



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020



Make your place better

Προδιαγραφές Λογισμικού και Υλοποίηση συστήματος

Del.1.2

Version 1.0

Καρατζάς Μιχάλης mikalaki@ece.auth.gr
Μαραντίδης Κυριάκος kyrmarkos@ece.auth.gr
Εμμανουήλ Μιχάλινας michalain@ece.auth.gr
Ιωάννης Χαρισιάδης charisiad@ece.auth.gr

Ημερομηνία: 9/6/2020



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2020

Μέλη της Ομάδας Ανάπτυξης

Όνομα	OA	Email
Καρατζάς Μιχάλης	myDimos	mikalaki@ece.auth.gr
Εμμανουήλ Μιχάλαϊνας	myDimos	michalain@ece.auth.gr
Μαραντίδης Κυριάκος	myDimos	kyrmarkos@ece.auth.gr
Ιωάννης Χαρισιάδης	myDimos	charisiad@ece.auth.gr



Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	3
Λίστα Σχημάτων	4
1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν	6
1.1 Πρότυπο Abstract Factory	6
1.2 Πρότυπο Proxy	7
1.3 Πρότυπο Adapter	8
2. Αρχιτεκτονική Συστήματος	9
2.1 Αναγνώριση Πόρων (Resources) Συστήματος	9
2.2 Τεκμηρίωση REST διεπαφής	9
2.2.1 Πόρος Issue	9
2.2.1.1 Μοντέλο δεδομένων Issue	9
2.2.1.2 Endpoint GET πόρου Issue, για διαχειριστή και συγκεκριμένο Issue id	10
2.2.1.3 Endpoint PUT πόρου Issue, για διαχειριστή και συγκεκριμένο Issue id	11
2.2.1.4 Endpoint DELETE πόρου Issue, για διαχειριστή και συγκεκριμένο Issue id	12
2.2.1.5 Endpoint GET πόρου Issue, για απλό χρήστη και για συγκεκριμένο Issue id	13
2.2.1.6 Endpoint PUT πόρου Issue, για απλό χρήστη και για συγκεκριμένο Issue id	14
2.2.1.7 Endpoint DELETE πόρου Issue, για απλό χρήστη και συγκεκριμένο Issue id	15
2.2.1.8 Endpoint GET πόρου Issue, για διαχειριστή	16
2.2.1.9 Endpoint GET πόρου Issue, για απλό χρήστη	17
2.2.1.10 Endpoint POST πόρου Issue, για απλό χρήστη	18
2.2.2 Πόρος Category	19
2.2.2.1 Μοντέλο δεδομένων Category	19
2.2.2.2 Endpoint POST πόρου Category	19
2.2.2.3 Endpoint GET πόρου Category	20
2.2.2.4 Endpoint GET πόρου Category, για συγκεκριμένο Category id	21
2.2.2.5 Endpoint DELETE πόρου Category, για συγκεκριμένο Category id	22
2.2.2.6 Endpoint PUT πόρου Category, για συγκεκριμένο Category id	23
3. Υλοποίηση Συστήματος με Node-RED	24
3.1 Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed	24
3.1.1 Ροές πόρου Issue	24
3.1.2 Ροές πόρου Category	27
3.2 Υλοποίηση Ιστοριών χρήστη	29
3.2.1 Ιστορία Χρήστη User add issue	29
3.2.2 Ιστορία Χρήστη User views issues	29



3.2.3	Ιστορία Χρήστη User views full content issue.....	29
3.2.4	Ιστορία Χρήστη User updates an issue.....	30
3.2.5	Ιστορία Χρήστη User deletes an issue.....	30
3.2.6	Ιστορία Χρήστη Admin views issues.....	30
3.2.7	Ιστορία Χρήστη Admin views full content issue.....	30
3.2.8	Ιστορία Χρήστη Admin updates an issue	31
3.2.9	Ιστορία Χρήστη Admin deletes an issue.....	31
3.2.10	Ιστορία Χρήστη Admin adds a category	31
3.2.11	Ιστορία Χρήστη Admin views all categories	32
3.2.12	Ιστορία Χρήστη Admin views a category.....	32
3.2.13	Ιστορία Χρήστη Admin deletes a category.....	32
3.2.14	Ιστορία Χρήστη Admin updates a category.....	33
Παράρτημα Ι – Ανοιχτά Θέματα.....		34

Λίστα Σχημάτων

Σχήμα 1: Εφαρμογή προτύπου Abstract Factory.....	6
Σχήμα 2: Εφαρμογή προτύπου Proxy.....	7
Σχήμα 3: Εφαρμογή προτύπου Adapter.....	8
Σχήμα 4: Μοντέλο δεδομένων Issue.....	9
Σχήμα 5: Εξήγηση και παράθεση των παραμέτρων του πόρου	10
Σχήμα 6: Αποκρίσεις του συστήματος.....	10
Σχήμα 7: Δοκιμή του endpoint.....	10
Σχήμα 8: Εξήγηση και παράθεση των παραμέτρων του πόρου.....	11
Σχήμα 9: Αποκρίσεις του συστήματος.....	11
Σχήμα 10: Δοκιμή του endpoint.....	11
Σχήμα 11: Εξήγηση και παράθεση των παραμέτρων του πόρου.....	12
Σχήμα 12: Αποκρίσεις του συστήματος.....	12
Σχήμα 13: Δοκιμή του endpoint.....	12
Σχήμα 14: Εξήγηση και παράθεση των παραμέτρων του πόρου.....	13
Σχήμα 15: Αποκρίσεις του συστήματος.....	13
Σχήμα 16: Δοκιμή του endpoint.....	13
Σχήμα 17: Εξήγηση και παράθεση των παραμέτρων του πόρου.....	14
Σχήμα 18: Αποκρίσεις του συστήματος.....	14
Σχήμα 19: Δοκιμή του endpoint.....	14
Σχήμα 20: Εξήγηση και παράθεση των παραμέτρων του πόρου.....	15
Σχήμα 21: Αποκρίσεις του συστήματος.....	15
Σχήμα 22: Δοκιμή του endpoint.....	15
Σχήμα 23: Εξήγηση και παράθεση των παραμέτρων του πόρου.....	16



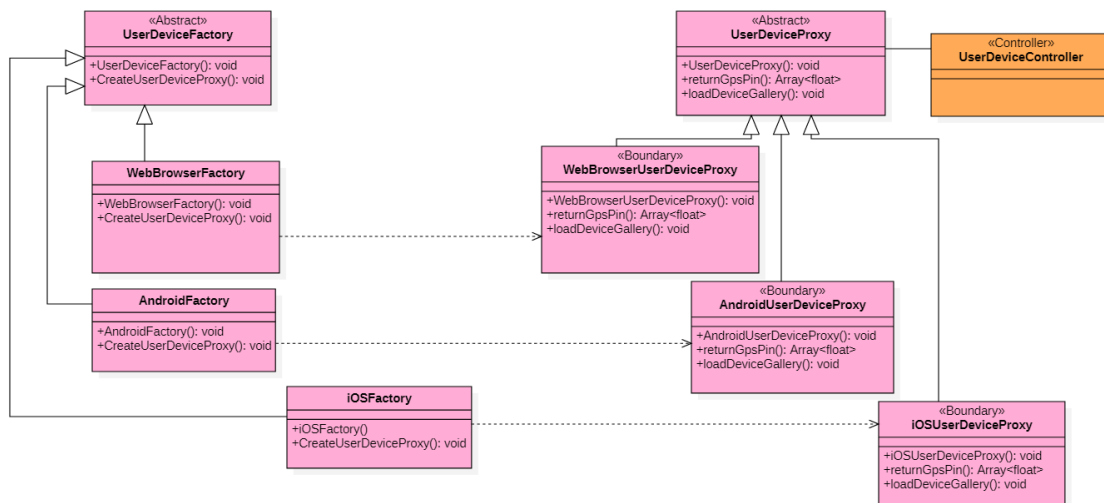
Σχήμα 24: Αποκρίσεις του συστήματος.	16
Σχήμα 25: Δοκιμή του endpoint.	16
Σχήμα 26: Εξήγηση και παράθεση των παραμέτρων του πόρου.	17
Σχήμα 27: Αποκρίσεις του συστήματος.	17
Σχήμα 28: Δοκιμή του endpoint.	17
Σχήμα 29: Εξήγηση και παράθεση των παραμέτρων του πόρου.	18
Σχήμα 30: Αποκρίσεις του συστήματος.	18
Σχήμα 31: Δοκιμή του endpoint.	18
Σχήμα 32: Μοντέλο δεδομένων Category.....	19
Σχήμα 33: Εξήγηση και παράθεση των παραμέτρων του πόρου.	19
Σχήμα 34: Αποκρίσεις του συστήματος.	19
Σχήμα 35: Δοκιμή του endpoint.	19
Σχήμα 36: Εξήγηση και παράθεση των παραμέτρων του πόρου.	20
Σχήμα 37: Αποκρίσεις του συστήματος.	20
Σχήμα 38: Δοκιμή του endpoint.	20
Σχήμα 39: Εξήγηση και παράθεση των παραμέτρων του πόρου.	21
Σχήμα 40: Αποκρίσεις του συστήματος.	21
Σχήμα 41: Δοκιμή του endpoint.	21
Σχήμα 42: Εξήγηση και παράθεση των παραμέτρων του πόρου.	22
Σχήμα 43: Αποκρίσεις του συστήματος.	22
Σχήμα 44: Δοκιμή του endpoint.	22
Σχήμα 45: Εξήγηση και παράθεση των παραμέτρων του πόρου.	23
Σχήμα 46: Αποκρίσεις του συστήματος.	23
Σχήμα 47: Δοκιμή του endpoint.	23



