



Enhance your ideas

New World Order Control

Enhance your ideas

Προδιαγραφές Λογισμικού και Υλοποίηση συστήματος

Del.1.2

Version 1.2

Αχιλλέας Αραμπατζής araachapo@ece.auth.gr
Θεμιστοκλής Αρβανίτης tarvanit@ece.auth.gr
Κωνσταντίνος Κράντας kkrantas@ece.auth.gr
Κωνσταντίνος Στρίκος strikosk@ece.auth.gr

Ημερομηνία: 21/06/2023



Μέλη της Ομάδας Ανάπτυξης

Όνομα	OA	Email
Α. Συμεωνίδης	*	asymeon@issel.ee.auth.gr
Αχιλλέας Αραμπατζής	16	araachapo@ece.auth.gr
Θεμιστοκλής Αρβανίτης	16	tarvanit@ece.auth.gr
Κωνσταντίνος Κράντας	16	kkrantas@ece.auth.gr
Κωνσταντίνος Στρίκος	16	strikosk@ece.auth.gr



Πίνακας Περιεχομένων

Λίστα Σχημάτων	5
1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν	6
1.1. Πρότυπο Proxy	6
1.2. Πρότυπο Abstract Factory	7
1.3. Πρότυπο Adapter	8
2. Αρχιτεκτονική Συστήματος	9
2.1. Αναγνώριση Πόρων (Resources) Συστήματος	9
2.2. Τεκμηρίωση REST διεπαφής	9
2.2.1. Πόρος User	9
2.2.1.1. Μοντέλο δεδομένων User	9
2.2.1.2. Endpoint POST πόρου User για δημιουργία νέου συμβολαίου	10
2.2.1.3. Endpoint POST πόρου User για αποστολή feedback	11
2.2.2. Πόρος Contract	12
2.2.2.1. Μοντέλο δεδομένων Contract	12
2.2.2.2. Endpoint PUT του πόρου Contract	12
2.2.2.3. Endpoint GET του πόρου Contract	14
2.2.3. Πόρος Post	15
2.2.3.1. Μοντέλο δεδομένων post	15
2.2.3.2. Endpoint GET πόρου Post	16
2.2.3.3. Endpoint PUT πόρου Post	17
2.2.4. Πόρος Graph	19
2.2.4.1. Μοντέλο δεδομένων Graph	19
2.2.4.2. Endpoint GET πόρου Graph	19
2.2.5. Πόρος Bot	21
2.2.5.1. Μοντέλο δεδομένων Bot	21
2.2.5.3. Endpoint GET πόρου Bot	21
3. Υλοποίηση Συστήματος με Node-RED	23
3.1. Αντιστοίχιση των REST Υπηρεσιών σε Ροές Node-RED	23
3.1.1. Ροές πόρου User	23
3.1.1.1. Ροή endpoint post /user/:UserID	23
3.1.1.2. Ροή endpoint post /user/feedback	23
3.1.2. Ροές πόρου Contract	23
3.1.2.1. Ροή endpoint put/user/:UserID/contract/:ContractID	23
3.1.2.2. Ροή endpoint get /user/:UserID/contract	23
3.1.3. Ροές πόρου Post	24
3.1.3.1. Ροή endpoint get /user/:UserID/contract/:ContractID/post	24



3.1.3.2. Ροή endpoint put /user/:UserID/contract/:ContractID/post	24
3.1.4. Ροές πόρου Graph	24
3.1.4.1. Ροή endpoint get /user/:UserID/contract/:ContractID/graph/:GraphID	24
3.1.5. Ροές πόρου Bot	24
3.1.5.1. Ροή endpoint get /user/:UserID/contract/:ContractID/bot/:botid	24
3.2. Υλοποίηση Ιστοριών χρήστη	25
3.2.1. Ιστορία Χρήστη user creates contract	25
3.2.2. Ιστορία Χρήστη user sends Feedback	25
3.2.3. Ιστορία Χρήστη user accepts/declines contract	26
3.2.4. Ιστορία Χρήστη user views contracts	26
3.2.5. Ιστορία Χρήστη user sees statistics	27
3.2.6. Ιστορία Χρήστη user sees bot's actions	27
3.2.7. Ιστορία Χρήστη user searches through bot's actions	28
3.2.8. Ιστορία Χρήστη user sees a bot's profile	28
Παράρτημα II – Ανοιχτά Θέματα	29



