



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2021



PET MY PET

TAKING CARE OF YOUR PETS

Προδιαγραφές Λογισμικού και Υλοποίηση συστήματος

ΟΜΑΔΑ 12

ΤΣΟΥΣΗΣ ΠΑΝΑΓΙΩΤΗΣ - pitsousis@ece.auth.gr

ΣΑΜΑΡΑΣ ΔΗΜΗΤΡΙΟΣ - dzsamaras@ece.auth.gr

ΚΑΣΙΜΙΑΔΗΣ ΓΕΩΡΓΙΟΣ - gkasimiad@ece.auth.gr

ΠΟΥΓΑΡΙΔΗΣ ΝΙΚΟΛΑΟΣ - pougaridi@ece.auth.gr



Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
Α. Συμεωνίδης	17/05/2007	Δημιουργία εγγράφου. Προσαρμογή των προτύπων του K. E. Wiegers ^[7] 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
ΣΑΜΑΡΑΣ ΔΗΜΗΤΡΙΟΣ	12	dzsamaras@ece.auth.gr
ΤΣΟΥΣΗΣ ΠΑΝΑΓΙΩΤΗΣ	12	pitsousis@ece.auth.gr
ΚΑΣΙΜΙΑΔΗΣ ΓΕΩΡΓΙΟΣ	12	gkasimiad@ece.auth.gr
ΠΟΥΓΑΡΙΔΗΣ ΝΙΚΟΛΑΟΣ	12	pougaridi@ece.auth.gr



Πίνακας Περιεχομένων

Περιεχόμενα

Πίνακας Περιεχομένων	3
Λίστα Σχημάτων	4
1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν	5
1.1 Πρότυπο Σχεδίασης 1	5
1.2 Πρότυπο Σχεδίασης 2	6
1.3 Πρότυπο Σχεδίασης 3	7
2. Αρχιτεκτονική Συστήματος	8
2.1 Αναγνώριση Πόρων (Resources) Συστήματος	8
2.2 Τεκμηρίωση REST διεπαφής	8
2.2.1 Πόρος User	8
2.2.2 Πόρος Evaluation	13
2.2.3 Πόρος ServiceList	18
2.2.4 Πόρος Order	27
3. Υλοποίηση Συστήματος με Node-RED	31
3.1 Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed	31
3.1.1 Ροές πόρου user	31
3.1.2 Ροές πόρου evaluation	32
3.1.3 Ροές πόρου serviceList	33
3.1.4 Ροές πόρου order	34
3.2 Υλοποίηση Ιστοριών χρήστη	34
3.2.1 Ιστορία Χρήστη 3 – Edit user information	34
3.2.2 Ιστορία Χρήστη 4 – Evaluate provider	35
3.2.3 Ιστορία Χρήστη 5 – See evaluations	36
3.2.4 Ιστορία Χρήστη 6 – Search provider	36
3.2.5 Ιστορία Χρήστη 7 – Select provider	37
3.2.6 Ιστορία Χρήστη 8 – Post a service	37
3.2.7 Ιστορία Χρήστη 9 – Delete a service	38
3.2.8 Ιστορία Χρήστη 12 – See order	38
3.2.9 Ιστορία Χρήστη 13 – Delete order	39
Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας	40
Παράρτημα ΙΙ – Ανοιχτά Θέματα	41



Λίστα Σχημάτων

Figure 1: Εφαρμογή προτύπου Proxy Design Pattern.	5
Figure 2: Εφαρμογή Adapter Design Pattern.	6
Figure 3: Εφαρμογή Observer Design Pattern.	7
Figure 4: Μοντέλο δεδομένων User.	8
Figure 5: Παράθεση παραμέτρων και αποκρίσεις του συστήματος.	9
Figure 6: Δοκιμή του endpoint.	10
Figure 7: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.	11
Figure 8: Δοκιμή του endpoint.	12
Figure 9: Μοντέλο δεδομένων Evaluation.	13
Figure 10: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.	13
Figure 11: Δοκιμή του endpoint.	14
Figure 12: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.	15
Figure 13: Δοκιμή του endpoint.	16
Figure 14: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.	17
Figure 15: Δοκιμή του Endpoint.	18
Figure 16: Μοντέλο δεδομένων Evaluation.	18
Figure 17: Μοντέλο δεδομένων Evaluation.	19
Figure 18: Μοντέλο δεδομένων Evaluation.	19
Figure 19: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.	20
Figure 20: Δοκιμή του endpoint.	21
Figure 21: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.	22
Figure 22: Δοκιμή του endpoint.	23
Figure 23: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.	23
Figure 24: Δοκιμή του endpoint.	24
Figure 25: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.	25
Figure 26: Δοκιμή του endpoint.	26
Figure 27: Μοντέλο δεδομένων Evaluation.	27
Figure 28: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.	27
Figure 29: Δοκιμή του endpoint.	28
Figure 30: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.	29
Figure 31: Δοκιμή του endpoint.	30



1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν

1.1 Πρότυπο Σχεδίασης 1

Αναφορά σε ΜΛΑ	ΜΛΑ-5
Κατηγορία Προτύπου Σχεδίασης	Δομικό Πρότυπο
Πρότυπο Σχεδίασης	Proxy Pattern

Με το πρότυπο αυτό, δημιουργείται ένα προσωρινό αντικείμενο αντικαταστάτη μιας λίστας αντικειμένων, ώσπου να προβληθεί το ζητούμενο. Η διαδικασία αυτή εκτελείται όταν το σύστημα δε μπορεί να ανταποκριθεί σε πραγματικό χρόνο, και κρατάει το χρήστη στην αναμονή των αποτελεσμάτων, προβάλλοντάς του μια απεικόνιση.

Το συγκεκριμένο πρότυπο, θα εφαρμόζεται σε 2 σημεία. Το ένα είναι για την εμφάνιση της λίστας των διαθέσιμων υπηρεσιών, και το άλλο για τη λίστα των αξιολογήσεων. Παρακάτω, περιγράφεται η λειτουργία του, στην πρώτη περίπτωση (search&selectService package).

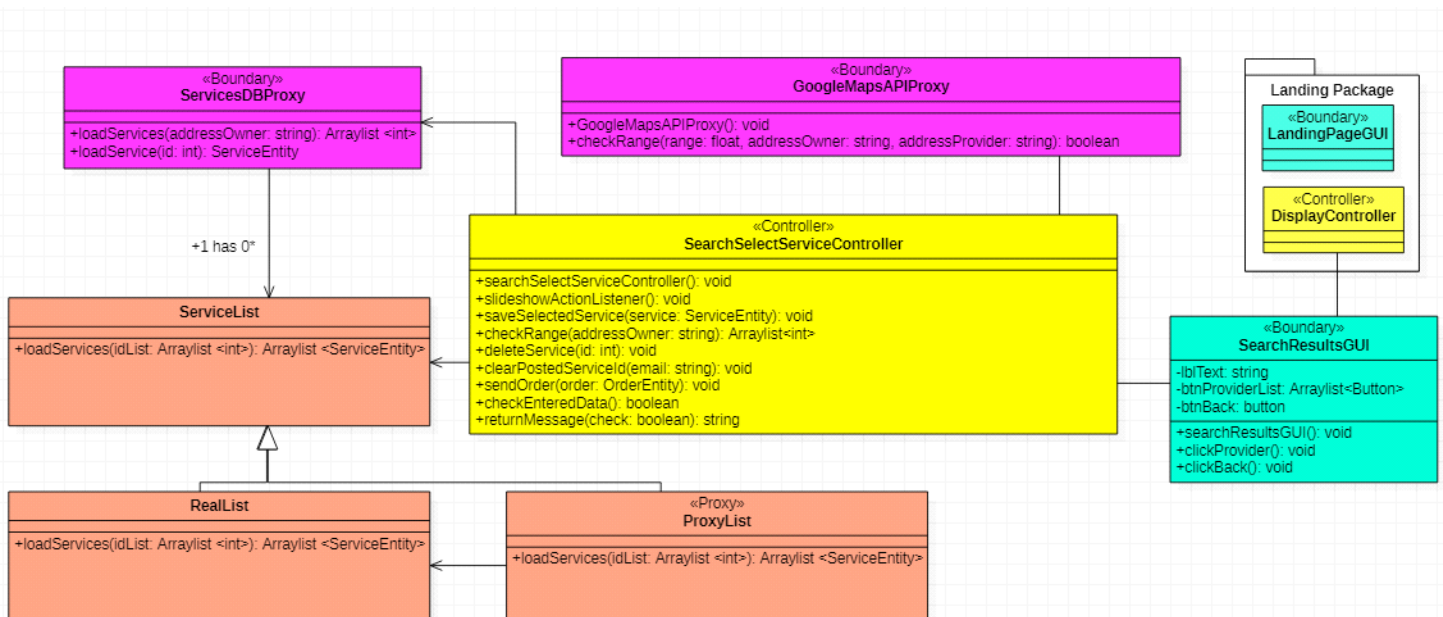


Figure 1: Εφαρμογή προτύπου Proxy Design Pattern.

