



FITNOW

Let's make our lives healthier

Προδιαγραφές Λογισμικού και Υλοποίηση συστήματος

Version 1.0

Αμπατζίδη Κωνσταντίνα
Μαστρακούλης Συμεών
Μπουλέτσης Αλέξης
Πρωτοψάλτης Παναγιώτης

14/06/2023



Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
Μπουλέτσας Αλέξης	03/06/2023	Δημιουργία των προτύπων σχεδιασμού	0.1
Μπουλέτσας Αλέξης	08/06/2023	Δημιουργία των endpoints στο swagger	0.2
Πρωτοψάλτης Παναγιώτης	09/06/2020	Αναθεώρηση του 1ου κεφαλαίου	0.2.1
Πρωτοψάλτης Παναγιώτης	10/06/2020	Δημιουργία των endpoints στο node-red	0.3
Κωνσταντίνα Αμπατζίδη	11/06/2020	Αναθεώρηση του 1ου και 2ου κεφαλαίου	0.3.1
Συμεών Μαστρακούλης	12/06/2020	Αναθεώρηση του 3ου κεφαλαίου	0.3.2

Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Μπουλέτσας Αλέξης	3	
Πρωτοψάλτης Παναγιώτης	3	
Κωνσταντίνα Αμπατζίδη	3	
Συμεών Μαστρακούλης	3	



Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	3
Λίστα Σχημάτων.....	4
1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν	5
2. Αρχιτεκτονική Συστήματος.....	9
2.1. Αναγνώριση Πόρων (Resources) Συστήματος.....	9
2.2. Τεκμηρίωση REST διεπαφής	10
2.2.1. Πόρος User	10
2.2.2. Πόρος Workout	11
2.2.3. Πόρος Exercise.....	16
2.2.4. Πόρος UserWorkout.....	18
2.2.5. Πόρος NotificationEvent.....	19
3. Υλοποίηση Συστήματος με Node-RED	20
3.1. Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed	20
3.1.1. Ροές πόρου User	20
3.1.2. Ροές πόρου Workout	21
3.1.3. Ροές πόρου Exercise.....	24
3.1.4. Ροές πόρου UserWorkout.....	25
3.1.5. Ροές πόρου NotificationEvent.....	25
3.2. Υλοποίηση Ιστοριών χρήστη	26
3.2.1. Ιστορία Χρήστη Add personal detailsz	26
3.2.2. Ιστορία Χρήστη Edit user information.....	26
3.2.3. Ιστορία Χρήστη Choose a workout plan.....	27
3.2.4. Ιστορία Χρήστη Edit the workout plan	27
3.2.5. Ιστορία Χρήστη Create a new workout plan.....	27
3.2.6. Ιστορία Χρήστη Filter workout plans	28
3.2.7. Ιστορία Χρήστη Add to favorites	28
3.2.8. Ιστορία Χρήστη Show execution of exercise.....	29
3.2.9. Ιστορία Χρήστη Rate the workout plan	29
3.2.10. Ιστορία Χρήστη The System notifies user to exercise	30
Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας	31
Παράρτημα ΙΙ – Ανοιχτά Θέματα	32



Άνοιξη 2023

Λίστα Σχημάτων

Σχήμα 1: Εφαρμογή προτύπου Proxy

Σχήμα 2: Εφαρμογή προτύπου Decorator

Σχήμα 3: Εφαρμογή προτύπου Abstract Factory

Σχήμα 4.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 4.2: Αποκρίσεις του συστήματος

Σχήμα 4.3: Δοκιμή του endpoint.

Σχήμα 5.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 5.2: Αποκρίσεις του συστήματος.

Σχήμα 5.3: Δοκιμή του endpoint.

Σχήμα 6.1: Εξήγηση και παράθεση των παραμέτρων του πόρου

Σχήμα 6.2: Αποκρίσεις του συστήματος.

Σχήμα 6.3: Δοκιμή του endpoint.

Σχήμα 7.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 7.2: Αποκρίσεις του συστήματος.

Σχήμα 7.3: Δοκιμή του endpoint.

Σχήμα 8.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 8.2: Αποκρίσεις του συστήματος.

Σχήμα 8.3: Δοκιμή του endpoint.

Σχήμα 9.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 9.2: Αποκρίσεις του συστήματος.

Σχήμα 9.3: Δοκιμή του endpoint.

Σχήμα 10.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 10.2: Αποκρίσεις του συστήματος.

Σχήμα 10.3: Δοκιμή του endpoint.

Σχήμα 11.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 11.2: Αποκρίσεις του συστήματος.

Σχήμα 11.3: Δοκιμή του endpoint.

Σχήμα 12.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 12.2: Αποκρίσεις του συστήματος.

Σχήμα 12.3: Δοκιμή του endpoint.

Σχήμα 13.1: Εξήγηση και παράθεση των παραμέτρων του πόρου

Σχήμα 13.2: Αποκρίσεις του συστήματος.

Σχήμα 13.3: Δοκιμή του endpoint.

Σχήμα 14.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 14.2: Αποκρίσεις του συστήματος.

Σχήμα 14.3: Δοκιμή του endpoint.



1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν

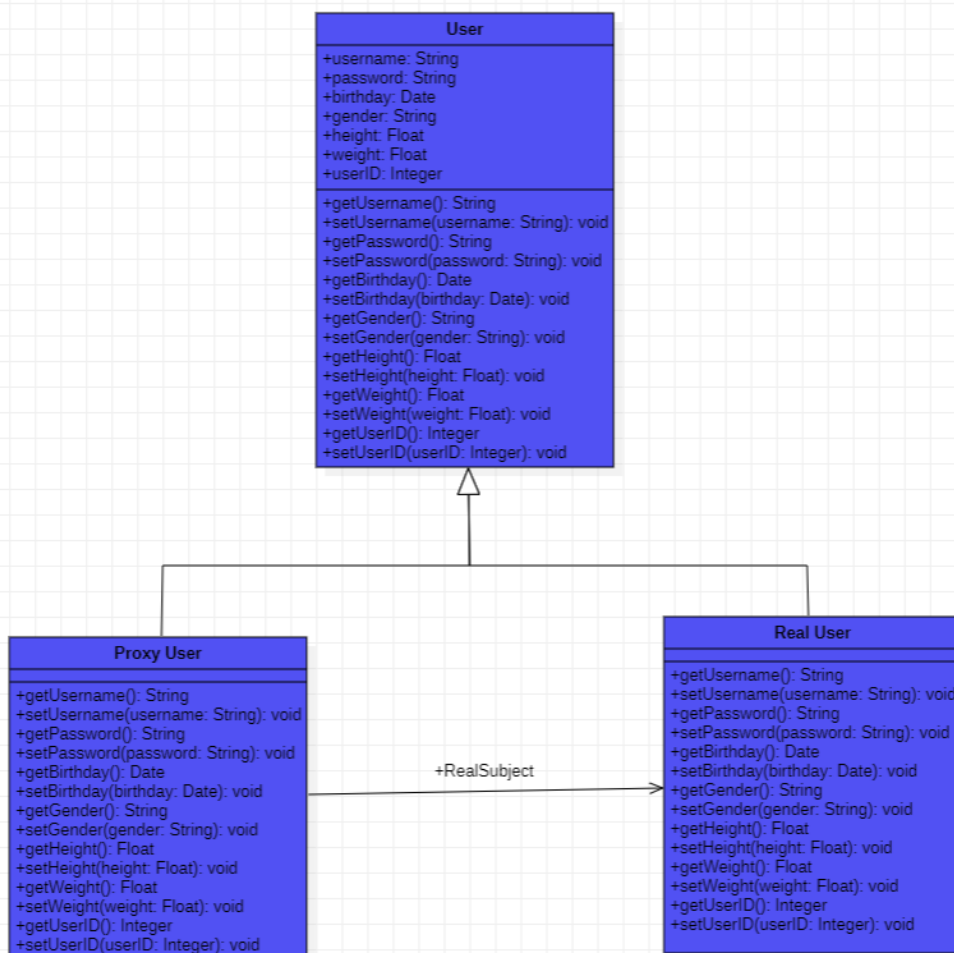
1.1 Πρότυπο Proxy

Το πρότυπο σχεδίασης Proxy είναι ένα δομικό πρότυπο σχεδίασης. Το Proxy ελέγχει την πρόσβαση στο αρχικό αντικείμενο, επιτρέποντας να εκτελεστεί κάτι πριν ή μετά το αίτημα να περαστεί στο αρχικό αντικείμενο. Χρησιμοποιούμε αυτό το πρότυπο σχεδίασης για τις μη λειτουργικές απαιτήσεις 1, 3, 5.

