

# Team Tactix

Where Speed Meets Community!

Panther Racing AUTH App

## Προδιαγραφές Λογισμικού και Υλοποίηση συστήματος

Version 0.6

Παναγιώτης Καρβουνάρης [panakarv@ece.auth.gr](mailto:panakarv@ece.auth.gr)

Έλενα Μαχμουτάι [elenmach@ece.auth.gr](mailto:elenmach@ece.auth.gr)

Παναγιώτης Σιταρίδης [psitarid@ece.auth.gr](mailto:psitarid@ece.auth.gr)

Άννα Τσιτσάνου [tsitsanou@ece.auth.gr](mailto:tsitsanou@ece.auth.gr)

Ημερομηνία 12/06/2023



Μηχανική Λογισμικού Ι  
Τομέας Ηλεκτρονικής και Υπολογιστών  
Τμήμα ΗΜΜΥ

Α.Π.Θ

8<sup>ο</sup> Εξάμηνο  
Άνοιξη 2023

## Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Π. Σιταρίδης	40	psitarid@ece.auth.gr
Π. Καρβουνάρης	40	panakarv@ece.auth.gr
Ε. Μαχμουτάι	40	elenmach@ece.auth.gr
Α. Τσιτσάνου	40	tsitsanou@ece.auth.gr



## Πίνακας Περιεχομένων

<b>Πίνακας Περιεχομένων</b>	<b>2</b>
<b>Λίστα Σχημάτων</b>	<b>3</b>
<b>1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν</b>	<b>6</b>
1.1 Πρότυπο Adapter	6
1.2 Abstract Factory	7
1.3 Remote Proxy	8
<b>2. Αρχιτεκτονική Συστήματος</b>	<b>13</b>
2.1. Αναγνώριση Πόρων (Resources) Συστήματος	13
2.2 Τεκμηρίωση REST διεπαφής	13
2.2.1 Πόρος Weather	14
2.2.1.1 Μοντέλο δεδομένων Weather	14
2.2.1.2 Endpoint πόρου Weather	14
2.2.2 Πόρος Proposal	16
2.2.2.1 Μοντέλο δεδομένων Proposal	16
2.2.2.2 Endpoint POST πόρου Proposal	16
2.2.2.3 Endpoint PUT πόρου Proposal (Αλλαγή της κατάστασης του proposal)	18
2.2.2.4 Endpoint PUT πόρου Proposal(Αποδοχή ή Απόρριψη proposal)	19
2.2.3 Πόρος VehicleSetup	20
2.2.3.1 Μοντέλο δεδομένων VehicleSetup	20
2.2.3.2 Endpoint GET πόρου VehicleSetup	20
2.2.3.3 Endpoint POST πόρου VehicleSetup	22
2.2.3.4 Endpoint PUT πόρου VehicleSetup	24
2.2.4 Πόρος Charts	26
2.2.4.1 Μοντέλο δεδομένων Charts	26
2.2.4.2 Endpoint GET πόρου Charts	27
2.2.4.3 Endpoint POST πόρου Charts	28
2.2.5 Πόρος ChatRoom	31
2.2.5.1 Μοντέλο δεδομένων ChatRoom	31
2.2.5.2 Endpoint POST πόρου ChatRoom	32
2.2.5.3 Endpoint POST πόρου ChatRoom	34
2.2.6 Πόρος User	36
2.2.6.1 Μοντέλο δεδομένων User	36
2.2.6.2 Endpoint POST πόρου User	36
2.2.6.3 Endpoint PUT πόρου User	37
2.2.7 Πόρος profile	39
2.2.7.1 Μοντέλο δεδομένων User	39
2.2.7.2 Endpoint PUT πόρου Profile	39



<b>3. Υλοποίηση Συστήματος με Node-RED</b>	<b>41</b>
3.1. Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed	41
3.1.1. Ροή πόρου Weather	41
3.1.2. Ροές πόρου Proposal	41
3.1.3. Ροές πόρου Vehicle Setup	44
3.1.4. Ροές πόρου ChatRoom	46
3.1.5. Ροές πόρου Chart	47
3.1.6. Ροές πόρου User	49
3.1.7. Ροή Πόρου Profile	50
3.2. Υλοποίηση Ιστοριών χρήστη	51
3.2.1. Ιστορία Χρήστη Get The Weather Forecast	51
3.2.2. Ιστορία Χρήστη Create New Proposal	51
3.2.3. Ιστορία Χρήστη Accept or Decline Proposal	52
3.2.4. Ιστορία Χρήστη Change Proposal Status to Done	52
3.2.5. Ιστορία Χρήστη Create Vehicle Setup	53
3.2.6. Ιστορία Χρήστη Modify Vehicle Setup	53
3.2.7. Ιστορία Χρήστη View Vehicle Setup	54
3.2.8. Ιστορία Χρήστη Create ChatRoom	54
3.2.9. Ιστορία Χρήστη Send Message to Chat	54
3.2.10. Ιστορία Χρήστη Send Chart to Chat	55
3.2.11. Ιστορία Χρήστη View a Chart	55
3.2.12. Ιστορία Χρήστη Assign Roles	56
3.2.13. Ιστορία Χρήστη Upload File	56
3.2.14. Ιστορία Χρήστη Edit Profile Page	57
<b>Παράρτημα Ι – Ανοιχτά Θέματα</b>	<b>58</b>

## Λίστα Σχημάτων

Σχήμα 1: Πρότυπο Adapter	6
Σχήμα 2: Πρότυπο Abstract Factory	7
Σχήμα 3: Πρότυπο Proxy	12
Σχήμα 4: Πόρος Weather	14
Σχήμα 5: Endpoint GET πόρου Weather	15
Σχήμα 6: Πόρος Proposal	16
Σχήμα 7: Endpoint POST πόρου Proposal	17
Σχήμα 8: Endpoint PUT πόρου Proposal	18
Σχήμα 9: Endpoint PUT πόρου Proposal	19



Σχήμα 10: Πόρος VehicleSetup	20
Σχήμα 11: Endpoint GET πόρου VehicleSetup	21
Σχήμα 12: Endpoint POST πόρου VehicleSetup	24
Σχήμα 13: Endpoint PUT πόρου VehicleSetup	26
Σχήμα 14: Πόρος Chart	26
Σχήμα 15: Endpoint GET πόρου Chart	28
Σχήμα 16: Endpoint POST πόρου Chart	30
Σχήμα 17: Πόρος ChatRoom	31
Σχήμα 18: Endpoint POST πόρου Chart (Δημιουργία Chat Room)	33
Σχήμα 19: Endpoint POST πόρου ChatRoom (Αποστολή μηνύματος με βάση το id του ChatRoom)	35
Σχήμα 20: Πόρος User	36
Σχήμα 21: Endpoint POST πόρου User	36
Σχήμα 22: Endpoint PUT πόρου User	38
Σχήμα 23: Πόρος Profile	39
Σχήμα 24: Endpoint POST πόρου Profile	40
<b>Ποές NodeRed:</b>	
Σχήμα 25: GET /user/:userID/weather	41
Σχήμα 26: POST /user/engineer/:userID/proposals	41
Σχήμα 27: PUT /user/chiefEngineer/:userID/proposals/:proposalID	42
Σχήμα 28: PUT /user/handsOnEngineer/:userID/proposals/:proposalID	43
Σχήμα 29: POST /user/chiefEngineer/:userID/vehicleSetup	44
Σχήμα 30: PUT /user/chiefEngineer/:userID/vehicleSetup/:elementID	45
Σχήμα 31: GET /user/:userID/vehicleSetup	45
Σχήμα 32: POST /user/:userID/chat	46
Σχήμα 33: POST /user/:userID/chat/:chatRoomID	47
Σχήμα 34: POST /user/:userID/chart/:chartID/chat/:chatRoomID	47
Σχήμα 35: GET /user/:userID/chart/:chartID	48
Σχήμα 36: PUT /user/chiefEngineer/:userID/adminPanel/:userID	49
Σχήμα 37: POST /user/engineer/:userID/file	49
Σχήμα 38: PUT /user/:userID/profilePage	50
Σχήμα 39: Ιστορία χρήστη "Get the Weather Forecast"	51
Σχήμα 40: Ιστορία χρήστη "Create New Proposal"	51
Σχήμα 41: Ιστορία χρήστη "Accept or Decline Proposal"	52
Σχήμα 42: Ιστορία χρήστη "Change Proposal Status to Done"	52
Σχήμα 43: Ιστορία χρήστη "Create Vehicle Setup"	53
Σχήμα 44: Ιστορία χρήστη "Modify Vehicle Setup"	53
Σχήμα 45: Ιστορία χρήστη "View Vehicle Setup"	54
Σχήμα 46: Ιστορία χρήστη "Create ChatRoom"	54
Σχήμα 47: Ιστορία χρήστη "Send Message to Chat"	55



Σχήμα 48: Ιστορία χρήστη "Send Chart to Chat"	55
Σχήμα 49: Ιστορία χρήστη "View a Chart"	55
Σχήμα 50: Ιστορία χρήστη "Assign Roles"	56
Σχήμα 51: Ιστορία χρήστη "Upload File"	56
Σχήμα 52: Ιστορία χρήστη "Edit profile page"	57



Adapter χρησιμοποιούμε όταν θέλουμε να καταστήσουμε συμβατά δύο ΗΔΗ υπάρχοντα συστήματα. Το δικό σας το σχεδιάζετε τώρα, επομένως πάμε σε bridge.

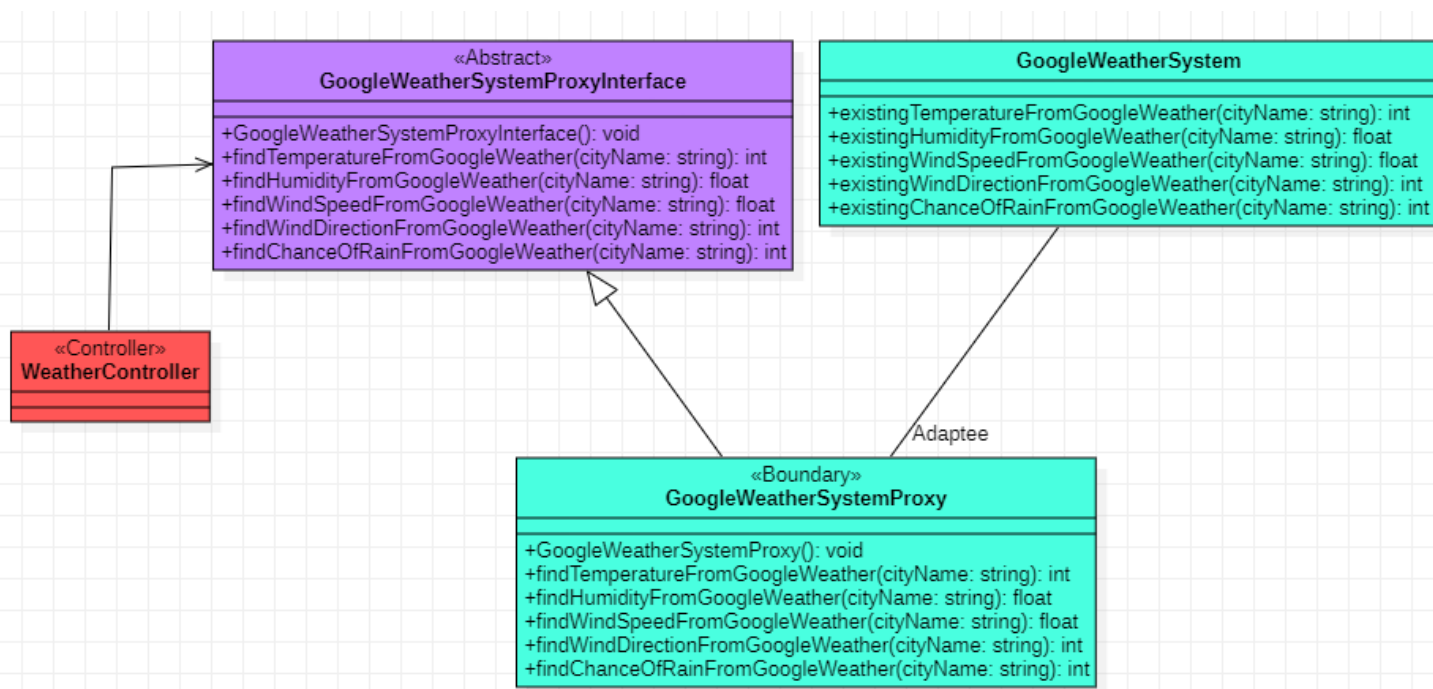
# 1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν

## 1.1 Πρότυπο Adapter

Δομικό πρότυπο. Με το πρότυπο adapter ικανοποιήθηκε η ΜΛΑ “The system must be compatible with Google Weather”. Το adapter επιτρέπει στο σύστημα να επικοινωνεί με εξωτικά συστήματα, στην δική μας περίπτωση η επικοινωνία γίνεται με το Google Weather, ώστε να υπάρχει συνεχή ενημέρωση για τις καιρικές συνθήκες που επικρατούν στην πίστα και τι αναμένεται τις επόμενες ώρες.