Κλάσεις:

- ProxyList: Η κλάση που δημιουργεί τον αντικαταστάτη αντικείμενο, ώσπου να φορτωθεί το επιθυμητό.
- RealList: Η κλάση που φορτώνει μέσω των ServicesDBProxy και GoogleMapsAPIProxy, την επιθυμητή λίστα των υπηρεσιών.
- ServiceList: Η abstract κλάση που απεικονίζει την εκάστοτε λίστα (proxy, επιθυμητή), για την εμφάνισή της στο GUI.



1.2 Πρότυπο Σχεδίασης 2

Αναφορά σε ΜΛΑ	ΜΛΑ-1
Κατηγορία Προτύπου Σχεδίασης	Δομικό Πρότυπο
Πρότυπο Σχεδίασης	Adapter Pattern

Το σύστημα πρέπει να είναι λειτουργικό σε διαφορετικά λειτουργικά συστήματα. Για τον λόγο αυτό, χρησιμοποιείται ένα Adapter Pattern, έτσι ώστε ανάλογα με το λειτουργικό της συσκευής που έχει εγκατασταθεί η εφαρμογή, θα εφαρμόζονται οι διαφοροποιήσεις που απαιτεί το λειτουργικό αυτό.

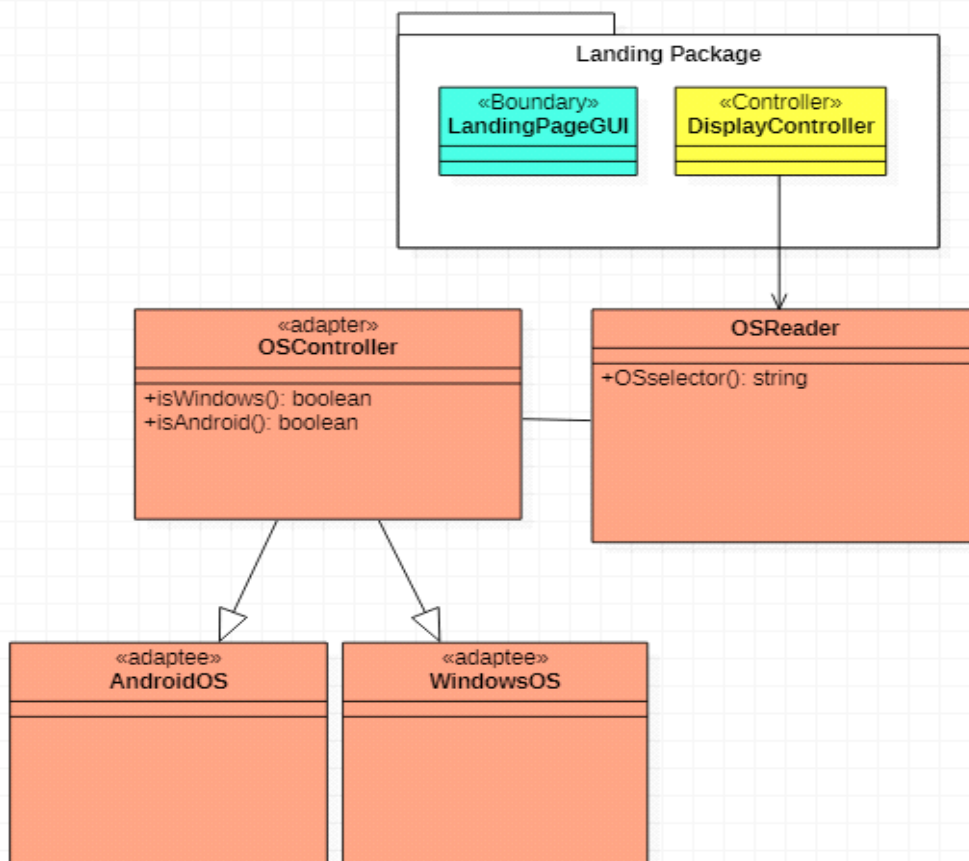


Figure 2: Εφαρμογή Adapter Design Pattern.

Κλάσεις:

- OSReader: Η κλάση που ανιχνεύει το λογισμικό στην εγκατεστημένη συσκευή.
- OSController: Η κλάση που λειτουργεί ως αντάπτορας. Μέσω αυτής επιλέγεται ο τρόπος λειτουργίας της εφαρμογής, ανάλογα με το λειτουργικό.
- AndroidOS / WindowsOS: Οι κλάσεις με τις αντίστοιχες συναρτήσεις τους, για το κάθε λειτουργικό.



1.3 Πρότυπο Σχεδίασης 3

ΝΕΑ ΜΗ-ΛΕΙΤΟΥΡΓΙΚΗ:

(Μ.Λ.Α.-6) <Ενημέρωση Οθονών Διεπαφής>

Το σύστημα θα πρέπει να ενημερώνει τις οθόνες διεπαφής.

Περιγραφή: Το σύστημα μας θα πρέπει να ενημερώνει αυτόματα τις οθόνες διεπαφής όταν αλλάζουν τα δεδομένα στη βάση δεδομένων.

User Priority {5/5}: Είναι σημαντικό για τον χρήστη έτσι ώστε η εφαρμογή του, να είναι πάντα ενημερωμένη με τα ορθά στοιχεία.

Technical Priority {5/5}: Είναι σημαντικό για το σύστημα για να μην παρουσιάζονται σφάλματα στην περίπτωση που τα δεδομένα έχουν αλλάξει.

Αναφορά σε ΜΛΑ	ΜΛΑ-6
Κατηγορία Προτύπου Σχεδίασης	Πρότυπο Συμπεριφοράς
Πρότυπο Σχεδίασης	Observer Pattern

Με το πρότυπο αυτό, παρατηρούνται οι αλλαγές στη βάση δεδομένων και ανάλογα ενημερώνονται οι διεπαφές της εφαρμογής. Πιο συγκεκριμένα, όταν θα γίνεται μια αλλαγή στα στοιχεία ενός χρήστη, αυτόματα ενημερώνεται η βάση δεδομένων, και γίνεται ειδοποίηση σε μια κλάση παρατηρητή για την εκάστοτε αλλαγή. Τέλος, μέσω του παρατηρητή, αλλάζουν οι οθόνες διεπαφής που χρησιμοποιούν τα στοιχεία αυτά, μέσω των αντίστοιχων controller.

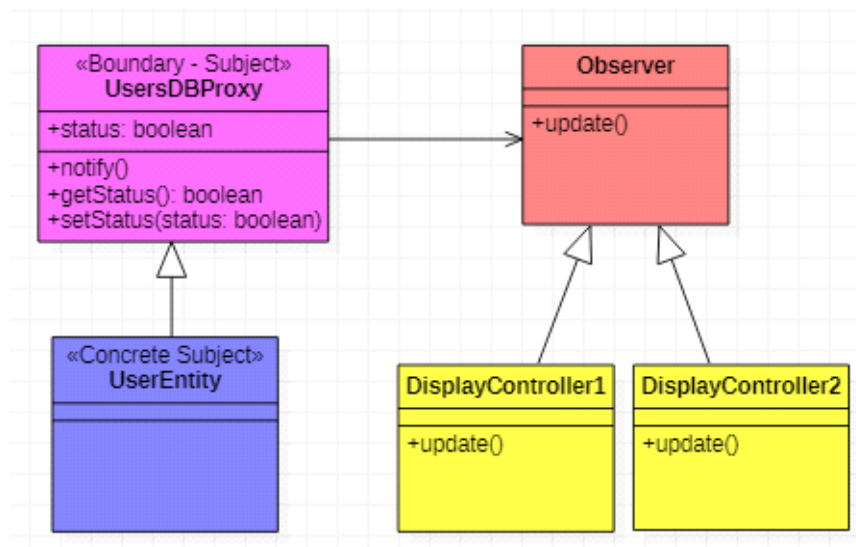


Figure 3: Εφαρμογή Observer Design Pattern.