1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν

1.1 Πρότυπο Abstract Factory

Πρόκειται για ένα **Δημιουργικό Πρότυπο**. Στο σύστημα μας με την χρήση του προτύπου Abstract Factory, το σύστημα μας μπορεί να υποστηρίξει διαφορετικού τύπου συσκευές (Android, iOS καθώς και web browsers) και να επικοινωνεί μαζί τους, κάτι το οποίο είναι απαραίτητο για την κάλυψη της ΜΛΑ-2 που ορίστηκε στο έγγραφο απαιτήσεων χρήστη. Συγκεκριμένα, μέσω των υλοποιήσεων του *Abstract Product* User Device Proxy, είναι εφικτή η λήψη του GPS στίγματος της τοποθεσίας του χρήστη και η πρόσβαση στα αρχεία πολυμέσων σε συσκευές διάφορων τύπων. Επίσης με χρήση αυτού του προτύπου δίνεται η δυνατότητα, εάν στο μέλλον υπάρξουν και νέες δημοφιλείς συσκευές, να μπορεί το σύστημα μας να αλληλοεπιδρά και με αυτές με την δημιουργία κατάλληλης concrete κλάσης στο κατάλληλο *Abstract Product*.

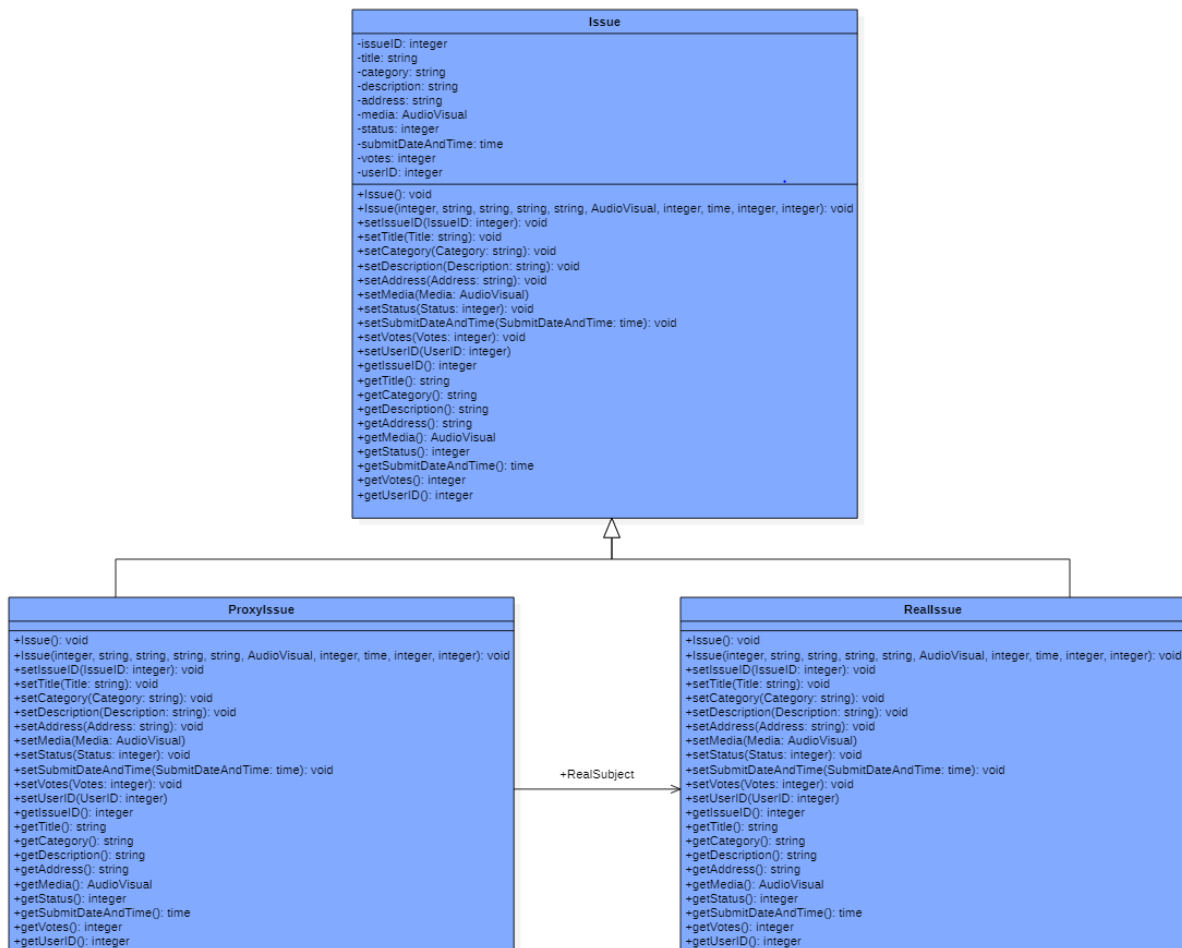


Σχήμα 1: Εφαρμογή προτύπου Abstract Factory.



1.2 Πρότυπο Proxy

Το πρότυπο proxy είναι ένα **δομικό πρότυπο**. Στο σύστημα myDimos, εφαρμόζεται με στόχο την κάλυψη των ΜΛΑ-1 και ΜΛΑ-5, που πραγματεύονται την **αξιοπιστία** και **ασφάλεια** του συστήματος έναντι κακόβουλων επιθέσεων και την προστασία των προσωπικών δεδομένων των χρηστών κατά GDPR, μέσω του ελέγχου πρόσβασης σε ένα αντικείμενο της κλάσης issue. Πιο συγκεκριμένα, ένα ζήτημα μπορεί να προβληθεί μόνο από ταυτοποιημένους χρήστες, οι οποίοι μάλιστα έχουν διαφορετικά δικαιώματα πρόσβασης ανάλογα με την ιδιότητα τους (απλός χρήστης ή διαχειριστής) ή εάν είναι δημιουργοί του ίδιου του ζητήματος. Επομένως, στην κλάση Issue εφαρμόσαμε ένα **protection proxy**. Επίσης το Proxy πρότυπο, με κατάλληλη υλοποίηση μπορεί να λειτουργήσει και ως virtual proxy, προκειμένου να φορτώνει ένα προσωρινό αντικείμενο (εικόνα), έως ότου δοθεί πρόσβαση στο πραγματικό περιεχόμενο ενός αντικειμένου. Η περίπτωση αυτή συμβάλει στην ομαλότερη εμπειρία χρήστη.

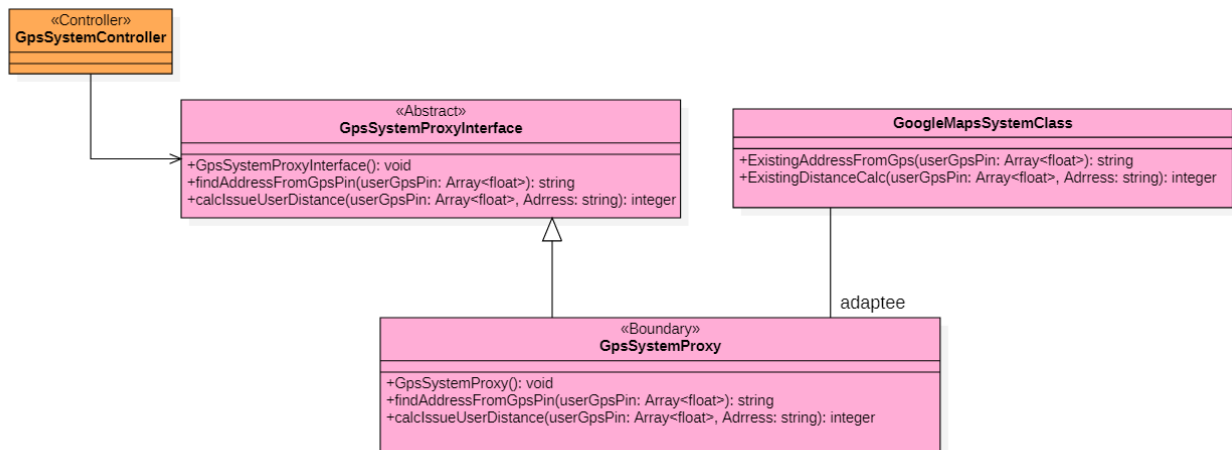


Σχήμα 2: Εφαρμογή προτύπου Proxy.