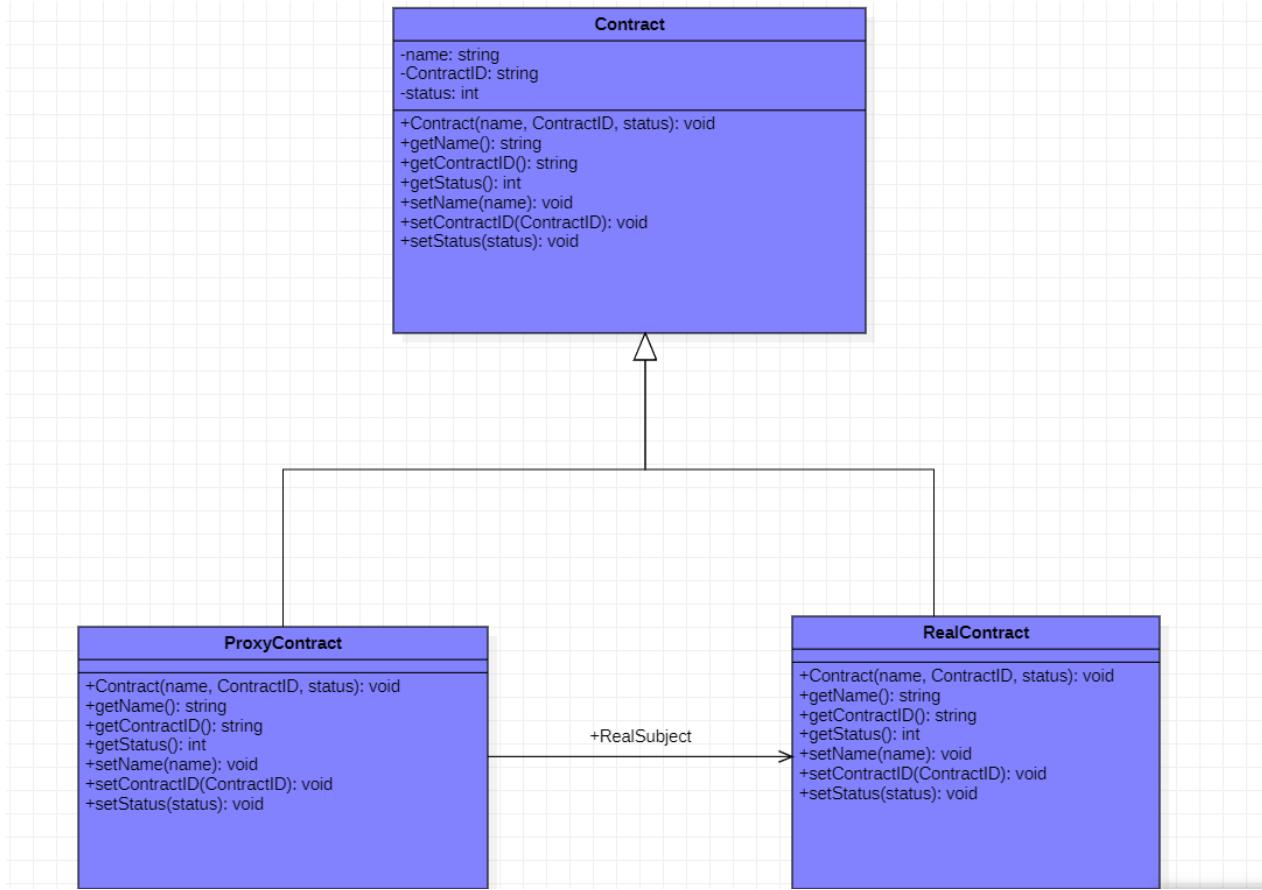
Λίστα Σχημάτων

Σχήμα 1.1: Εφαρμογή προτύπου Proxy	6
Σχήμα 1.2 Εφαρμογή προτύπου Abstract Factory	7
Σχήμα 1.3 Εφαρμογή προτύπου Abstract	8
Σχήμα 2.1 Μοντέλο δεδομένων πόρου User	9
Σχήμα 2.2 Εξήγηση και παράθεση των παραμέτρων του πόρου	10
Σχήμα 2.3 Αποκρίσεις του συστήματος	10
Σχήμα 2.4 Δοκιμή του endpoint	11
Σχήμα 2.5 Εξήγηση και παράθεση των παραμέτρων του πόρου	11
Σχήμα 2.6 Αποκρίσεις του συστήματος	11
Σχήμα 2.7 Δοκιμή του endpoint	12
Σχήμα 2.8 Μοντέλο δεδομένων πόρου Contract	12
Σχήμα 2.9 Εξήγηση και παράθεση των παραμέτρων του πόρου	12
Σχήμα 2.10 Αποκρίσεις του συστήματος	13
Σχήμα 2.11 Δοκιμή του endpoint	13
Σχήμα 2.12 Εξήγηση και παράθεση των παραμέτρων του πόρου	14
Σχήμα 2.13 Αποκρίσεις του συστήματος	14
Σχήμα 2.14 Δοκιμή του endpoint	15
Σχήμα 2.15 Μοντέλο δεδομένων πόρου Post	15
Σχήμα 2.16 Εξήγηση και παράθεση των παραμέτρων του πόρου	16
Σχήμα 2.17 Αποκρίσεις του συστήματος	16
Σχήμα 2.18 Δοκιμή του endpoint	17
Σχήμα 2.19 Εξήγηση και παράθεση των παραμέτρων του πόρου	17
Σχήμα 2.20 Αποκρίσεις του συστήματος	18
Σχήμα 2.21 Δοκιμή του endpoint	18
Σχήμα 2.22 Μοντέλο δεδομένων πόρου Graph	19
Σχήμα 2.23 Εξήγηση και παράθεση των παραμέτρων του πόρου	19
Σχήμα 2.24 Αποκρίσεις του συστήματος	20
Σχήμα 2.25 Δοκιμή του endpoint	20
Σχήμα 2.26 Μοντέλο δεδομένων πόρου Bot	21
Σχήμα 2.27 Εξήγηση και παράθεση των παραμέτρων του πόρου	21
Σχήμα 2.28 Αποκρίσεις του συστήματος	22
Σχήμα 2.29 Δοκιμή του endpoint	22