### Εφαρμογή του προτύπου σχεδίασης στις κλάσεις:

- WeatherController: Είναι ο ελεγκτής που ελέγχει το weather forecast.
- GoogleWeatherSystemProxyInterface: Λαμβάνει και επεξεργάζεται τα καιρικά δεδομένα.
- GoogleWeatherSystemProxy: Αποτελεί τον κόμβο σύνδεσης μεταξύ των δύο συστημάτων, της δικής μας εφαρμογής και του google weather.
- GoogleWeatherSystem: Αναπαριστά την διεπαφή με το google weather.



Σχήμα 1 Πρότυπο Adapter

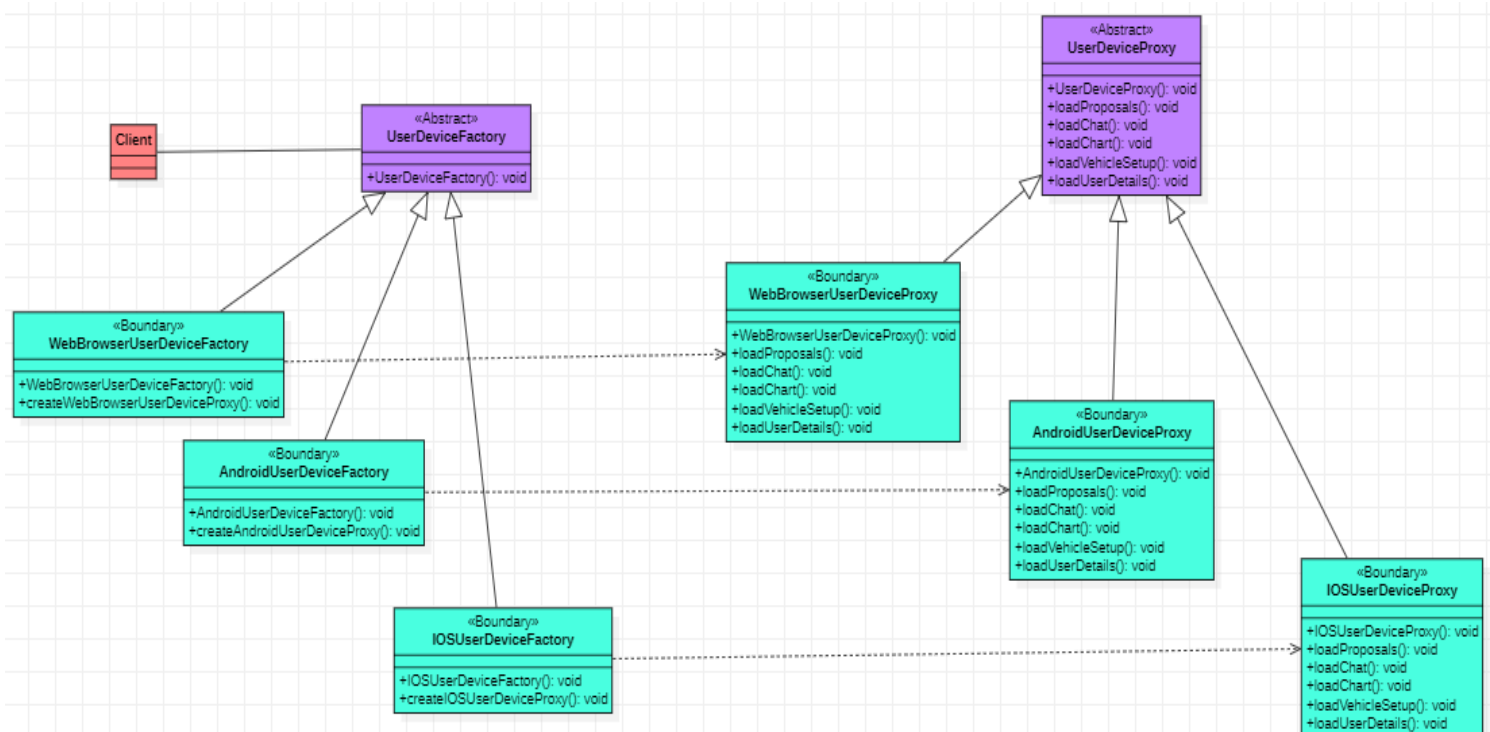


## 1.2 Abstract Factory

Δομικό πρότυπο. Με το πρότυπο abstract factory ικανοποιήθηκε η ΜΛΑ “The application must be cross platform”. Το abstract factory επιτρέπει στο σύστημά μας να λειτουργεί σε διαφορετικά περιβάλλοντα, χωρίς να χρειάζονται διαφοροποιήσεις.

### Εφαρμογή του προτύπου σχεδίασης στις κλάσεις:

- Client: Αντιπροσωπεύει τον χρήστη.
- UserDeviceFactory: Αναπαριστά το abstract class που είναι υπεύθυνο για την δημιουργία των μεθόδων για το σχετικό αντικείμενο.
- WebBrowserUserDeviceFactory: Αναπαριστά την κλάση που είναι υπεύθυνη για την δημιουργία του device proxy σε Web Browser environment.
- AndroidUserDeviceFactory: Αναπαριστά την κλάση που είναι υπεύθυνη για την δημιουργία του device proxy σε Android environment.
- IOSUserDeviceFactory: Αναπαριστά την κλάση που είναι υπεύθυνη για την δημιουργία του device proxy σε IOS environment.
- WebBrowserUserDeviceProxy: Αναπαριστά το product device proxy για web browser environment.
- AndroidUserDeviceProxy: Αναπαριστά το product device proxy για Android environment.
- IOSUserDeviceProxy: Αναπαριστά το product device proxy για IOS environment.
- UserDeviceProxy: Αναπαριστά το abstract class για την δημιουργία του product device proxy.



Σχήμα 2 Πρότυπο Abstract Factory



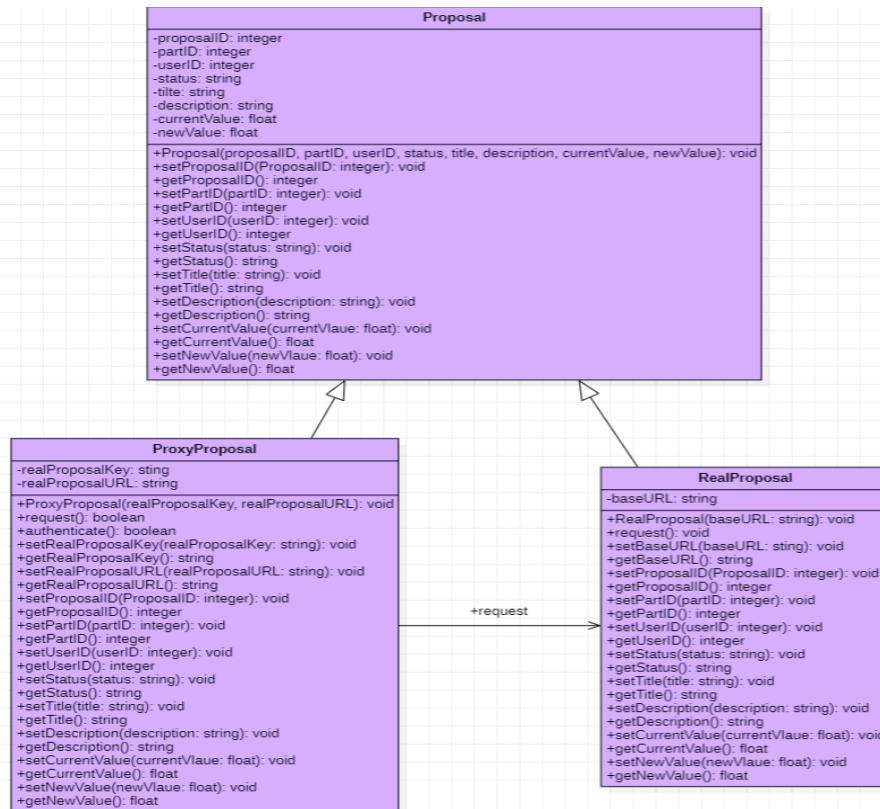


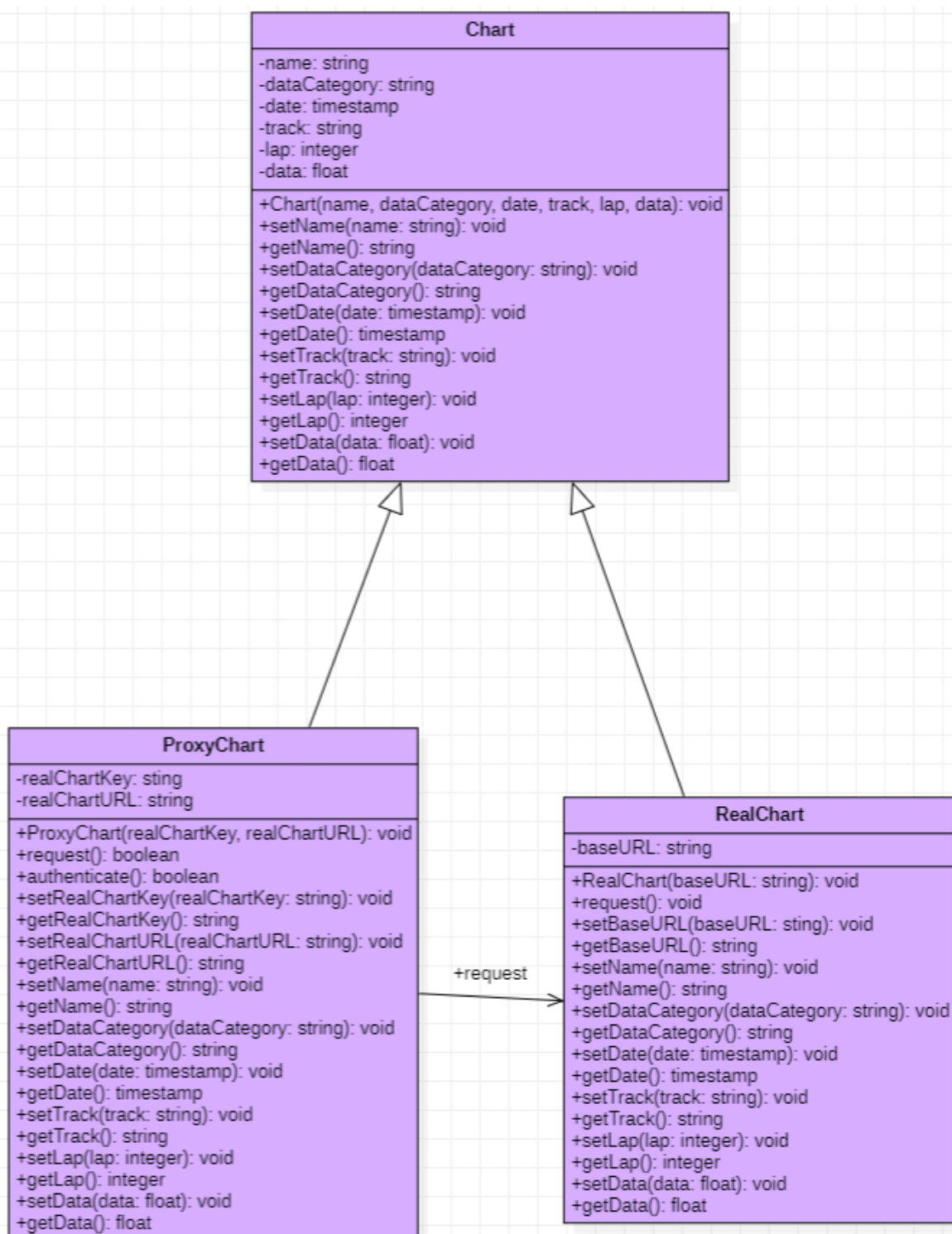
### 1.3 Remote Proxy

Δομικό πρότυπο. Με το πρότυπο remote proxy ικανοποιήθηκαν οι ΜΛΑ “The System must adhere to the European GDPR security protocol” “The system must have a response time of a few ms”. Η υλοποίηση του συγκεκριμένου σχεδιαστικού προτύπου τύπου διαμεσολαβητή ικανοποιεί την ανάγκη της γρήγορης φόρτωσης και παρουσίας της διεπαφής χρήστη. Αυτό συμβαίνει επειδή φορτώνουμε τα “πραγματικά” αντικείμενα (τα οποία είναι ακριβά να δημιουργηθούν) μόνο όταν τα ζητήσει/χρειαστεί ο χρήστης. Το remote proxy επιτρέπει στο σύστημά μας να είναι γρήγορο και να ασφαλές.