1.3 Πρότυπο Adapter

Το πρότυπο adapter είναι ένα **δομικό πρότυπο**. Στο σύστημα μας, το πρότυπο αυτό χρησιμοποιείται για την συνεργασία των διεπαφών του συστήματος μας, με το εξωτερικό σύστημα GoogleMaps, με στόχο την κάλυψη της της ΜΛΑ-3, κατά την οποία πρέπει να γίνεται ορθή διαχείριση των ιχνών GPS του χρήστη. Συγκεκριμένα χρησιμοποιείται για να μπορέσει να χρησιμοποιηθεί η λειτουργικότητα του συστήματος Google maps για τις ανάγκες του δικού μας συστήματος (λήψη διεύθυνσης από GPS στίγμα, υπολογισμός απόστασης μεταξύ στίγματος GPS χρήστη και συγκεκριμένης διεύθυνσης). Η κλάση **GoogleMapsSystemClass** αντιπροσωπεύει το API του συστήματος του οποίου θέλουμε να αξιοποιήσουμε την λειτουργικότητα. Επίσης με την χρήση αυτού του προτύπου επιτυγχάνεται, σε περίπτωση που το σύστημα διαχείρισης GPS (google maps) μεταβληθεί να είναι εφικτή η επαναχρησιμοποίηση του, απλά μεταβάλλοντας κατάλληλα την concrete κλάση **GpsSystemProxy**, χωρίς να χρειαστεί να μεταβληθεί κάποια άλλη κλάση ή υλοποίηση στην πλευρά του συστήματος μας.



Σχήμα 3: Εφαρμογή προτύπου Adapter.



2. Αρχιτεκτονική Συστήματος

Χρησιμοποιήθηκε το εργαλείο Swaggerhub.

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

2.1 Αναγνώριση Πόρων (Resources) Συστήματος

Κλάση BEC	Πόρος REST-	Endpoints (HTTP Verbs)
Issue, (User)	/user/{username}/issue	POST, GET
Issue, (User)	/user/{username}/issue/{issueID}	GET, PUT, DELETE
Issue, (AdminEntity)	/admin/{adminUsername}/issue	GET
Issue, (AdminEntity)	/admin/{adminUsername}/issue/{issueID}	GET, PUT, DELETE
CategoryEntity, (AdminEntity)	/admin/{adminUsername}/category/	GET, POST
CategoryEntity, (AdminEntity)	/admin/{adminUsername}/category/{categoryID}	GET, PUT, DELETE

2.2 Τεκμηρίωση REST διεπαφής

2.2.1 Πόρος Issue

2.2.1.1 Μοντέλο δεδομένων Issue

```
Issue {
  issueID      integer
  title        string
  category     string
  description  string
  address      string
  media        string($byte)
  status       integer
  submitDateAndTime string($date-time)
  votes        integer
  userID       integer
}
```

Σχήμα 4: Μοντέλο δεδομένων Issue.



2.2.1.2 Endpoint GET πόρου Issue, για διαχειριστή και συγκεκριμένο Issue id

GET `/admin/{adminUsername}/issue/{issueID}` Admin Retrieves An Issue

FR8 - An administrator must be able to view all issues.

Parameters

Name	Description
adminUsername * required string (path)	The username of the admin viewing the issue.
issueID * required string (path)	The ID of the issue that is retrieved.

Execute Clear

Responses	
Code	Description
200	Successful operation Example Value Model <pre>{ "issueID": 0, "title": "string", "category": "string", "description": "string", "address": "string", "media": "string", "status": 0, "submitDateAndTime": "2020-06-06T08:35:20.527Z", "votes": 0, "userID": 0}</pre>
400	Bad input parameter
404	Issue not found

Σχήμα 5: Εξήγηση και παράθεση των παραμέτρων του πόρου

Σχήμα 6: Αποκρίσεις του συστήματος.

Curl

```
curl -X GET "https://virtserver.swaggerhub.com/manolismi3/myDimos/1.0.0/admin/michalain/issue/1221" -H "accept: application/json"
```

Request URL

```
https://virtserver.swaggerhub.com/manolismi3/myDimos/1.0.0/admin/michalain/issue/1221
```

Server response

Code	Details
200	Response body <pre>{ "issueID": 0,</pre>

Σχήμα 7: Δοκιμή του endpoint.



2.2.1.3 Endpoint PUT πόρου Issue, για διαχειριστή και συγκεκριμένο Issue id

PUT

/admin/{adminUsername}/issue/{issueID} Admin issue management

FR11 - An administrator must be able to manage an issue.

Parameters

Name	Description
adminUsername * required	The username of the admin that managing the issue.
string (path)	<input type="text" value="michalain"/>
issueID * required	The ID of the issue that is being updated.
string (path)	<input type="text" value="1234"/>
Issue body * required	Issue model
object (body)	Edit Value Model
	<pre>{ "issueID": 1234, "title": "Pothole in <u>tsimiski</u>", "category": "roadworks", "description": "Needs urgent attention", "address": "Tsimiski 109, 54989, Thessaloniki, Greece", "media": "img.com/id=11235", "status": 0, "submitDateAndTime": "2020-06-06T09:13:08.232Z", "votes": 34, "userID": 124 }</pre>

Responses

Code	Description
200	Successful operation
	Example Value Model
	<pre>{ "issueID": 0, "title": "string", "category": "string", "description": "string", "address": "string", "media": "string", "status": 0, "submitDateAndTime": "2020-06-06T09:15:34.990Z", "votes": 0, "userID": 0 }</pre>
400	Bad input parameter
404	Issue not found

Σχήμα 8: Εξήγηση και παράδοση των παραμέτρων του πόρου.

Σχήμα 9: Αποκρίσεις του συστήματος.

Responses

Response content type application/json

Curl

```
curl -X PUT "https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/admin/michalain/issue/1234" -H "accept: application/json" -H "Content-Type: application/json" -d '{"issueID": 1234, "title": "Pothole in tsimiski", "category": "roadworks", "description": "Needs urgent attention", "address": "Tsimiski 109, 54989, Thessaloniki, Greece", "media": "img.com/id=11235", "status": 0, "submitDateAndTime": "2020-06-06T09:13:08.232Z", "votes": 34, "userID": 124}'
```

Request URL

https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/admin/michalain/issue/1234

Server response

Code	Details
200	Response body
	<pre>{ "issueID": 0, "title": "string", </pre>

Σχήμα 10: Δοκιμή του endpoint.



2.2.1.4 Endpoint DELETE πόρου Issue, για διαχειριστή και συγκεκριμένο Issue id

DELETE /admin/{adminUsername}/issue/{issueID} Delete an issue (admin)

FR11 - An administrator must be able to manage an issue.

Parameters

Name	Description
adminUsername * required string (path)	The username of the admin that deletes the issue.
issueID * required string (path)	The ID of the issue that is being deleted.

Responses

Code	Description
200	Successful operation
400	Bad input parameter
404	Issue not found

Σχήμα 11: Εξήγηση και παράδοση των παραμέτρων του πόρου.

Σχήμα 12: Αποκρίσεις του συστήματος.

Responses Response content type: application/json

Curl

```
curl -X DELETE "https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/admin/michalain/issue/1234" -H "accept: application/json"
```

Request URL

```
https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/admin/michalain/issue/1234
```

Server response

Code	Details
200	Response headers <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With, Content-Type, Accept, Origin access-control-allow-methods: *</pre>

Σχήμα 13: Δοκιμή του endpoint.



2.2.1.5 Endpoint GET πόρου Issue, για απλό χρήστη και για συγκεκριμένο Issue id

GET

/user/{username}/issue/{issueID} Retrieve an Issue's full content

FR4 - A logged in user must be able to view the full content of a public issue.

Parameters

Name	Description
username * required	The username of the user retrieving the issue.
string (path)	<input type="text" value="kiriakos"/>
issueID * required	The ID of the issue that is being retrieved.
string (path)	<input type="text" value="19937"/>

Responses

Code	Description
200	Successful operation
400	Bad input parameter
404	Issue not found

Example Value | Model

```
{  "issueID": 0,  "title": "string",  "category": "string",  "description": "string",  "address": "string",  "media": "string",  "status": 0,  "submitDateAndTime": "2020-06-06T14:24:07.687Z",  "votes": 0,  "userID": 0}
```

Σχήμα 14: Εξήγηση και παράδοση των παραμέτρων του πόρου.

Σχήμα 15: Αποκρίσεις του συστήματος.

Responses Response content type application/json

Curl

```
curl -X GET "https://virtserver.swaggerhub.com/manolism3/myDimos/1.0.0/user/kiriakos/issue/19937" -H "accept: application/json"
```

Request URL

```
https://virtserver.swaggerhub.com/manolism3/myDimos/1.0.0/user/kiriakos/issue/19937
```

Server response

Code	Details
200	Response body

```
{  "issueID": 0,  "title": "string",  "category": "string",
```

Σχήμα 16: Δοκιμή του endpoint.



2.2.1.6 Endpoint PUT πόρου Issue, για απλό χρήστη και για συγκεκριμένο Issue id

PUT /user/{username}/issue/{issueID} Update issue

FR3 - A logged in user must be able to vote a public issue.
FR7 - A logged in user must be able to edit one of his pending issues.

Parameters

Name	Description
username * required string (path)	The username of the user that applies an action to the existing issue.
issueID * required string (path)	The ID of the issue that is updated.
body * required object (body)	Issue model

Edit Value | Model

```
{  "issueID": 12234,  "title": "pothole",  "category": "roadwork",  "description": "A minor problem of low urgency",  "address": "Stenou 11, 54643, Thessaloniki, Greece",  "media": "img.com/1234567",  "status": 0,  "submitDateAndTime": "2020-06-06T14:28:17.163Z",  "votes": 10,  "userID": 1035634}
```

Σχήμα 17: Εξήγηση και παράδοση των παραμέτρων του πόρου.