Μπορούμε να εφαρμόσουμε ένα protection proxy για να ικανοποιήσουμε την ΜΛΑ-1 που αφορά την προστασία των προσωπικών δεδομένων των χρηστών σύμφωνα με τον GDPR. Αυτό το πρότυπο θα παρέχει ένα πρόσθετο επίπεδο προστασίας στην δική μας εφαρμογή επαληθεύοντας εάν ένας πραγματικός χρήστης έχει ή όχι πρόσβαση στο κατάλληλο περιεχόμενο.

Μπορούμε να εφαρμόσουμε ένα smart reference proxy για την ικανοποίηση της ΜΛΑ-3. Μας βοηθά να εξοικονομήσουμε μνήμη αποφεύγοντας την αντιγραφή αντικειμένων που καταναλώνουν περισσότερη μνήμη. Αυτό το πρότυπο σχεδίασης χρησιμοποιείται για τη δημιουργία μιας wrapper κλάσης που περικλείει μια αναφορά σε ένα αντικείμενο. Η κλάση wrapper περιέχει επίσης λογική για την παρακολούθηση πόσες αναφορές υπάρχουν για το αντικείμενο. Διαγράφει αυτόματα το αντικείμενο όταν αφαιρεθεί η τελευταία αναφορά. Με αυτόν τον τρόπο το πρότυπο σχεδίασης Proxy εξοικονομεί μνήμη από περιττές εκχωρήσεις (ΜΛΑ-3) και επιταχύνει την εφαρμογή (ΜΛΑ-5).

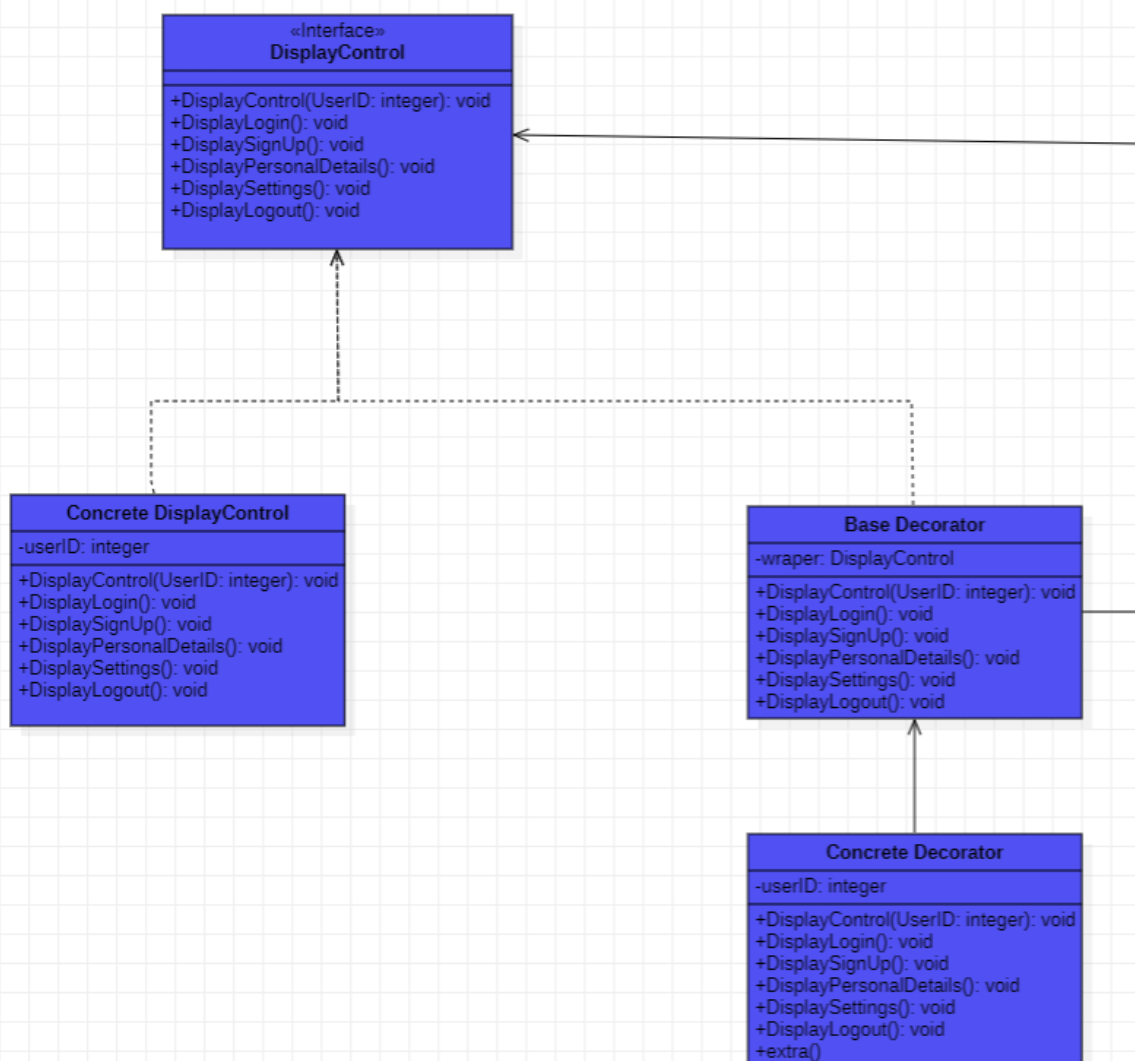
Μπορούμε να εφαρμόσουμε έναν caching proxy για να ικανοποιήσουμε την ΜΛΑ-5. Είναι μια τεχνική internet caching που επιτρέπει στον Proxy Server να αποθηκεύει πρόσφατα και συχνά αιτήματα ή δεδομένα τοπικά για γρήγορη πρόσβαση.



Σχήμα 1: Εφαρμογή προτύπου Proxy

1.2 Πρότυπο Decorator

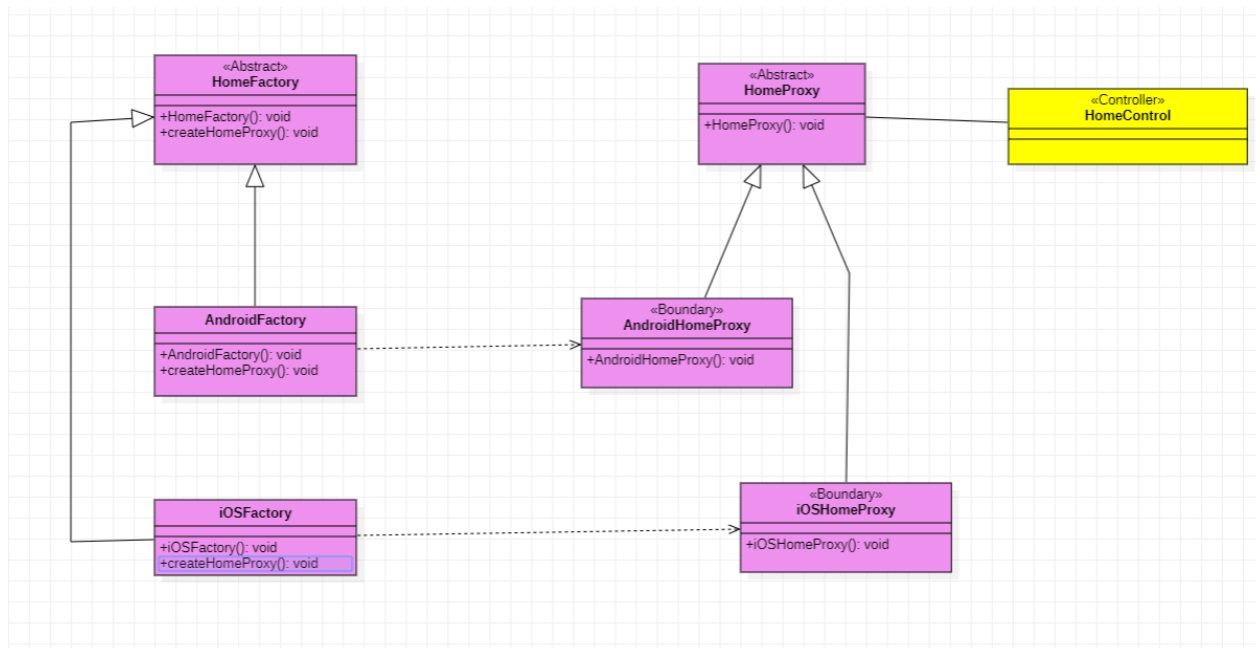
Το πρότυπο σχεδίασης Decorator είναι ένα δομικό πρότυπο σχεδίασης. Μπορούμε να το εφαρμόσουμε για να ικανοποιήσουμε την ΜΛΑ-4 που είναι για την ικανότητα να εμπλουτίζεται εύκολα η εφαρμογή με νέα χαρακτηριστικά. Το πρότυπο σχεδίασης Decorator επιτρέπει να ενισχύονται δυναμικά οι υπάρχουσες κλάσεις με νέες λειτουργικότητες και συμπεριφορά χωρίς τροποποιούμε τον αρχικό κώδικα. Αυτό γίνεται δημιουργώντας μια νέα κλάση (the decorator) που τυλίγει την αρχική κλάση και προσθέτει την επιθυμητή συμπεριφορά.