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 σέρβερ της εφαρμογής.

2.1 Αναγνώριση Πόρων (Resources) Συστήματος

Κλάση BEC	Πόρος REST	Endpoints (HTTP Verbs)
UserEntity	user/{email}	PUT, GET
EvaluationEntity,(OwnerEntity)	user/owner/{email}/evaluations	GET
EvaluationEntity,(OwnerEntity)	user/owner/{email}/evaluations/{evalid}	PUT, DELETE
EvaluationEntity,(ProviderEntity)	user/provider/{email}/evaluations	GET
ServiceEntity,(OwnerEntity)	user/owner/{email}/serviceList	GET
ServiceEntity,(OwnerEntity)	user/owner/{email}/serviceList/{serviceId}	GET, DELETE
ServiceEntity,(ProviderEntity)	user/provider/{email}/serviceList	POST
ServiceEntity,(ProviderEntity)	user/provider/{email}/serviceList/{serviceId}	DELETE
OrderEntity,(ProviderEntity)	user/provider/{email}/order	GET, PUT

2.2 Τεκμηρίωση REST διεπαφής

2.2.1 Πόρος User

2.2.1.1 Μοντέλο δεδομένων User

```
user {  
  fullName      string  
  email         string  
  phone        string  
  address       string  
  category      string  
  photo        > {...}  
}
```

Figure 4: Μοντέλο δεδομένων User



2.2.1.2 Endpoint PUT πόρου User

PUT `/user/{email}` The user must be able to edit their personal information ↩

Parameters

Try it out

Name	Description
email * required	email of the user
string (path)	<input type="text" value="pougar@gmail.com"/>

Responses

Response content type **application/json** ▼

Code	Description
200	OK
	Example Value Model
	<pre>{ "fullName": "string", "email": "string", "phone": "string", "address": "string", "category": "string", "photo": {} }</pre>
404	Error

Figure 5: Παράθεση παραμέτρων και αποκρίσεις του συστήματος.



Responses Response content type **application/json**

Curl

```
curl -X 'PUT' \
'https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/pougar%40gmail.com' \
-H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/pougar%40gmail.com
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "fullName": "string", "email": "string", "phone": "string", "address": "string", "category": "string", "photo": {} }</pre> <p>Download</p> <p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With,Content-Type,Accept,Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 89 content-type: application/json; charset=utf-8 date: Mon,31 May 2021 16:04:35 GMT etag: W/"59-1DjxoiIXXILwrVnNpjlew52+8fk" expires: -1 server: Jetty(9.4.29.v20200521) status: 200 OK via: 1.1 5f9d907c5eee8903615fffc7d8961200.cloudfront.net (CloudFront) x-amz-cf-id: 31XkT0xv9p2jcd-SSNbz0XaM3xcuba6dUt1JXFixycH6gifgAA1uRA== x-amz-cf-pop: OTP50-C1 x-cache: Miss from cloudfront x-powered-by: Express,Phusion Passenger(R) 6.0.8</pre> <p>Request duration</p> <pre>850 ms</pre>

Figure 6: Δοκιμή του endpoint



2.2.1.3 Endpoint GET πόρου User

GET `/user/{email}` The user must be able to view their personal information ↩

Parameters Cancel

Name	Description
email * required	email of the user
string	
(path)	<input type="text" value="pougar@gmail.com"/>

Execute

Responses Response content type application/json ▼

Code	Description
200	OK
	Example Value Model
	<pre>{ "fullName": "string", "email": "string", "phone": "string", "address": "string", "category": "string", "photo": {} }</pre>
404	Error

Figure 7: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.



Responses Response content type **application/json**

Curl

```
curl -X 'GET' \
'https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/pougar%40gmail.com' \
-H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/pougar%40gmail.com
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "fullName": "string", "email": "string", "phone": "string", "address": "string", "category": "string", "photo": {} }</pre> <p>Download</p> <p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With,Content-Type,Accept,Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 89 content-type: application/json; charset=utf-8 date: Mon,31 May 2021 16:05:48 GMT etag: W/"59-1DjxoiDXXILwrVnNpj1ew52+8fk" expires: -1 server: Jetty(9.4.29.v20200521) status: 200 OK via: 1.1 5f9d907c5eee8903615fffc7d8961200.cloudfront.net (CloudFront) x-amz-cf-id: o0eRyWd-22Gky9vuEIM7W_WfpsF1InQR_eavZaCCGhLUfzPENmWr7A== x-amz-cf-pop: OTP50-C1 x-cache: Miss from cloudfront x-powered-by: Express,Phusion Passenger(R) 6.0.8</pre> <p>Request duration</p> <pre>341 ms</pre>

Figure 8: Δοκιμή του endpoint.



2.2.2 Πόρος Evaluation

2.2.2.1 Μοντέλο δεδομένων Evaluation

```
evaluation {
  id                integer($int64)
  emailOwner        string
  emailProvider      string
  serviceType        string
  datePurchased      string($date)
  timePurchased      string($time)
  rating             number
  comment            string
  evaluated           boolean
}
```

Figure 9: Μοντέλο δεδομένων Evaluation.

2.2.2.2 Endpoint GET πόρου Evaluation, για Providers.

The screenshot displays a REST client interface for the GET endpoint `/user/provider/{email}/evaluation`. The URL bar shows the endpoint, and a note indicates that the provider must be able to view all their evaluations. The parameters section includes an 'email' field, which is required and has the value 'pougar@gmail.com'. The responses section shows two responses: a successful 200 response with a JSON body and a 400 error response with a JSON body.

GET `/user/provider/{email}/evaluation` The provider must be able to view all their evaluations

Parameters

Name **Description**

email * required

string (path)

Responses Response content type `application/json`

Code **Description**

200 Success

Example Value | Model

```
[
  {
    "id": 0,
    "emailOwner": "string",
    "emailProvider": "string",
    "serviceType": "string",
    "datePurchased": "2021-05-31",
    "timePurchased": "string",
    "rating": 0,
    "comment": "string",
    "evaluated": true
  }
]
```

400 Error

Example Value | Model

```
{
  "message": "string"
}
```

Figure 10: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.



Responses

Response content type **application/json**

Curl

```
curl -X 'GET' \
'https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/provider/pougar%40gmail.com/evaluation' \
-H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/provider/pougar%40gmail.com/evaluation
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>[{ "id": 0, "emailOwner": "string", "emailProvider": "string", "serviceType": "string", "datePurchased": "2015-07-20", "timePurchased": "string", "rating": 1.5, "comment": "string", "evaluated": true }]</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><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: 151 content-type: application/json; charset=utf-8 date: Mon,31 May 2021 16:09:26 GMT etag: W/"97-VrurK/jDWM5h199X+B7rxSyRttA" expires: -1 server: Jetty(9.4.29.v20200521) status: 200 OK via: 1.1 5f9d907c5eee8903615fffc7d8961200.cloudfront.net (CloudFront) x-amz-cf-id: c0J_h4AgTUofK91c2CEJgxwEgej4BZLjT91u6fUwBIxexT7qa0_01Q== x-amz-cf-pop: OTP50-C1 x-cache: Miss from cloudfront x-powered-by: Express,Phusion Passenger(R) 6.0.8</pre></div><div><div>Request duration</div><div><pre>543 ms</pre></div></div></div>

Figure 11: Δοκιμή του endpoint.



2.2.2.3 Endpoint Get πόρου Evaluation, για Owners.

GET

/user/owner/{email}/evaluation

The owner must be able to view all their available evaluations ↩

Parameters

Try it out

Name	Description
email * required	
string	pougar@gmail.com
(path)	

Responses

Response content type application/json

Code	Description
200	Success
Example Value Model	
<pre>[{ "id": 0, "emailOwner": "string", "emailProvider": "string", "serviceType": "string", "datePurchased": "2021-05-31", "timePurchased": "string", "rating": 0, "comment": "string", "evaluated": true }]</pre>	
400	Error
Example Value Model	
<pre>{ "message": "string" }</pre>	

Figure 12: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος.



Responses Response content type **application/json**

Curl