1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν

1.1. Πρότυπο Proxy

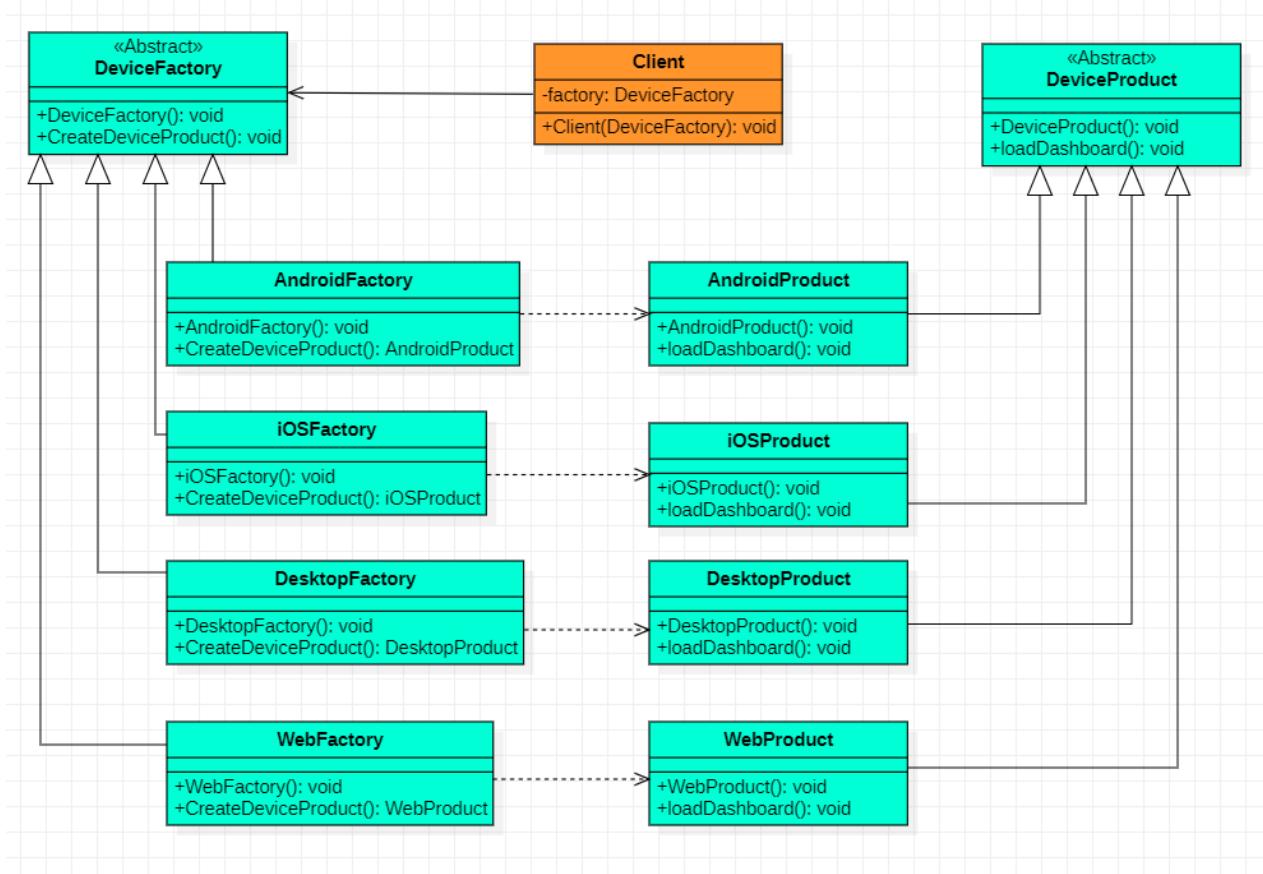


Σχήμα 1.1: Εφαρμογή προτύπου Proxy

Το πρότυπο Proxy είναι ένα δομικό πρότυπο. Στο σύστημά μας εφαρμόζεται με στόχο την κάλυψη των απαιτήσεων ασφάλειας (ΜΛΑ-2), με στόχο την προστασία του συστήματος από κακόβουλες επιθέσεις καθώς και την εξασφάλιση της διατήρησης της ιδιωτικότητας του χρήστη του συστήματος. Για να συμβεί αυτό γίνεται έλεγχος στο αντικείμενο **Contract**, το συμβόλαιο του χρήστη, στο οποίο μόνο αυτός μπορεί να έχει πρόσβαση και περιλαμβάνει όλες τις κρίσιμες πληροφορίες και λεπτομέρειες της καμπάνιας που έχει ξεκινήσει. Έτσι, χρησιμοποιώντας το πρότυπο σχεδίασης proxy και ειδικότερα τον τύπο proxy protection, με το συγκεκριμένο αντικείμενο ενισχύεται το επίπεδο ασφάλειας των ευαίσθητων δεδομένων που μόνο ο χρήστης πρέπει να έχει πρόσβαση.



1.2. Πρότυπο Abstract Factory

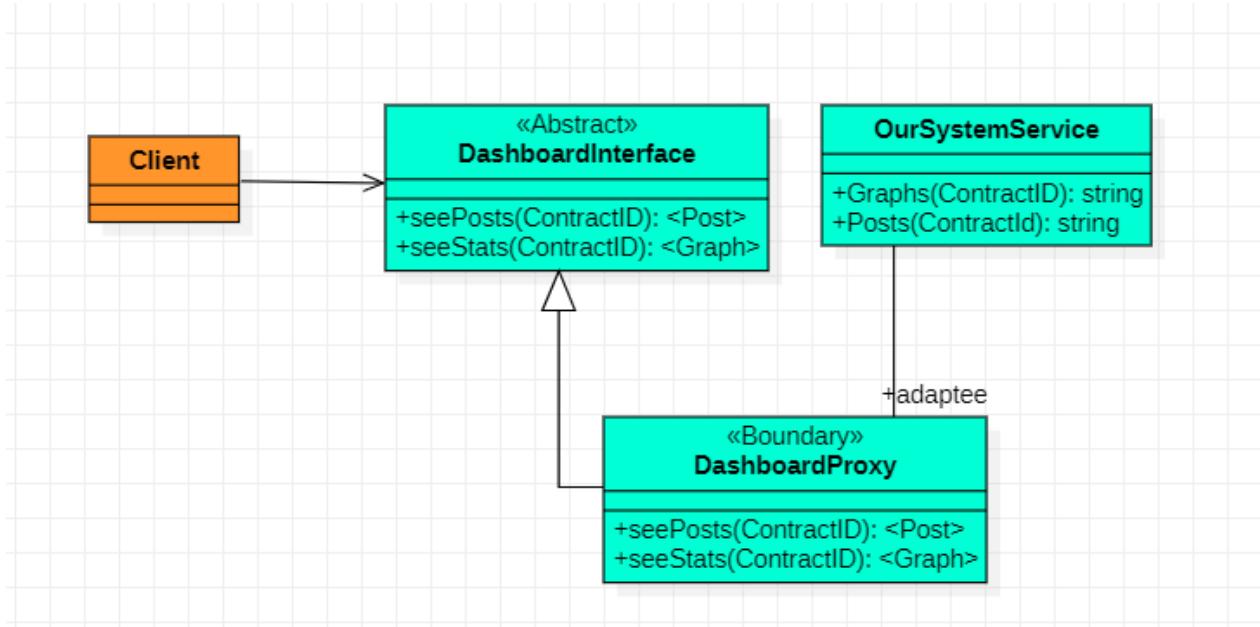


Σχήμα 1.2 Εφαρμογή προτύπου Abstract Factory

Το Abstract Factory αποτελεί ένα δημιουργικό πρότυπο, το οποίο αξιοποιείται στο σύστημα μας για να εξασφαλίσει την ΜΛΑ-1 σχετικά με τις απαιτήσεις φορητότητας και χρηστικότητας. Με αυτό το πρότυπο, το σύστημα μας έχει τη δυνατότητα να υποστηρίζει iOS και Android λειτουργικά, web browser χρήση, καθώς και χρήση μέσω desktop, καθώς και να επικοινωνεί μαζί τους. Ειδικότερα, μέσω του Abstract DeviceProduct, είναι εφικτή η πρόσβαση στο dashboard της εφαρμογής για τις διάφορες συσκευές, ενώ αν υπάρξει η ανάγκη χρηστικότητας από νέα συσκευή στο μέλλον, με προσθήκη ενός νέου αντίστοιχου Product αυτό θα είναι εφικτό.