Σχήμα 2: Εφαρμογή προτύπου Decorator

1.3 Πρότυπο Abstract Factory

Το πρότυπο σχεδίασης Abstract Factory είναι ένα δημιουργικό πρότυπο σχεδίασης. Μπορούμε να το εφαρμόσουμε για να ικανοποιήσουμε την ΜΛΑ-2 που είναι η εφαρμογή μας να μπορεί να υποστηρίζεται σε διαφορετικές συσκευές (android και iOS) και να μπορεί το σύστημα να επικοινωνεί μαζί τους.



Σχήμα 3: Εφαρμογή προτύπου Abstract Factory



2. Αρχιτεκτονική Συστήματος

- 1) [Σύνδεσμος](#) με το json αρχείο.
- 2) [Σύνδεσμος](#) με το zip αρχείο.
- 3) [Σύνδεσμος](#) σε μορφή yaml.

2.1. Αναγνώριση Πόρων (Resources) Συστήματος

Κλάση BEC	Πόρος REST	Endpoints (HTTP Verbs)
User	/user	POST
User	/user/{userID}	PUT
Workout	/user/{userID}/workout	POST
Workout	/user/{userID}/workout/{workoutID}	GET, PUT, DELETE
Workout	/user/{userID}/workout/{workoutID}/favorites	POST
Exercise	/user/{userID}/exercise	GET
Exercise	/user/{userID}/exercise/{exerciseID}	GET
UserWorkout	/user/{userID}/userWorkout/{workoutID}	PUT
NotificationEvent	/user/{userID}/notificationEvent	POST



2.2. Τεκμηρίωση REST διεπαφής

2.2.1. Πόρος User

2.2.1.1. Μοντέλο δεδομένων User

```
User {  
  username      string  
  password      string  
  birthday      string($date)  
  gender        string  
  height        number($float)  
  weight        number($float)  
  userID        integer  
}
```

2.2.1.2. Endpoint POST πόρου User

POST /user 1-Create a new user Account

FR-1 The user must be able to add his personal details.

Parameters

No parameters

Request body **required**

```
{  
  "username": "Παναγιώτης",  
  "password": "12345",  
  "birthday": "2013-06-12",  
  "gender": "Αντρας",  
  "height": 180,  
  "weight": 80,  
  "userID": 10  
}
```

Σχήμα 4.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Responses	
Code	Description
200	Successful operation
Media type <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "username": "string", "password": "string", "birthday": "2023-06-17", "gender": "string", "height": 0, "weight": 0, "userID": 0 }</pre></div>	
400	Bad input parameter
404	User not found

Σχήμα 4.2: Αποκρίσεις του συστήματος.

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

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{  
  "username": "Παναγιώτης",  
  "password": "12345",  
  "birthday": "2013-06-12",  
  "gender": "Αντρας",  
  "height": 180,  
  "weight": 80,  
  "userID": 10  
}'
```

Request URL

https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user

Σχήμα 4.3: Δοκιμή του endpoint.

2.2.1.3. Endpoint PUT πόρου User, για χρήστη με συγκεκριμένο user ID



PUT /user/{userID} 2-Edit personal details

FR-2 The user must be able to edit his personal details.

Parameters

Name	Description
userID * required	The ID of the user that updates his personal details
integer (path)	
	<input type="text" value="11"/>

Request body required

```
{  "username": "Παναγιώτης",  "password": "12345",  "birthday": "2013-06-12",  "gender": "Άντρας",  "height": 180,  "weight": 80,  "userID": 11}
```

Σχήμα 5.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Responses

Code	Description
200	Successful operation
400	Bad input parameter

Media type

Controls Accept header.

Example Value | Schema

```
{  "username": "string",  "password": "string",  "birthday": "2023-06-17",  "gender": "string",  "height": 0,  "weight": 0,  "userID": 0}
```

Σχήμα 5.2: Αποκρίσεις του συστήματος.

Snippets

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

```
curl -X 'PUT' \  'https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11' \  -H 'accept: application/json' \  -H 'Content-Type: application/json' \  -d '{  "username": "Παναγιώτης",  "password": "12345",  "birthday": "2013-06-12",  "gender": "Άντρας",  "height": 180,  "weight": 80,  "userID": 11}'
```

Request URL

https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11

Σχήμα 5.3: Δοκιμή του endpoint.

2.2.2. Πόρος Workout

2.2.2.1. Μοντέλο δεδομένων Workout

```
Workout {  workoutID      integer  name          string  duration       integer  category      string}
```

2.2.2.2.Endpoint POST πόρου Workout



POST /user/{userID}/workout 5-Define workout

FR-5 The user must be able to define a workout plan.

Parameters

Name	Description
userID * required	The ID of the user that defines a new workout
integer (path)	
	<input type="text" value="11"/>

Request body required

```
{
  "workoutID": 10,
  "name": "ABS2",
  "duration": 10,
  "category": "ABS"
}
```

Σχήμα 6.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Code	Description
200	Successfull operation
Media type	
<input type="text" value="application/json"/>	
Controls Accept header.	
Example Value Schema	
<pre>{ "workoutID": 0, "name": "string", "duration": 0, "category": "string" }</pre>	
400	Bad input parameter
404	User not found

Σχήμα 6.2: Αποκρίσεις του συστήματος.

Snippets

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

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/workout' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "workoutID": 10,
    "name": "ABS2",
    "duration": 10,
    "category": "ABS"
  }'
```

Request URL

https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/workout

Σχήμα 6.3: Δοκιμή του endpoint.

2.2.2.3. Endpoint GET πόρου Workout για συγκεκριμένο Workout ID



GET /user/{userID}/workout/{workoutID} 3-Choose workout

FR-3 The user must be able to choose a workout plan.

Parameters

Name	Description
userID * required integer (path)	The ID of the user that chooses a workout plan
workoutID * required integer (path)	The ID of the chosen workout

Σχήμα 7.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Code	Description
200	Successful operation
Media type <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "workoutID": 0, "name": "string", "duration": 0, "category": "string" }</pre></div>	
400	Bad input parameter
404	Workout plan not found

Σχήμα 7.2: Αποκρίσεις του συστήματος.

Snippets

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

```
curl -X 'GET' \
'https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/workout/11' \
-H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/workout/11
```

Σχήμα 7.3: Δοκιμή του endpoint.



2.2.2.4. Endpoint PUT πόρου Workout για συγκεκριμένο Workout ID

PUT /user/{userID}/workout/{workoutID} 4-Edit workout

FR-4 The user must be able to edit a workout plan.

Parameters

Name	Description
userID * required	The ID of the user that edits the workout
integer (path)	<input type="text" value="11"/>
workoutID * required	The ID of the workout to be edited
integer (path)	<input type="text" value="11"/>

Responses

Code	Description
200	Successfull operation
Media type <input type="text" value="application/json"/> Controls Accept header.	
Example Value Schema	
<pre>{ "workoutID": 0, "name": "string", "duration": 0, "category": "string"}</pre>	
400	Bad input parameter
404	Workout plan not found

Σχήμα 8.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 8.2: Αποκρίσεις του συστήματος.

Snippets

cURL (bash)

cURL (PowerShell)

cURL (CMD)

```
curl -X 'PUT' \
  'https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/workout/11' \
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/workout/11
```

Σχήμα 8.3: Δοκιμή του endpoint.



2.2.2.5. Endpoint DELETE πόρου Workout για συγκεκριμένο Workout ID

DELETE /user/{userID}/workout/{workoutID} 4-Edit(delete) workout

FR-4 The user must be able to edit a workout plan.