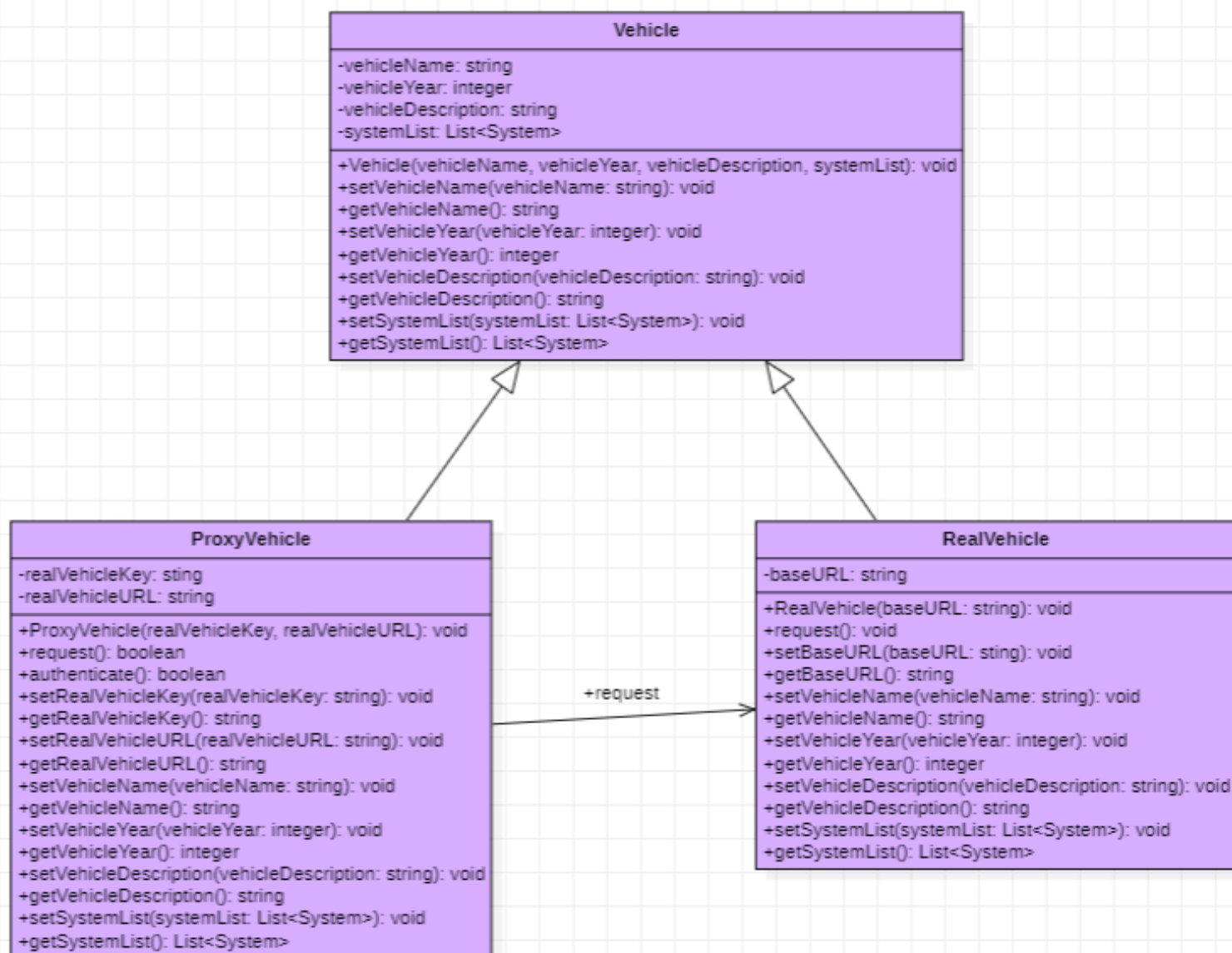
#### Εφαρμογή του προτύπου σχεδίασης στις κλάσεις:

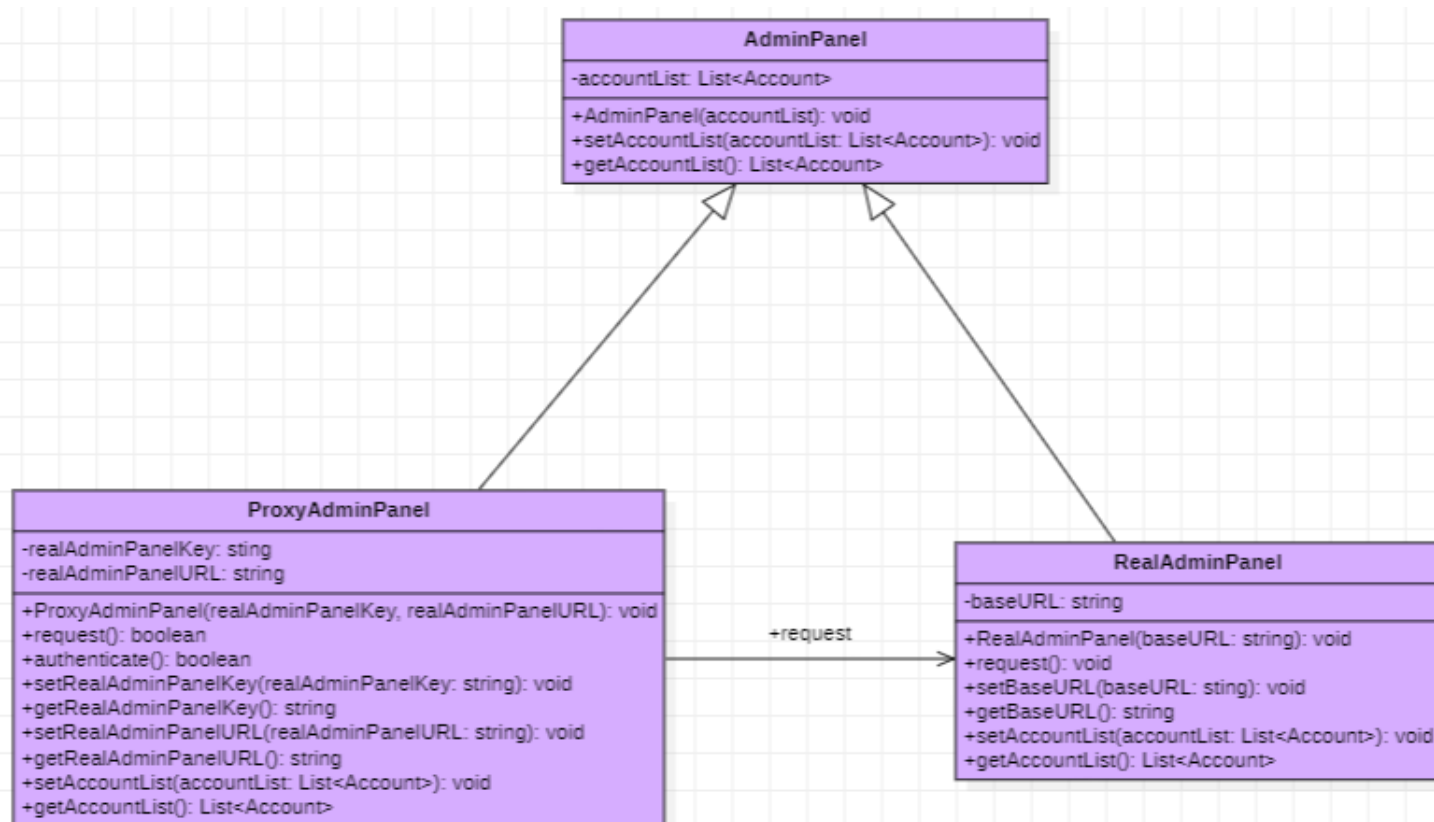
- Proposal/Chart/ChatRoom/Account/Vehicle/AdminPanel: Είναι ένα abstract class που έχει όλες τις μεθόδους που ο χρήστης μπορεί να χρησιμοποιήσει για να αλλάξει το αντικείμενο που βρίσκεται στο database.
- ProxyProposal/ProxyChart/ProxyChatRoom/ProxyAccount/ProxyVehicle/ProxyAdminPanel: Αποτελεί ένα αντίγραφο του πραγματικού αντικειμένου κληρονομεί το abstract class και έχει άμεση σύνδεση με το πραγματικό αντικείμενο μέσω κλειδιού και για λόγους ασφαλείας δηλαδή ταυτοποίησης του χρήστη.
- RealProposal/RealChart/RealChatRoom/RealAccount/RealVehicle/RealAdminPanel: Αποτελεί το πραγματικό αντικείμενο που βρίσκεται στο database και οι πληροφορίες του αλλάζουν μετά από αίτηση και ταυτοποίηση του χρήστη.











Σχήμα 3 Πρότυπο Proxy



## 2. Αρχιτεκτονική Συστήματος

Χρησιμοποιήθηκε το εργαλείο Swaggerhub.

- [Σύνδεσμος](#) για το αρχείο τύπου JSON με τις προδιαγραφές του API.
- [Σύνδεσμος](#) για το swagger στο useReq.

### 2.1. Αναγνώριση Πόρων (Resources) Συστήματος

Κλάση BEC	Πόρος REST	Endpoints (HTTP Verbs)
Weather,(Account Entity)	/user/{userID}/weather	GET
Proposals,(Account Entity)	/user/engineer/{userID}/proposals	POST
Proposals,(Account Entity)	/user/handsOnEngineer/{userID}/proposals/{proposalID}	PUT
VehicleSetUp.(Account Entity)	/user/{userID}/vehicleSetUp	GET
VehicleSetUp.(Account Entity)	/user/chiefEngineer/{userID}/vehicleSetUp/{elementID}	PUT
VehicleSetUp.(Account Entity)	/user/chiefEngineer/{userID}/vehicleSetUp	POST
Charts,(Account Entity)	/user/{userID}/chart/{chartID}	GET
Charts,(Account Entity), Chat	/user/{userID}/chart/{chartID}/chat/{chatRoomID}	PUT
Chat,(Account Entity)	/user/{userID}/chat/{chatRoomID}	POST
Chat,(Account Entity)	/user/{userID}/chat	POST
(Account Entity)	/user/engineer/{userID}/file	POST
Admin Panel(Account Entity)	/user/chiefEngineer/{userID}/adminPanel/{userID}	PUT
Profile(Account Entity)	/user/{userID}/profilePage	PUT
Proposals,(Account Entity)	/user/chiefEngineer/{userID}/proposals/{proposalID}	PUT

### 2.2 Τεκμηρίωση REST διεπαφής



## 2.2.1 Πόρος Weather

### 2.2.1.1 Μοντέλο δεδομένων Weather

```
Weather {  
  temperature integer  
  humidity number  
  windSpeed number  
  windDirection integer  
  chanceOfRain integer  
}
```

Σχήμα 4 Πόρος Weather

### 2.2.1.2 Endpoint πύρου Weather

GET

/user/{userID}/Weather

Gets the weather forecast

All assigned users must be able to view the weather report

Parameters

Cancel

Name	Description
userID <small>required</small>	This is the unique identifier of the user
integer (path)	
<input type="text" value="234"/>	

Execute

Clear

Responses

Curl

curl -X GET "https://virtserver.swaggerhub.com/TSITSANOU\_1/Ergasia/1.0/user/234/Weather" -H "accept: application/json"

Request URL

https://virtserver.swaggerhub.com/TSITSANOU\_1/Ergasia/1.0/user/234/Weather

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{   "temperature": 0,   "humidity": 1.5,   "windSpeed": 1.5,   "windDirection": 0,   "chanceOfRain": 0 }</pre></div><div><div></div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 95 content-type: application/json; charset=utf-8</pre></div></div>





Server response

Code	Details
200	<div>Response body</div> <div><pre>{  "temperature": 0,  "humidity": 1.5,  "windSpeed": 1.5,  "windDirection": 0,  "chanceOfRain": 0}</pre></div> <div>Response headers</div> <div><pre>content-length: 95  content-type: application/json; charset=utf-8</pre></div>

Response

Code	Description	Links
200	<div>Successful operation</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div><pre>{  "temperature": 0,  "humidity": 0,  "windSpeed": 0,  "windDirection": 0,  "chanceOfRain": 0}</pre></div>	No links
500	Server error	No links

Σχήμα 5 Endpoint GET πόρου Weather



## 2.2.2 Πόρος Proposal

### 2.2.2.1 Μοντέλο δεδομένων Proposal

```
proposal {  
  id integer  
  prposalID integer  
  partID integer  
  userID integer  
  status string  
  title string  
  description string  
  currentValue number  
  newValue number  
}
```

Σχήμα 6 Πόρος Proposal

### 2.2.2.2 Endpoint POST πόρου Proposal

**POST** /user/engineer/{userID}/proposals the engineer makes a new proposal

Engineers must be able to make proposals

**Parameters** Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user

Request body required

application/json

Proposal model

```
{  
  "id": 2,  
  "proposalID": 1,  
  "partID": 15,  
  "userID": 2,  
  "status": "pending",  
  "title": "Health check",  
  "description": "change tension and lubrication",  
  "currentValue": 10,  
  "newValue": 20  
}
```

Execute

Clear

**Responses**

Curl

```
curl -X POST "https://virtserver.swaggerhub.com/TSITSANOU_1/Ergasia/1.0/user/engineer/456/proposals" -H "accept: application/json" -H "Content-Type: application/json" -d '{"id":2,"proposalID":1,"partID":15,"userID":2,"status":"pending","title":"Health check","description":"change tension and lubrication","currentValue":10,"newValue":20}'
```

Request URL

```
https://virtserver.swaggerhub.com/TSITSANOU_1/Ergasia/1.0/user/engineer/456/proposals
```



Server response

Code	Details						
201	<p>Response body</p> <pre>{   "id": 0,   "proposalID": 0,   "partID": 0,   "userID": 0,   "status": "string",   "title": "string",   "description": "string",   "currentValue": 1.5,   "newValue": 1.5 }</pre> <p>Response headers</p> <pre>content-length: 117 content-type: application/json; charset=utf-8</pre>						
<p>Responses</p> <table border="1"><thead><tr><th>Code</th><th>Description</th><th>Links</th></tr></thead><tbody><tr><td>201</td><td>New Proposal added</td><td>No links</td></tr></tbody></table> <p>Media type: <input type="text" value="application/json"/></p> <p>Controls Accept header.</p> <p>Example Value   Schema</p> <pre>{   "id": 0,   "proposalID": 0,   "partID": 0,   "userID": 0,   "status": "string",   "title": "string",   "description": "string",   "currentValue": 0,   "newValue": 0 }</pre>		Code	Description	Links	201	New Proposal added	No links
Code	Description	Links					
201	New Proposal added	No links					
400	Invalid value or missing required parameter	No links					

