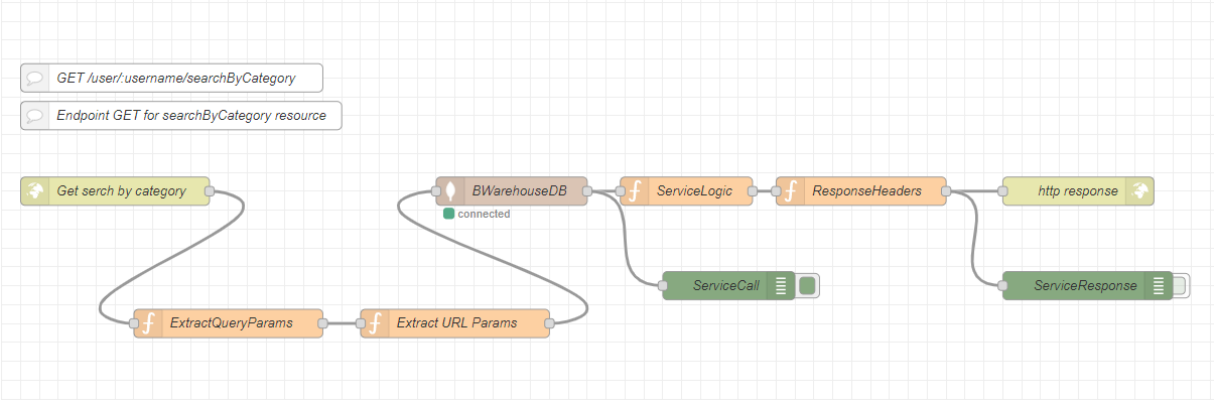
Ροή του endpoint GET /user/{username}/searchByTitle:
Με την συγκεκριμένη υπηρεσία ο χρήστης μπορεί να κάνει αναζήτηση ενός βιβλίου χρησιμοποιώντας τον τίτλο του.



Σχήμα 59: Ροή endpoint GET /user/{username}/searchByTitle.

3.1.8 Ροή αλγοριθμικού πόρου SearchByCategory

Ροή του endpoint GET /user/{username}/searchByCategory:
Με την συγκεκριμένη υπηρεσία ο χρήστης μπορεί να κάνει αναζήτηση βιβλίων χρησιμοποιώντας την κατηγορία στην οποία ανήκουν.

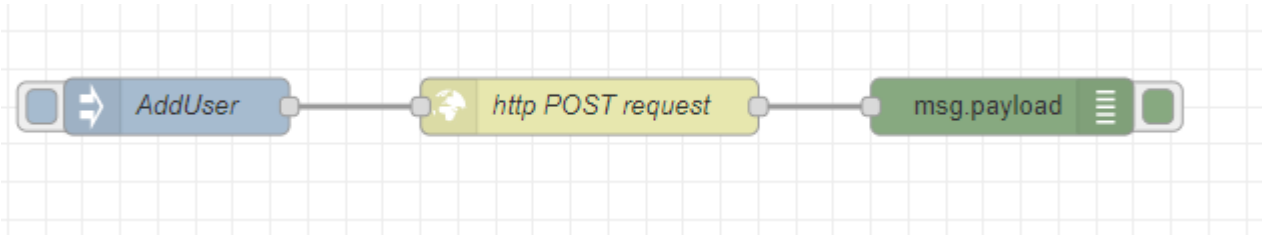


Σχήμα 60: Ροή endpoint GET /user/{username}/searchByCategory.

3.2 Υλοποίηση Ιστοριών χρήστη

3.2.1 Ιστορία Χρήστη: User create a new account

Ροή μέσω της οποίας ο χρήστης μπορεί να δημιουργήσει έναν καινούργιο λογαριασμό στο σύστημα, παρέχοντας ένα όνομα χρήστη και έναν κωδικό πρόσβασης.