Parameters

Name	Description
userID * required integer (path)	The ID of the user that edits(deletes) the workout
workoutID * required integer (path)	The ID of the workout to be edited(deleted)

Responses

Code	Description
200	Succesfull operation
400	Bad input parameter
404	Workout plan not found

Σχήμα 9.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 9.2: Αποκρίσεις του συστήματος.

cURL (bash)

cURL (PowerShell)

cURL (CMD)

```
curl -X 'DELETE' \
'https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/workout/11' \
-H 'accept: */*'
```

Request URL

```
https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/workout/11
```

Σχήμα 9.3: Δοκιμή του endpoint.

2.2.2.6. Endpoint POST πόρου Favorite

POST /user/{userID}/workout/{workoutID} /favorites 7-add workout to favorites

FR-7 The user must be able to add a workout plan on favorites

Parameters

Name	Description
userID * required integer (path)	The ID of the user that adds a workout on favorites
workoutID * required integer (path)	The ID of the workout that the user adds on favorites

Code

Description

200	Succesfull operation
400	Bad input parameter
404	Workout not found

Media type
application/json

Controls Accept header.

Example Value | Schema

```
{
  "workoutID": 0,
  "name": "string",
  "duration": 0,
  "category": "string"
}
```

Σχήμα 10.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 10.2: Αποκρίσεις του συστήματος.



Σχήμα 10.3: Δοκιμή του endpoint.

2.2.3. Πόρος Exercise

2.2.3.1. Μοντέλο δεδομένων Exercise

```
Exercise ▼ {  
  exerciseID      integer  
  name            string  
  description      string  
  imageURL        string  
  videoURL        string  
  nrRepetitions   integer  
  category        string  
}
```

2.2.3.2. Endpoint GET πόρου Exercise

GET /user/{userID}/exercise 6-filter exercises

FR-6 The user must be able to filter the exercises.

Parameters

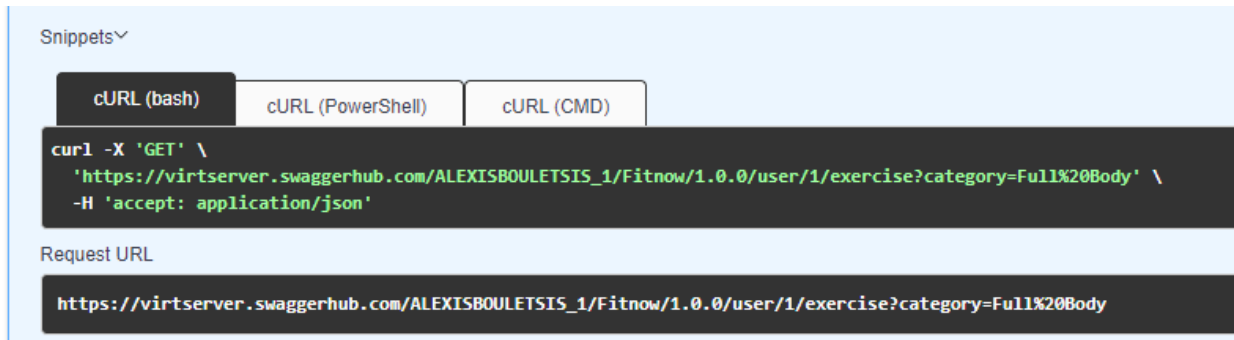
Name	Description
userID * required integer (path)	The ID of the user that filters the exercises
category * required array[string] (query)	Show the exercises that belong to this/these category/categories

Input fields:
userID: 1
category: Full Body (selected from dropdown: Full Body, Abs, Arms, Legs)

Σχήμα 11.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

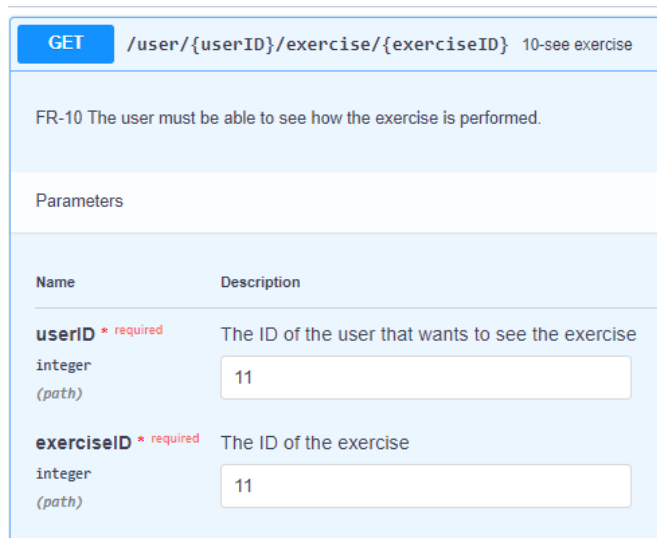
Code	Description
200	Successfull operation
Media type application/json	
Controls Accept header.	
Example Value Schema	
<pre>{ "exerciseID": 0, "name": "string", "description": "string", "imageURL": "string", "videoURL": "string", "nrRepetitions": 0, "category": "string" }</pre>	
400	Bad input parameter
404	User not found

Σχήμα 11.2: Αποκρίσεις του συστήματος.

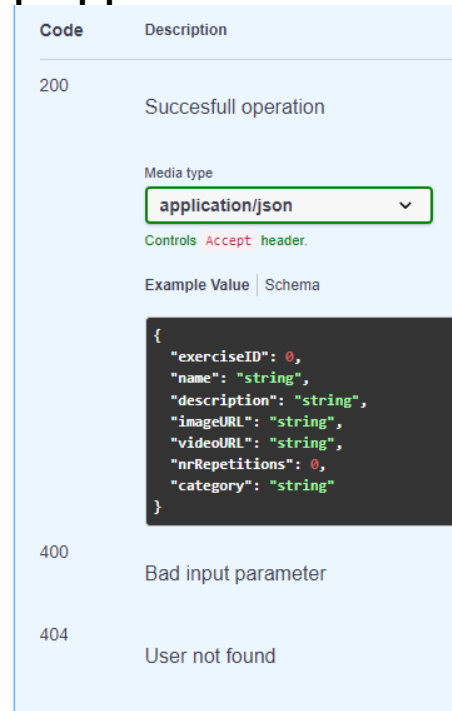


Σχήμα 11.3: Δοκιμή του endpoint.

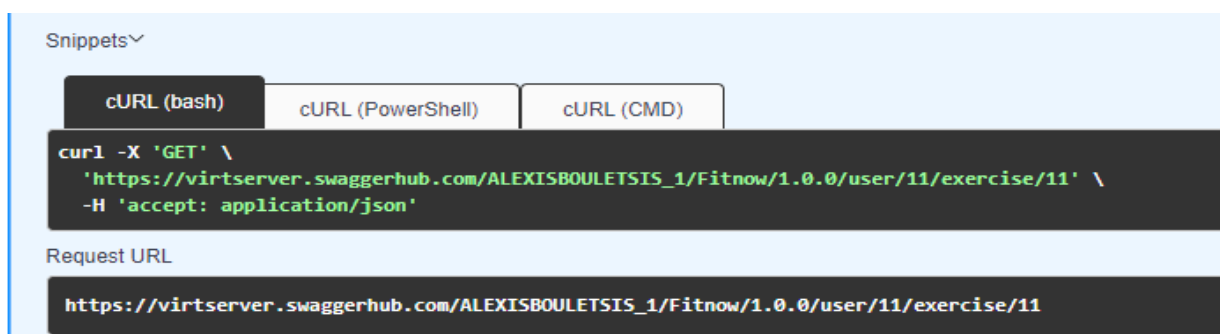
2.2.3.3. Endpoint GET πόρου Exercise για συγκεκριμένο Exercise ID



Σχήμα 12.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.



Σχήμα 12.2: Αποκρίσεις του συστήματος.



Σχήμα 12.3: Δοκιμή του endpoint.



2.2.4. Πόρος UserWorkout

2.2.4.1. Μοντέλο δεδομένων UserWorkout

```
UserWorkout {  
  userWorkoutID integer  
  userID integer  
  workoutID integer  
  statsFinish boolean  
  rating integer  
    minimum: 1  
    maximum: 5  
  spentTime integer  
}
```

Endpoint PUT πόρου UserWorkout για συγκεκριμένο workout ID

PUT /user/{userID}/userWorkout/{workoutID} 8-rate a workout

FR-8 The user must be able to rate a workout plan.

Parameters

Name	Description
userID * required integer (path)	The ID of the user who rates a workout
workoutID * required integer (path)	The ID of the workout that user rated

Request body required

Rate the workout from 1 to 5

```
{  
  "rating": 1  
}
```

Code	Description
200	Successful operation
Media type <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "workoutID": 0, "name": "string", "duration": 0, "category": "string" }</pre></div>	
400	Bad input parameter
404	User not found

Σχήμα 13.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 13.2: Αποκρίσεις του συστήματος.