1.3. Πρότυπο Adapter



Σχήμα 1.3 Εφαρμογή προτύπου Abstract

Το πρότυπο adapter αποτελεί ένα δομικό πρότυπο, το οποίο χρησιμοποιείται στο σύστημα μας για την κάλυψη της ΜΛΑ-3 η οποία απαιτεί τη διασύνδεση του εσωτερικού συστήματος της εταιρείας με την εφαρμογή που χρησιμοποιεί ο χρήστης. Συγκεκριμένα, το εσωτερικό σύστημα μπορεί να αποστέλλει δεδομένα σε μορφή `string` και ο adapter να μετατρέπει αυτά τα δεδομένα σε αντικείμενα, τα οποία είναι αναγνώσιμα και προσβάσιμα από τον χρήστη της εφαρμογής. Στην προκειμένη περίπτωση, για να επιτευχθεί ο στόχος χρησιμοποιείται ο προσαρμογέας αντικειμένου (Object Adapter), ενώ με τη συγκεκριμένη σχεδίαση, εάν τροποποιηθούν οι ιδιότητες του dashboard, με μια αλλαγή στο πρότυπο εξασφαλίζεται ευκολότερα η ορθή λειτουργία του.



2. Αρχιτεκτονική Συστήματος

Η ανάπτυξη έγινε στο εργαλείο Postman API Platform, αλλά για λόγους ευκολίας και προτυποποίησης επιλέξαμε να διαμοιραστούμε παρακάτω το documentation σε Swaggerhub.

- [Σύνδεσμος](#) για το API στο Swaggerhub.

2.1. Αναγνώριση Πόρων (Resources) Συστήματος

Κλάση BEC	Πόρος REST	Endpoints (HTTP Verbs)
User	/user/{UserID}	POST
Contract	/user/{UserID}/contract	GET
Contract	/user/{UserID}/contract/{ContractID}	PUT
Post	/user/{UserID}/contract/{ContractID}/post	GET, PUT
Graph	/user/{UserID}/contract/{ContractID}/graph/{GraphID}	GET
Bot	/user/{UserID}/contract/{ContractID}/post/{BotID}	GET
Feedback	/user/feedback	POST

2.2. Τεκμηρίωση REST διεπαφής

2.2.1. Πόρος User

2.2.1.1. Μοντέλο δεδομένων User

```
user ▼ {  
    UserID          string  
    username       string  
    password       string  
  
}
```

Σχήμα 2.1 Μοντέλο δεδομένων πόρου User



2.2.1.2. Endpoint POST πόρου User για δημιουργία νέου συμβολαίου

The screenshot shows a POST request to the endpoint /user/{UserID}. A note at the top states: "FR1/FR2 - The user should be able to select the parameters and the campaign." Below this is a "Parameters" section with a table. The first row has "Name" and "Description" columns. The second row contains "UserID" (marked as required, type string, path), with the value "themis1234" in the input field. There is also a "Cancel" button. Below the parameters is a "Request body" section with a dropdown set to "application/json". The JSON example provided is:

```
{  
  "select": "string",  
  "comments": "string",  
  "platforms": 0,  
  "intensity": 0  
}
```

Σχήμα 2.2 Εξήγηση και παράθεση των παραμέτρων του πόρου

The screenshot displays two API response entries. The first entry for code 200 OK shows a "Media type" dropdown set to "application/json" and an "Example Value" block containing a JSON array of objects with fields name, ContractID, and status.

Code	Description	Links
200	OK	No links
400	unexpected error	No links

The second entry for code 400 unexpected error shows a "Media type" dropdown set to "application/json" and an "Example Value" block containing a JSON object with code and message fields.

Σχήμα 2.3 Αποκρίσεις του συστήματος



Snippets▼

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

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/themis1234' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "select": "string",
    "comments": "string",
    "platforms": 0,
    "intensity": 0
  }'
```

Request URL

```
https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/themis1234
```

Server response

Σχήμα 2.4 Δοκιμή του endpoint.

2.2.1.3. Endpoint POST πόρου User για αποστολή feedback

POST /user/feedback

User gives feedback

Parameters

No parameters

Cancel

Execute Clear

Σχήμα 2.5 Εξήγηση και παράθεση των παραμέτρων του πόρου

200 sent feedback No links

Media type application/json ▾
Controls Accept header.

Example Value Schema

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

400 unexpected error No links

Media type application/json ▾
Controls Accept header.

Example Value Schema

```
{ "code": 0, "message": "string" }
```

Σχήμα 2.6 Αποκρίσεις του συστήματος



Snippets▼

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

```
curl -X 'POST' \
'https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/feedback' \
-H 'accept: application/json' \
-d ''
```

Request URL

<https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/feedback>

Server response

Code Details

Σχήμα 2.7 Δοκιμή του endpoint.

2.2.2. Πόρος Contract

2.2.2.1. Μοντέλο δεδομένων Contract

```
contract ▾ {  
    name          string  
    ContractID   string  
    status        integer  
  
}
```

Σχήμα 2.8 Μοντέλο δεδομένων πόρου Contract

2.2.2.2. Endpoint PUT του πόρου Contract

PUT /user/{UserID}/contract/{ContractID}

fr4

Parameters

Name	Description
UserID <small>* required</small> string (path)	ID of the user strikos1234
ContractID <small>* required</small> string (path)	ID of the contract get political

Request body required

application/json

```
{ "Decision": true }
```

Σχήμα 2.9 Εξήγηση και παράθεση των παραμέτρων του πόρου



The screenshot shows a Swagger UI interface with two main sections. The top section is for a successful response (200 OK) with the status code 'OK'. It includes a dropdown for 'Media type' set to 'application/json', a note about the 'Accept' header, and example values for the schema. The bottom section is for an unexpected error (400), with the status code 'unexpected error'. It also has a dropdown for 'Media type' set to 'application/json' and displays an error message schema.

Σχήμα 2.10 Αποκρίσεις του συστήματος

The screenshot shows a Swagger UI interface with a 'Snippets' section. It features three tabs for cURL (bash), cURL (PowerShell), and cURL (CMD). The 'cURL (bash)' tab is selected and contains a command to perform a PUT request to a specific URL with JSON data. Below the tabs, there is a 'Request URL' field containing the same URL and a 'Server response' field which is currently empty.

Σχήμα 2.11 Δοκιμή του endpoint.



2.2.2.3. Endpoint GET του πόρου Contract

GET /user/{UserID}/contract

fr5

Parameters

Name Description

UserID * required
string
(path)
strikos1234

Cancel

Execute Clear

Σχήμα 2.12 Εξήγηση και παράθεση των παραμέτρων του πόρου

Code	Description	Links
200	get contracts	No links
	Media type application/json	
	Controls <code>Accept</code> header.	
	Example Value Schema	
	[{ "name": "string", "ContractID": "string", "status": 0 }]	
400	unexpected error	No links
	Media type application/json	
	Example Value Schema	
	{ "code": 0, "message": "string" }	

Σχήμα 2.13 Αποκρίσεις του συστήματος



Snippets▼

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