Σχήμα 7 Endpoint POST πόρου Proposal



### 2.2.2.3 Endpoint PUT πόρου Proposal (Αλλαγή της κατάστασης του proposal)

PUT

/user/handsOnEngineer/{userID}  
/proposals/{proposalID}

the hands-on engineer changes the proposal status

Hands-on engineers must be able to change the proposal status to DONE

Parameters

Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user
<b>proposalID</b> * required integer (path)	this is the unique identifier of the proposal

Request body required

application/json

Proposal model

```
{
  "id": 2,
  "proposalID": 1,
  "partID": 15,
  "userID": 2,
  "status": "pending",
  "title": "Health check ",
  "description": "change tension and lubrication",
  "currentValue": 10,
  "newValue": 20
}
```

Execute

Clear

Responses

Curl

```
curl -X PUT
https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/handsOnEngineer/573/proposals/1
-H 'accept: application/json' -H 'Content-Type: application/json' -d '{
  "id": 2, "proposalID": 1, "partID": 15, "userID": 2, "status": "pending", "title": "Health
check ", "description": "change tension and lubrication", "currentValue": 10, "newValue": 20
}'
```

Request URL

https://virtserver.swaggerhub.com/TSITSANOI\_1/Ergasia/1.0/user/handsOnEngineer/573/proposals/1

Server response

Code	Details
200	<div>Response body</div> <pre>{   "id": 0,   "proposalID": 0,   "partID": 0,   "userID": 0,   "status": "string",   "title": "string",   "description": "string",   "currentValue": 1.5,   "newValue": 1.5 }</pre> <div>Response headers</div> <pre>content-length: 117 content-type: application/json;charset=utf-8</pre>

Responses

Code	Description	Links
200	Successful Proposal Status Update	No links
400	Bad Request	No links
404	Not found	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
  "proposalID": 0,
  "partID": 0,
  "userID": 0,
  "status": "string",
  "title": "string",
  "description": "string",
  "currentValue": 0,
  "newValue": 0
}
```

Σχήμα 8 Endpoint PUT πόρου Proposal  
(Αλλαγή της κατάστασης του proposal)



#### 2.2.2.4 Endpoint PUT πόρου Proposal(Αποδοχή ή Απόρριψη proposal)

**PUT** `/user/chiefEngineer/{userID}/proposals/{proposalID}` Accepts or Declines a proposal  
`{proposalID}`

Chief engineer must be able to accept or decline proposals

**Parameters** Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user
<input type="text" value="189"/>	
<b>proposalID</b> * required integer (path)	This is a unique identifier of the proposal
<input type="text" value="1"/>	

**Request body** \* required application/json

Proposal model

```
{  "id": 2,  "proposalID": 1,  "partID": 15,  "userID": 2,  "status": "pending",  "title": "Health check ",  "description": "change tension and lubrication",  "currentValue": 10,  "newValue": 20}
```

**Execute** Clear

Responses

Curl

```
curl -X PUT "https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/chiefEngineer/189/proposals/1" -H "accept: application/json" -H "Content-Type: application/json" -d "{ \"id\":2, \"proposalID\":1, \"partID\":15, \"userID\":2, \"status\": \"pending\", \"title\": \"Health check \", \"description\": \"change tension and lubrication\", \"currentValue\":10, \"newValue\":20 }"
```

Request URL

```
https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/chiefEngineer/189/proposals/1
```

Server response

Code	Details
200	<p>Response body</p> <pre>{  "id": 0,  "proposalID": 0,  "partID": 0,  "userID": 0,  "status": "string",  "title": "string",  "description": "string",  "currentValue": 1.5,  "newValue": 1.5}</pre> <p>Response headers</p> <pre>content-length: 117 content-type: application/json; charset=utf-8</pre>

Responses

Code	Description	Links
200	Successfully Accepted or Declined a Proposal	No links
400	Bad Request	No links
404	Not found	No links

Media type application/json

Controls Accept header.

Example Value | Schema

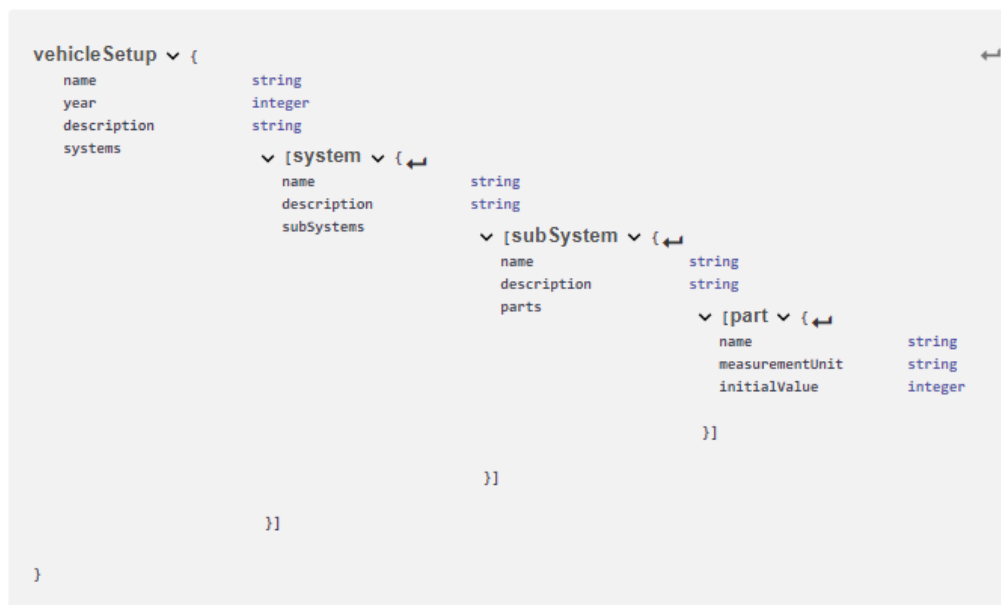
```
{  "id": 0,  "proposalID": 0,  "partID": 0,  "userID": 0,  "status": "string",  "title": "string",  "description": "string",  "currentValue": 0,  "newValue": 0}
```



Σχήμα 9 Endpoint PUT πόρου Proposal  
(Αποδοχή ή Απόρριψη του proposal)

## 2.2.3 Πόρος VehicleSetup

### 2.2.3.1 Μοντέλο δεδομένων VehicleSetup



Σχήμα 10 Πόρος VehicleSetup

### 2.2.3.2 Endpoint GET πόρου VehicleSetup

GET

/user/{userID}/vehicleSetup

View vehicle setup

This endpoint displays vehicle setup details

Parameters

Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user

Execute

Clear

Responses

Curl

curl -X GET "https://virtserver.swaggerhub.com/TSITSANOU\_1/Ergasia/1.0/user/234/vehicleSetup" -H "accept: application/json"

Request URL

https://virtserver.swaggerhub.com/TSITSANOU\_1/Ergasia/1.0/user/234/vehicleSetup



Server response

Code	Details
200	<div>Response body<pre>{   "name": "string",   "year": 0,   "description": "string",   "systems": [     {       "name": "string",       "description": "string",       "subsystems": [         {           "name": "string",           "description": "string",           "parts": [             {               "name": "string",               "measurementUnit": "string",               "initialValue": 0             }           ]         }       ]     }   ] }</pre></div> <div>Response headers<pre>content-length: 146 content-type: application/json;charset=utf-8</pre></div>

Responses

Code	Description	Links
200	Successful operation	No links
<div>Media type application/json</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <pre>{   "name": "string",   "year": 0,   "description": "string",   "systems": [     {       "name": "string",       "description": "string",       "subsystems": [         {           "name": "string",           "description": "string",           "parts": [             {               "name": "string",               "measurementUnit": "string",               "initialValue": 0             }           ]         }       ]     }   ] }</pre>		
400	Invalid input parameter	No links
404	Vehicle setup not found	No links

Σχήμα 11 Endpoint GET πόρον VehicleSetup



### 2.2.3.3 Endpoint POST πύρου VehicleSetup

POST

/user/chiefEngineer/{userID}  
/vehicleSetUp

Create vehicle setup

This endpoint allows the chief engineer to create a new vehicle setup

Parameters

Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user
<input type="text" value="189"/>	

Request body required

application/json

Vehicle Setup model

```
{
  "name": "Black Panther",
  "year": 2023,
  "description": "Racing legend",
  "systems": [
    {
      "name": "Vehicle Geometry",
      "description": "",
      "subSystems": [
        {
          "name": "Geometry Setup",
          "description": "",
          "parts": [
            {
              "name": "Trail",
              "measurementUnit": "mm",
              "initialValue": 78
            },
            {
              "name": "Rear Wing Length",
              "measurementUnit": "mm",
              "initialValue": 540
            }
          ]
        }
      ]
    }
  ]
}
```





Execute

Clear

Responses

Curl

```
curl -X POST "https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/chiefEngineer/189/vehicleSetup" -H "accept: application/json" -H "Content-Type: application/json" -d '{"name": "Black Panther", "year": 2023, "description": "Racing legend", "systems": [{"name": "Vehicle Geometry", "description": "", "subSystems": [{"name": "Geometry Setup", "description": "", "parts": [{"name": "Trail", "measurementUnit": "mm", "initialValue": 78}, {"name": "Rear Wing Length", "measurementUnit": "mm", "initialValue": 540}]}]}
```

Request URL

```
https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/chiefEngineer/189/vehicleSetup
```

Server response

Code

Details

201

Response body

```
{  "name": "string",  "year": 0,  "description": "string",  "systems": [    {      "name": "string",      "description": "string",      "subSystems": [        {          "name": "string",          "description": "string",          "parts": [            {              "name": "string",              "measurementUnit": "string",              "initialValue": 0            }          ]        }      ]    }  ]}
```

Download

Response headers

```
content-length: 146  content-type: application/json; charset=utf-8
```



Responses		
Code	Description	Links
201	Vehicle setup created	No links
<div>Media type application/json</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <pre>{  "name": "string",  "year": 0,  "description": "string",  "systems": [    {      "name": "string",      "description": "string",      "subSystems": [        {          "name": "string",          "description": "string",          "parts": [            {              "name": "string",              "measurementUnit": "string",              "initialValue": 0            }          ]        }      ]    }  ]}</pre>		
400	Invalid action	No links
500	Server error	No links

Σχήμα 12 Endpoint POST πόρου VehicleSetup

#### 2.2.3.4 Endpoint PUT πόρου VehicleSetup

PUT

/user/chiefEngineer/{userID}  
/vehicleSetup/{elementID}

Update vehicle setup

This endpoint allows the chief engineer to update the vehicle setup

Parameters

Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user
<input type="text" value="189"/>	
<b>elementID</b> * required integer (path)	This is the unique identifier of the element
<input type="text" value="10"/>	

Request body required

application/json



# Μηχανική Λογισμικού Ι

## Τομέας Ηλεκτρονικής και Υπολογιστών

### Τμήμα ΗΜΜΥ

Α.Π.Θ

8<sup>ο</sup> Εξάμηνο  
Άνοιξη 2023

Vehicle Setup model

```
"name": "Black Panther",
"year": 2023,
"description": "Racing legend",
"systems": [
  {
    "name": "Vehicle Geometry",
    "description": "",
    "subSystems": [
      {
        "name": "Geometry Setup",
        "description": "",
        "parts": [
          {
            "name": "Trail",
            "measurementUnit": "mm",
            "initialValue": 78
          },
          {
            "name": "Rear Wing Length",
            "measurementUnit": "mm",
            "initialValue": 540
          }
        ]
      }
    ]
  }
]
```

Execute Clear

Responses

Curl

```
curl -X PUT "https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/chiefEngineer/189/vehicleSetup/10" -H "accept: application/json" -H "Content-Type: application/json" -d "{ \"name\": \"Black Panther\", \"year\": 2023, \"description\": \"Racing legend\", \"systems\": [ { \"name\": \"Vehicle Geometry\", \"description\": \"\", \"subSystems\": [ { \"name\": \"Geometry Setup\", \"description\": \"\", \"parts\": [ { \"name\": \"Trail\", \"measurementUnit\": \"mm\", \"initialValue\": 78 }, { \"name\": \"Rear Wing Length\", \"measurementUnit\": \"mm\", \"initialValue\": 540 } ] } ] } ] } }
```

Request URL

https://virtserver.swaggerhub.com/TSITSANOI\_1/Ergasia/1.0/user/chiefEngineer/189/vehicleSetup/10

Server response

Code	Details
200	<p>Response body</p> <pre>{   "name": "string",   "year": 0,   "description": "string",   "systems": [     {       "name": "string",       "description": "string",       "subSystems": [         {           "name": "string",           "description": "string",           "parts": [             {               "name": "string",               "measurementUnit": "string",               "initialValue": 0             }           ]         }       ]     }   ] }</pre> <p>Response headers</p> <pre>content-length: 146 content-type: application/json;charset=utf-8</pre>



Responses		
Code	Description	Links
200	Successful operation	No links
<div>Media type application/json</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <pre>{   "name": "string",   "year": 0,   "description": "string",   "systems": [     {       "name": "string",       "description": "string",       "subSystems": [         {           "name": "string",           "description": "string",           "parts": [             {               "name": "string",               "measurementUnit": "string",               "initialValue": 0             }           ]         }       ]     }   ] }</pre>		
400	Invalid input parameter	No links
404	Vehicle setup not found	No links

Σχήμα 13 Endpoint PUT πόρου VehicleSetup

## 2.2.4 Πόρος Charts

### 2.2.4.1 Μοντέλο δεδομένων Charts