Snippets

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

```
curl -X 'PUT' \
  'https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/userWorkout/11' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{  
  "rating": 1  
}'
```

Request URL

```
https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/userWorkout/11
```

Σχήμα 13.3: Δοκιμή του endpoint.



2.2.5. Πόρος NotificationEvent

2.2.5.1. Μοντέλο δεδομένων NotificationEvent

```
NotificationEvent {  
  notificationTime string($datetime)  
  notificationDate string($date)  
  userID integer  
  notificationEventID integer  
}
```

2.2.5.2. Endpoint POST πόρου NotificationEvent

POST

/user/{userID}/notificationEvent

9-notify user

FR-9 The system must notify the user when to exercise.

Parameters

Name	Description
userID * required	The ID of the user to be notified
integer	
(path)	

Code	Description
200	Successfull operation
400	Bad input parameter
404	User not found

Media type
application/json

Controls Accept header.

Example Value | Schema

```
{  
  "notificationTime": "string",  
  "notificationDate": "2023-06-17",  
  "userID": 0,  
  "notificationEventID": 0  
}
```

Σχήμα 14.1: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Σχήμα 14.2: Αποκρίσεις του συστήματος.

cURL (bash)

cURL (PowerShell)

cURL (CMD)

```
curl -X 'POST' \  
  'https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/notificationEvent' \  
  -H 'accept: application/json' \  
  -d ''
```

Request URL

```
https://virtserver.swaggerhub.com/ALEXISBOULETSIS_1/Fitnow/1.0.0/user/11/notificationEvent
```

Σχήμα 14.3: Δοκιμή του endpoint.



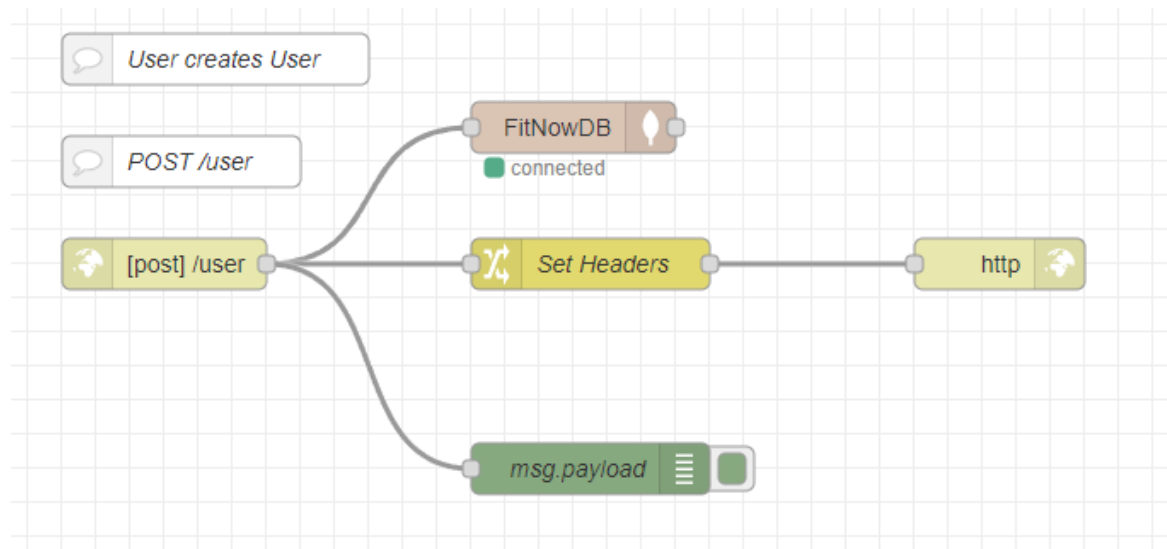
3. Υλοποίηση Συστήματος με Node-RED

3.1. Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed

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

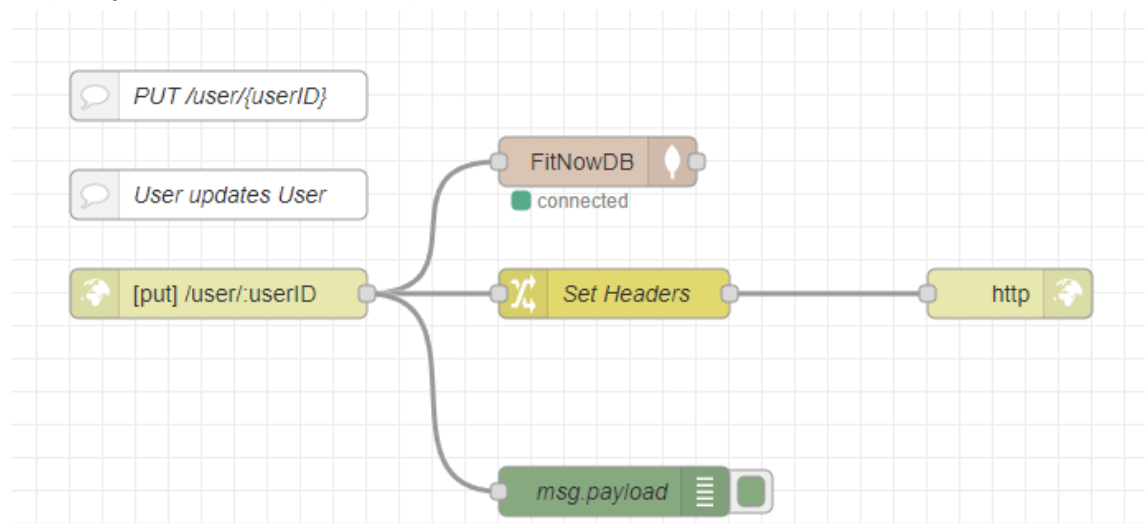
3.1.1. Ροές πόρου User

Ροή endpoint POST /user



Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας μπορεί ένας χρήστης να δημιουργήσει ένα νέο προφίλ και να καταχωρήσει τα προσωπικά του στοιχεία.

Ροή endpoint PUT /user/{userID}

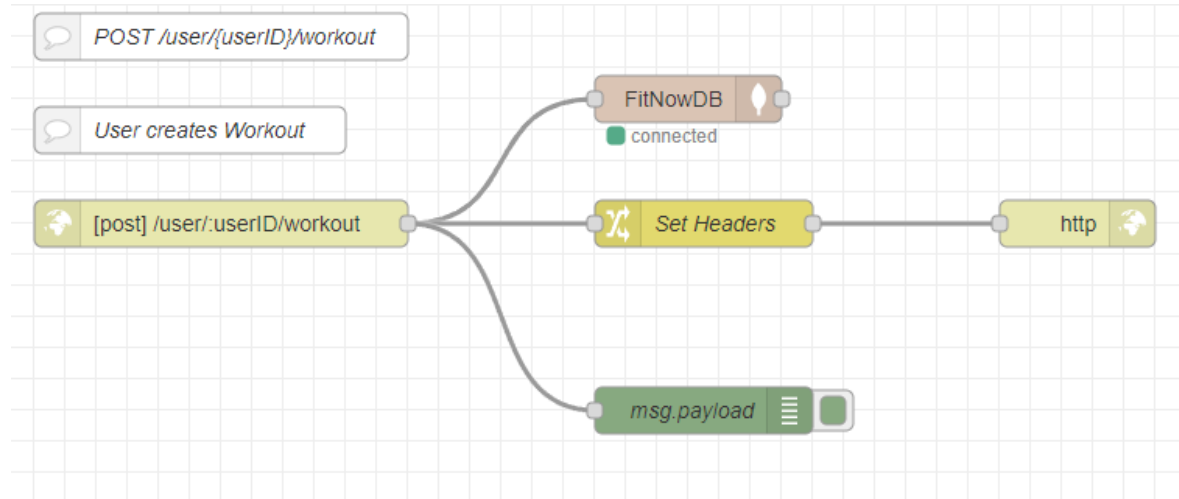


Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας ένας χρήστης (με βάση το userID) μπορεί να ενημερώσει τα προσωπικά του στοιχεία στην βάση δεδομένων.