```
curl -X 'GET' \
  'https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/strikos1234/contract' \
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/strikos1234/contract
```

Server response

Σχήμα 2.14 Δοκιμή του endpoint

2.2.3. Πόρος Post

2.2.3.1. Μοντέλο δεδομένων post

```
post ▾ {  
  postID*           string  
  BotID*           string  
  postLink         string($uri)  
  imgPost          string($uri)  
  
}
```

Σχήμα 2.15 Μοντέλο δεδομένων πόρου Post



2.2.3.2. Endpoint GET πόρου Post

GET /user/{UserID}/contract/{ContractID}/post

FR 8 - The user should be able to search and view the actions of the bots.

View the posts the Bots have done

Parameters

Name Description

ContractID * required
string
(path)
randomhashcontracct

UserID * required
string
(path)
randomhashuser

Execute Clear

Σχήμα 2.16 Εξήγηση και παράθεση των παραμέτρων του πόρου

Code	Description	Links
200	OK	No links
	Media type application/json	
	Controls Accept header.	
	Example Value Schema	
	[{ "postID": "string", "BotID": "string", "postLink": "string", "imgPost": "string" }]	
400	The provided parameters are incorrect	No links
	Media type application/json	
	Example Value Schema	
	{ "code": 0, "message": "string" }	

Σχήμα 2.17 Αποκρίσεις του συστήματος



Snippets▼

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

```
curl -X 'GET' \
  'https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/randomhashuser/contract/randomhashcontracct/post'
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/randomhashuser/contract/randomhashcontracct/post
```

Server response

Σχήμα 2.18 Δοκιμή του endpoint

2.2.3.3. Endpoint PUT πόρου Post

PUT /user/{UserID}/contract/{ContractID}/post FR 8 - The user should be able to search and view the actions of the bots.

Edit the filters of the search

Parameters

Name	Description
ContractID * required	ID of the contract
string (path)	randomhashcontracct
UserID * required	ID of the user
string (path)	randomhashuser

Cancel

Request body required

application/json

```
{
  "SortBy": 0,
  "Platforms": 0
}
```

Σχήμα 2.19 Εξήγηση και παράθεση των παραμέτρων του πόρου



Code	Description	Links
200	OK	No links
400	The provided parameters are incorrect	No links

Σχήμα 2.20 Αποκρίσεις του συστήματος

Snippets ▾

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

```
curl -X 'PUT' \
'https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/randomhashuser/contract/randomhashcontract/post' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "SortBy": 0,
  "Platforms": 0
}'
```

Request URL

```
https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/randomhashuser/contract/randomhashcontract/post
```

Server response

Σχήμα 2.21 Δοκιμή του endpoint



2.2.4. Πόρος Graph

2.2.4.1. Μοντέλο δεδομένων Graph

```
Graph {  
    xData  
    yData  
    color  
    labels  
    size  
  
}
```

Σχήμα 2.22 Μοντέλο δεδομένων πόρου Graph

2.2.4.2. Endpoint GET πόρου Graph

GET /user/{UserID}/contract/{ContractID}/graph/{GraphID}

Returns all statistical graphs for the specific campaign

Parameters

Name	Description
ContractID * required	ID of the contract
string (path)	randomhashcontracct
UserID * required	ID of the user
string (path)	randomhashuser
GraphID * required	ID of a specific graph
string (path)	randomhasggraph

Execute Clear

Σχήμα 2.23 Εξίγηση και παράθεση των παραμέτρων του πόρου



Code	Description	Links
200	chosen graph	No links
	<p>Media type</p> <p>application/json</p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>[{ "xData": [0], "yData": [0], "color": "string", "labels": ["string"], "size": 0 }]</pre>	
400	unexpected error	No links
	<p>Media type</p> <p>application/json</p> <p>Example Value Schema</p> <pre>{ "code": 0, "message": "string" }</pre>	

Σχήμα 2.24 Αποκρίσεις του συστήματος

Snippets▼

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

```
curl -X 'GET' \  
  'https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/randomhashuser/contract/randomhashcontracct/graph/randomhasggraph'  
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/strikosk/NW0/1.0.0/user/randomhashuser/contract/randomhashcontracct/graph/randomhasggraph
```

Server response

Σχήμα 2.25 Δοκιμή του endpoint



2.2.5. Πόρος Bot

2.2.5.1. Μοντέλο δεδομένων Bot

```
Bot {  
    Name      string  
    id        integer  
    Profiles  [string]  
    Stats     [string]  
    top5     [post {  
        postID*   string  
        BotID*    string  
        postLink  string($uri)  
        imgPost   string($uri)  
    }]  
}
```

Σχήμα 2.26 Μοντέλο δεδομένων πόρου Bot

2.2.5.2. Endpoint GET πόρου Bot

GET	/user/{UserID}/contract /{ContractID}/post/{BotID}	FR9	↔																																
Returns a bots' profile																																			
Parameters																																			
<table border="1"><tr><td colspan="2">Name</td><td>Description</td><td>Cancel</td></tr><tr><td>ContractID</td><td>* required</td><td>ID of the contract</td><td></td></tr><tr><td>string</td><td>(path)</td><td>randomhashcontracct</td><td></td></tr><tr><td>UserID</td><td>* required</td><td>ID of the user</td><td></td></tr><tr><td>string</td><td>(path)</td><td>randomhashuser</td><td></td></tr><tr><td>BotID</td><td>* required</td><td>ID of the bots</td><td></td></tr><tr><td>string</td><td>(path)</td><td>randomhashbot</td><td></td></tr><tr><td colspan="2">Execute</td><td colspan="2">Clear</td></tr></table>				Name		Description	Cancel	ContractID	* required	ID of the contract		string	(path)	randomhashcontracct		UserID	* required	ID of the user		string	(path)	randomhashuser		BotID	* required	ID of the bots		string	(path)	randomhashbot		Execute		Clear	
Name		Description	Cancel																																
ContractID	* required	ID of the contract																																	
string	(path)	randomhashcontracct																																	
UserID	* required	ID of the user																																	
string	(path)	randomhashuser																																	
BotID	* required	ID of the bots																																	
string	(path)	randomhashbot																																	
Execute		Clear																																	

Σχήμα 2.27 Εξήγηση και παράθεση των παραμέτρων του πόρου



Code	Description	Links
200	chosen bot	No links
	<p>Media type</p> <p>application/json</p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>[{ "Name": "string", "id": 0, "Profiles": ["string"], "Stats": ["string"], "tops": [{ "postID": "string", "botID": "string", "postLink": "string", "imgPost": "string" }] }]</pre>	
400	unexpected error	No links
	<p>Media type</p> <p>application/json</p> <p>Example Value Schema</p> <pre>{ "code": 0, "message": "string" }</pre>	

Σχήμα 2.28 Αποκρίσεις του συστήματος

Snippets▼

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

