|  |
| --- |
|  |

# Icon Description automatically generated

VOLTApp  
*Your local tour guide*

# Προδιαγραφές Λογισμικού και Υλοποίηση συστήματος

# Del.1.2

Version 0.6

**Αντώνιος Αντωνίου** aantonii@ece.auth.gr

**Γιαννουρής Πολύδωρος** polydoros@ece.auth.gr

**Γρηγοράκης Ευθύμιος** eegrigor@ece.auth.gr

**Καϊμακαμίδης Ανέστης** anestisk@ece.auth.gr

04/06/2022

Ιστορικό Αλλαγών

|  |  |  |  |
| --- | --- | --- | --- |
| **Όνομα** | **Ημερομηνία** | **Αλλαγή** | **Έκδοση** |
| Α. Συμεωνίδης | 17/05/2007 | Δημιουργία εγγράφου. Προσαρμογή των προτύπων του K. E. Wiegers[[1]](#footnote-1) και του M. Smialek’s. | 0.1 |
| Α. Συμεωνίδης | 29/3/2014 | Μικρή αναθεώρηση – τροποποίηση ενοτήτων | 0.1.3 |
| Χ. Ζολώτας | 10/4/2020 | Μεγάλη αναθεώρηση – αφαίρεση ενοτήτων | 0.4 |
| Χ. Ζολώτας | 15/4/2020 | Μεγάλη αναθεώρηση – προσθήκη ενότητας REST προδιαγραφών | 0.5.3 |
| Κ. Παναγιώτου | 25/4/2020 | Μεγάλη αναθεώρηση – προσθήκη ενότητας Nodered περιγραφής | 0.5.7 |
| Α. Συμεωνίδης | 30/4/2020 | Αναθεώρηση και τελική δομή προτύπου | 0.6 |

Μέλη της Ομάδας Ανάπτυξης

|  |  |  |
| --- | --- | --- |
| **Όνομα** | **OA** | **Email** |
| Α. Αντωνίου | Voltapp | aantonii@ece.auth.gr |
| Π. Γιαννουρής | Voltapp | polydoros@ece.auth.gr |
| Ε. Γρηγοράκης | Voltapp | eegrigor@ece.auth.gr |
| Α. ΚαΪμακαμίδης | Voltapp | anestisk@ece.auth.gr |
| Α. Αντωνίου | Voltapp | aantonii@ece.auth.gr |
|  |  |  |

# Πίνακας Περιεχομένων

Πίνακας Περιεχομένων [3](#__RefHeading___Toc39140971)

Λίστα Σχημάτων [3](#__RefHeading___Toc39140972)

1. Πρότυπα Σχεδιασμού που υιοθετήθηκαν [4](#__RefHeading___Toc39140973)

2. Αρχιτεκτονική Συστήματος [5](#__RefHeading___Toc39140974)

2.1 Αναγνώριση Πόρων (Resources) Συστήματος [5](#__RefHeading___Toc39140975)

2.2 Τεκμηρίωση REST διεπαφής [5](#__RefHeading___Toc39140976)

2.2.1 Πόρος Pet [6](#__RefHeading___Toc39140977)

2.2.1.1 Μοντέλο δεδομένων Pet [6](#__RefHeading___Toc39140978)

2.2.1.2 Endpoint πόρου Pet [6](#__RefHeading___Toc39140979)

3. Υλοποίηση Συστήματος με Node-RED [9](#__RefHeading___Toc39140980)

3.1 Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed [9](#__RefHeading___Toc39140981)

3.1.1 Ροές Πόρου ΧΧ [9](#__RefHeading___Toc39140982)

3.2 Υλοποίηση της Εφαρμογής Πελάτη [9](#__RefHeading___Toc39140983)

3.2.1 Σενάριο Χρήσης ΧΧ [10](#__RefHeading___Toc39140984)

Παράρτημα I – Πίνακας Ιχνηλασιμότητας [11](#__RefHeading___Toc39140985)