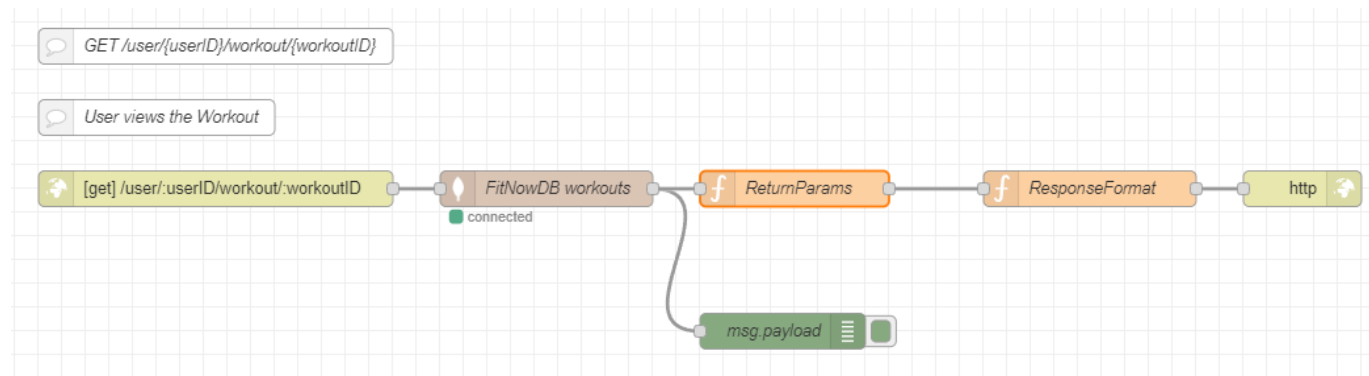
3.1.2. Ροές πόρου Workout

Ροή endpoint POST /user/{userID}/workout



Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας μπορεί ένας χρήστης να δημιουργήσει ένα workout και να το αποθηκεύσει στην βάση δεδομένων.

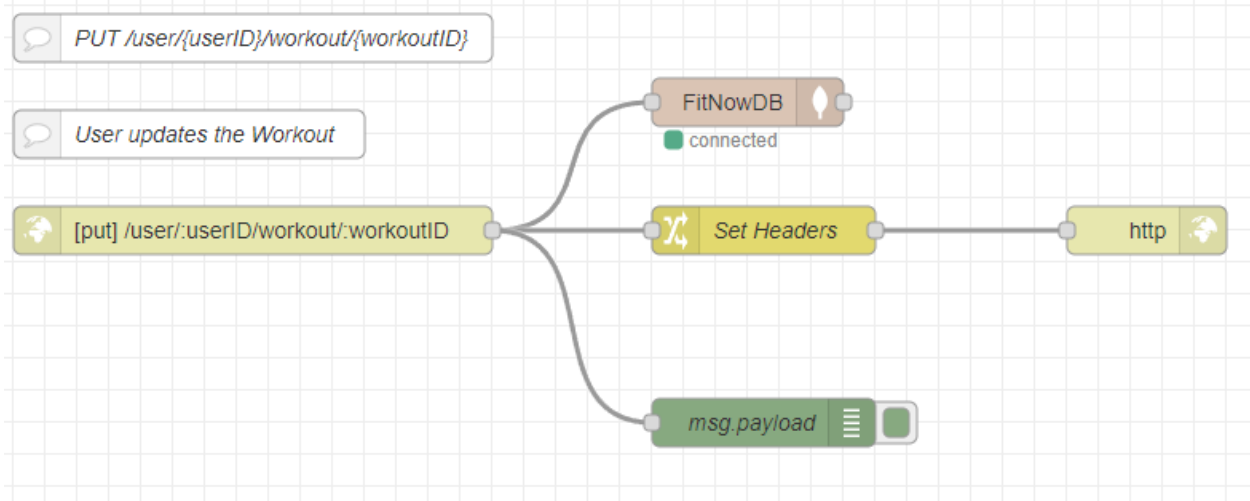
Ροή endpoint GET /user/{userID}/workout/{workoutID}



Η ροή που υλοποιεί την υπηρεσία, που είναι υπεύθυνη για την επιστροφή ενός συγκεκριμένου workout σε έναν χρήστη. Μέσω της συνάρτησης ReturnParams, επιστρέφεται το workout με συγκεκριμένο workoutID, όπως ορίζεται στο URL parameter και μέσω της συνάρτησης CreateResponseFormat, επιστρέφεται το workout στο format που απαιτείται.

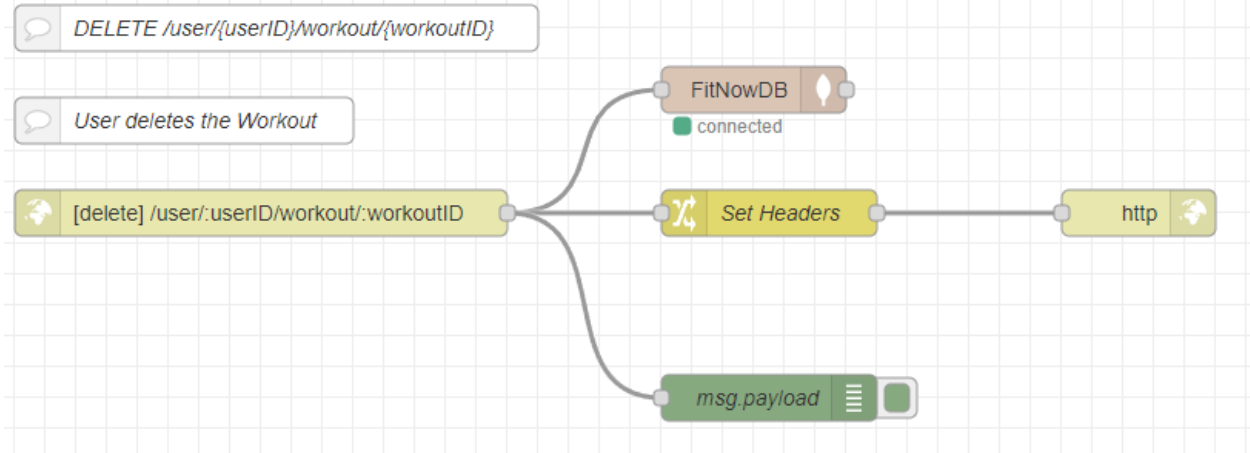


Ροή endpoint PUT /user/{userID}/workout/{workoutID}



Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας ένας χρήστης (με βάση το userID) μπορεί να τροποποιήσει το workout που επιθυμεί (με βάση το workoutID), στην βάση δεδομένων.

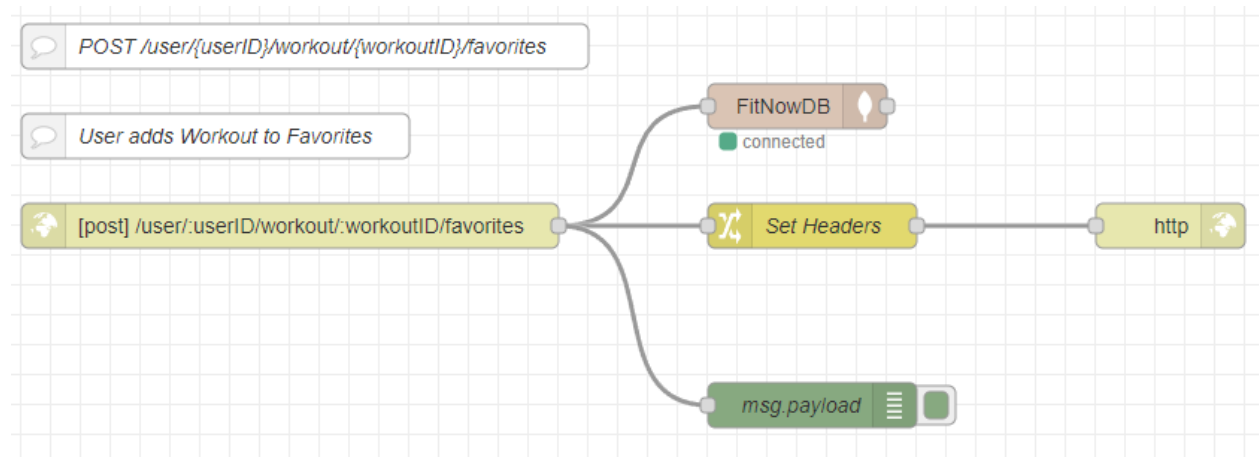
Ροή endpoint DELETE /user/{userID}/workout/{workoutID}



Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας ένας χρήστης (με βάση το userID) μπορεί να διαγράψει το workout που επιθυμεί (με βάση το workoutID), στην βάση δεδομένων.