```
chart {
  id          integer
  name        string
  dataCategory string
  date        integer
  track       string
  lap         integer
  data        number
}
```

Σχήμα 14 Πόρος Chart



#### 2.2.4.2 Endpoint GET Πόρου Charts

**GET** `/user/{userID}/chart/{chartID}` View the data in data charts

Returns a data chart based on a single ID

Parameters

Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user
<input type="text" value="234"/>	
<b>chartID</b> * required integer (path)	this is the unique identifier of the chart
<input type="text" value="5"/>	

Execute

Clear

Responses

Curl

```
curl -X GET "https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/234/chart/5" -H "accept: application/json"
```

Request URL

```
https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/234/chart/5
```

Server response

Code	Details
200	<div><div>Response body</div><pre>{  "id": 0,  "name": "string",  "dataCategory": "string",  "date": "2015-07-20",  "track": "string",  "lap": 0,  "data": 1.5}</pre><div><div></div><div>Download</div></div></div> <div><div>Response headers</div><pre>content-length: 99  content-type: application/json;charset=utf-8</pre></div>



Responses		
Code	Description	Links
200	Chart successfully fetched.	No links
<div>Media type application/json</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <pre>{  "id": 0,  "name": "string",  "dataCategory": "string",  "date": 0,  "track": "string",  "lap": 0,  "data": 0}</pre>		
400	Invalid data details. Data don't exist.	No links

Σχήμα 15 Endpoint GET πόρου Chart

#### 2.2.4.3 Endpoint POST πόρου Charts

POST

/user/{userID}/chart/{chartID}/chat/{chatRoomID}

Send data chart to chat

Send a data chart based on a user ID

Parameters

Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user
<b>chartID</b> * required integer (path)	This is the unique identifier of the chart
<b>chatRoomID</b> * required integer (path)	This is the unique identifier of the chatroom

Request body required

application/json



Μηχανική Λογισμικού Ι  
Τομέας Ηλεκτρονικής και Υπολογιστών  
Τμήμα ΗΜΜΥ

Α.Π.Θ

8<sup>ο</sup> Εξάμηνο  
Άνοιξη 2023

User model

```
{
  "chatRoomName": "Team Low Voltage",
  "userList": [
    {
      "userID": 45,
      "name": "John",
      "department": "Low Voltage",
      "role": "Engineer",
      "joined": "2023-06-10T11:28:09.125Z",
      "last-modified": "2023-06-10T11:28:09.125Z"
    },
    {
      "userID": 46,
      "name": "Bob",
      "department": "Low Voltage",
      "role": "Hands-On Engineer",
      "joined": "2023-06-10T11:28:09.125Z",
      "last-modified": "2023-06-10T11:28:09.125Z"
    },
    {
      "userID": 47,
      "name": "Jenny",
      "department": "",
      "role": "Hands-On Engineer",
      "joined": "2023-06-10T11:28:09.125Z",
      "last-modified": "2023-06-10T11:28:09.125Z"
    }
  ],
  "messageList": [
    {
      "text": "string",
      "messageID": 0,
      "userID": 0,
      "timeSent": "2023-06-10T11:28:09.125Z",
      "image": "string",
      "chart": {
        "id": 12,
        "name": "Acceleration Data",
        "dataCategory": "Acceleration",
        "date": "05-05-2023",
        "track": "Aragon",
        "lap": 9,
        "data": 1235236345745799
      }
    }
  ],
  "chatRoomID": 45,
  "chatRoomIcon": "string"
}
```

Execute Clear

Responses

Curl

```
curl -X POST "https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/45/chart/5/chat/45" -H "accept: application/json" -H "Content-Type: application/json" -d '{"chatRoomName":"Team Low Voltage","userList":[{"userID":"45","name":"John","department":"Low Voltage","role":"Engineer","joined":"2023-06-10T11:28:09.125Z","last-modified":"2023-06-10T11:28:09.125Z"}, {"userID":"46","name":"Bob","department":"Low Voltage","role":"Hands-On Engineer","joined":"2023-06-10T11:28:09.125Z","last-modified":"2023-06-10T11:28:09.125Z"}, {"userID":"47","name":"Jenny","department":"","role":"Hands-On Engineer","joined":"2023-06-10T11:28:09.125Z","last-modified":"2023-06-10T11:28:09.125Z"}],"messageList":[{"text":"string","messageID":0,"userID":0,"timeSent":"2023-06-10T11:28:09.125Z","image":"string","chart":{"id":12,"name":"Acceleration Data","dataCategory":"Acceleration","date":"05-05-2023","track":"Aragon","lap":9,"data":1235236345745799}}],"chatRoomID":"45","chatRoomIcon":"string"}'
```

Request URL

https://virtserver.swaggerhub.com/TSITSANOI\_1/Ergasia/1.0/user/45/chart/5/chat/45



Server response

Code	Details
405 <i>Undocumented</i>	Error:

Responses

Code	Description	Links
200	Chart successfully fetched.	No links

Media type

▾

Controls Accept header.

Example Value | Schema

```
{
  "chatRoomName": "string",
  "userList": [
    {
      "userID": 0,
      "name": "string",
      "department": "string",
      "role": "string",
      "joined": "2023-06-12T20:07:49.462Z",
      "last-modified": "2023-06-12T20:07:49.462Z"
    }
  ],
  "messageList": [
    {
      "text": "string",
      "messageID": 0,
      "userID": 0,
      "timesSent": "2023-06-12T20:07:49.462Z",
      "image": "string"
    }
  ],
  "chatRoomID": 0,
  "chatRoomIcon": "string"
}
```

400	Invalid data details. Data don't exist.	No links
-----	---	----------

Σχήμα 16 Endpoint POST πόρου Chart





## 2.2.5 Πόρος ChatRoom

### 2.2.5.1 Μοντέλο δεδομένων ChatRoom



Σχήμα 17 Πόρος ChatRoom



### 2.2.5.2 Endpoint POST πύρου ChatRoom

**POST** `/user/{userID}/chat` Create a chatRoom

Creates a chatRoom for the users to communicate

**Parameters** Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user

**Request body** required application/json

User model

```
{
  "chatRoomName": "Team Low Voltage",
  "userList": [
    {
      "userID": 45,
      "name": "John",
      "department": "Low Voltage",
      "role": "Engineer",
      "joined": "2023-06-10T11:28:09.125Z",
      "last-modified": "2023-06-10T11:28:09.125Z"
    },
    {
      "userID": 46,
      "name": "Bob",
      "department": "Low Voltage",
      "role": "Hands-On Engineer",
      "joined": "2023-06-10T11:28:09.125Z",
      "last-modified": "2023-06-10T11:28:09.125Z"
    },
    {
      "userID": 47,
      "name": "Jenny",
      "department": "",
      "role": "Hands-On Engineer",
      "joined": "2023-06-10T11:28:09.125Z",
      "last-modified": "2023-06-10T11:28:09.125Z"
    }
  ]
}
```

Execute Clear

**Responses**

Curl

```
curl -X POST "https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/234/chat" -H "accept: application/json" -H "Content-Type: application/json" -d '{"chatRoomName": "Team Low Voltage", "userList": [{"userID": 45, "name": "John", "department": "Low Voltage", "role": "Engineer", "joined": "2023-06-10T11:28:09.125Z", "last-modified": "2023-06-10T11:28:09.125Z"}, {"userID": 46, "name": "Bob", "department": "Low Voltage", "role": "Hands-On Engineer", "joined": "2023-06-10T11:28:09.125Z", "last-modified": "2023-06-10T11:28:09.125Z"}, {"userID": 47, "name": "Jenny", "department": "", "role": "Hands-On Engineer", "joined": "2023-06-10T11:28:09.125Z", "last-modified": "2023-06-10T11:28:09.125Z"}]}'
```

Request URL

```
https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/234/chat
```



Request URL

https://virtserver.swaggerhub.com/TSITSANOI\_1/Ergasia/1.0/user/234/chat

Server response

Code Details

200

Response body

```
{
  "chatRoomName": "string",
  "userList": [
    {
      "userID": 0,
      "name": "string",
      "department": "string",
      "role": "string",
      "joined": "2015-07-20T15:49:04-07:00",
      "last-modified": "2015-07-20T15:49:04-07:00"
    }
  ],
  "messageList": [
    {
      "text": "string",
      "messageID": 0,
      "userID": 0,
      "timeSent": "2015-07-20T15:49:04-07:00",
      "image": "string"
    }
  ],
  "chatRoomID": 0,
  "chatRoomIcon": "string"
}
```

Response headers

```
content-length: 201
content-type: application/json;charset=utf-8
```

Responses

Code	Description	Links
200	Chat successfully created.	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "chatRoomName": "string",
  "userList": [
    {
      "userID": 0,
      "name": "string",
      "department": "string",
      "role": "string",
      "joined": "2023-06-12T20:26:45.292Z",
      "last-modified": "2023-06-12T20:26:45.292Z"
    }
  ],
  "messageList": [
    {
      "text": "string",
      "messageID": 0,
      "userID": 0,
      "timeSent": "2023-06-12T20:26:45.292Z",
      "image": "string"
    }
  ],
  "chatRoomID": 0,
  "chatRoomIcon": "string"
}
```

400

Invalid chatRoom name or Error: Empty user list.

No links

Σχήμα 18 Endpoint POST πόρου Chat



(Δημιουργία Chat Room)

### 2.2.5.3 Endpoint POST πύρου ChatRoom

**POST** `/user/{userID}/chat/{chatRoomID}` send message to chat

Send message based on a chatroom ID

**Parameters** Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user
	<input type="text" value="234"/>
<b>chatRoomID</b> * required integer (path)	This is the unique identifier of the chatroom
	<input type="text" value="1"/>

**Request body** required application/json

User model

```
{
  "chatRoomName": "Racing Team",
  "userList": [
    {
      "userID": 234,
      "name": "Anna",
      "department": "brakes",
      "role": "engineer",
      "joined": "2023-06-09T12:42:03.740Z",
      "last-modified": "2023-06-09T12:42:03.740Z"
    }
  ],
  "messageList": [
    {
      "text": "hello",
      "messageID": 3,
      "userID": 2,
      "timeSent": "2023-06-09T12:42:03.740Z",
      "image": "string"
    }
  ],
  "chatRoomID": 1,
  "chatRoomIcon": "string"
}
```

**Execute** **Clear**

Responses

Curl

```
curl -X POST "https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/234/chat/1" -H "accept: application/json" -H "Content-type: application/json" -d "{ \"chatRoomName\": \"Racing Team\", \"userList\": [{ \"userID\": 234, \"name\": \"Anna\", \"department\": \"brakes\", \"role\": \"engineer\", \"joined\": \"2023-06-09T12:42:03.740Z\", \"last-modified\": \"2023-06-09T12:42:03.740Z\" }], \"messageList\": [{ \"text\": \"hello\", \"messageID\": 3, \"userID\": 2, \"timeSent\": \"2023-06-09T12:42:03.740Z\", \"image\": \"string\" }], \"chatRoomID\": 1, \"chatRoomIcon\": \"string\" }"
```

Request URL

```
https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/234/chat/1
```



Server response

Code	Details
200	<p>Response body</p> <pre>{   "chatRoomName": "string",   "userList": [     {       "userID": 0,       "name": "string",       "department": "string",       "role": "string",       "joined": "2015-07-20T15:49:04-07:00",       "last-modified": "2015-07-20T15:49:04-07:00"     }   ],   "messageList": [     {       "text": "string",       "messageID": 0,       "userID": 0,       "timeSent": "2015-07-20T15:49:04-07:00",       "image": "string"     }   ],   "chatRoomID": 0,   "chatRoomIcon": "string" }</pre> <p>Response headers</p> <pre>content-length: 201 content-type: application/json;charset=utf-8</pre>

Responses		
Code	Description	Links
200	<p>Message send successfully.</p> <p>Media type</p> <p><input type="text" value="application/json"/></p> <p>Controls Accept header.</p> <p>Example Value   Schema</p> <pre>{   "chatRoomName": "string",   "userList": [     {       "userID": 0,       "name": "string",       "department": "string",       "role": "string",       "joined": "2023-06-12T20:34:40.251Z",       "last-modified": "2023-06-12T20:34:40.251Z"     }   ],   "messageList": [     {       "text": "string",       "messageID": 0,       "userID": 0,       "timeSent": "2023-06-12T20:34:40.251Z",       "image": "string"     }   ],   "chatRoomID": 0,   "chatRoomIcon": "string" }</pre>	No links
400	Invalid action	No links
500	Server error	No links

Σχήμα 19 Endpoint POST πόρου ChatRoom  
(Αποστολή μηνύματος με βάση το id του ChatRoom)