```
curl -X 'GET' \
'https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/owner/pougar%40gmail.com/evaluation' \
-H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/owner/pougar%40gmail.com/evaluation
```

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "id": 0, "emailOwner": "string", "emailProvider": "string", "serviceType": "string", "datePurchased": "2015-07-20", "timePurchased": "string", "rating": 1.5, "comment": "string", "evaluated": true }]</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: * cache-control: no-cache content-encoding: gzip content-length: 151 content-type: application/json; charset=utf-8 date: Mon,31 May 2021 16:12:04 GMT etag: W/"97-VrurK/j0NM5h199X+B7rxSyRttA" expires: -1 server: Jetty(9.4.29.v20200521) status: 200 OK via: 1.1 5f9d907c5eee8903615fffc7d8961200.cloudfront.net (CloudFront) x-amz-cf-id: 8i2q-ZZQ3kocjzi39j3A1ESwJ7Ksw7x4yYLVKNREoDuGAm9tMBvfvjA== x-amz-cf-pop: OTP50-C1 x-cache: Miss from cloudfront x-powered-by: Express,Phusion Passenger(R) 6.0.8</pre> <p>Request duration</p> <pre>408 ms</pre>

Figure 13: Δοκιμή του endpoint.



2.2.2.4 Endpoint Put πόρου Evaluation, για χρήση Owner.

PUT `/user/owner/{email}/evaluation/{evalid}` The owner must be able to fill an evaluation

FR4 - An Owner evaluates a Provider for his service

Parameters Try it out

Name	Description
evalid * required string (path)	<input type="text" value="evalid"/>
email * required string (path)	<small>the email of the owner</small> <input type="text" value="email - the email of the owner"/>
comments * required string (query)	<small>The owners comment to the provider evaluation</small> <input type="text" value="comments - The owners comment to the provider evalua"/>
rating * required number (query)	<small>The stars given to the provider by an owner</small> <input type="text" value="rating - The stars given to the provider by an owner"/>

Responses Response content type application/json

Code	Description
200	successful evaluation
400	An error has occurred

Figure 14: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος



Responses Response content type: application/json

Curl

```
curl -X 'PUT' \
'https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/owner/pougar%40gmail.com/evaluation/1?comments=good&rating=5' \
-H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/owner/pougar%40gmail.com/evaluation/1?comments=good&rating=5
```

Server response

Code	Details
200	<p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With,Content-Type,Accept,Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json; charset=utf-8 date: Mon,31 May 2021 16:13:51 GMT etag: W/"14-4DhJ6nhrn3soo1wXlJ6FqT6xz/E" expires: -1 server: Jetty(9.4.29.v20200521) status: 200 OK via: 1.1 5f9d907c5eee8903615fffc7d8961200.cloudfront.net (CloudFront) x-amz-cf-id: B43J-120uTBs7-YbtABfcPV007_N4oQ9EdDuxu0QyTZH45Ybi0mcLQ== x-amz-cf-pop: OTP50-C1 x-cache: Miss from cloudfront x-powered-by: Express,Phusion Passenger(R) 6.0.8</pre> <p>Request duration</p> <pre>493 ms</pre>

Figure 15: Δοκιμή του Endpoint.

2.2.3 Πόρος ServiceList

2.2.3.1 Μοντέλο δεδομένων ServiceCreate

```
serviceCreate {
  id                integer($int64)
  serviceType       string
  petType           string
  petSize           string
  days              string
  time              string($time)
  range             number
  price             number
}
```

Figure 16: Μοντέλο δεδομένων Evaluation.



2.2.3.2 Μοντέλο δεδομένων ServiceSearch

```
serviceSearch {  
  id           integer($int64)  
  serviceType  string  
  petType      string  
  petSize      string  
  date         string($date)  
  time         string($time)  
}
```

Figure 17: Μοντέλο δεδομένων Evaluation.

2.2.3.3 Μοντέλο δεδομένων SearchParams

```
searchParams {  
  serviceType  string  
  petType      string  
  petSize      string  
  address      string  
  date         string($date-time)  
  time         string($date-time)  
}
```

Figure 18: Μοντέλο δεδομένων Evaluation.

Χρησιμοποιούμε τα τρία αυτά μοντέλα για την εύκολη μοντελοποίηση των υπηρεσιών που διαχειρίζονται τα services.



2.2.3.4 Endpoint Post πόρου ServiceList, για χρήση Provider

POST `/user/provider/{email}/serviceList`

The provider must be able to create a service listing

Parameters

Try it out

Name	Description
email * required	
string	<input type="text" value="pougar@gmail.com"/>
(path)	
order * required	order object
object	
(body)	<div>Example Value Model</div> <pre>{ "id": 0, "serviceType": "string", "petType": "string", "petSize": "string", "days": "string", "time": "string", "range": 0, "price": 0}</pre>
Parameter content type	
<div>application/json</div>	

Responses

Response content type

application/json

Code	Description
200	Success
400	Error

Figure 19: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος



Responses Response content type **application/json**

Curl

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/provider/pougar%40gmail.com/servicelist' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 0,
    "serviceType": "string",
    "petType": "string",
    "petSize": "string",
    "days": "string",
    "time": "string",
    "range": 0,
    "price": 0
  }'
```

Request URL

```
https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/provider/pougar%40gmail.com/servicelist
```

Server response

Code	Details
200	<p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With,Content-Type,Accept,Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json; charset=utf-8 date: Mon,31 May 2021 16:27:11 GMT etag: W/"14-40hJ6nhm3soo1wXlJ6FqT6xz/E" expires: -1 server: Jetty(9.4.29.v20200521) status: 200 OK via: 1.1 2d1e1e8dc0f3eb7773ec9d89a7d50ce2.cloudfront.net (CloudFront) x-amz-cf-id: D24Bb4qRyiIlganyhKbGEssVTbh5BIoqjPe3-_Get40bDaNksVIX00Q== x-amz-cf-pop: FRA53-C1 x-cache: Miss from cloudfront x-powered-by: Express,Phusion Passenger(R) 6.0.8</pre> <p>Request duration</p> <pre>535 ms</pre>

Figure 20: Δοκιμή του endpoint.



2.2.3.5 Endpoint GET πόρου ServiceList, για χρήση Owner

The image displays the Swagger UI for the endpoint `GET /user/owner/{email}/serviceList`. The left pane shows the responses, and the right pane shows the parameters.

Responses:

Code	Description
200	OK
404	Wrong

Parameters:

Name	Description
email * required	
string	
(path)	
searchParams * required	The owner search for a service
object	
(body)	

Example Value (for searchParams):

```
{  "serviceType": "string",  "petType": "string",  "petSize": "string",  "date": "2021-05-31",  "time": "string"}
```

Figure 21: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος



Responses Response content type: application/json

Curl

```
curl -X 'GET' \
  'https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/owner/pougar%40gmail.com/serviceList' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "serviceType": "string",
    "petType": "string",
    "petSize": "string",
    "address": "string",
    "date": "2021-05-31T16:27:40.622Z",
    "time": "2021-05-31T16:27:40.622Z"
  }'
```

Request URL

```
https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/owner/pougar%40gmail.com/serviceList
```

Figure 22: Δοκιμή του endpoint.