Παράρτημα IΙ – Ανοιχτά Θέματα [12](#__RefHeading___Toc39140986)

# Λίστα Σχημάτων

Σχήμα 1. Λεζάντα Σχήματος

# Πρότυπα Σχεδιασμού που υιοθετήθηκαν

## Πρότυπο Observer

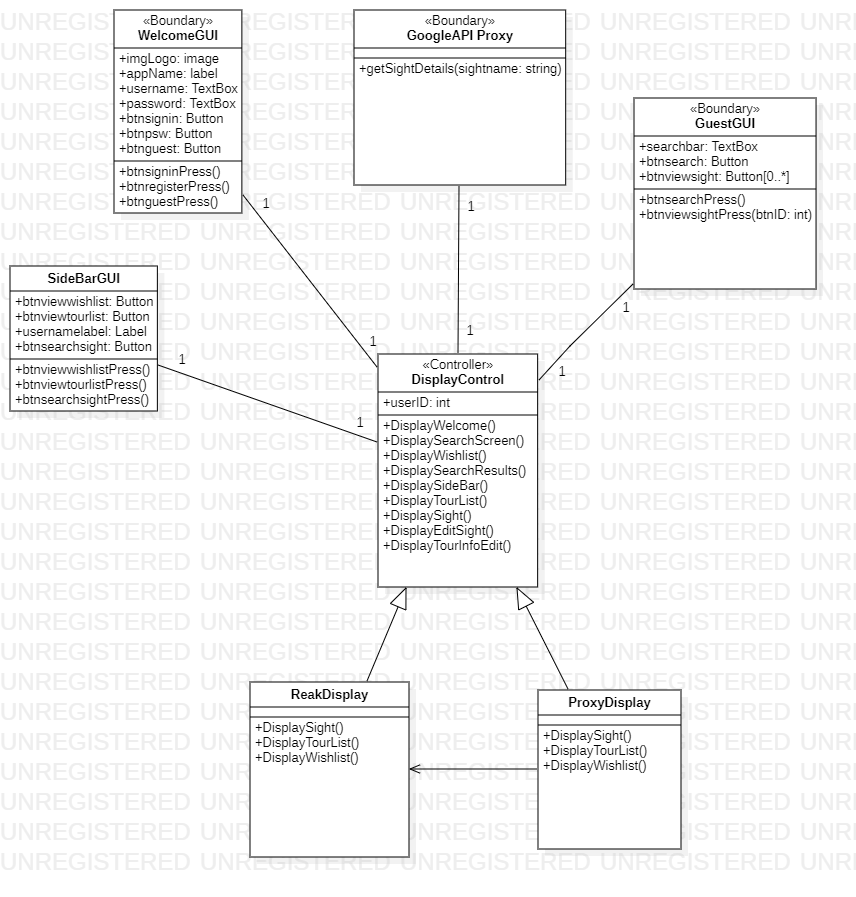
Το πρότυπο Observer αποτελεί ένα πρότυπο συμπεριφοράς. Η εφαρμογή του αφορά την απαίτηση για αυτόματη ενημέρωση του ρόλου ενός εγγεγραμμένου χρήστη σε ταξιδιωτικό οδηγό έπειτα από την εγγραφή σε συγκεκριμένο αριθμό από tours. Ως subject χρησιμοποιείται ο signed in controller, ενώ καθώς η αλλαγή του ρόλου αποτυπώνεται στη βάση δεδομένων προστέθηκε η αντίστοιχη συνάρτηση update και σε αυτή.

**

Σχήμα 1: Εφαρμογή προτύπου Observer

## Πρότυπο Proxy

Το πρότυπο proxy αποτελεί ένα δομικό πρότυπο. Η χρήση του ενδείκνυται για πολλά προβλήματα, στη συγκεκριμένη όμως εφαρμογή έχει ως σκοπό την εκπλήρωση της μη λειτουργικής απαίτησης για χρόνο απόκρισης μικρότερο των 150ms. Συγκεκριμένα η κλάση proxy θα περιέχει placeholder σχήματα τα οποία θα δίνουν την εντύπωση της γρήγορης απόκρισης, μέχρι να φορτώσουν τα επιθυμητά.