## 2.2.6 Πόρος User

### 2.2.6.1 Μοντέλο δεδομένων User

```
user {  
  userID      integer  
  name        string  
  department  string  
  role        string  
  joined      string($date-time)  
  last-modified string($date-time)  
}
```

Σχήμα 20 Πόρος User

### 2.2.6.2 Endpoint POST πόρου User

**POST** `/user/engineer/{userID}/file` Uploads a file containing data from the sensors

Engineers must be able to upload a file containing data from the sensors

Parameters

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user

userID - This is the unique identifier of the us

Request body required

text/plain

Example Value | Schema

string

Responses

Code	Description	Links
201	New file added	No links
400	Invalid action	No links
500	Server error	No links

Σχήμα 21 Endpoint POST πόρου User



### 2.2.6.3 Endpoint PUT πύρου User

**PUT** `/user/chiefEngineer/{userID}/adminPanel/{userID}` Assign roles

The chief engineer must be able to edit the roles

**Parameters** Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user
<input type="text" value="234"/>	

**Request body** required application/json

User model

```
{
  "userID": 257,
  "name": "Anna",
  "department": "brakes",
  "role": "hands-on engineer",
  "joined": "2023-06-12T20:54:54.733Z",
  "last-modified": "2023-06-12T20:54:54.733Z"
}
```

Execute Clear



Responses

Curl

```
curl -X PUT "https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/chiefEngineer/234/adminPanel/234" -H "accept: application/json" -H "Content-Type: application/json" -d '{"userID":257,"name":"Anna","department":"brakes","role":"hands-on engineer","joined":"2023-06-12T20:54:54.733Z","last-modified":"2023-06-12T20:54:54.733Z"}'
```

Request URL

https://virtserver.swaggerhub.com/TSITSANOI\_1/Ergasia/1.0/user/chiefEngineer/234/adminPanel/234

Server response

Code Details

200

Response body

```
{  "userID": 0,  "name": "string",  "department": "string",  "role": "string",  "joined": "2015-07-20T15:49:04-07:00",  "last-modified": "2015-07-20T15:49:04-07:00"}
```

Response headers

```
content-length: 120  content-type: application/json; charset=utf-8
```

Responses		
Code	Description	Links
200	Successful Changed Users Role	No links
	<div>Media type application/json</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <pre>{  "userID": 0,  "name": "string",  "department": "string",  "role": "string",  "joined": "2023-06-12T20:55:50.140Z",  "last-modified": "2023-06-12T20:55:50.140Z"}</pre>	
404	User not found	No links
500	Server Error	No links

Σχήμα 22 Endpoint PUT πόρου User





## 2.2.7 Πόρος profile

### 2.2.7.1 Μοντέλο δεδομένων User

```
profile {  
  username      string  
  role          string  
  department    string  
  description   string  
  googleLink    string($uri)  
  githubLink    string($uri)  
  linkedinLink  string($uri)  
  profileImage  string($uri)  
}
```

Σχήμα 23 Πόρος Profile

### 2.2.7.2 Endpoint PUT πόρου Profile

**PUT** /user/{userID}/profilePage Edit Profile Page

All assigned users must be able to edit their Personal Profile page

**Parameters** Cancel

Name	Description
<b>userID</b> * required integer (path)	This is the unique identifier of the user
<input type="text" value="234"/>	

**Request body** \* required application/json

**User Profile**

```
{  
  "username": "Panagiotiskarv",  
  "role": "Default",  
  "department": "Cleaning",  
  "description": "I am a very good student",  
  "googleLink": "link",  
  "githubLink": "link",  
  "linkedinLink": "link",  
  "profileImage": "link"  
}
```

Execute Clear



Responses

Curl

```
curl -X PUT "https://virtserver.swaggerhub.com/TSITSANOI_1/Ergasia/1.0/user/234/profilePage" -H "accept: application/json" -H "Content-Type: application/json" -d '{"username": "\PanagiotisKarv\","\role": "\Default","\department": "\Cleaning","\description": "\I am a very good student","\googleLink": "\link","\githubLink": "\link","\linkedinLink": "\link","\profileImage": "\link"}'
```

Request URL

https://virtserver.swaggerhub.com/TSITSANOI\_1/Ergasia/1.0/user/234/profilePage

Server response

Code Details

200

Response body

```
{  "username": "string",  "role": "string",  "department": "string",  "description": "string",  "googleLink": "string",  "githubLink": "string",  "linkedinLink": "string",  "profileImage": "string"}
```

Response headers

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

Responses		
Code	Description	Links
200	Successful Changed User's Profile	No links
<div>Media type application/json</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <pre>{  "username": "string",  "role": "string",  "department": "string",  "description": "string",  "googleLink": "string",  "githubLink": "string",  "linkedinLink": "string",  "profileImage": "string"}</pre>		
400	Invalid image size	No links
404	User not found	No links
500	Server Error	No links

Σχήμα 24 Endpoint POST πόρου Profile



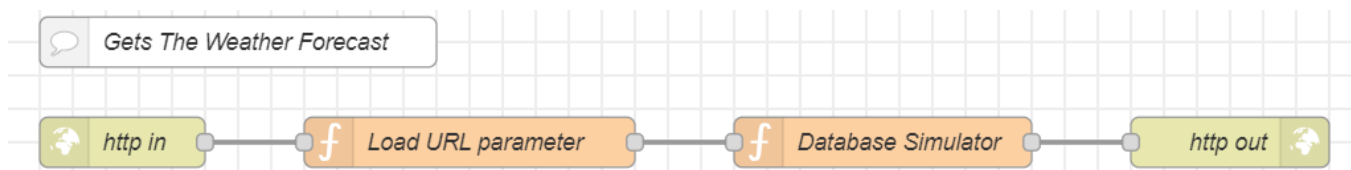
### 3. Υλοποίηση Συστήματος με Node-RED

#### 3.1. Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed

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

##### 3.1.1. Ροή πόρου Weather

Ροή Endpoint GET /user/:userID/weather



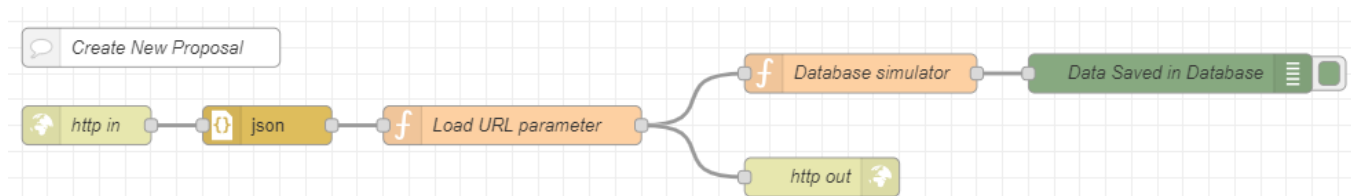
Σχήμα 25 Ροή NodeRed που αντιστοιχεί στο API call GET /user/:userID/weather

Η ροή που υλοποιεί την υπηρεσία, που επιστρέφει το δελτίο καιρού στην τοποθεσία του χρήστη.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint GET /user/:userID/weather για τη δημιουργία του web service που επιστρέφει το δελτίο καιρού .
Load URL parameter	function	Συνάρτηση που δημιουργεί μια μεταβλητή userID και την θέτει ίση με τη παράμετρο userID του path. Μετά την προσθέτει τη νέα μεταβλητή στο msg.payload.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το δελτίο καιρού στο msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.

##### 3.1.2. Ροές πόρου Proposal

Ροή Endpoint POST /user/engineer/:userID/proposals



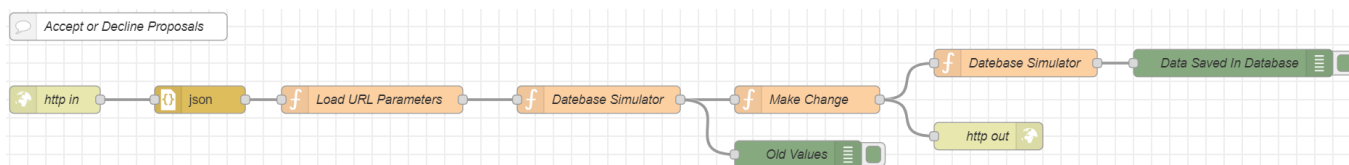
Σχήμα 26 Ροή NodeRed που αντιστοιχεί στο API call POST /user/engineer/:userID/proposals

Η ροή που υλοποιεί την υπηρεσία, η οποία είναι υπεύθυνη για την δημιουργία μιας νέας πρότασης από έναν χρήστη engineer.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint POST /user/engineer/:userID/proposals για τη δημιουργία του web service που δημιουργεί ένα proposal .
json	json	Μετατρέπει το JSON string που δέχεται από τον κόμβο inject σε JavaScript object representation.
Load URL parameter	function	Συνάρτηση που δημιουργεί μια μεταβλητή userID, την θέτει ίση με τη παράμετρο userID του path και την προσθέτει τη νέα μεταβλητή στο msg.topic. Έπειτα, προσθέτει το JavaScript object σε μια μεταβλητή proposal στο msg.payload.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το msg.
Data saved in Database	debug	Εκτυπώνει το περιεχόμενο του msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.

### Ροή Endpoint PUT /user/chiefEngineer/:userID/proposals/:proposalID



Σχήμα 27 Ροή NodeRed που αντιστοιχεί στο API call PUT /user/chiefEngineer/:userID/proposals/:proposalID

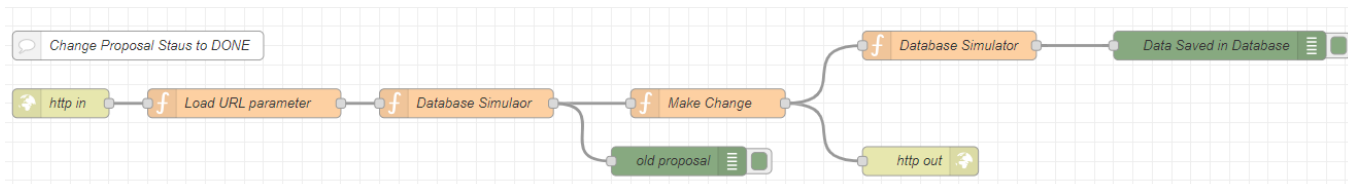
Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει σε ένα χρήστη chief engineer να αποδεχτεί ή να απορρίψει μία πρόταση.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint PUT /user/chiefEngineer/:userID/proposals/:proposalID για τη δημιουργία του web service που αλλάζει το status ενός proposal .
json	json	Μετατρέπει το JSON string που δέχεται από τον κόμβο inject σε JavaScript object representation.
Load URL parameter	function	Συνάρτηση που δημιουργεί δύο μεταβλητές userID, proposalID, τις θέτει ίσες με τις παραμέτρους userID, proposalID του path αντίστοιχα και τις προσθέτει στο msg.payload. Έπειτα, προσθέτει το JavaScript object σε μια μεταβλητή status στο msg.topic.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το παλιό αντικείμενο proposal με το συγκεκριμένο proposalID.
Old Values	debug	Εκτυπώνει το παλιό αντικείμενο.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Make Change	function	Αλλάζει τη τιμή της μεταβλητής status του παλιού αντικειμένου στη τιμή της μεταβλητής status.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το msg.
Data saved in Database	debug	Εκτυπώνει το περιεχόμενο του msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.

### Ροή Endpoint PUT /user/handsOnEngineer/:userID/proposals/:proposalID



Σχήμα 28 Ροή NodeRed που αντιστοιχεί στο API call PUT /user/handsOnEngineer/:userID/proposals/:proposalID

Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει σε ένα χρήστη hands-on engineer να αλλάξει την κατάσταση μιας πρότασης σε DONE.

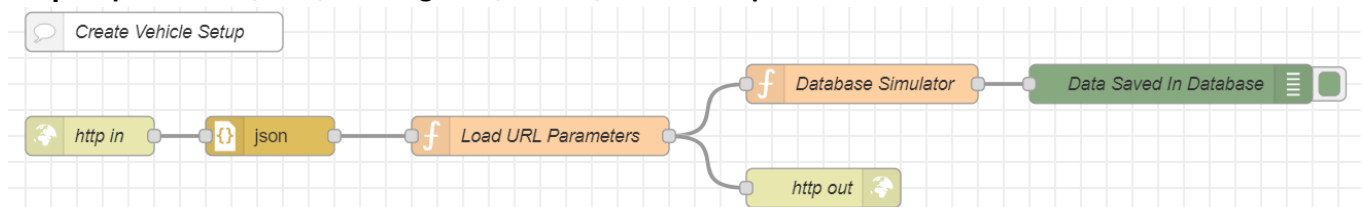
Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint PUT /user/handsOnEngineer/:userID/proposals/:proposalID για τη δημιουργία του web service που αλλάζει το status ενός proposal .
json	json	Μετατρέπει το JSON string που δέχεται από τον κόμβο inject σε JavaScript object representation.
Load URL parameter	function	Συνάρτηση που δημιουργεί δύο μεταβλητές userID, proposalID, τις θέτει ίσες με τις παραμέτρους userID, proposalID του path αντίστοιχα και τις προσθέτει στο msg.payload. Έπειτα, προσθέτει το JavaScript object σε μια μεταβλητή newStatus στο msg.topic.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το παλιό αντικείμενο proposal με το συγκεκριμένο proposalID.
Old Values	debug	Εκτυπώνει το παλιό αντικείμενο.
Make Change	function	Αλλάζει τη τιμή της μεταβλητής status του παλιού αντικειμένου στη τιμή της μεταβλητής newStatus.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το msg.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Data saved in Database	debug	Εκτυπώνει το περιεχόμενο του msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.