Responses	
Code	Description
200	Successful operation
Example Value Model	
<pre>{ "issueID": 0, "title": "string", "category": "string", "description": "string", "address": "string", "media": "string", "status": 0, "submitDateAndTime": "2020-06-06T15:11:56.917Z", "votes": 0, "userID": 0}</pre>	
400	Bad input parameter
404	Issue not found

Σχήμα 18: Αποκρίσεις του συστήματος.

Execute Clear

Responses Response content type application/json

Curl

```
curl -X PUT "https://virtserver.swaggerhub.com/manolis3/myOimos/1.0.0/user/michalis/issue/12234" -H "accept: application/json" -H "Content-Type: application/json" -d '{"issueID": 12234, "title": "pothole", "category": "roadwork", "description": "A minor problem of low urgency", "address": "Stenou 11, 54643, Thessaloniki, Greece", "media": "img.com/1234567", "status": 0, "submitDateAndTime": "2020-06-06T14:28:17.163Z", "votes": 10, "userID": 1035634}'
```

Request URL

https://virtserver.swaggerhub.com/manolis3/myOimos/1.0.0/user/michalis/issue/12234

Server response

Code	Details
200	Response body

```
{  "issueID": 0,  "title": "string",
```

Σχήμα 19: Δοκιμή του endpoint.



2.2.1.7 Endpoint DELETE πόρου Issue, για απλό χρήστη και συγκεκριμένο Issue id

DELETE /user/{username}/issue/{issueID} Delete an own pending issue

FR7 - A logged in user must be able to edit(delete) one of his pending issues.

Parameters

Name	Description
username * required	The username of the user that deletes the issue
string (path)	<input type="text" value="giannakis"/>
issueID * required	The ID of the issue that is being deleted.
string (path)	<input type="text" value="1140"/>

Σχήμα 20: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Responses	
Code	Description
200	Successful operation
400	Bad input parameter
404	Issue not found

Σχήμα 21: Αποκρίσεις του συστήματος.

Execute Clear

Responses Response content type application/json

Curl

```
curl -X DELETE "https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/user/giannakis/issue/1140" -H "accept: application/json"
```

Request URL

```
https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/user/giannakis/issue/1140
```

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: *</pre>

Σχήμα 22: Δοκιμή του endpoint.



2.2.1.8 Endpoint GET πόρου Issue, για διαχειριστή

GET /admin/{adminUsername}/issue Admin retrieves all issues

FR8 - An administrator must be able to view all issues.

Parameters

Name	Description
adminUsername * required string (path)	The username of the administrator that views the issues
title string (query)	A string that corresponds to the issue's title the admin searches.
category array[string] (query)	The categories of the issues, the admin wants to view.
status array[integer] (query)	The status of the issues, the admin wants to view.

Σχήμα 23: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Responses	
Code	Description
200	Successful operation
Example Value Model	
<pre>[{ "issueID": 0, "title": "string", "category": "string", "description": "string", "address": "string", "media": "string", "status": 0, "submitDateAndTime": "2020-06-06T15:32:47.641Z", "votes": 0, "userID": 0 }]</pre>	
400	Bad input parameter

Σχήμα 24: Αποκρίσεις του συστήματος.

address
string
(query)

The address(location) the issue happened.

Tsimiski

dateFrom
string(\$date)
(query)

The beginning date of the issues the admin wants to see.

10/5/2020

dateTo
string(\$date)
(query)

The end date of the issues the admin wants to see.

6/6/2020

Execute Clear

Responses

Response content type application/json

Curl

```
curl -X GET "https://virtserver.swaggerhub.com/manolis3/myOimos/1.0.0/admin/giannakis/issue?category=roadwork&status=0,1,2&address=Tsimiski&dateFrom=10%2F5%2F2020&dateTo=6%2F6%2F2020" -H "accept: application/json"
```

Request URL

```
https://virtserver.swaggerhub.com/manolis3/myOimos/1.0.0/admin/giannakis/issue?category=roadwork&status=0,1,2&address=Tsimiski&dateFrom=10%2F5%2F2020&dateTo=6%2F6%2F2020
```

Σχήμα 25: Δοκιμή του endpoint.



2.2.1.9 Endpoint GET πόρου Issue, για απλό χρήστη

GET /user/{username}/issue Retrieve issues

FR2 - A logged in user must be able to view all public issues.
FR6 - A logged in user must be able to view his pending issues.
FR5 - A logged in user must be able to apply filters on public issues.

Parameters

Name	Description
username * required string (path)	The username of the user that views the issues
title string (query)	A string that corresponds to the issue's title the user searches.
category array[string] (query)	The categories of the issues, the user wants to view.
status array[integer] (query)	The status of the issues, the user wants to view.

Example values for query parameters:

- username: kiriakos
- title: broken sign
- category: roadwork
- status: 0

Σχήμα 26: Εξήγηση και παράδοση των παραμέτρων του πόρου.

Responses

Code	Description
200	Successful operation
400	Bad input parameter

Example Value | Model

```
[  
  {  
    "issueID": 0,  
    "title": "string",  
    "category": "string",  
    "description": "string",  
    "address": "string",  
    "media": "string",  
    "status": 0,  
    "submitDateAndTime": "2020-06-06T16:02:24.190Z",  
    "votes": 0,  
    "userID": 0  
  }  
]
```

Σχήμα 27: Αποκρίσεις του συστήματος.

maxDistance
number(float)
(query)

dateFrom
string(date)
(query)

dateTo
string(date)
(query)

Execute Clear

Responses

Response content type: application/json

Curl

```
curl -X GET "https://virtserver.swaggerhub.com/manolismi3/myOimos/1.0.0/user/kiriakos/issue?title=broken%20sign&category=roadwork&status=0&maxDistance=0.4" -H "accept: application/json"
```

Request URL

```
https://virtserver.swaggerhub.com/manolismi3/myOimos/1.0.0/user/kiriakos/issue?title=broken%20sign&category=roadwork&status=0&maxDistance=0.4
```

Σχήμα 28: Δοκιμή του endpoint.



2.2.1.10 Endpoint POST πόρου Issue, για απλό χρήστη

POST `/user/{username}/issue` Add a new Issue to the system.

FR1 - A logged in user must be able to add a new issue.

Parameters

Name	Description
username * required	The username of the user that submits a new issue
string (path)	<input type="text" value="kiriakos"/>
body * required	Issue object that is going to be submitted.
object (body)	<div>Edit Value Model</div> <pre>{ "issueID": 12345, "title": "Αδέσποτες αγέλες σκύλων", "category": "Ζώα", "description": "Μια αγέλη 10 αδέσποτων σκύλων απειλεί τους κατοίκους", "address": "Κάτω τζουμαγιάς 5, 54322, Θεσσαλονίκη, Ελλάδα", "media": "img.com/12345", "status": 0, "submitDateAndTime": "2020-06-06T14:07:20.782Z", "votes": 10, "userID": 11450 }</pre>

Responses

Code	Description
201	Issue created
400	Bad input parameter
405	Invalid input

Example Value | Model

```
{
  "issueID": 0,
  "title": "string",
  "category": "string",
  "description": "string",
  "address": "string",
  "media": "string",
  "status": 0,
  "submitDateAndTime": "2020-06-06T14:17:56.648Z",
  "votes": 0,
  "userID": 0
}
```

Σχήμα 29: Εξήγηση και παράδοση των παραμέτρων του πόρου.

Σχήμα 30: Αποκρίσεις του συστήματος.

Curl

```
curl -X POST "https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/user/kiriakos/issue" -H "accept: application/json" -H "Content-Type: application/json" -d '{"issueID": 12345, "title": "Αδέσποτες αγέλες σκύλων", "category": "Ζώα", "description": "Μια αγέλη 10 αδέσποτων σκύλων απειλεί τους κατοίκους", "address": "Κάτω τζουμαγιάς 5, 54322, Θεσσαλονίκη, Ελλάδα", "media": "img.com/12345", "status": 0, "submitDateAndTime": "2020-06-06T14:07:20.782Z", "votes": 10, "userID": 11450}'
```

Request URL

`https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/user/kiriakos/issue`

Server response

Code	Details
201	Response headers

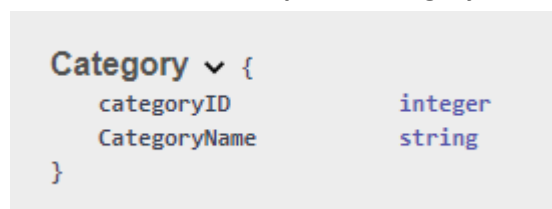
```
access-control-allow-credentials: true
access-control-allow-headers: X-Requested-With, Content-Type, Accept, Origin
```

Σχήμα 31: Δοκιμή του endpoint.



2.2.2 Πόρος Category

2.2.2.1 Μοντέλο δεδομένων Category



Σχήμα 32: Μοντέλο δεδομένων Category.

2.2.2.2 Endpoint POST πόρου Category

POST /admin/{adminUsername}/category Add a new Category to the system.

FR10 - An administrator must be able to manage categories.

Parameters

Name	Description
adminUsername * required string (path)	The username of the administrator adding a new category. <input type="text" value="michalis"/>
body * required object (body)	Category object that is being added to the system. Edit Value Model <pre>{ "categoryID": 13, "CategoryName": "Καθαριότητα"}</pre>

Σχήμα 33: Εξήγηση και παράδοση των παραμέτρων του πόρου.