```
curl -X 'GET' \  
'https://virtserver.swaggerhub.com/strikosk/NIO/1.0.0/user/randomhashuser/contract/randomhashcontracct/post/randomhashbot' \  
-H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/strikosk/NIO/1.0.0/user/randomhashuser/contract/randomhashcontracct/post/randomhashbot
```

Server response

Σχήμα 2.29 Δοκιμή του endpoint



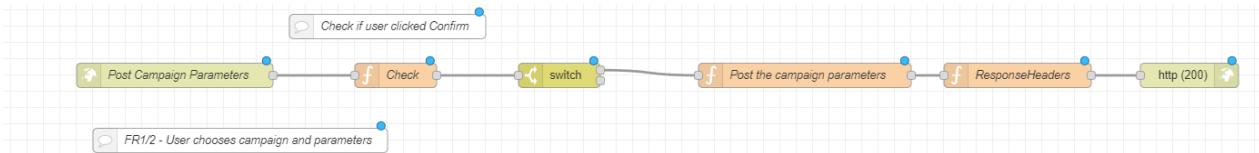
3. Υλοποίηση Συστήματος με Node-RED

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

3.1. Αντιστοίχιση των REST Υπηρεσιών σε Ροές Node-RED

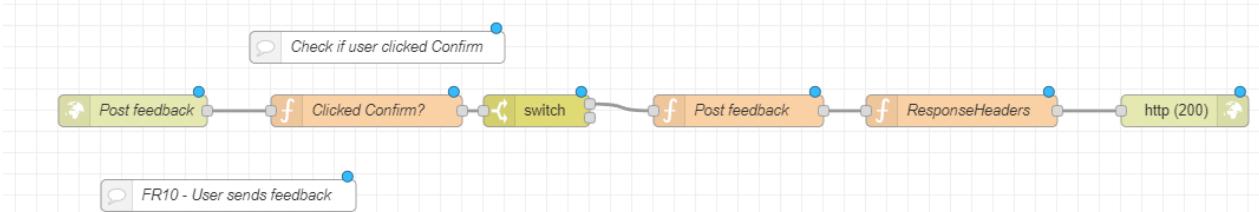
3.1.1. Ροές πόρου User

3.1.1.1. Ροή endpoint post /user/:UserID



Η ροή που υλοποιεί την υπηρεσία, η οποία δημιουργεί το συμβόλαιο του χρήστη.

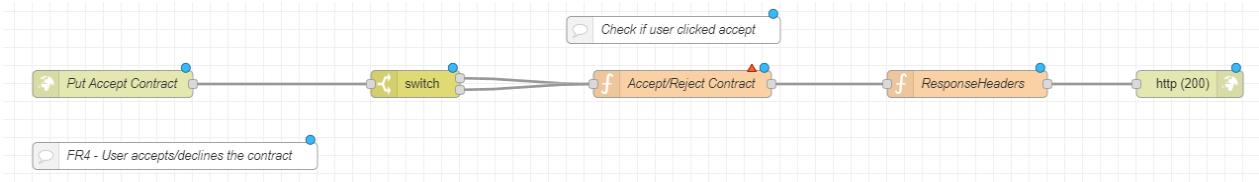
3.1.1.2. Ροή endpoint post /user/feedback



Η ροή που υλοποιεί την υπηρεσία, η οποία στέλνει την ανατροφοδότηση του χρήστη στο σύστημα.

3.1.2. Ροές πόρου Contract

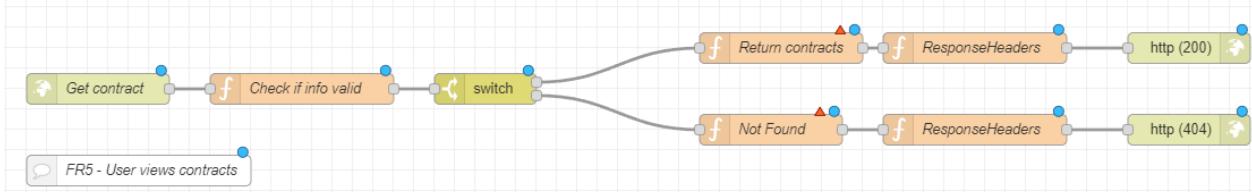
3.1.2.1. Ροή endpoint put/user/:UserID/contract/:ContractID



Η ροή η οποία υλοποιεί την υπηρεσία, η οποία ελέγχει αν ο χρήστης δέχεται ή απορρίπτει το συμβόλαιο.



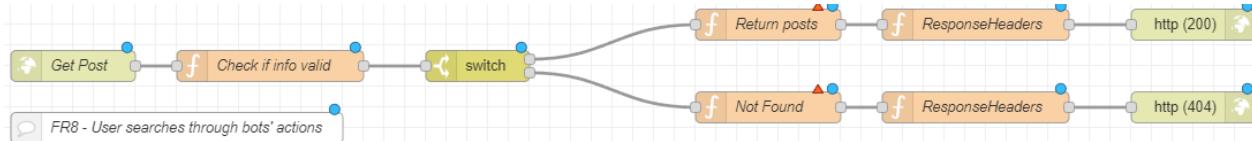
3.1.2.2. Ροή endpoint get /user/:UserID/contract



Η ροή η οποία υλοποιεί την υπηρεσία, η οποία επιστρέφει τα συμβόλαια του χρήστη.

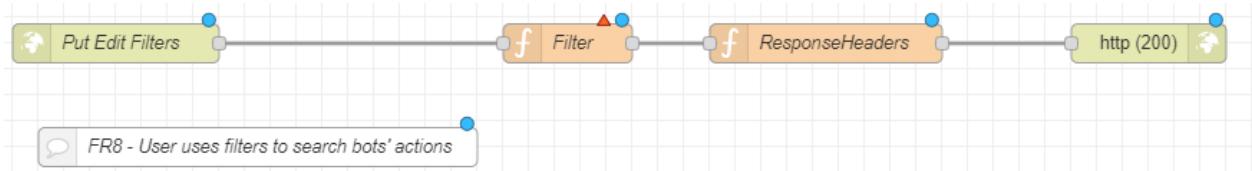
3.1.3. Ροές πόρου Post

3.1.3.1. Ροή endpoint get /user/:UserID/contract/:ContractID/post



Η ροή η οποία υλοποιεί την υπηρεσία, η οποία επιστρέφει τις ενέργειες των bot.

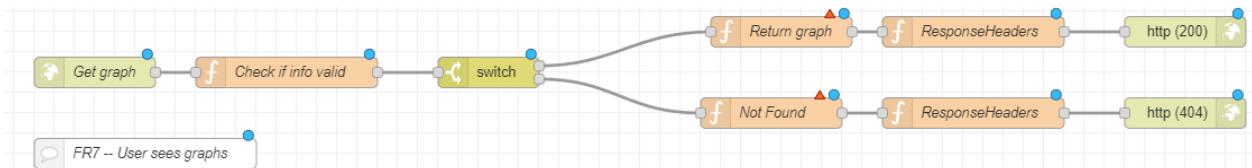
3.1.3.2. Ροή endpoint put /user/:UserID/contract/:ContractID/post



Η ροή η οποία υλοποιεί την υπηρεσία, μέσω της οποίας ο χρήστης αλλάζει τα φίλτρα αναζήτησης

3.1.4. Ροές πόρου Graph

3.1.4.1. Ροή endpoint get /user/:UserID/contract/:ContractID/graph/:GraphID

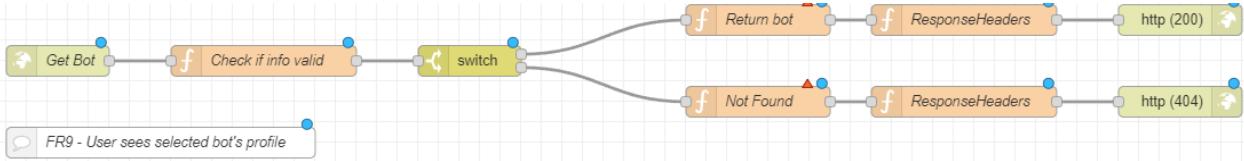


Η ροή η οποία υλοποιεί την υπηρεσία, η οποία επιστρέφει και εμφανίζει τα γραφήματα στατιστικών στον χρήστη.



3.1.5. Ροές πόρου Bot

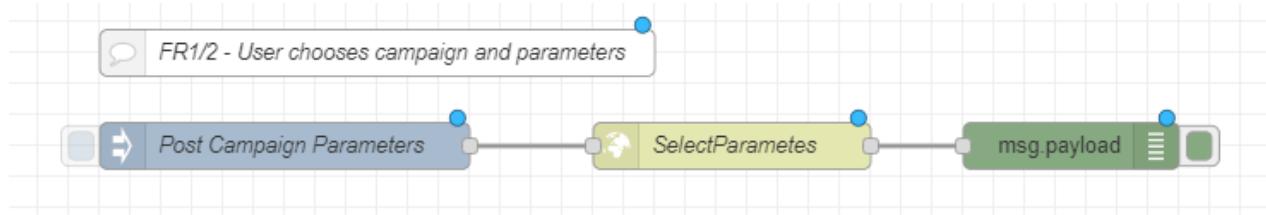