2.2.3.6 Endpoint GET πόρου ServiceList, για χρήση Owner με serviceId

GET /user/owner/{email}/serviceList The owner must be able to choose a service

Parameters Try it out

Name	Description
email * required	
string	email
(path)	
serviceId * required	
integer	serviceId
(path)	

Responses Response content type: application/json

Code	Description
200	OK Example Value Model <pre>{ "id": { "id": 0, "serviceType": "string", "petType": "string", "petSize": "string", "date": "2021-05-31", "time": "string" } }</pre>
404	Wrong

Figure 23: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος



Responses Response content type: application/json

Curl

```
curl -X 'GET' \
'https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/owner/pougar%40gmail.com/serviceList/1' \
-H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/owner/pougar%40gmail.com/serviceList/1
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "id": { "id": 0, "serviceType": "string", "petType": "string", "petSize": "string", "date": "2015-07-20", "time": "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: * cache-control: no-cache content-encoding: gzip content-length: 99 content-type: application/json; charset=utf-8 date: Mon, 31 May 2021 16:31:18 GMT etag: W/"63-h9nswqDzKB6vVs96Zru8AK1hgNM" expires: -1 server: Jetty(9.4.29.v20200521) status: 200 OK via: 1.1 ab39b007ab81966ada6e7fb1536bf377.cloudfront.net (CloudFront) x-amz-cf-id: T27d579MytBytiJmEkzINpJ_BsFRalX1K6jM3VhXuJ2ywk4-_Pz3A== x-amz-cf-pop: FRA53-C1 x-cache: Miss from cloudfront x-powered-by: Express,Phusion Passenger(R) 6.0.8</pre> <p>Request duration</p> <pre>643 ms</pre>

Figure 24: Δοκιμή του endpoint.



2.2.3.7 Endpoint DELETE πόρου ServiceList, για χρήση Provider με serviceId

DELETE

`/user/provider/{email}`
`/serviceList/{serviceId}`

The provider must be able to delete their service

Parameters

Try it out

Name	Description
email * required	<input type="text" value="email"/>
string (path)	
serviceId * required	<input type="text" value="serviceId"/>
integer (path)	

Responses

Response content type: **application/json**

Code	Description
200	OK
404	Error

Figure 25: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος



Responses

Response content type **application/json**

Curl

```
curl -X 'DELETE' \
  'https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/provider/pougar%40gmail.com/serviceList/1' \
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/provider/pougar%40gmail.com/serviceList/1
```

Server response

Code	Details
200	<div><div>Response headers</div><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: Mon,31 May 2021 16:33:42 GMT etag: W/"14-4DhJ6nhrn3soo1wXlJ6FqT6xz/E" expires: -1 server: Jetty(9.4.29.v20200521) status: 200 OK via: 1.1 ab39b007ab81966ada6e7fb1536bf377.cloudfront.net (CloudFront) x-amz-cf-id: Me9aI-1s41dTdAN-e6eXwvsQm5byvpFBfr-tZ7IPA0LvBZT35kDExQ== x-amz-cf-pop: FRA53-C1 x-cache: Miss from cloudfront x-powered-by: Express,Phusion Passenger(R) 6.0.8</pre></div> <div><div>Request duration</div><pre>414 ms</pre></div>

Figure 26: Δοκιμή του endpoint.



2.2.4 Πόρος Order

2.2.4.1 Μοντέλο δεδομένων Order

```
order {  
  ownerName      string  
  ownerAddress   string  
  dateSelected   string($date)  
  timeSelected   string($time)  
  serviceTypeSelected string  
  petTypeSelected string  
  petSizeSelected string  
  price          number  
                example: 3.14  
  id            string  
}
```

Figure 27: Μοντέλο δεδομένων Evaluation.

2.2.4.2 Endpoint GET πόρου Order, για χρήση Provider

Responses Response content type application/json

Code	Description
200	OK Example Value Model <pre>{ "id": { "ownerName": "string", "ownerAddress": "string", "dateSelected": "2021-05-31", "timeSelected": "string", "serviceTypeSelected": "string", "petTypeSelected": "string", "petSizeSelected": "string", "price": 3.14, "id": "string" } }</pre>
404	Failed

Figure 28: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος



Responses Response content type: application/json

Curl

```
curl -X 'GET' \
  'https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/provider/pougar%40gmail.com/order' \
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/provider/pougar%40gmail.com/order
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "id": { "ownerName": "string", "ownerAddress": "string", "dateSelected": "2015-07-20", "timeSelected": "string", "serviceTypeSelected": "string", "petTypeSelected": "string", "petSizeSelected": "string", "price": 3.14, "id": "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: * cache-control: no-cache content-encoding: gzip content-length: 141 content-type: application/json; charset=utf-8 date: Mon, 31 May 2021 16:37:05 GMT etag: W/"8d-dMyctSRT1cYyiHa25pnMQFbUZg4" expires: -1 server: Jetty(9.4.29.v20200521) status: 200 OK via: 1.1 ab39b007ab81966ada6e7fb1536bf377.cloudfront.net (CloudFront) x-amz-cf-id: 5XfadJqCobLwUzuih14k1nx19nIxWtkZ0GnQEdMiFu0F0qWlKj2riA== x-amz-cf-pop: FRA53-C1 x-cache: Miss from cloudfront x-powered-by: Express, Phusion Passenger(R) 6.0.8</pre> <p>Request duration</p> <pre>453 ms</pre>

Figure 29: Δοκιμή του endpoint.



2.2.4.3 Endpoint DELETE πόρου Order, για χρήση Provider

DELETE `/user/provider/{email}/order` The provider must be able to delete his order. ↩

FR12 - A Provider clears an order

Parameters Try it out

Name	Description
email * required	the email of the provider
string (path)	<input type="text" value="email - the email of the provider"/>

Responses Response content type application/json ▼

Code	Description
200	Successful clearance of order
400	An error has occurred

Figure 30: Παράθεση παραμέτρων πόρου και αποκρίσεις του συστήματος



Responses Response content type: application/json

Curl

```
curl -X 'DELETE' \
'https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/provider/pougar%40gmail.com/order' \
-H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/jimsam1/PetMyPet/1.0.0/user/provider/pougar%40gmail.com/order
```

Server response

Code	Details
200	<p>Response headers</p> <pre>access-control-allow-credentials: true access-control-allow-headers: X-Requested-With,Content-Type,Accept,Origin access-control-allow-methods: * access-control-allow-origin: * cache-control: no-cache content-encoding: gzip content-length: 20 content-type: application/json; charset=utf-8 date: Mon,31 May 2021 16:38:51 GMT etag: W/"14-4DhJ6nhn3soo1wXlJ6FqT6xz/E" expires: -1 server: Jetty(9.4.29.v20200521) status: 200 OK via: 1.1 ab39b007ab81966ada6e7fb1536bf377.cloudfront.net (CloudFront) x-amz-cf-id: 0bsXTYfEz05RKnJ20CDzyM73X70eB50ym6DhACnxpp7tVkk2ZRwjfA== x-amz-cf-pop: FRA53-C1 x-cache: Miss from cloudfront x-powered-by: Express,Phusion Passenger(R) 6.0.8</pre> <p>Request duration</p> <pre>289 ms</pre>

Figure 31: Δοκιμή του endpoint.



3. Υλοποίηση Συστήματος με Node-RED

3.1 Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed

Στο κεφάλαιο αυτό περιγράφονται οι REST υπηρεσίες της εφαρμογής PetMyPet μέσω ροών NodeRed.

Υπερσύνδεσμος:

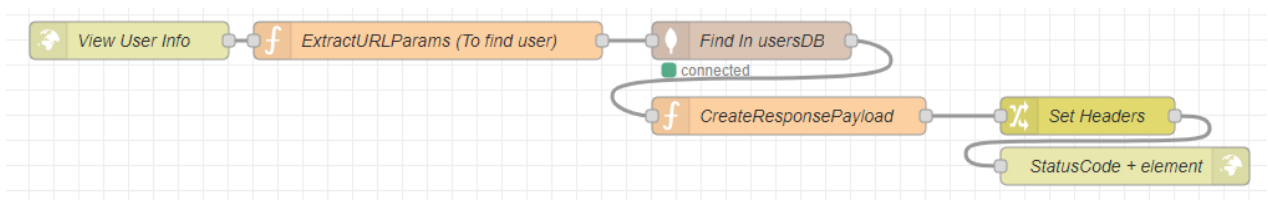
Παρατήρηση 1: Μετά από οδηγίες που δόθηκαν, τα flows για τις ιστορίες χρήστη “Login” και “Signup” δεν υλοποιήθηκαν.

Παρατήρηση 2: Για ευκολία στην υλοποίηση, έχει αλλάξει ελαφρώς η σχεδίαση της εφαρμογής σε σχέση με την πρώτη εργασία. Συγκεκριμένα, τα evaluation entities έχουν πλέον το δικό τους database καθώς και ένα id. Επίσης, αντί να αποδίδονται σαν ολόκληρο αντικείμενο στον κάθε χρήστη, πλέον ο κάθε χρήστης έχει τα id με τα evaluations που του αντιστοιχούν.