Responses	
Code	Description
201	Category created Example Value Model <pre>{ "categoryID": 0, "CategoryName": "string"}</pre>
400	Bad input parameter

Σχήμα 34: Αποκρίσεις του συστήματος.

Responses Response content type application/json

Curl

```
curl -X POST "https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/admin/michalis/category" -H "accept: application/json" -H "Content-Type: application/json" -d '{"categoryID": 13, "CategoryName": "Καθαριότητα"}'
```

Request URL

```
https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/admin/michalis/category
```

Server response

Code	Details
201	Response body <pre>{ "categoryID": 0, "CategoryName": "string"}</pre> Download Response headers

Σχήμα 35: Δοκιμή του endpoint.



2.2.2.3 Endpoint GET πόρου Category

GET /admin/{adminUsername}/category View all issues' categories of the system.

FR10 - An administrator must be able to manage categories.

Parameters

Name	Description
adminUsername * required	The username of the administrator that views the categories
string (path)	<input type="text" value="michalis"/>

Σχήμα 36: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Responses

Code	Description
200	Successful operation
400	Bad input parameter

Example Value | Model

```
[  
  {  
    "categoryID": 0,  
    "CategoryName": "string"  
  }  
]
```

Σχήμα 37: Αποκρίσεις του συστήματος.

Responses

Response content type: application/json

Curl

```
curl -X GET "https://virtserver.swaggerhub.com/manolismi3/myDimos/1.0.0/admin/michalis/category" -H "accept: application/json"
```

Request URL

```
https://virtserver.swaggerhub.com/manolismi3/myDimos/1.0.0/admin/michalis/category
```

Server response

Code	Details
200	Response body

```
[  
  {  
    "categoryID": 0,  
    "CategoryName": "string"  
  }  
]
```

Σχήμα 38: Δοκιμή του endpoint.



2.2.2.4 Endpoint GET πόρου Category, για συγκεκριμένο Category id

GET /admin/{adminUsername}/category/{categoryID} Retrieve category

FR10 - An administrator must be able to manage categories.

Parameters

Name	Description
adminUsername * required string (path)	The username of the admin retrieving the category.
categoryID * required string (path)	The ID of the category.

Σχήμα 39: Εξήγηση και παράδοση των παραμέτρων του πόρου.

Responses	
Code	Description
200	Successful operation
Example Value Model	
<pre>{ "categoryID": 0, "CategoryName": "string"}</pre>	
400	Bad input parameter
404	Category not found

Σχήμα 40: Αποκρίσεις του συστήματος.

Responses Response content type application/json

Curl

```
curl -X GET "https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/admin/michalain/category/11" -H "accept: application/json"
```

Request URL

```
https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/admin/michalain/category/11
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "categoryID": 0, "CategoryName": "string"}</pre> <p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With,Content-Type,Accept,Origin</pre>

Σχήμα 41: Δοκιμή του endpoint.



2.2.2.5 Endpoint DELETE πόρου Category, για συγκεκριμένο Category id

DELETE /admin/{adminUsername}/category/{categoryID} Delete Category from the system

FR10 - An administrator must be able to manage categories.

Parameters

Name	Description
adminUsername * required string (path)	The username of the admin deleting the category.
categoryID * required string (path)	The ID of the category, being deleted.

Responses

Code	Description
200	Successful operation
400	Bad input parameter
404	Category not found

Σχήμα 42: Εξήγηση και παράδοση των παραμέτρων του πόρου.

Σχήμα 43: Αποκρίσεις του συστήματος.

Responses Response content type application/json

Curl

```
curl -X DELETE "https://virtserver.swaggerhub.com/manolismi3/myDimos/1.0.0/admin/michalain/category/9" -H "accept: application/json"
```

Request URL

```
https://virtserver.swaggerhub.com/manolismi3/myDimos/1.0.0/admin/michalain/category/9
```

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: Sat, 06 Jun 2020 16:31:29 GMT etag: W/"14-40h36nhrn3soo1wXl36FqT6xz/E" expires: -1 server: nginx/1.14.0 + Phusion Passenger 5.3.4 status: 200, 200 OK via: 1.1 013a789b7ec978dd950fd5eef80d2e6e.cloudfront.net (CloudFront) x-amz-cf-id: NQAgepttr2RqgoyzdVvtrK4R1ghNQ66VKx/Dxjv82F5DEkQw_Bjng== x-amz-cf-pop: ATH50-C1 x-cache: Miss from cloudfront x-powered-by: Express, Phusion Passenger 5.3.4</pre></div>

Σχήμα 44: Δοκιμή του endpoint.



2.2.2.6 Endpoint PUT πόρου Category, για συγκεκριμένο Category id

PUT /admin/{adminUsername}/category/{categoryID} Update a category

FR10 - An administrator must be able to manage categories.

Parameters

Name	Description
adminUsername * required string (path)	The username of the admin updating a category. <input type="text" value="kiriakos"/>
categoryID * required string (path)	The ID of the category being updated. <input type="text" value="4"/>
body * required object (body)	Category object that is being updated. Edit Value Model <pre>{ "categoryID": 4, "CategoryName": "Έργα Οδοποιίας"}</pre>

Σχήμα 45: Εξήγηση και παράθεση των παραμέτρων του πόρου.

Responses

Code	Description
200	Successful operation Example Value Model <pre>{ "categoryID": 0, "CategoryName": "string"}</pre>
400	Bad input parameter
404	Category not found

Σχήμα 46: Αποκρίσεις του συστήματος.

Responses Response content type application/json

Curl

```
curl -X PUT "https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/admin/kiriakos/category/4" -H "accept: application/json" -H "Content-Type: application/json" -d '{"categoryID": 4, "CategoryName": "Έργα Οδοποιίας"}'
```

Request URL

```
https://virtserver.swaggerhub.com/manolis3/myDimos/1.0.0/admin/kiriakos/category/4
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "categoryID": 0, "CategoryName": "string"}</pre> <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: *</pre>

Download

Σχήμα 47: Δοκιμή του endpoint.



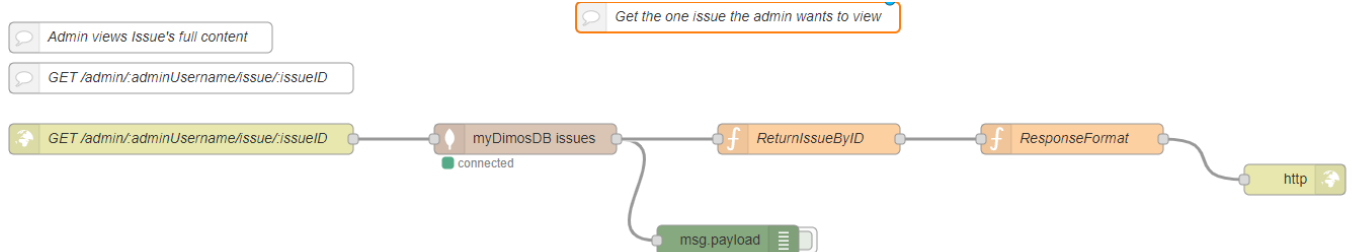
3. Υλοποίηση Συστήματος με Node-RED

3.1 Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed

- [Σύνδεσμος](#) για το αρχείο τύπου **zip** που περιέχει τα flows.
- [Σύνδεσμος](#) για το αρχείο τύπου **zip** που περιέχει την βάση δεδομένων που χρησιμοποιήθηκε στις δοκιμές.

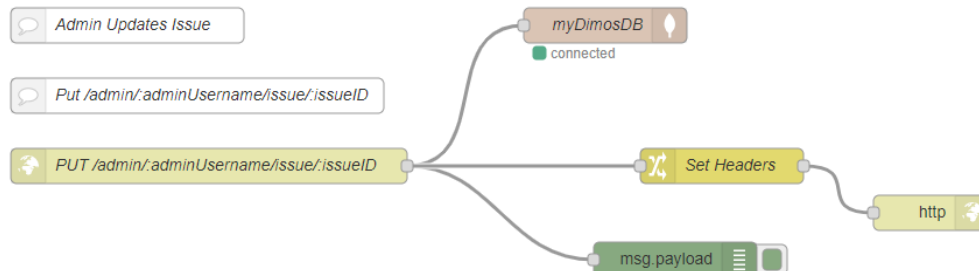
3.1.1 Ροές πόρου Issue

Ροή endpoint GET /admin/{adminUsername}/issue/{issueID}



Η ροή που υλοποιεί την υπηρεσία, που είναι υπεύθυνη για την επιστροφή ενός συγκεκριμένου ζητήματος σε έναν διαχειριστή. Μέσω της συνάρτησης ReturnIssueById, επιστρέφεται το ζήτημα με συγκεκριμένο issueID, όπως ορίζεται στο URL parameter, ενώ η ResponseFormat, φέρνει το HTTP response που επιστρέφεται στην επιθυμητή μορφή.