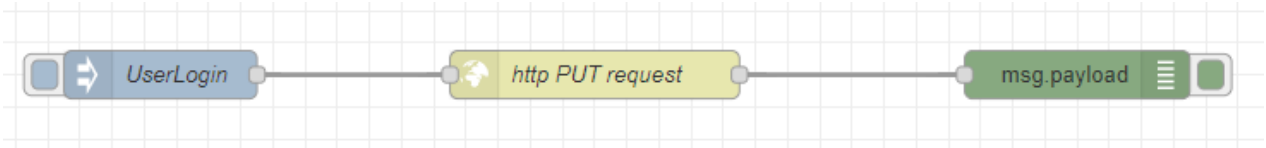


Σχήμα 61: Ροή ιστορίας user create account.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
AddUser	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http POST request	http-request	Κάνει κλήση προς την υπηρεσία POST /user η οποία δίνει στο χρήστη τη δυνατότητα να δημιουργήσει έναν λογαριασμό.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη.

3.2.2 Ιστορία Χρήστη: User login to account

Ροή μέσω της οποίας ο χρήστης μπορεί να συνδεθεί στο λογαριασμό του, παρέχοντας το όνομα χρήστη και τον κωδικό πρόσβασης.



Σχήμα 62: Ροή ιστορίας user login to account.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
UserLogin	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http PUT request	http-request	Κάνει κλήση προς την υπηρεσία PUT /user/login η οποία δίνει στο χρήστη τη δυνατότητα να συνδεθεί σε ένα λογαριασμό.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη.

3.2.3 Ιστορία Χρήστη: User select a book

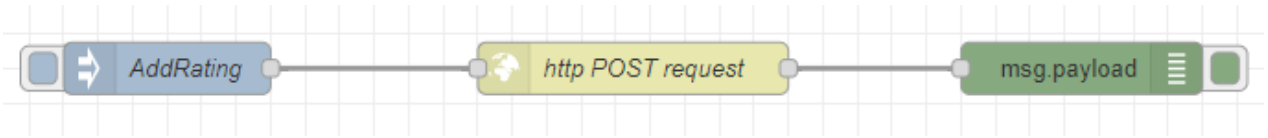
Ροή μέσω της οποίας ο χρήστης μπορεί να επιλέξει ένα βιβλίο, δηλαδή να του επιστραφούν από τον server τα χαρακτηριστικά του βιβλίου με id το {bookID}.

Σχήμα 63: Ροή ιστορίας user select book.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
GetBook	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http GET request	http-request	Κάνει κλήση προς την υπηρεσία GET /user/{username}/book/{bookID} η οποία δίνει στο χρήστη τη δυνατότητα να επιλέξει ένα βιβλίο για να δει τα χαρακτηριστικά του.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη, δηλαδή τον κατάλληλο κωδικό και το σώμα του επιλεγμένου βιβλίου.

3.2.4 Ιστορία Χρήστη: User rate a book

Ροή μέσω της οποίας ο χρήστης μπορεί να αξιολογήσει ένα βιβλίο.



Σχήμα 64: Ροή ιστορίας user rate book.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
AddRating	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http POST request	http-request	Κάνει κλήση προς την υπηρεσία POST /user/{username}/book/{bookID}/rating η οποία δίνει στο χρήστη τη δυνατότητα να αξιολογήσει το συγκεκριμένο βιβλίο.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη.

3.2.5 Ιστορία Χρήστη: User see book's ratings

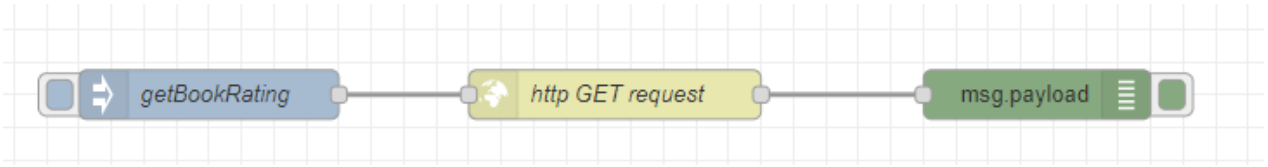
Ροή μέσω της οποίας ο χρήστης μπορεί να δει τις αξιολογήσεις ενός βιβλίου.