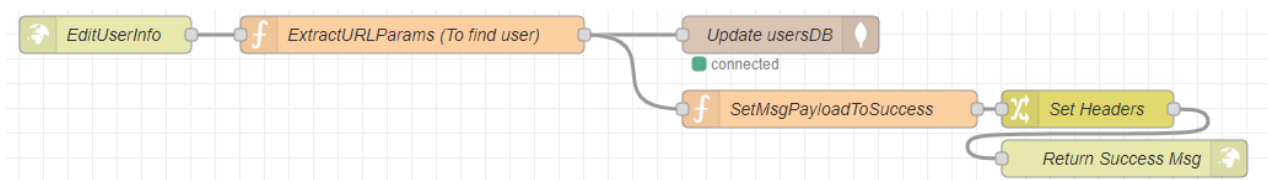
Παρατήρηση 3: Μερικές υπηρεσίες απαιτούσαν την κλήση εξωτερικών API (όπως κάποια πληρωμή ή η χρήση google maps). Σε αυτές τις περιπτώσεις η υλοποίηση έγινε μέχρι το σημείο πριν την κλήση του εξωτερικού API.

3.1.1 Ροές πόρου user

Ροή endpoint [GET] /user/{email}



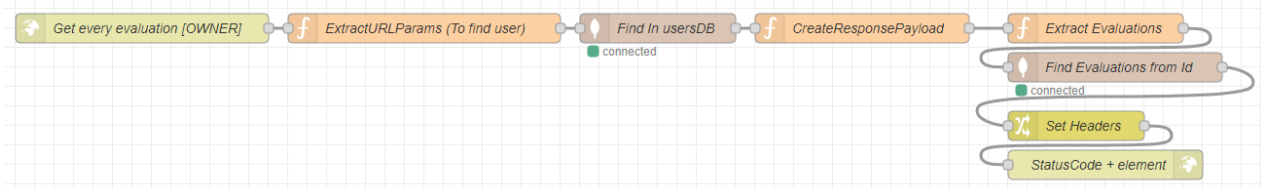
Ροή endpoint [PUT] /user/{email}



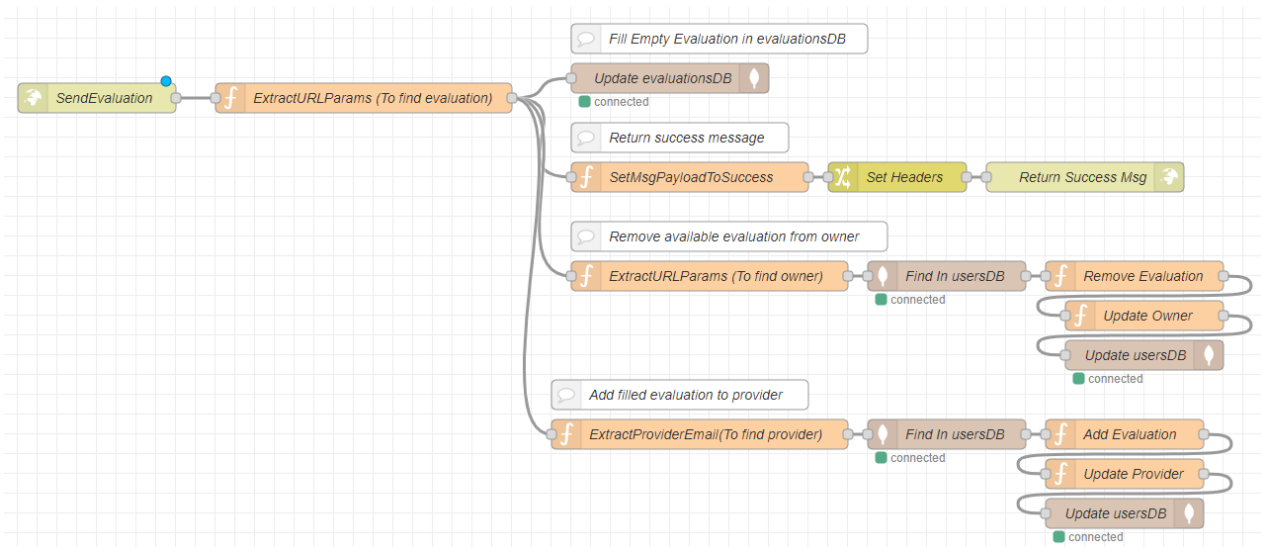


3.1.2 Ροές πόρου evaluation

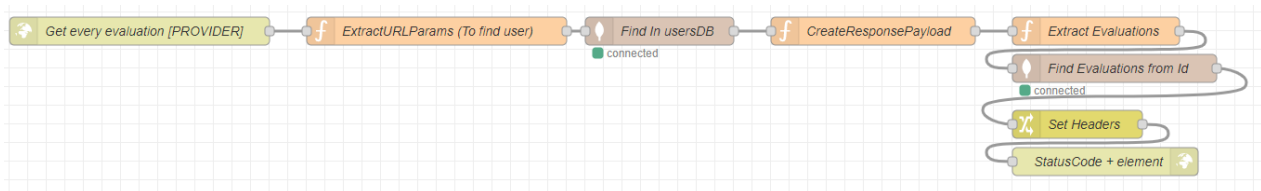
Ροή endpoint [GET] /user/owner/{email}/evaluation



Ροή endpoint [PUT] /user/owner/{email}/evaluation/{evalid}



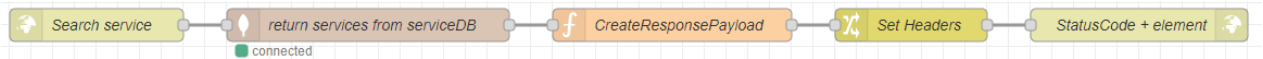
Ροή endpoint [GET] /user/provider/{email}/evaluation



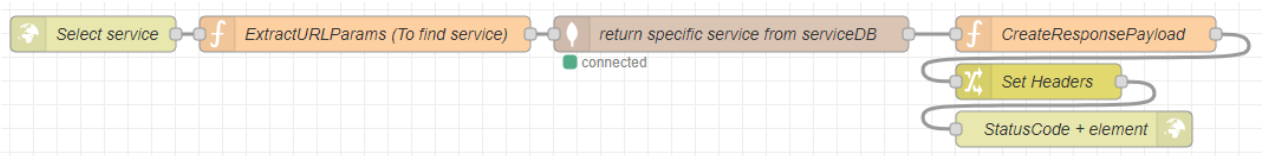


3.1.3 Ροές πόρου serviceList

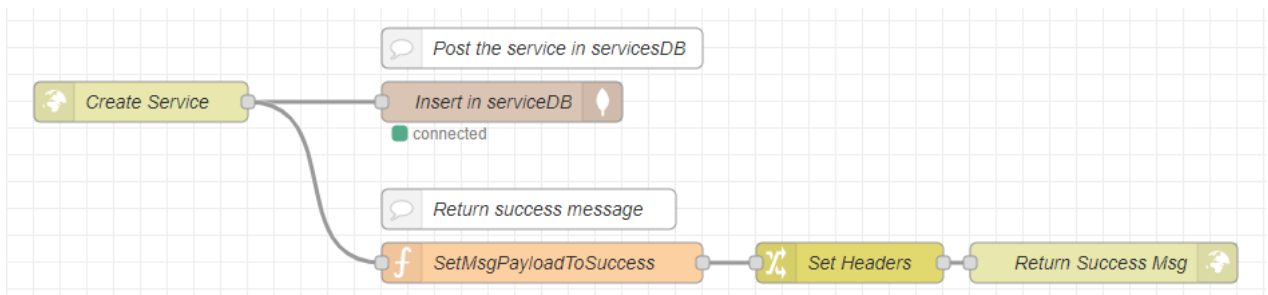
Ροή endpoint [GET] /user/owner/{email}/serviceList



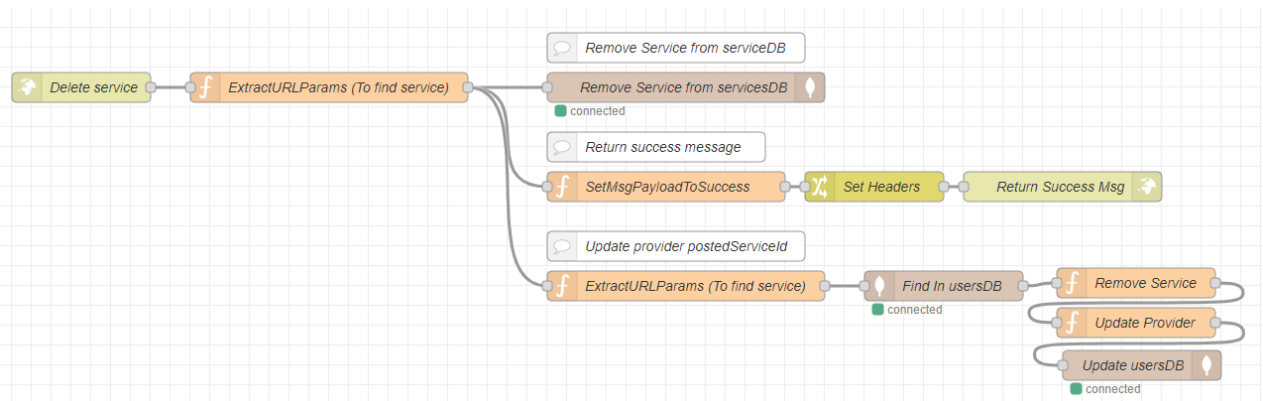
Ροή endpoint [GET] /user/owner/{email}/serviceList/{id}