### 3.1.3. Ροές πόρου Vehicle Setup

Ροή Endpoint POST /user/chiefEngineer/:userID/vehicleSetup



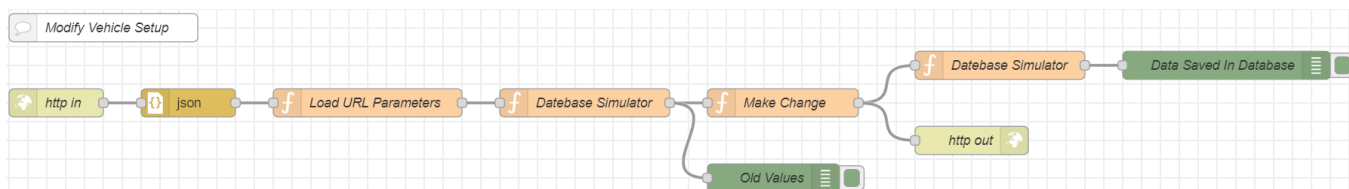
Σχήμα 29 Ροή NodeRed που αντιστοιχεί στο API call POST /user/chiefEngineer/:userID/vehicleSetup

Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει σε ένα χρήστη chief engineer να δημιουργήσει ένα νέο vehicle set up.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint POST /user/chiefEngineer/:userID/vehicleSetup για τη δημιουργία του web service που δημιουργεί ένα νέο vehicle set up.
json	json	Μετατρέπει το JSON string που δέχεται από τον κόμβο inject σε JavaScript object representation.
Load URL parameter	function	Συνάρτηση που δημιουργεί μια μεταβλητή userID, την θέτει ίση με τη παράμετρο userID του path και την προσθέτει τη νέα μεταβλητή στο msg.topic. Έπειτα, προσθέτει το JavaScript object σε μια μεταβλητή vehicleSetup στο msg.payload.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το msg.
Data saved in Database	debug	Εκτυπώνει το περιεχόμενο του msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.



### Ροή Endpoint PUT /user/chiefEngineer/:userID/vehicleSetUp/:elementID

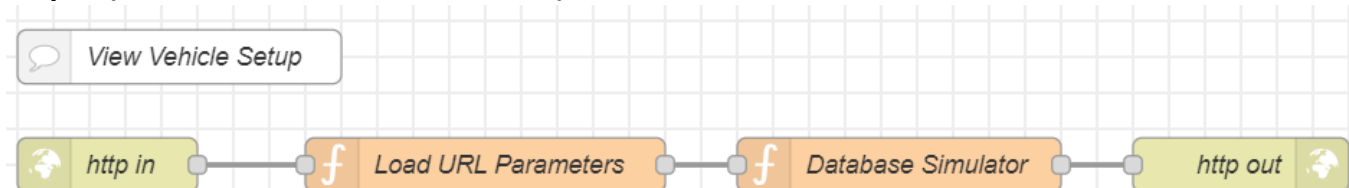


Σχήμα 30 Ροή NodeRed που αντιστοιχεί στο API call PUT /user/chiefEngineer/:userID/vehicleSetup/:elementID

Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει σε ένα χρήστη chief engineer να αλλάξει το value ενός element στο vehicle set up.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint PUT /user/chiefEngineer/:userID/vehicleSetUp/:elementID για τη δημιουργία του web service που αλλάζει το value ενός element του vehicle set up.
json	json	Μετατρέπει το JSON string που δέχεται από τον κόμβο inject σε JavaScript object representation.
Load URL parameter	function	Συνάρτηση που δημιουργεί δύο μεταβλητές userID, elementID, τις θέτει ίσες με τις παραμέτρους userID, elementID του path αντίστοιχα και τις προσθέτει στο msg.payload. Έπειτα, προσθέτει το JavaScript object σε μια μεταβλητή newValue στο msg.topic.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το παλιό αντικείμενο element με το συγκεκριμένο elementID.
Old Values	debug	Εκτυπώνει το παλιό αντικείμενο.
Make Change	function	Αλλάζει τη τιμή της μεταβλητής initialValue του παλιού αντικείμενου στη τιμή της μεταβλητής newValue.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το msg.
Data saved in Database	debug	Εκτυπώνει το περιεχόμενο του msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.

### Ροή Endpoint GET /user/:userID/vehicleSetUp



Σχήμα 31 Ροή NodeRed που αντιστοιχεί στο API call GET /user/:userID/vehicleSetup

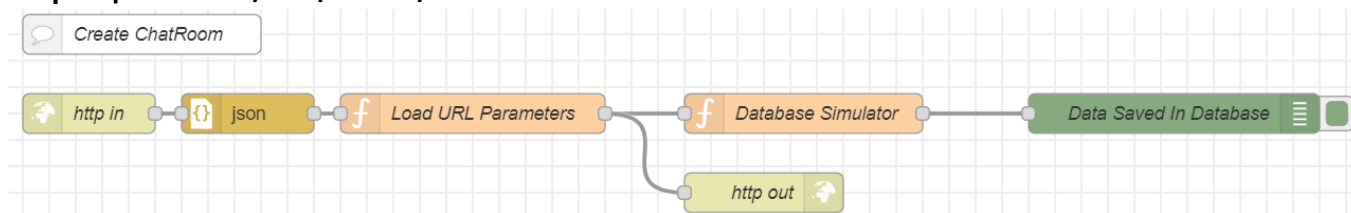


Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει στους χρήστες να δουν το vehicle set up.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint GET /user/:userID/vehicleSetUp για τη δημιουργία του web service που επιστρέφει το vehicle set up.
Load URL parameter	function	Συνάρτηση που δημιουργεί μια μεταβλητή userID και την θέτει ίση με τη παράμετρο userID του path. Μετά την προσθέτει τη νέα μεταβλητή στο msg.payload.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το vehicle set up στο msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.

### 3.1.4. Ροές πόρου ChatRoom

Ροή Endpoint POST /user/:userID/chat



Σχήμα 32 Ροή NodeRed που αντιστοιχεί στο API call POST /user/:userID/chat

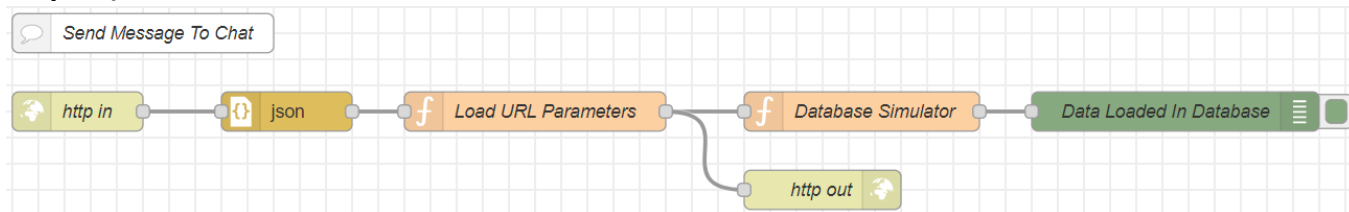
Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει στους χρήστες να δημιουργήσουν ένα νέο chat room.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint POST /user/:userID/chat για τη δημιουργία του web service που δημιουργεί ένα νέο chat room.
json	json	Μετατρέπει το JSON string που δέχεται από τον κόμβο inject σε JavaScript object representation.
Load URL parameter	function	Συνάρτηση που δημιουργεί μια μεταβλητή userID, την θέτει ίση με τη παράμετρο userID του path και την προσθέτει τη νέα μεταβλητή στο msg.topic. Έπειτα, προσθέτει το JavaScript object σε μια μεταβλητή ChatRoom στο msg.payload.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το msg.
Data saved in Database	debug	Εκτυπώνει το περιεχόμενο του msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.





### Ροή Endpoint POST /user/:userID/chat/:chatRoomID



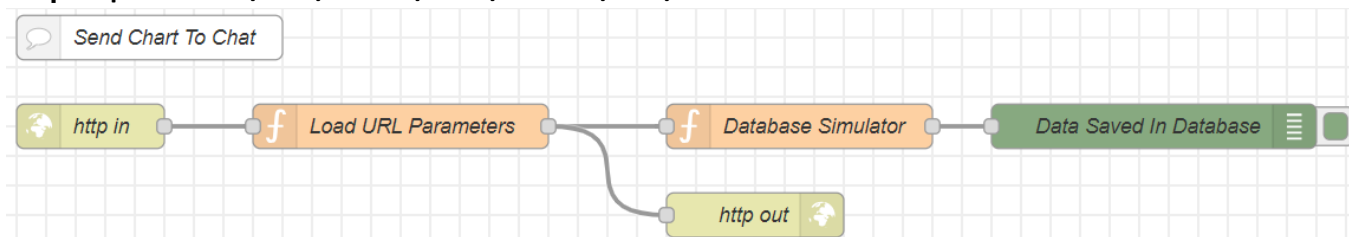
Σχήμα 33 Ροή NodeRed που αντιστοιχεί στο API call POST /user/:userID/chat/:chatRoomID

Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει τους χρήστες να δημιουργήσουν και να στείλουν μήνυμα στο επιλεγμένο ChatRoom.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint POST /user/:userID/chat/:chatRoomID για τη δημιουργία του web service που δημιουργεί ένα νέο μήνυμα στο Chat room.
json	json	Μετατρέπει το JSON string που δέχεται από τον κόμβο inject σε JavaScript object representation.
Load URL parameter	function	Συνάρτηση που δημιουργεί δύο μεταβλητές userID, chatRoomID, τις θέτει ίσες με τις παραμέτρους userID, chatRoomID του path αντίστοιχα και τις προσθέτει στο msg.topic. Έπειτα, προσθέτει το JavaScript object σε μια μεταβλητή Message στο msg.payload.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το msg.
Data saved in Database	debug	Εκτυπώνει το περιεχόμενο του msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.

### 3.1.5. Ροές πόρου Chart

#### Ροή Endpoint POST /user/:userID/chart/:chartID/chat/:chatRoomID



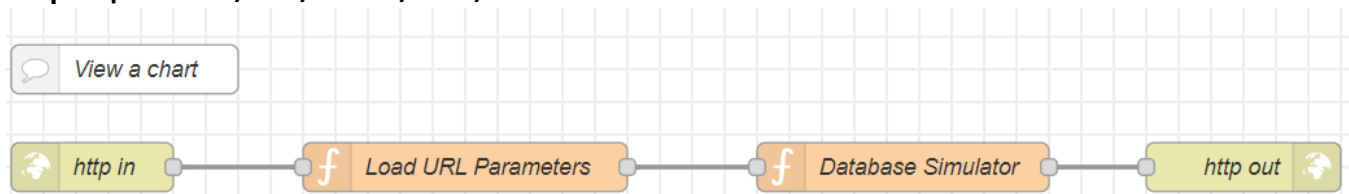
Σχήμα 34 Ροή NodeRed που αντιστοιχεί στο API call POST /user/:userID/chart/:chartID/chat/:chatRoomID



Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει στους χρήστες να στείλουν ένα telemetry data chart σε chat room.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint POST /user/:userID/chart/:chartID/chat/:chatRoomID για τη δημιουργία του web service που στέλνει ένα chart σε ένα Chat room.
Load URL parameter	function	Συνάρτηση που δημιουργεί τρεις μεταβλητές userID, chartID, chatRoomID, τις θέτει ίσες με τις παραμέτρους userID, chartID chatRoomID του path αντίστοιχα και τις προσθέτει στο msg.topic. Έπειτα, προσθέτει το JavaScript object σε μια μεταβλητή chartMessage στο msg.payload.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το msg.
Data saved in Database	debug	Εκτυπώνει το περιεχόμενο του msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.

#### Ροή Endpoint GET /user/:userID/chart/:chartID



Σχήμα 35 Ροή NodeRed που αντιστοιχεί στο API call GET /user/:userID/chart/:chartID

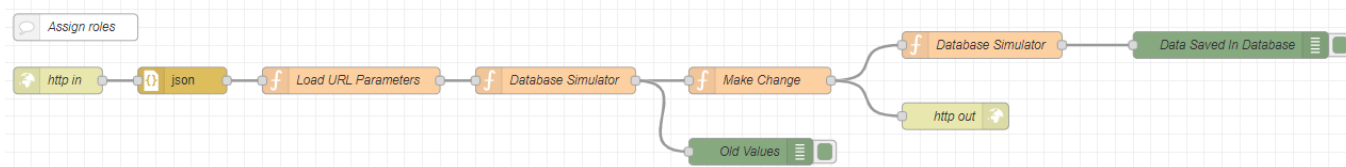
Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει στους χρήστες να δουν ένα telemetry data chart.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint GET /user/:userID/chart/:chartID για τη δημιουργία του web service που επιστρέφει ένα.
Load URL parameter	function	Συνάρτηση που δημιουργεί δύο μεταβλητές userID, chartID και τις θέτει ίσες με τις παραμέτρους userID, chartID του path. Μετά την προσθέτει τη νέα μεταβλητή στο msg.payload.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το chart στο msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.