3.1.5.1. Ροή endpoint get /user/:UserID/contract/:ContractID/bot/:botid



Η ροή μέσω της οποίας υλοποιείται η υπηρεσία, η οποία επιστρέφει το προφίλ του επιλεγμένου bot.

3.2. Υλοποίηση Ιστοριών χρήστη

3.2.1. Ιστορία Χρήστη user creates contract



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Campaign Parameters	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
SelectParameters	http-request	Κάνει κλήση προς την υπηρεσία /user/:UserID για την δημιουργία συμβολαίου.
msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος .

3.2.2. Ιστορία Χρήστη user sends Feedback

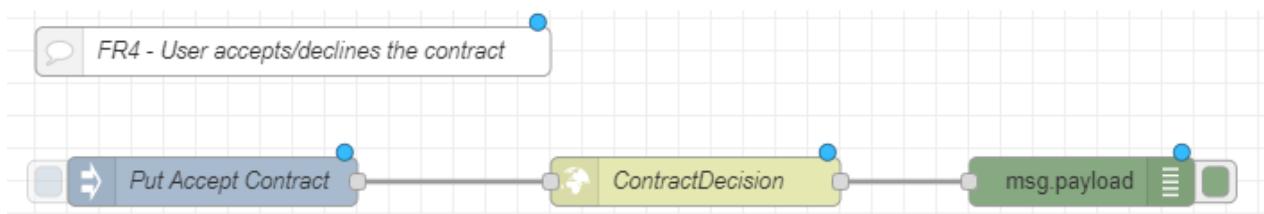


Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Post feedback	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
Send feedback	http-request	Κάνει κλήση προς την υπηρεσία /user/feedback και αποστέλλει το feedback του χρήστη.



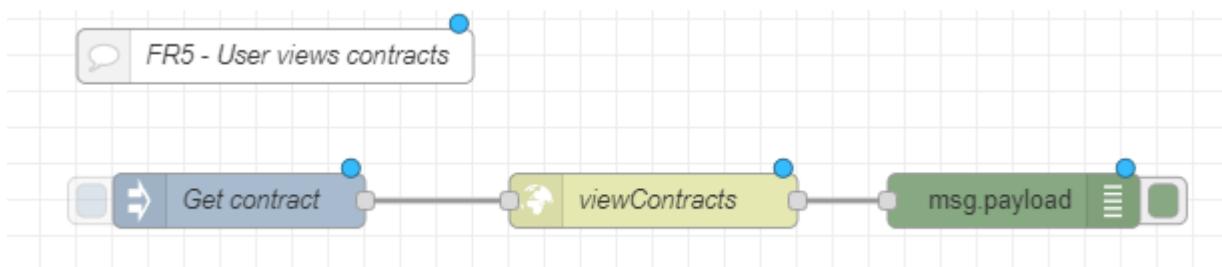
Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος .

3.2.3. Ιστορία Χρήστη user accepts/declines contract



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Put Accept Contract	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
ContractDecision	http-request	Κάνει κλήση προς την υπηρεσία /user/:UserID/contract/:ContractID, η οποία εμφανίζει το συμβόλαιο με το συγκεκριμένο ID.
msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος στην κλήση του χρήστη, δηλαδή το συμβόλαιο που θα δεχτεί ή όχι ο χρήστης.

3.2.4. Ιστορία Χρήστη user views contracts

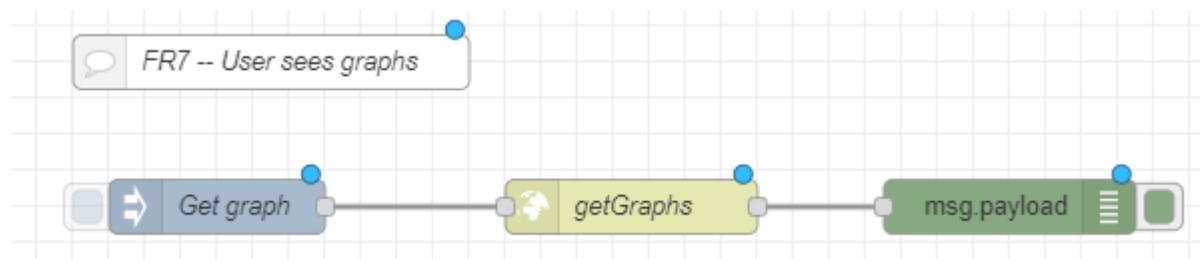


Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Get Contract	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της



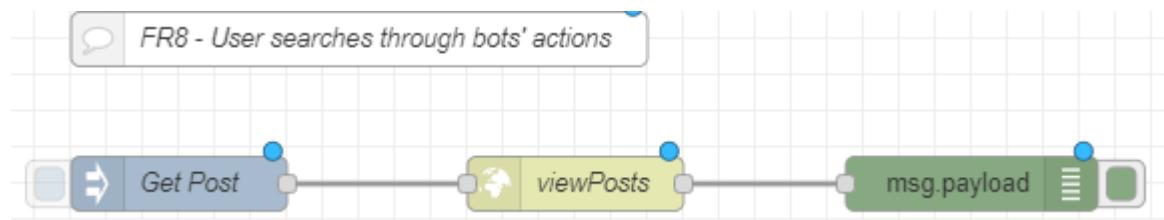
Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
		ροής.
viewContracts	http-request	Κάνει κλήση προς την υπηρεσία /user/:UserID/contract.
msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος., δηλαδή τα συμβόλαια του χρήστη με το συγκεκριμένο UserID

3.2.5. Ιστορία Χρήστη user sees statistics



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Get graph	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