Ροή endpoint [POST] /user/provider/{email}/serviceList



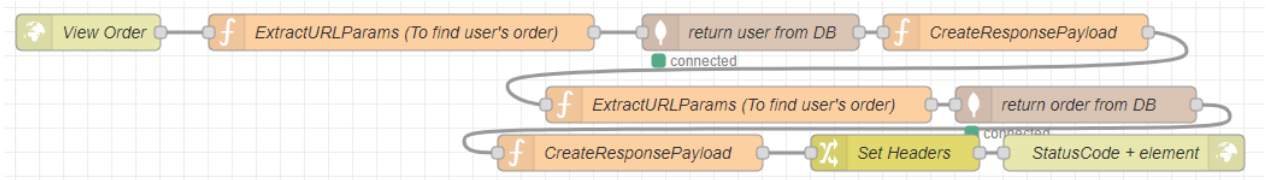
Ροή endpoint [DELETE] /user/provider/{email}/serviceList/{serviceId}



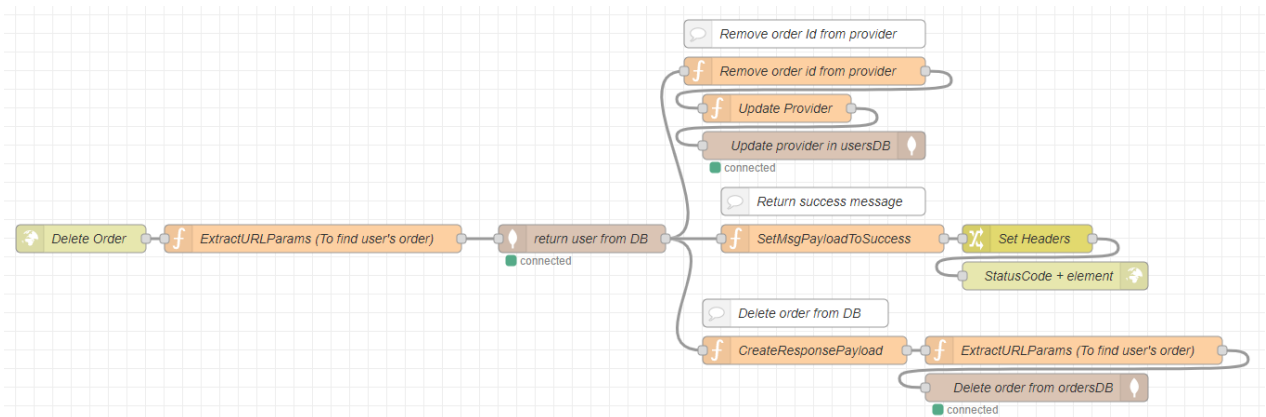


3.1.4 Ροές πόρου order

Ροή endpoint [GET] /user/provider/{email}/order



Ροή endpoint [DELETE] /user/provider/{email}/order

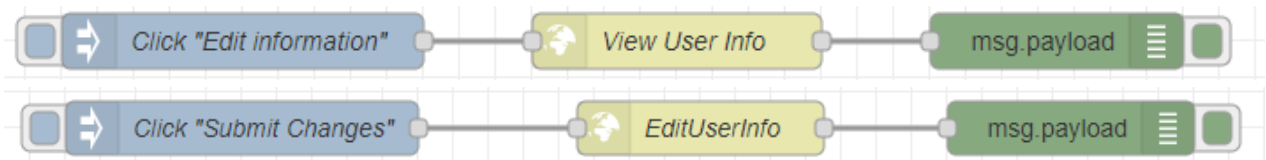


3.2 Υλοποίηση Ιστοριών χρήστη

Περιγράψτε εδώ τις Ροές που υλοποιήσατε για τις ιστορίες χρήστη που καταγράψατε στο πρώτο παραδοτέο (δείτε οδηγίες στο elearning και το σχετικό εργαστήριο).

Για κάθε Ιστορία Χρήστη δημιουργήστε μια υπό-ενότητα στην οποία θα πρέπει να αποτυπώσετε και να περιγράψετε την υλοποίηση, με αντίστοιχη αναφορά στο σενάριο.

3.2.1 Ιστορία Χρήστη 3 – Edit user information



Ο χρήστης αρχικά βρίσκεται στην αρχική οθόνη. Πατάει το κουμπί «Edit Information» για να οδηγηθεί στη σελίδα με τις πληροφορίες του και ταυτόχρονα αποστέλλεται request στο server ώστε να επιστραφούν οι πληροφορίες του και να εμφανιστούν στην οθόνη. Στη συνέχεια, αφού ο χρήστης αλλάξει τις πληροφορίες του, πατάει το κουμπί «Submit changes», ώστε να αποσταλούν στο server και να ενημερωθεί η βάση



δεδομένων.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Click "Edit information"	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής επιστροφής των προσωπικών δεδομένων του χρήστη.
View user info	http-request	Κάνει κλήση προς την υπηρεσία [GET] /user/{email} η οποία επιστρέφει τα δεδομένα του χρήστη από τη βάση δεδομένων users.
msg.payload	debug	Τυπώνει στην κονσόλα τα δεδομένα του χρήστη.
Click "Submit changes"	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής ενημέρωσης των προσωπικών δεδομένων χρήστη.
EditUserInfo	http-request	Κάνει κλήση προς την υπηρεσία [PUT] /user/{email} η οποία ενημερώνει τα δεδομένα του χρήστη στη βάση δεδομένων users.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιτυχίας «success».

3.2.2 Ιστορία Χρήστη 4 – Evaluate provider

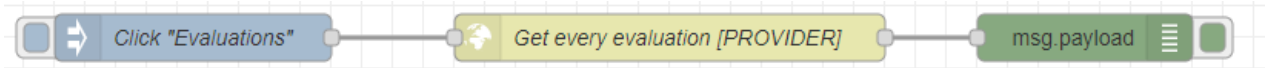


Ο owner αρχικά βρίσκεται στην αρχική οθόνη. Πατάει το κουμπί «Evaluate» για να οδηγηθεί στη σελίδα με τις διαθέσιμες αξιολογήσεις του και ταυτόχρονα αποστέλλεται request στο server ώστε να επιστραφούν οι πληροφορίες των διαθέσιμων αξιολογήσεων του και να εμφανιστούν στην οθόνη. Στη συνέχεια, αφού ο owner επιλέξει μια αξιολόγηση και την τροποποιήσει, πατάει το κουμπί «Submit», ώστε να αποσταλεί στο server και να ενημερωθούν οι βάσεις δεδομένων evaluations και users.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Click "Evaluate"	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής επιστροφής των διαθέσιμων αξιολογήσεων του owner.
Get every evaluation [OWNER]	http-request	Κάνει κλήση προς την υπηρεσία [GET] /user/owner/{email}/evaluation η οποία επιστρέφει τις διαθέσιμες αξιολογήσεις του owner από τη βάση δεδομένων evaluations.
msg.payload	debug	Τυπώνει στην κονσόλα τις διαθέσιμες αξιολογήσεις του owner.
Click "Submit"	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής ενημέρωσης των αξιολογήσεων του owner και του provider.
Send evaluation	http-request	Κάνει κλήση προς την υπηρεσία [PUT] /user/owner/{email}/evaluation/{evalid} η οποία ενημερώνει το evaluation στη βάση δεδομένων evaluations, τις διαθέσιμες αξιολογήσεις του owner στη βάση δεδομένων users και τις αξιολογήσεις του αντίστοιχου provider στη βάση δεδομένων users.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιτυχίας «success».