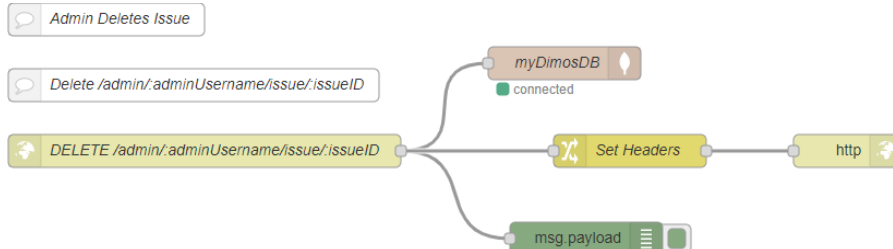
Ροή endpoint PUT /admin/{adminUsername}/issue/{issueID}



Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας ένας διαχειριστής μπορεί να ενημερώνει την κατάσταση ενός συγκεκριμένου (με βάση issueID) ζητήματος στην βάση δεδομένων.

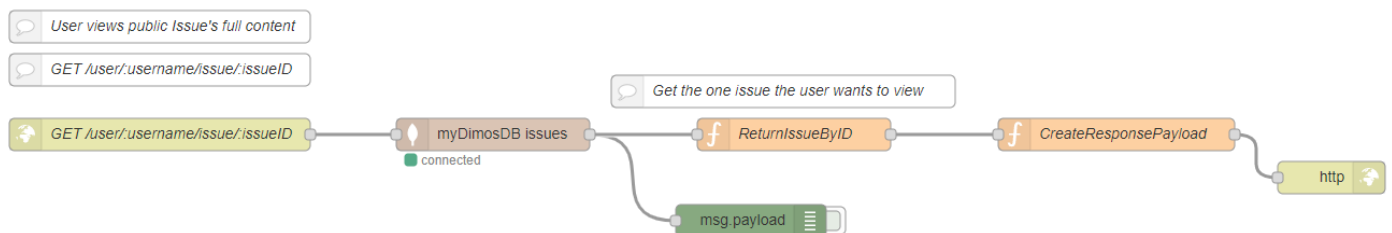


Ροή endpoint DELETE /admin/{adminUsername}/issue/{issueID}



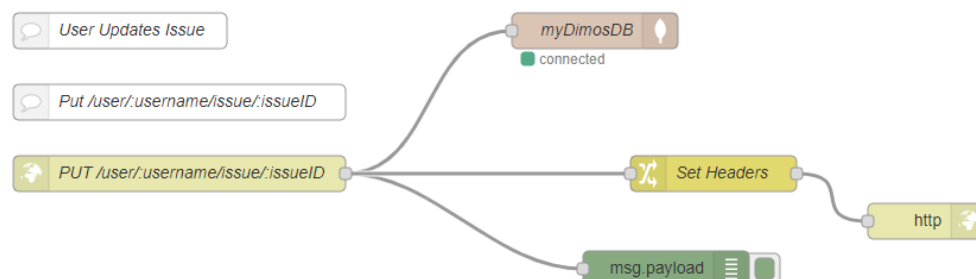
Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας ένας διαχειριστής μπορεί να διαγράψει ένα συγκεκριμένο ζήτημα από την βάση δεδομένων.

Ροή endpoint GET /user/{username}/issue/{issueID}



Η ροή που υλοποιεί την υπηρεσία, που είναι υπεύθυνη για την επιστροφή ενός συγκεκριμένου ζητήματος σε έναν απλό χρήστη. Μέσω της συνάρτησης ReturnIssueByID, επιστρέφεται το ζήτημα με συγκεκριμένο issueID, όπως ορίζεται στο URL parameter, ενώ η CreateResponsePayload, φέρνει το HTTP response που επιστρέφεται στην επιθυμητή μορφή.

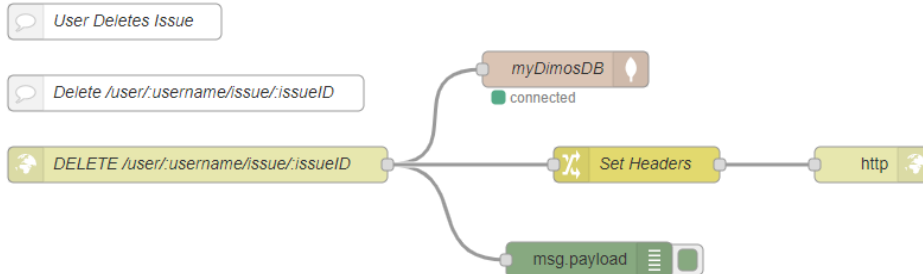
Ροή endpoint PUT /user/{username}/issue/{issueID}



Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας μπορεί ένας χρήστης να ενημερώσει το περιεχόμενο ενός ζητήματος που καταχώρησε νωρίτερα στο σύστημα.

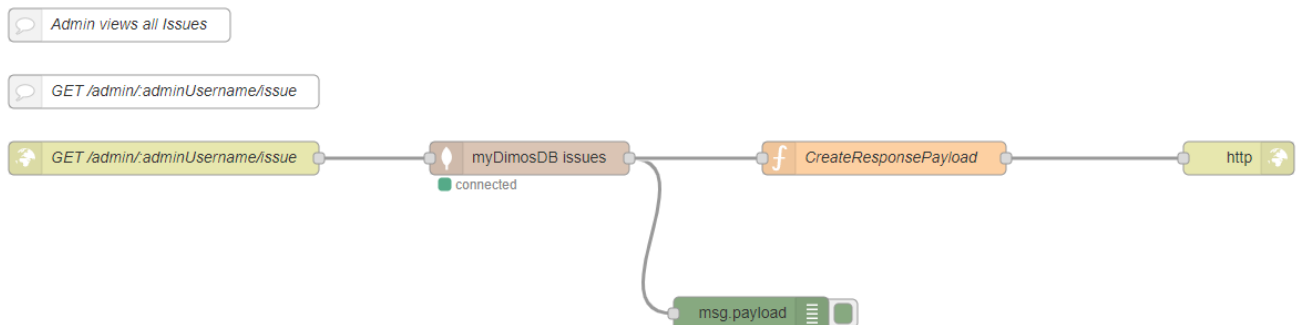


Ροή endpoint DELETE /user/{username}/issue/{issueID}



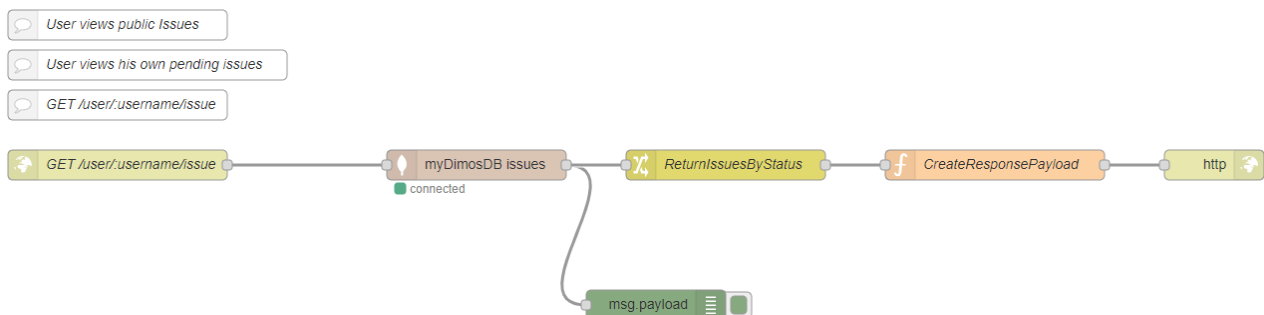
Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας μπορεί ένας χρήστης να διαγράψει ένα ζήτημα που καταχώρησε νωρίτερα στο σύστημα.

Ροή endpoint GET /admin/{adminUsername}/issue



Η ροή που υλοποιεί την υπηρεσία, που είναι υπεύθυνη για την επιστροφή της λίστας των ζητημάτων σε έναν διαχειριστή. Μέσω της συνάρτησης ReturnIssuesByStatus, επιστρέφονται τα ζητήματα που καλύπτουν το request του χρήστη, ενώ η CreateResponsePayload, φέρνει το HTTP response που επιστρέφεται στην επιθυμητή μορφή.

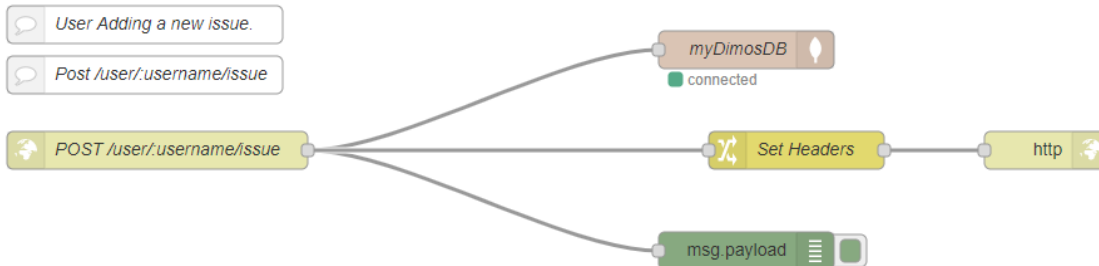
Ροή endpoint GET /user/{username}/issue



Η ροή που υλοποιεί την υπηρεσία, που είναι υπεύθυνη για την επιστροφή της λίστας των ζητημάτων (δημόσιων ή αναμενόμενων δικών του) στον απλό χρήστη. Μέσω της συνάρτησης ReturnIssuesByStatus, επιστρέφονται τα ζητήματα που καλύπτουν το request του χρήστη, ενώ η CreateResponsePayload, φέρνει το HTTP response που επιστρέφεται στην επιθυμητή μορφή.