Σχήμα 65: Ροή ιστορίας user see book's ratings.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
getBookRatings	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http GET request	http-request	Κάνει κλήση προς την υπηρεσία GET /user/{username}/book/{bookID}/rating η οποία δίνει στο χρήστη τη δυνατότητα να δει τις αξιολογήσεις για το συγκεκριμένο βιβλίο.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη, δηλαδή τον κατάλληλο κωδικό και το σώμα των Rating του βιβλίου.

3.2.6 Ιστορία Χρήστη: User see a book's rating

Ροή μέσω της οποίας ο χρήστης μπορεί να δει μία συγκεκριμένη αξιολόγηση ενός βιβλίου.



Σχήμα 66: Ροή ιστορίας user see a book's rating.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
getBookRating	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http GET request	http-request	Κάνει κλήση προς την υπηρεσία GET /user/{username}/book/{bookID}/rating/{ratingID} η οποία δίνει στο χρήστη τη δυνατότητα να δει την αξιολόγηση με id to {ratingID} του συγκεκριμένου βιβλίου.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη, δηλαδή τον κατάλληλο κωδικό και το σώμα του Rating του βιβλίου.

3.2.7 Ιστορία Χρήστη: User get recommended

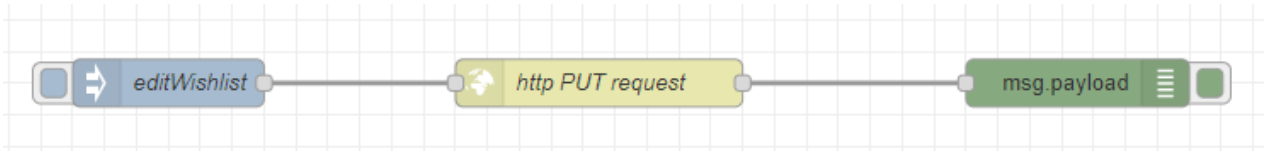
Ροή μέσω της οποίας ο χρήστης μπορεί να δει τα προτεινόμενα για αυτόν βιβλία.

Σχήμα 67: Ροή ιστορίας user get recommended.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
getRecommended	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http GET request	http-request	Κάνει κλήση προς την υπηρεσία GET /user/{username}/recommended η οποία δίνει στο χρήστη τη δυνατότητα να δει τα προτεινόμενα για αυτόν βιβλία.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη, δηλαδή τον κατάλληλο κωδικό και τα προτεινόμενα βιβλία.

3.2.8 Ιστορία Χρήστη: User edit wishlist

Ροή μέσω της οποίας ο χρήστης μπορεί να τροποποιήσει τα περιεχόμενα του wishlist του.

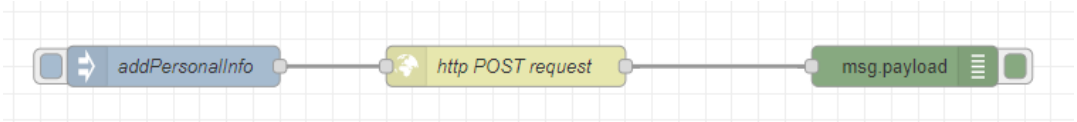


Σχήμα 68: Ροή ιστορίας user edit wishlist.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
editWishlist	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http PUT request	http-request	Κάνει κλήση προς την υπηρεσία PUT /user/{username}/wishlist η οποία δίνει στο χρήστη τη δυνατότητα να τροποποιήσει το wishlist του.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη.

3.2.9 Ιστορία Χρήστη: User add personal information

Ροή μέσω της οποίας ο χρήστης μπορεί να εισάγει τα προσωπικά του στοιχεία στο σύστημα.



Σχήμα 69: Ροή ιστορίας user add personal information.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
addPersonalInfo	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http POST request	http-request	Κάνει κλήση προς την υπηρεσία POST /user/{username}/personalInfo η οποία δίνει στο χρήστη τη δυνατότητα να εισάγει τα προσωπικά του στοιχεία.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη.