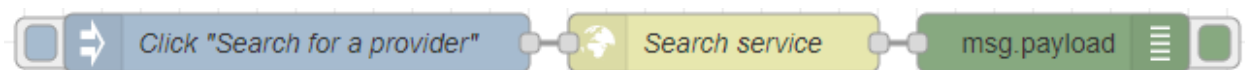
3.2.3 Ιστορία Χρήστη 5 – See evaluations



Ο provider αρχικά βρίσκεται στην αρχική οθόνη. Πατάει το κουμπί «Evaluations» για να οδηγηθεί στη σελίδα με τις αξιολογήσεις του και ταυτόχρονα αποστέλλεται request στο server ώστε να επιστραφούν οι πληροφορίες των αξιολογήσεων του και να εμφανιστούν στην οθόνη.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Click "Evaluations"	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής επιστροφής των αξιολογήσεων του provider.
Get every evaluation [PROVIDER]	http-request	Κάνει κλήση προς την υπηρεσία [GET] /user/provider/{email}/evaluation η οποία επιστρέφει τις αξιολογήσεις του provider από τη βάση δεδομένων evaluations.
msg.payload	debug	Τυπώνει στην κονσόλα τις αξιολογήσεις του provider.

3.2.4 Ιστορία Χρήστη 6 – Search provider

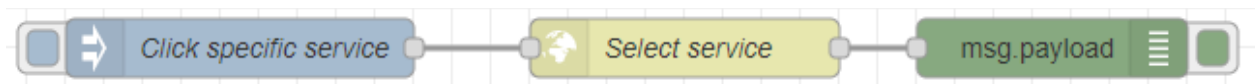


Ο owner αρχικά βρίσκεται στην οθόνη όπου εισάγει τις παραμέτρους για την εύρεση μιας υπηρεσίας που παρέχεται από providers. Αφού έχει εισαγάγει τις παραμέτρους, πατάει το κουμπί «Search for a provider» ώστε να στείλει request στο server για να ψάξει στη βάση δεδομένων services, όπου και θα γίνει αναζήτηση για υπηρεσίες που ταιριάζουν με αυτές και θα επιστραφούν όσες είναι κατάλληλες.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Click "Search for a provider"	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής αναζήτησης υπηρεσιών.
Search service	http-request	Κάνει κλήση προς την υπηρεσία [GET] /user/owner/{email}/serviceList η οποία επιστρέφει τα αντικείμενα service που ταιριάζουν με την αναζήτηση.
msg.payload	debug	Τυπώνει στην κονσόλα τα αντικείμενα service που επιστράφηκαν .



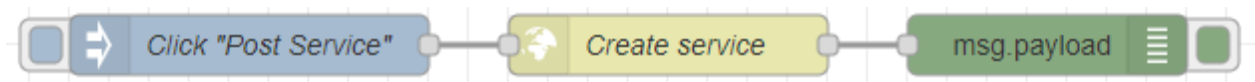
3.2.5 Ιστορία Χρήστη 7 – Select provider



Ο owner αρχικά βρίσκεται στην οθόνη όπου έχουν εμφανιστεί οι υπηρεσίες που παρέχονται από providers. Στη συνέχεια, επιλέγει μια συγκεκριμένη εφαρμογή με το “click specific service”, ώστε να στείλει request στο server, ο οποίος θα του επιστρέψει όλες τις πληροφορίες σχετικά με την υπηρεσία που επέλεξε από τη βάση δεδομένων services.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Click specific service	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής επιστροφής των δεδομένων μιας συγκεκριμένης υπηρεσίας.
Select service	http-request	Κάνει κλήση προς την υπηρεσία [GET] /user/owner/{email}/serviceList/{id} η οποία επιστρέφει τα δεδομένα μιας συγκεκριμένης υπηρεσίας.
msg.payload	debug	Τυπώνει στην κονσόλα το αντικείμενο service που επιστράφηκε .

3.2.6 Ιστορία Χρήστη 8 – Post a service



Ο provider αρχικά βρίσκεται στην οθόνη όπου συμπληρώνει τα στοιχεία της υπηρεσίας που θέλει να παρέχει. Στη συνέχεια, δημοσιεύει την υπηρεσία με το “click Post Service”, ώστε να στείλει request στο server, ο οποίος θα αποθηκεύσει τη νέα υπηρεσία στη βάση δεδομένων services.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Click “Post Service”	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής δημιουργίας μιας νέας υπηρεσίας.
Create service	http-request	Κάνει κλήση προς την υπηρεσία [POST] /user/provider/{email}/serviceList η οποία δημιουργεί τη νέα υπηρεσία στη βάση δεδομένων services.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιτυχίας «success».



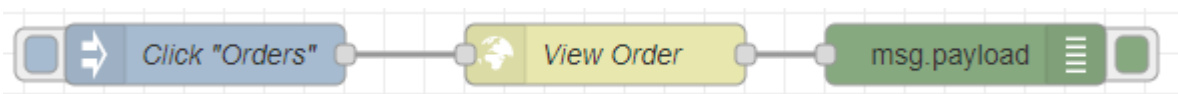
3.2.7 Ιστορία Χρήστη 9 – Delete a service



Ο provider αρχικά βρίσκεται στην οθόνη όπου βλέπει τα στοιχεία της υπηρεσίας που έχει δημοσιεύσει. Στη συνέχεια, διαγράφει την υπηρεσία με το “click Delete Service”, ώστε να στείλει request στο server, ο οποίος θα διαγράψει την υπηρεσία από τη βάση δεδομένων services.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Click “Delete Service”	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής διαγραφής μιας υπηρεσίας.
Create service	http-request	Κάνει κλήση προς την υπηρεσία [DELETE] /user/provider/{email}/serviceList/{serviceId} η οποία διαγράφει την υπηρεσία από τη βάση δεδομένων services.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιτυχίας «success».

3.2.8 Ιστορία Χρήστη 12 – See order

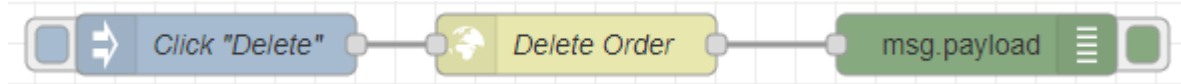


Ο provider αρχικά βρίσκεται στην αρχική οθόνη. Στη συνέχεια, ανοίγει την τρέχουσα παραγγελία του με το “click Orders”, ώστε να στείλει request στο server, ο οποίος θα επιστρέψει την τρέχουσα παραγγελία του από τη βάση δεδομένων orders.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Click “Orders”	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής επιστροφής των δεδομένων της παραγγελίας του provider.
Create service	http-request	Κάνει κλήση προς την υπηρεσία [GET] /user/provider/{email}/order η οποία επιστρέφει τα στοιχεία της παραγγελίας του provider από τη βάση δεδομένων orders.
msg.payload	debug	Τυπώνει στην κονσόλα το αντικείμενο order που επιστράφηκε.



3.2.9 Ιστορία Χρήστη 13 – Delete order



Ο provider αρχικά βρίσκεται στην οθόνη όπου βλέπει την τρέχουσα παραγγελία, την οποία έχει ολοκληρώσει. Στη συνέχεια, διαγράφει την τρέχουσα παραγγελία του με το “click Delete”, ώστε να στείλει request στο server, ο οποίος θα διαγράψει την τρέχουσα παραγγελία του από τη βάση δεδομένων orders και θα ενημερώσει και τον αντίστοιχο provider στη βάση δεδομένων users.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Click “Delete”	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής διαγραφής της παραγγελίας του provider.
Create service	http-request	Κάνει κλήση προς την υπηρεσία [DELETE] /user/provider/{email}/order η οποία διαγράφει την παραγγελία του provider από τη βάση δεδομένων orders και ενημερώνει τα στοιχεία του provider στη βάση δεδομένων users.
msg.payload	debug	Τυπώνει στην κονσόλα το μήνυμα επιτυχίας «success».



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2021

Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας

<Εφόσον χρειάζεται, ορίστε έναν πίνακα Ιχνηλασιμότητας>.



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Α.Π.Θ.

8^ο Εξάμηνο

Άνοιξη 2021

Παράρτημα II – Ανοιχτά Θέματα

<Μια δυναμική λίστα με ανοιχτά θέματα απαιτήσεων, θα δημιουργηθεί στο παράρτημα II>