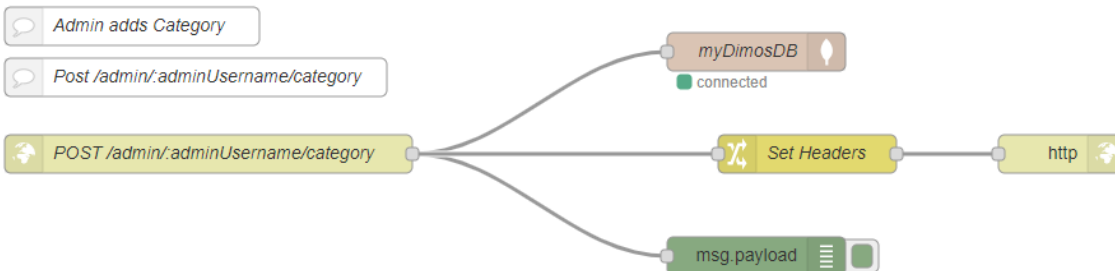
Ροή endpoint POST /user/{username}/issue



Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας μπορεί ένας χρήστης να καταχωρήσει ένα νέο ζήτημα στο σύστημα.

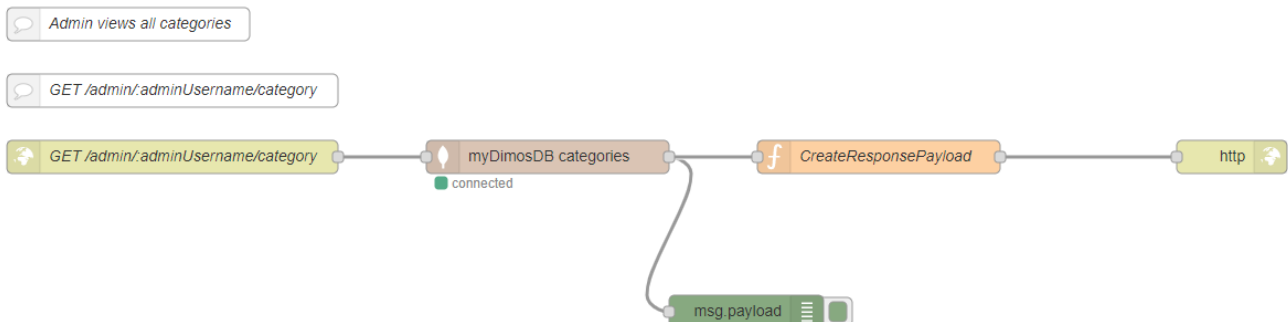
3.1.2 Ροές πόρου Category

Ροή endpoint POST /admin/{adminUsername}/category



Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας μπορεί ένας διαχειριστής να καταχωρήσει ένα νέο ζήτημα στο σύστημα.

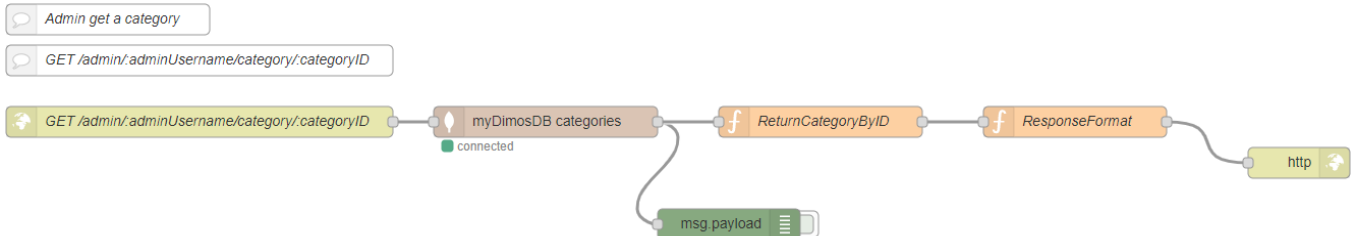
Ροή endpoint GET /admin/{adminUsername}/category



Η ροή που υλοποιεί την υπηρεσία, που είναι υπεύθυνη για την επιστροφή της λίστας των υφιστάμενων κατηγοριών σε έναν διαχειριστή.

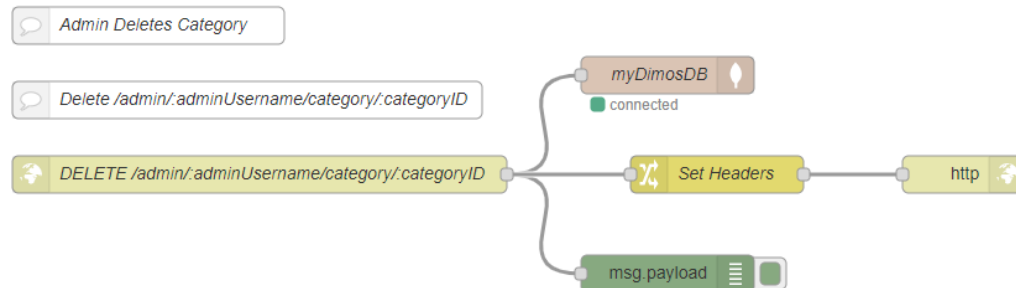


Ροή endpoint GET /admin/{adminUsername}/category/{categoryID}



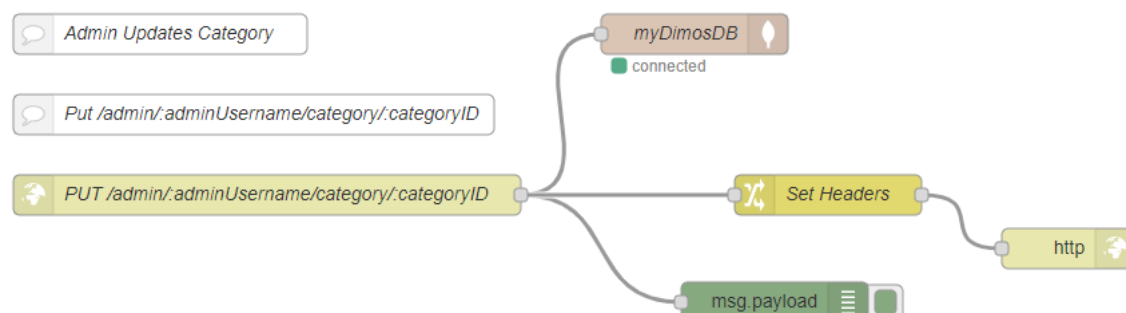
Η ροή που υλοποιεί την υπηρεσία, που είναι υπεύθυνη για την επιστροφή μιας συγκεκριμένης κατηγορίας σε έναν διαχειριστή. Μέσω της συνάρτησης ReturnCategoryByID, επιστρέφεται η κατηγορία με συγκεκριμένο categoryID, όπως ορίζεται στο URL parameter.

Ροή endpoint DELETE /admin/{adminUsername}/category/{categoryID}



Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας μπορεί ένας διαχειριστής να διαγράψει μια κατηγορία του συστήματος.

Ροή endpoint PUT /admin/{adminUsername}/category/{categoryID}

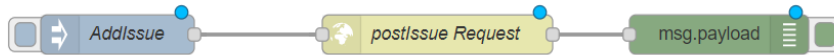


Η ροή που υλοποιεί την υπηρεσία, μέσω της οποίας μπορεί ένας διαχειριστής να ενημερώσει μια κατηγορία του συστήματος.



3.2 Υλοποίηση Ιστοριών χρήστη

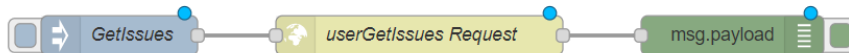
3.2.1 Ιστορία Χρήστη User add issue



Ροή μέσω της οποίας ο χρήστης, μπορεί να προσθέσει ένα νέο ζήτημα στο σύστημα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
AddIssue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
postIssue Request	http-request	Κάνει κλήση προς την υπηρεσία Post /user/{username}/issue η οποία, δίνει στον χρήστη {username} την δυνατότητα να καταχωρήσει ένα νέο ζήτημα στο σύστημα.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη, δηλαδή τον κατάλληλο κωδικό και το ζήτημα που δημιουργεί ο χρήστης.

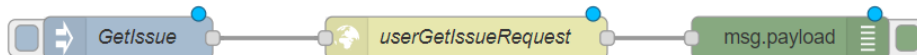
3.2.2 Ιστορία Χρήστη User views issues



Ροή μέσω της οποίας ο χρήστης μπορεί να δει 1) τα δημόσια ζητήματα καθώς και 2) τα δικά του εκκρεμή ζητήματα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
GetIssues	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
userGetIssues Request	http-request	Κάνει κλήση προς την υπηρεσία Get /user/{username}/issue η οποία επιστρέφει την λίστα με τα ζητήματα που αιτείται ο χρήστης {username}.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη, δηλαδή τα ζητήματα που αιτείται ο χρήστης.