### 3.1.6. Ροές πόρου User

Ροή Endpoint PUT /user/chiefEngineer/:userID/adminPanel/:userID

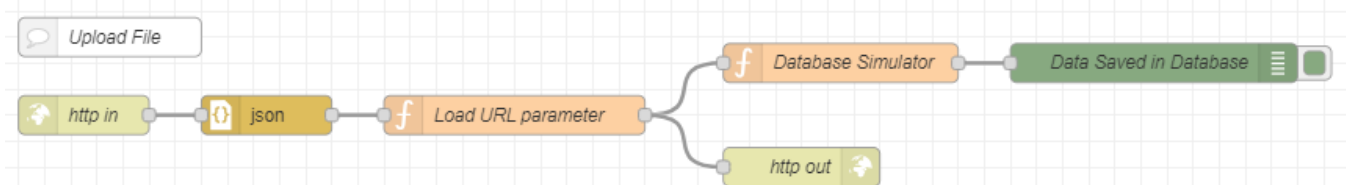


Σχήμα 36 Ροή NodeRed που αντιστοιχεί στο API call PUT /user/chiefEngineer/:userID/adminPanel/:userID

Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει σε ένα χρήστη chief engineer να αλλάξει το ρόλο ενός άλλου χρήστη.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint PUT /user/chiefEngineer/:userID/adminPanel/:userID για τη δημιουργία του web service που αλλάζει το value ενός element του vehicle set up.
json	json	Μετατρέπει το JSON string που δέχεται από τον κόμβο inject σε JavaScript object representation.
Load URL parameter	function	Συνάρτηση που δημιουργεί μία μεταβλητή userID, τη θέτει ίση με τη παράμετρο userID του path και τη προσθέτει στο msg.payload. Έπειτα, προσθέτει το JavaScript object σε μια μεταβλητή newRole στο msg.topic.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το παλιό αντικείμενο προφίλ του χρήστη με το συγκεκριμένο userID.
Old Values	debug	Εκτυπώνει το παλιό αντικείμενο.
Make Change	function	Αλλάζει τη τιμή της μεταβλητής role του παλιού αντικειμένου στη τιμή της μεταβλητής newRole.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το msg.
Data saved in Database	debug	Εκτυπώνει το περιεχόμενο του msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.

Ροή Endpoint POST /user/engineer/:userID/file



Σχήμα 37 Ροή NodeRed που αντιστοιχεί στο API call POST /user/engineer/:userID/file



Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει στον χρήστη να ανεβάσει ένα αρχείο.

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint POST /user/engineer/:userID/file για τη δημιουργία του web service που ανεβάζει ένα νέο αρχείο.
json	json	Μετατρέπει το JSON string που δέχεται από τον κόμβο inject σε JavaScript object representation.
Load URL parameter	function	Συνάρτηση που δημιουργεί μια μεταβλητή userID, την θέτει ίση με τη παράμετρο userID του path και την προσθέτει τη νέα μεταβλητή στο msg.topic. Έπειτα, προσθέτει το JavaScript object στο msg.payload.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το msg.
Data saved in Database	debug	Εκτυπώνει το περιεχόμενο του msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.

### 3.1.7. Ροή Πόρου Profile

**Ροή Endpoint PUT /user/:userID/profilePage**

*Σχήμα 38 Ροή NodeRed που αντιστοιχεί στο API call PUT /user/:userID/profilePage*

Η ροή που υλοποιεί την υπηρεσία, που επιτρέπει σε ένα χρήστη να αλλάξει τα προσωπικά του στοιχεία στη σελίδα profile.

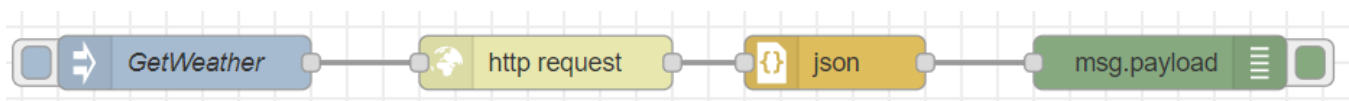
Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http in	http in	Δημιουργεί το HTTP endpoint PUT /user/:userID/profilePage για τη δημιουργία του web service που αλλάζει το profile page του.
json	json	Μετατρέπει το JSON string που δέχεται από τον κόμβο inject σε JavaScript object representation.
Load URL parameter	function	Συνάρτηση που δημιουργεί μία μεταβλητή userID, τη θέτει ίση με τη παράμετρο userID του path και τη προσθέτει στο msg.payload. Έπειτα, προσθέτει το JavaScript object στο msg.topic.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το παλιό αντικείμενο προφίλ του χρήστη με το συγκεκριμένο userID.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Old Values	debug	Εκτυπώνει το παλιό αντικείμενο.
Make Change	function	Αλλάζει τις τιμές των μεταβλητών του παλιού αντικειμένου στις τιμές των μεταβλητών αποθηκευμένων στο msg.payload.
Database Simulator	function	Στατική προσομοίωση του database που επιστρέφει το msg.
Data saved in Database	debug	Εκτυπώνει το περιεχόμενο του msg.payload.
http out	http response	Στέλνει τα responses πίσω στο request.

## 3.2. Υλοποίηση Ιστοριών χρήστη

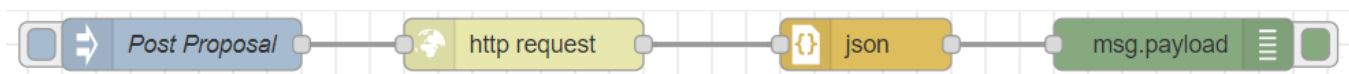
### 3.2.1. Ιστορία Χρήστη Get The Weather Forecast



Σχήμα 39 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη “Get the Weather Forecast”

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
GetWeather	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/:userID/weather η οποία επιστρέφει το δελτίο καιρού.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα το δελτίο καιρού.

### 3.2.2. Ιστορία Χρήστη Create New Proposal



Σχήμα 40 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη “Create New Proposal”



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Post Proposal	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/engineer/:userID/proposals η οποία δημιουργεί μία νέα πρόταση.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα την νέα πρόταση.

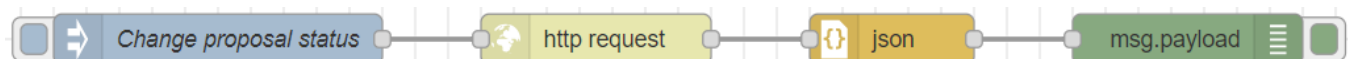
### 3.2.3. Ιστορία Χρήστη Accept or Decline Proposal



Σχήμα 41 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη “Accept or Decline Proposal”

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Accept or Decline Proposal	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/chiefEngineer/:userID/proposals/:proposalID η οποία επιτρέπει στο χρήστη την αποδοχή ή την απόρριψη μιας πρότασης.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα την πρόταση που αποδέχτηκε ή απέρριψε ο χρήστης.

### 3.2.4. Ιστορία Χρήστη Change Proposal Status to Done



Σχήμα 42 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη “Change Proposal Status to Done”

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Change proposal status	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/handsOnEngineer/:userID/proposals/:proposalID η οποία



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
		επιτρέπει στο χρήστη να αλλάξει την κατάσταση της πρότασης σε DONE..
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα την πρόταση με τη νέα κατάσταση.

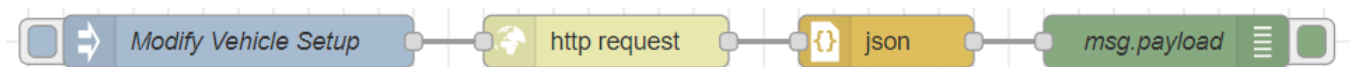
### 3.2.5. Ιστορία Χρήστη Create Vehicle Setup



Σχήμα 43 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη “Create Vehicle Setup”

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Create Vehicle Setup	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/chiefEngineer/:userID/vehicleSetup η οποία δημιουργεί ένα νέο vehicle setup.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα το νέο vehicle setup .

### 3.2.6. Ιστορία Χρήστη Modify Vehicle Setup



Σχήμα 44 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη “Modify Vehicle Setup”

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Modify Vehicle Setup	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/chiefEngineer/:userID/vehicleSetup/:elementID η οποία αλλάζει την τιμή ενός element του vehicle setup.
json	json	Μετατρέπει το JSON string του msg.payload σε object.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
msg.payload	debug	Τυπώνει στην κονσόλα τη νέα τιμή του element .

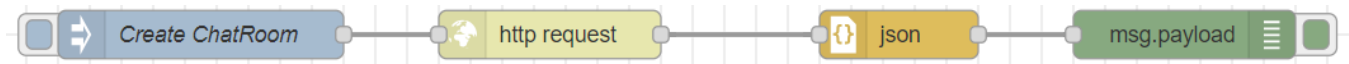
### 3.2.7. Ιστορία Χρήστη View Vehicle Setup



Σχήμα 45 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήση “View Vehicle Setup”

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
View Vehicle Setup	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/:userID/vehicleSetUp η οποία επιστρέφει το vehicle setup.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα το vehicle setup .

### 3.2.8. Ιστορία Χρήστη Create ChatRoom



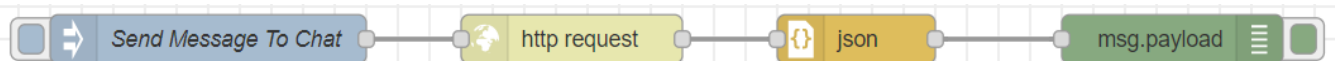
Σχήμα 46 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήση “Create ChatRoom”

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Create ChatRoom	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/:userID/chat η οποία δημιουργεί ένα νέο chat room.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα το νέο chat room .





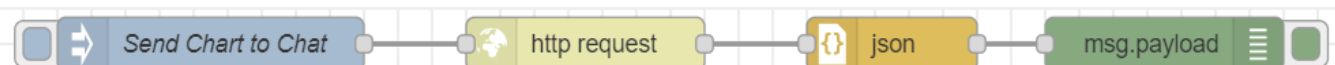
### 3.2.9. Ιστορία Χρήστη Send Message to Chat



Σχήμα 47 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη “Send Message to Chat”

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Send Message To Chat	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/:userID/chat/:chatRoomID η οποία στέλνει ένα νέο μήνυμα στο chat room.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα το νέο μήνυμα.

### 3.2.10. Ιστορία Χρήστη Send Chart to Chat



Σχήμα 48 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη “Send Chart to Chat”

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Send Chart To Chat	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/:userID/chart/:chartID/chat/:chatRoomID η οποία στέλνει ένα chart στο chat room.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα τα περιεχόμενα του chat room.

### 3.2.11. Ιστορία Χρήστη View a Chart

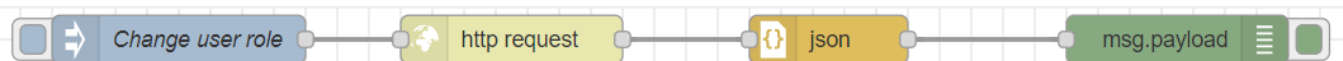


Σχήμα 49 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη “View a Chart”



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
View chart	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/:userID/chart/:chartID η οποία επιστρέφει ένα chart.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα τα στοιχεία του chart.

### 3.2.12. Ιστορία Χρήστη Assign Roles



Σχήμα 50 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη "Assign Roles"

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Change user role	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/chiefEngineer/:userID/adminPanel/:userID η οποία αλλάζει το ρόλο ενός χρήστη.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα το νέο ρόλο του χρήστη.

### 3.2.13. Ιστορία Χρήστη Upload File



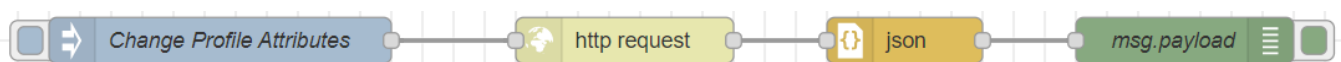
Σχήμα 51 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη "Upload File"

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Upload file	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/engineer/:userID/file η οποία ανεβάζει ένα νέο αρχείο.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα το περιεχόμενο του αρχείου.

### 3.2.14. Ιστορία Χρήστη Edit Profile Page



Σχήμα 52 Ροή NodeRed που αντιστοιχεί στην ιστορία χρήστη “Edit Profile Page”

Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Change Profile Attributes	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
http request	http-request	Κάνει κλήση προς την υπηρεσία /user/:userID/profilePage η οποία αλλάζει τα στοιχεία στο profile page ενός χρήστη.
json	json	Μετατρέπει το JSON string του msg.payload σε object.
msg.payload	debug	Τυπώνει στην κονσόλα τα νέα στοιχεία του χρήστη.



## Παράρτημα Ι – Ανοιχτά Θέματα

- Τα πρότυπα σχεδίασης που υλοποιήσαμε δεν καλύπτουν όλες τις λειτουργικές και μη λειτουργικές απαιτήσεις. Συγκεκριμένα, απαιτείται εμπειριστατωμένη ανάλυση των κλάσεων του προηγούμενου παραδοτέου προκειμένου η σχεδίαση να γίνει πιο απλή και να φανούν ξεκάθαρα οι περιπτώσεις στις οποίες χρησιμοποιούνται πρότυπα σχεδίασης
- Χρειάστηκε η προσθήκη της μεταβλητής elementID που δεν είχε υλοποιηθεί στο πρώτο παραδοτέο προκειμένου να διευκρινιστεί σε ποιό στοιχείο αναφέρεται ο χρήστης.