Σχήμα 2: Εφαρμογή προτύπου Proxy

## Πρότυπο Bridge

Το πρότυπο bridge αποτελεί ένα δομικό πρότυπο. Σκοπός του είναι να αποσυνδέσει ένα abstraction από την υλοποίησή του ώστε να μπορούν να μεταβάλλονται ανεξάρτητα. Στην εφαρμογή μας σκοπός του είναι η εκπλήρωση μίας νέας μη λειτουργικής απαίτησης, της χρήσης δεδομένων τόσο από το Google maps API όσο και από το Trip Advisor API.

Graphical user interface, text

Description automatically generated

Σχήμα 3: Εφαρμογή προτύπου Bridge

# Αρχιτεκτονική Συστήματος

Εδώ θα πρέπει να προδιαγράψετε το **REST API** της εφαρμογής που έχετε αναλύσει στα στάδια απαιτήσεων χρηστών και ιστοριών χρηστών. Το API σας θα πρέπει να καλύπτει όσο το δυνατόν μεγαλύτερο σύνολο των απαιτήσεων που καταγράψατε στις ιστορίες χρηστών.

Θα πρέπει να κατασκευάσετε και να παραδώσετε ένα λεπτομερές specification του API σας με βάση το swagger. Συγκεκριμένα, πρέπει να καταθέσετε ενεργούς υπερσυνδέσμους (Dropbox or Gdrive or Onedrive links) από τους οποίους θα μπορούμε να κατεβάσουμε τα εξής:

1. Ένα αρχείο τύπου **JSON** με τις προδιαγραφές του **API** σας όπως αυτό δημιουργείται από τον Swagger Editor (<https://editor.swagger.io/>) – οδηγίες στο elearning.
2. Ένα αρχείο τύπου **zip** το οποίο θα δημιουργηθεί από το Swagger Hub (<https://swagger.io/tools/swaggerhub/>) και θα περιέχει τον κώδικα για τον server της εφαρμογής σας – οδηγίες στο elearning.

## Αναγνώριση Πόρων (Resources) Συστήματος

Με βάση τις επιταγές της αρχιτεκτονικής REST αναγνωρίστε τους Πόρους του συστήματός σας (one-off, δεδομένων και αλγοριθμικούς – όπως ορίζονται στη διάλεξη «Εξειδικευμένοι Τύποι Αρχιτεκτονικής Client Server: REST») αξιοποιώντας τις κλάσεις που αναγνωρίσατε στο προηγούμενο παραδοτέο και καταγράψτε τους σε μορφή Πίνακα όπως ο ακόλουθος (χάρην σύνοψης).

|  |  |  |
| --- | --- | --- |
| **Κλάση BEC** | **Πόρος REST** | **Endpoints (HTTP Verbs)** |
|  |  |  |

## Τεκμηρίωση REST διεπαφής

Τεκμηριώστε εδώ τα διάφορα REST APIs για κάθε έναν πόρο που αναγνωρίσατε στην προηγούμενη ενότητα. Για την ανάπτυξη της συγκεκριμένης πληροφορίας προτείνεται να χρησιμοποιήσετε το <https://editor.swagger.io/>

Ένα καλό παράδειγμα σχετικής τεκμηρίωσης μπορείτε να δείτε φορτώνοντας και δοκιμάζοντας το αρχείο προδιαγραφών *Swagger Petstore* (προ-φορτώνεται κατά την εκκίνηση του Swagger editor).

Προσθέστε από μία υπό-ενότητα για κάθε Πόρο που αναγνωρίσατε. Σε κάθε μία από τις ενότητες αυτές προσθέστε την πληροφορία, όπως παράγεται από το Live documentation του Swagger editor, έχοντας εκτελέσει και κάθε endpoint του πόρου, καθώς και το αντίστοιχο μοντέλο δεδομένων του πόρου (Model section).