3.2.3 Ιστορία Χρήστη User views full content issue

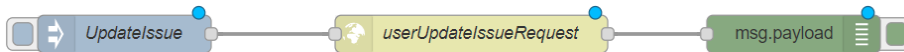


Ροή μέσω της οποίας ο χρήστης μπορεί να δει το πλήρες περιεχόμενο ενός ζητήματος.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
GetIssue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
userGetIssueRequest	http-request	Κάνει κλήση προς την υπηρεσία Get /user/{username}/issue/{issueID} η οποία επιστρέφει το πλήρες περιεχόμενο του ζητήματος με ID: issueID στον χρήστη {username}.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη, δηλαδή το ζήτημα που αιτείται ο χρήστης.



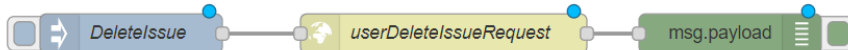
3.2.4 Ιστορία Χρήστη User updates an issue



Ροή μέσω της οποίας ο χρήστης μπορεί να επεξεργαστεί ένα δικό του εκκρεμές ζήτημα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
UpdateIssue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
userUpdateIssueRequest	http-request	Κάνει κλήση προς την υπηρεσία Put /user/{username}/issue/{issueID} η οποία δίνει στον χρήστη {username} την δυνατότητα ενημέρωσης του ζητήματος με ID: issueID .
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη, δηλαδή το ζήτημα που ανανεώνει ο χρήστης.

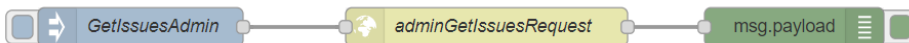
3.2.5 Ιστορία Χρήστη User deletes an issue



Ροή μέσω της οποίας ο χρήστης μπορεί να διαγράψει ένα δικό του εκκρεμές ζήτημα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
DeleteIssue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
userDeleteIssueRequest	http-request	Κάνει κλήση προς την υπηρεσία Delete /user/{username}/issue/{issueID} η οποία δίνει στον χρήστη {username} να διαγράψει το ζήτημα με ID: issueID .
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη.

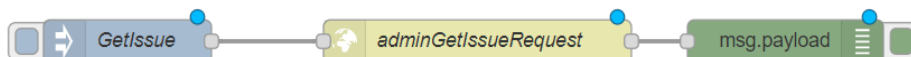
3.2.6 Ιστορία Χρήστη Admin views issues



Ροή μέσω της οποίας ο διαχειριστής μπορεί να δει τα ζητήματα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
GetIssuesAdmin	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
adminGetIssuesRequest	http-request	Κάνει κλήση προς την υπηρεσία Get /admin/{adminUsername}/issue η οποία επιστρέφει την λίστα με τα ζητήματα που αιτείται ο διαχειριστής {adminUsername} .
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη (διαχειριστή), δηλαδή τα ζητήματα που αιτείται ο χρήστης.

3.2.7 Ιστορία Χρήστη Admin views full content issue

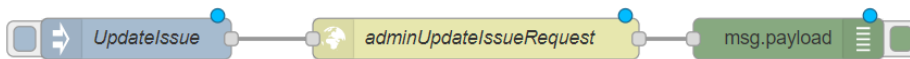


Ροή μέσω της οποίας ένας διαχειριστής μπορεί να δει το πλήρες περιεχόμενο ενός ζητήματος.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
GetIssue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
adminGetIssueRequest	http-request	Κάνει κλήση προς την υπηρεσία GET /admin/{adminUsername}/issue/{issueID} η οποία επιστρέφει το πλήρες περιεχόμενο του ζητήματος με ID: issueID στον διαχειριστή {adminUsername}.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη (διαχειριστή), δηλαδή το ζήτημα που αιτείται ο χρήστης.

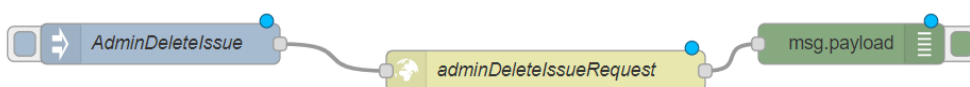
3.2.8 Ιστορία Χρήστη Admin updates an issue



Ροή μέσω της οποίας ένας διαχειριστής μπορεί να επεξεργαστεί ένα ζήτημα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
UpdateIssue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
adminUpdateIssueRequest	http-request	Κάνει κλήση προς την υπηρεσία PUT /admin/{adminUsername}/issue/{issueID} η οποία δίνει την δυνατότητα επεξεργασίας του ζητήματος με ID: issueID στον διαχειριστή {adminUsername}.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη (διαχειριστή).

3.2.9 Ιστορία Χρήστη Admin deletes an issue



Ροή μέσω της οποίας ένας διαχειριστής μπορεί να διαγράψει ένα ζήτημα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
AdminDeleteIssue	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
adminDeleteIssueRequest	http-request	Κάνει κλήση προς την υπηρεσία DELETE /admin/{adminUsername}/issue/{issueID} η οποία δίνει την δυνατότητα διαγραφής του ζητήματος με ID: issueID στον διαχειριστή {adminUsername}.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη (διαχειριστή).

3.2.10 Ιστορία Χρήστη Admin adds a category

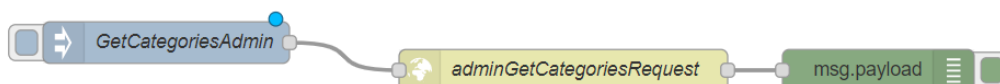




Ροή μέσω της οποίας ένας διαχειριστής μπορεί να προσθέσει μια νέα κατηγορία στο σύστημα.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
AddCategory	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
adminPostCategoryRequest	http-request	Κάνει κλήση προς την υπηρεσία POST /admin/{adminUsername}/category η οποία δίνει στον διαχειριστή {adminUsername} την δυνατότητα να προσθέσει νέα κατηγορία.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη (διαχειριστή).

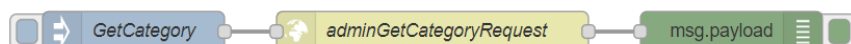
3.2.11 Ιστορία Χρήστη Admin views all categories



Ροή μέσω της οποίας ένας διαχειριστής μπορεί να δει τις κατηγορίες του συστήματος.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
GetCategoriesAdmin	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
AdminGetCategoriesRequest	http-request	Κάνει κλήση προς την υπηρεσία GET /admin/{adminUsername}/category η οποία δίνει στον διαχειριστή {adminUsername} την δυνατότητα να δει τις κατηγορίες του συστήματος.
Msg.payload	debug	Τυπώνει στην κονσόλα την λίστα με τους ενεργούς χρήστες .

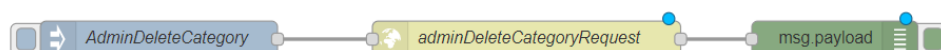
3.2.12 Ιστορία Χρήστη Admin views a category



Ροή μέσω της οποίας ένας διαχειριστής μπορεί να δει μια κατηγορία του συστήματος.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
GetCategory	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
adminGetCategory	http-request	Κάνει κλήση προς την υπηρεσία GET /admin/{adminUsername}/category/{categoryID} η οποία δίνει στον διαχειριστή {adminUsername} την δυνατότητα να δει την κατηγορία με ID categoryID.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη (διαχειριστή), δηλαδή την κατηγορία που αιτείται.

3.2.13 Ιστορία Χρήστη Admin deletes a category

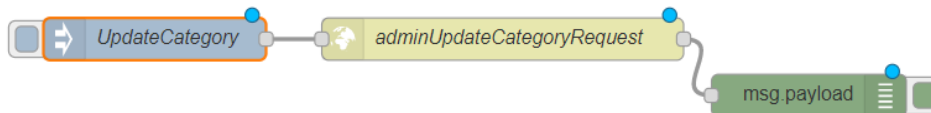


Ροή μέσω της οποίας ένας διαχειριστής μπορεί να διαγράψει μια κατηγορία από το σύστημα.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
AdminDeleteCategory	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
AdminDeleteCategoryRequest	http-request	Κάνει κλήση προς την υπηρεσία DELETE /admin/{adminUsername}/category/{categoryID} η οποία δίνει στον διαχειριστή {adminUsername} την δυνατότητα να διαγράψει την κατηγορία με ID categoryID.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη (διαχειριστή).

3.2.14 Ιστορία Χρήστη Admin updates a category



Ροή μέσω της οποίας ένας διαχειριστής μπορεί να επεξεργαστεί μια κατηγορία του συστήματος.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
UpdateCategory	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
adminUpdateCategoryRequest	http-request	Κάνει κλήση προς την υπηρεσία DELETE /admin/{adminUsername}/category/{categoryID} η οποία δίνει στον διαχειριστή {adminUsername} την δυνατότητα να επεξεργαστεί την κατηγορία με ID: categoryID.
Msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη (διαχειριστή).



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

8^ο Εξάμηνο
Άνοιξη 2020

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών

Α.Π.Θ.

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.
- Τα πρότυπα σχεδίασης που χρησιμοποιήθηκαν δεν καλύπτουν όλες τις απαιτήσεις. Επιπλέον, απαιτείται ένας εξορθολογισμός των κλάσεων του προηγούμενου παραδοτέου ώστε η σχεδίαση να γίνει πιο απλή και να φανούν ξεκάθαρα οι περιπτώσεις στις οποίες χρησιμοποιούνται πρότυπα σχεδίασης.