Ροή endpoint POST /user/{userID}/workout/{workoutID}/favorites

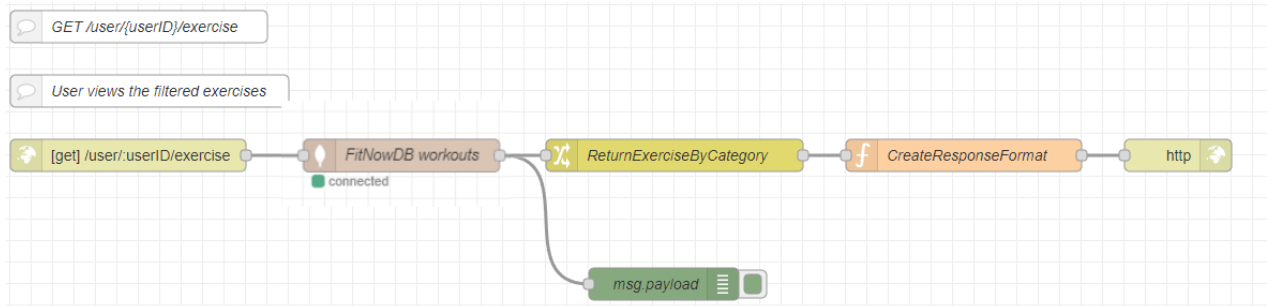


Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας μπορεί ένας χρήστης να προσθέσει ένα workout στα favorites.



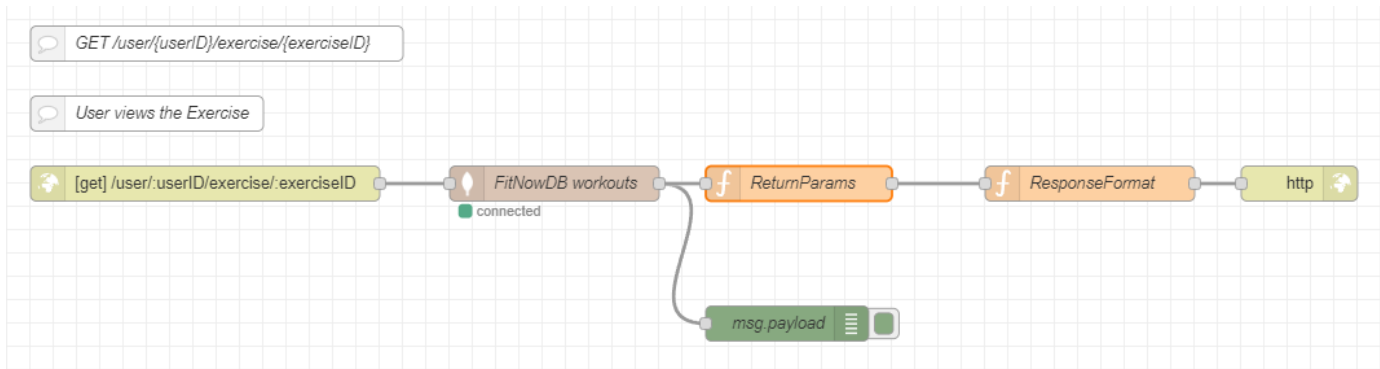
3.1.3. Ροές πόρου Exercise

Ροή endpoint GET /user/{userID}/exercise



Η ροή που υλοποιεί την υπηρεσία, που είναι υπεύθυνη για την επιστροφή των φιλτραρισμένων exercises σε έναν χρήστη. Μέσω της συνάρτησης ReturnExerciseByCategory, επιστρέφεται η άσκηση με βάση την κατηγορία της και με την συνάρτηση CreateResponseFormat, επιστρέφεται η άσκηση στο format που απαιτείται.

Ροή endpoint GET /user/{userID}/exercise/{exerciseID}

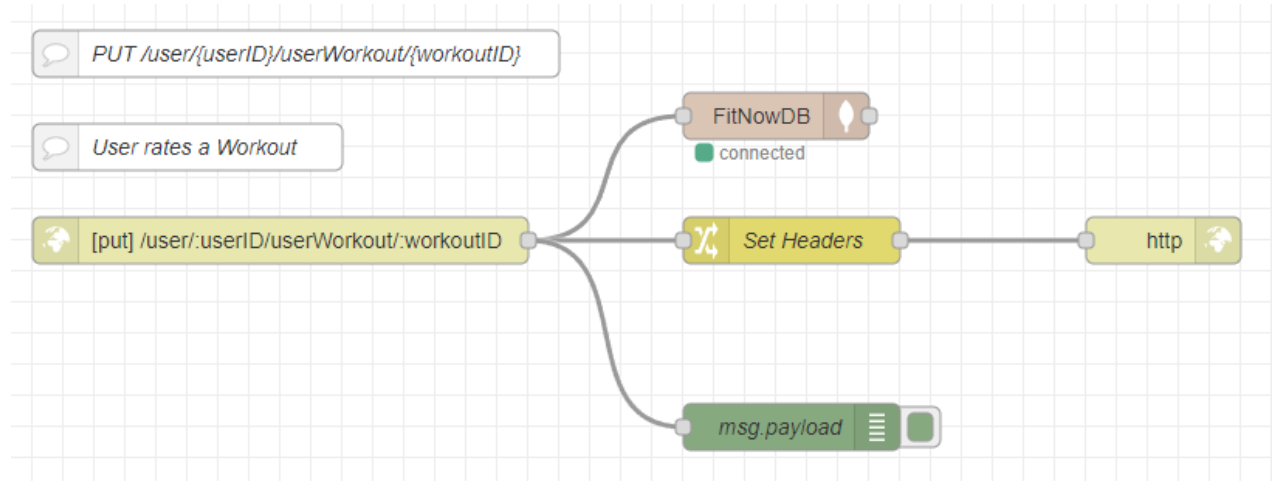


Η ροή που υλοποιεί την υπηρεσία, που είναι υπεύθυνη για την επιστροφή ενός συγκεκριμένου exercise σε έναν χρήστη. Μέσω της συνάρτησης ReturnParams, επιστρέφεται η άσκηση με συγκεκριμένο exerciseID, όπως ορίζεται στο URL parameter και μέσω της συνάρτησης CreateResponseFormat, επιστρέφεται η άσκηση στο format που απαιτείται.



3.1.4. Ροές πόρου UserWorkout

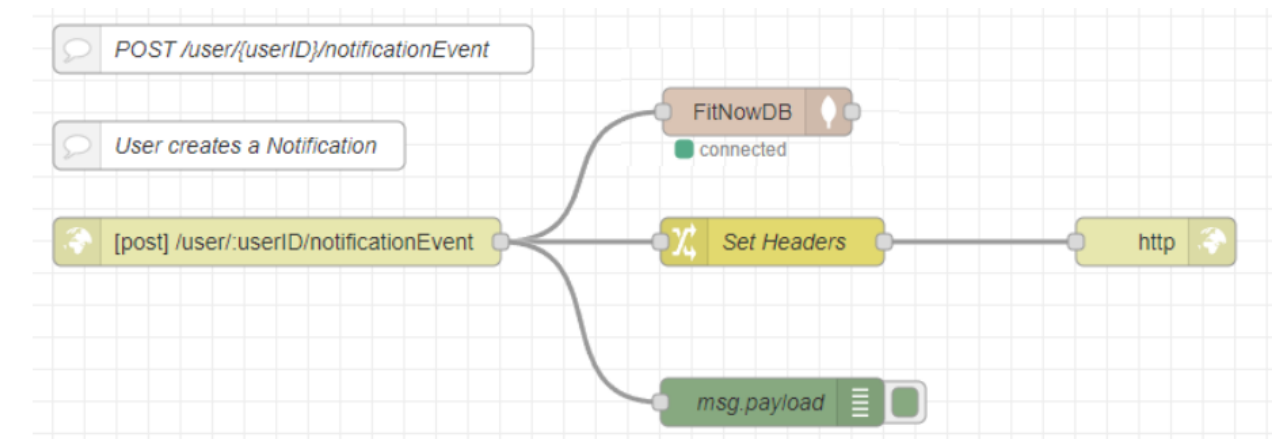
Ροή endpoint PUT /user/{userID}/userWorkout/{workoutID}



Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας ένας χρήστης (με βάση το userID) μπορεί να κάνει rate το workout που επιθυμεί (με βάση το workoutID), στην βάση δεδομένων.