getGraphs	http-request	Κάνει κλήση προς την υπηρεσία /user/:UserID/contract/:ContractID/graph/:GraphID.
msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος, δηλαδή το γράφημα που αντιστοιχεί στο συγκεκριμένο GraphID.

3.2.6. Ιστορία Χρήστη user sees bot's actions





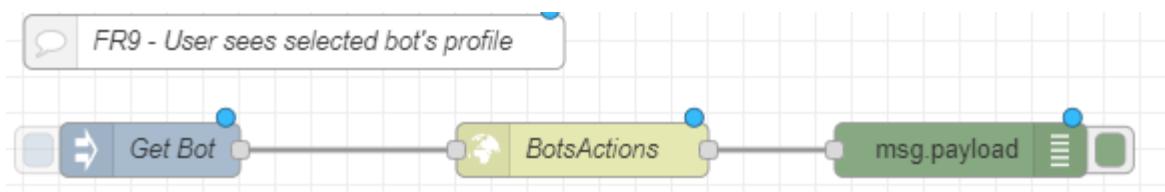
Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Get Post	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
viewPosts	http-request	Κάνει κλήση προς την υπηρεσία /user/:UserID/contract/:ContractID/post.
msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος, δηλαδή τα posts που αντιστοιχούν στο συγκεκριμένο ContractID.

3.2.7. Ιστορία Χρήστη user searches through bot's actions



Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Put Edit Filters	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
EditFilters	http-request	Κάνει κλήση προς την υπηρεσία /user/:UserID/contract/:ContractID/post και εισάγει τα φίλτρα που επιλέγει ο χρήστης.
msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος.

3.2.8. Ιστορία Χρήστη user sees a bot's profile





Όνομα κόμβου	Τύπος κόμβου	Περιγραφή
Get Bot	Inject	Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής.
BotsActions	http-request	Κάνει κλήση προς την υπηρεσία /user/:UserID/contract/:ContractID/post/:BotID.
msg.payload	debug	Τυπώνει στην κονσόλα το response του συστήματος, δηλαδή το συγκεκριμένο bot που αντιστοιχεί στο BotID.



Παράρτημα II – Ανοιχτά Θέματα

- Τα πρότυπα σχεδίασης που χρησιμοποιήθηκαν δεν καλύπτουν όλες τις μη λειτουργικές απαιτήσεις του συστήματός μας.
- Απαιτείται εξέλιξη των κλάσεων του προηγούμενου παραδοτέου, ώστε η σχεδίαση της εφαρμογής να γίνει πιο αποδοτική με βάση το προσδοκώμενο αποτέλεσμα.
- Στο σύστημα που αναπτύχθηκε δεν λάβαμε υπόψη, ακολουθώντας τις οδηγίες που δόθηκαν, τις τετριμμένες διεπαφές (εγγραφή χρήστη, αποσύνδεση χρήστη, αλλαγή κωδικού χρήστη κλπ.)
- Στο σύστημα δεν ελήφθησαν υπόψη σημαντικοί περιορισμοί που αφορούν κόστη χρήσης κλπ.)
- Στο σύστημα δεν λάβαμε υπόψη το σύστημα που θα ειδοποιεί τον χρήστη για το ανέβασμα του συμβολαίου του.