3.2.10 Ιστορία Χρήστη: User edit personal information

Ροή μέσω της οποίας ο χρήστης μπορεί να τροποποιεί τα προσωπικά του στοιχεία.

Σχήμα 70: Ροή ιστορίας user edit personal information.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
editPersonalInfo	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http PUT request	http-request	Κάνει κλήση προς την υπηρεσία PUT /user/{username}/personalInfo/{personalInfo} η οποία δίνει στο χρήστη τη δυνατότητα να τροποποιήσει τα προσωπικά του στοιχεία.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη.

3.2.11 Ιστορία Χρήστη: User search book by title

Ροή μέσω της οποίας ο χρήστης έχει την δυνατότητα να αναζητήσει ένα βιβλίο με βάση τον τίτλο του.

Σχήμα 71: Ροή ιστορίας user search book by title.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
getSearchByTitle	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http GET request	http-request	Κάνει κλήση προς την υπηρεσία GET /user/{username}/searchByTitle η οποία δίνει στο χρήστη τη δυνατότητα να κάνει κάνει αναζήτηση ενός βιβλίου με βάση τον τίτλο του.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη, δηλαδή τον κατάλληλο κωδικό και τα επιστρεφόμενα βιβλία της αναζήτησης.

3.2.12 Ιστορία Χρήστη: User search book by category

Ροή μέσω της οποίας ο χρήστης μπορεί να κάνει αναζήτηση βιβλίων με βάση την κατηγορία στην οποία ανήκουν.

Σχήμα 72: Ροή ιστορίας user search book by category.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
getSearchByCategory	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής και για τον καθορισμό των παραμέτρων.
http GET request	http-request	Κάνει κλήση προς την υπηρεσία GET /user/{username}/searchByCategory η οποία δίνει στο χρήστη τη δυνατότητα να κάνει κάνει αναζήτηση βιβλίων με βάση την κατηγορία τους.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος για τη συγκεκριμένη κλήση της υπηρεσίας από το χρήστη, δηλαδή τον κατάλληλο κωδικό και τα επιστρεφόμενα βιβλία της αναζήτησης.

Παράρτημα Ι – Ανοιχτά Θέματα

1. Ορισμένες από τις ροές, και επομένως και απο τις Λειτουργικές Απαιτήσεις, που έχουμε ορίσει και προδιαγράψει απαιτούν την υλοποίηση περαιτέρω ροών για να προσφέρουν λειτουργικότητα σε πραγματικές συνθήκες. Για παράδειγμα, στην μέχρι τώρα ανάπτυξη της εφαρμογής μας, ο χρήστης μπορεί να προσθαφαιρέσει βιβλία στο wishlist του, αλλά χωρίς την υλοποίηση μιας Λειτουργικής Απαίτησης για να μπορεί να δει τα βιβλία αυτά, καθώς και της αντίστοιχης ροής, η πρώτη δεν προσφέρει κάποιο πρακτικό όφελος. Παρόμοια θέματα υπάρχουν στο κομμάτι της αποσύνδεσης των χρηστών από την εφαρμογή, καθώς και των ροών που θα πραγματοποιούνται σε οριακές συνθήκες, δηλαδή κατά τις πρώτες στιγμές της λειτουργίας της εφαρμογής, τις διάφορες περιπτώσεις σφαλμάτων που μπορεί να προκύψουν, καθώς και την επαναλειτουργία της εφαρμογής μετά από αυτά. Επομένως, σε μία πραγματική εφαρμογή θα απαιτούνταν στη συνέχεια της διαδικασίας ανάπτυξής της, η προσθήκη περαιτέρω λειτουργικότητας και των αντίστοιχων ροών για την κάλυψη των συγκεκριμένων αναγκών.

2. Απαιτείται επίσης ο σχεδιασμός των ροών που υλοποιούν το front-end μέρος της εφαρμογής, δηλαδή το κομμάτι που ασχολείται με τη μετατροπή και την παρουσίαση των δεδομένων που παρέχονται από τον server σε κατάλληλη μορφή, έτσι ώστε οι χρήστες να μπορούν να δουν και να αλληλεπιδράσουν με τα δεδομένα αυτά.