3.1.5. Ροές πόρου NotificationEvent

Ροή endpoint POST /user/{userID}/notificationEvent

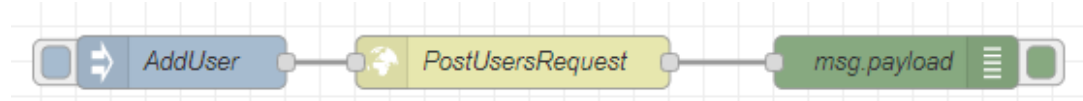


Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας το σύστημα μπορεί να στείλει notification σε έναν χρήστη.



3.2. Υλοποίηση Ιστοριών χρήστη

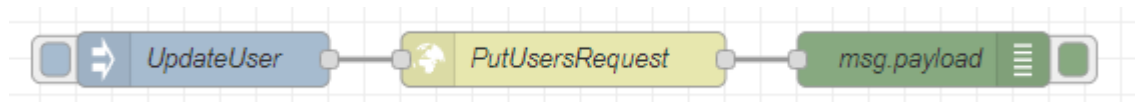
3.2.1. Ιστορία Χρήστη Add personal detailsz



Ροή μέσω της οποίας ο χρήστης, μπορεί να δημιουργήσει έναν λογαριασμό και τα προσθέσει τα προσωπικά του στοιχεία στο σύστημα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
AddUser	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
PostUsersRequest	http-request	Κάνει κλήση προς την υπηρεσία /user η οποία δημιουργεί την λίστα με τα στοιχεία του χρήστη στο σύστημα.
Msg.payload	debug	Τυπώνει στην κονσόλα την λίστα με τα στοιχεία του χρήστη.

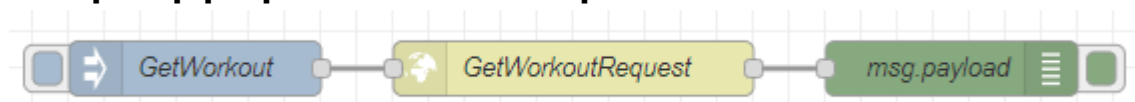
3.2.2. Ιστορία Χρήστη Edit user information



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

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
UpdateUser	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
PutUsersRequest	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID} η οποία τροποποιεί την λίστα με τα στοιχεία του χρήστη στο σύστημα.
Msg.payload	debug	Τυπώνει στην κονσόλα την λίστα με τα στοιχεία του χρήστη.

3.2.3. Ιστορία Χρήστη Choose a workout plan

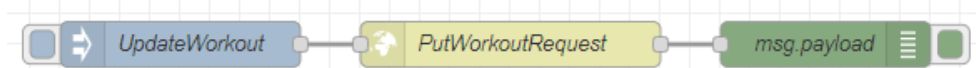


Ροή μέσω της οποίας ο χρήστης, μπορεί να επιλέγει το workout plan του και να βλέπει πληροφορίες για το workout.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
GetWorkout	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
GetWorkoutRequest	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/workout/{workoutID} η οποία επιστρέφει την λίστα με τα στοιχεία του workout του συστήματος.
Msg.payload	debug	Τυπώνει στην κονσόλα την λίστα με τα στοιχεία του workout που με το ID να είναι το workoutID.

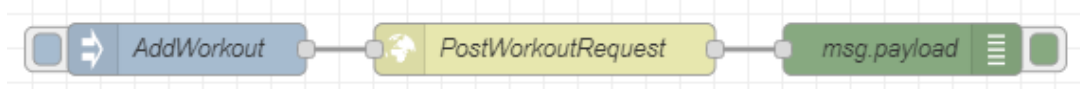
3.2.4. Ιστορία Χρήστη Edit the workout plan



Ροή μέσω της οποίας ο χρήστης, μπορεί να τροποποιήσει το workout plan του.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
UpdateWorkout	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
PutWorkoutRequest	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/workout/{workoutID} η οποία τροποποιεί την λίστα με τα στοιχεία του workout του συστήματος.
Msg.payload	debug	Τυπώνει στην κονσόλα την λίστα με τα στοιχεία του workout που με το ID να είναι το workoutID.

3.2.5. Ιστορία Χρήστη Create a new workout plan



Ροή μέσω της οποίας ο χρήστης, μπορεί να δημιουργεί το workout plan του.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
AddWorkout	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
PostWorkoutRequest	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/workout/{workoutID} η οποία δημιουργεί την λίστα με τα στοιχεία του workout του συστήματος.
Msg.payload	debug	Τυπώνει στην κονσόλα την λίστα με τα στοιχεία του workout που με το ID να είναι το workoutID.



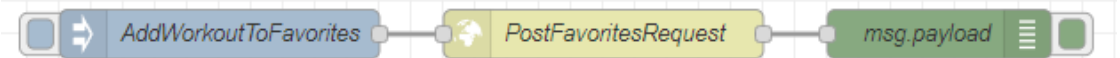
3.2.6. Ιστορία Χρήστη Filter workout plans



Ροή μέσω της οποίας ο χρήστης, μπορεί να φιλτράρει τις ασκήσεις των workout plan του.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
GetFilteredExercises	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
GetFilteredExercises Request	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/exercise η οποία επιστρέφει την λίστα με τις φιλτραρισμένες ασκήσεις του συστήματος.
Msg.payload	debug	Τυπώνει στην κονσόλα την λίστα με τα στοιχεία των φιλτραρισμένων ασκήσεων.

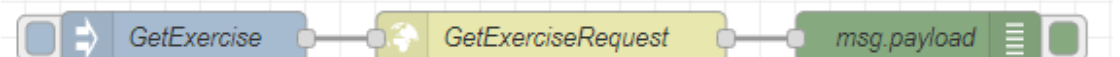
3.2.7. Ιστορία Χρήστη Add to favorites



Ροή μέσω της οποίας ο χρήστης, μπορεί να φιλτράρει τις ασκήσεις των workout plan του.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
AddWorkoutToFavorites	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
PostFavoritesRequest	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/workout/{workoutID}/favorites η οποία δημιουργεί μία λίστα στα αγαπημένα workout του χρήστη.
Msg.payload	debug	Τυπώνει στην κονσόλα την λίστα με τα αγαπημένα workout του χρήστη.

3.2.8. Ιστορία Χρήστη Show execution of exercise

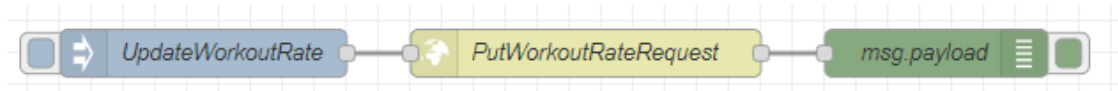


Ροή μέσω της οποίας ο χρήστης, μπορεί να δει την εκτέλεση της άσκησης.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
GetExercise	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
GetExerciseRequest	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/exercise/{exerciseID} η οποία επιστρέφει μία λίστα με την άσκηση με ID το exerciseID.
Msg.payload	debug	Τυπώνει στην κονσόλα την λίστα με τα αγαπημένα workout του χρήστη.

3.2.9. Ιστορία Χρήστη Rate the workout plan

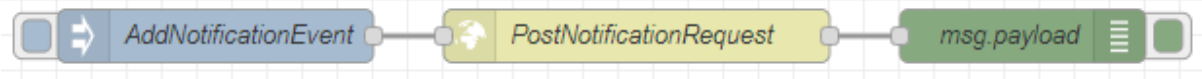


Ροή μέσω της οποίας ο χρήστης, μπορεί να βαθμολογήσει το workout του.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
UpdateWorkoutRate	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
PutWorkoutRateRequest	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/userWorkout/{workoutID} η οποία τροποποιεί μία λίστα για το workout με ID το workoutID.
Msg.payload	debug	Τυπώνει στην κονσόλα την λίστα για το workout με ID το workoutID.



3.2.10. Ιστορία Χρήστη The System notifies user to exercise



Ροή μέσω της οποίας το σύστημα, μπορεί να ειδοποιήσει τον χρήστη να ασκηθεί.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
AddNotificationEvent	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
PostNotificationRequest	http-request	Κάνει κλήση προς την υπηρεσία /user/{userID}/notificationEvent η οποία δημιουργεί μία λίστα με την οποία θα ειδοποιήσει τον χρήστη.
Msg.payload	debug	Τυπώνει στην κονσόλα την λίστα για το notification event.



Παράρτημα Ι – Ανοιχτά Θέματα

- Οι πόροι που σχετίζονται με την διαχείριση χρηστών δεν τεκμηριώνονται, αφού δεν αποτελούν μέρος της εργασίας, σε ένα ρεαλιστικό σύστημα όμως είναι απαραίτητοι.
- Απαιτείται επίσης σχεδιασμός των ροών που θα χρησιμοποιούνται για το front-end μέρος του συστήματος.