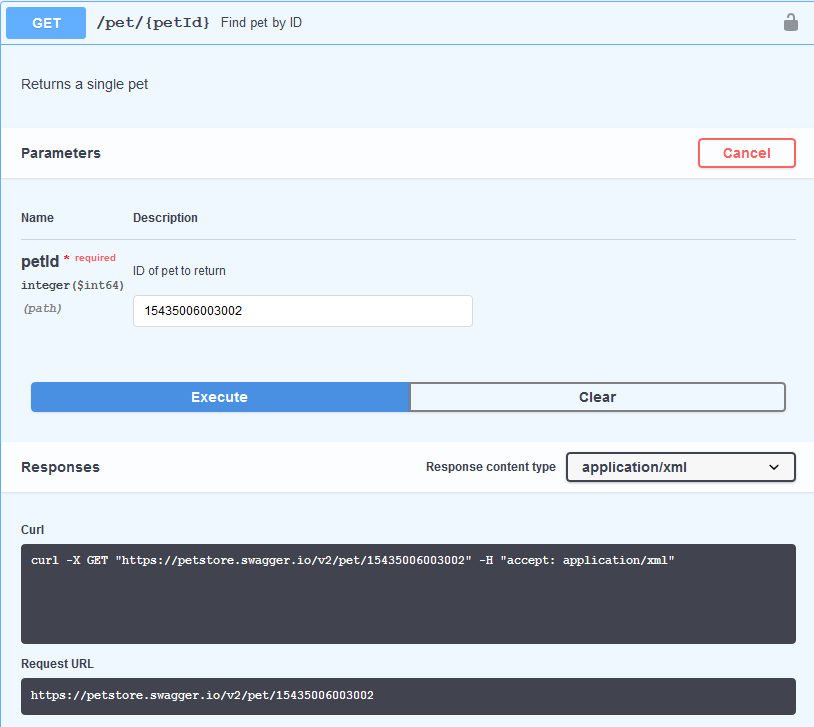
Δείτε χαρακτηριστικό παράδειγμα παρακάτω από το Petstore API (να διαγραφεί κατά την εισαγωγή της δική σας πληροφορίας):

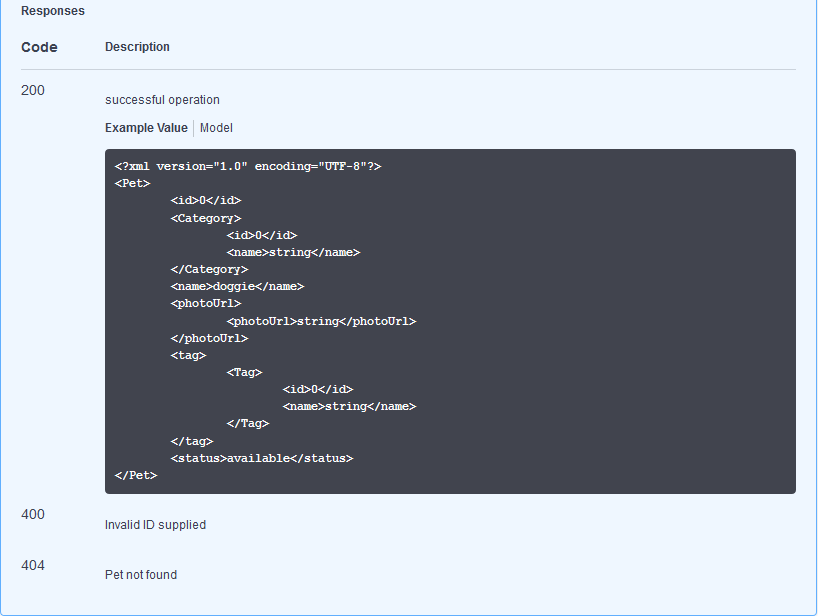
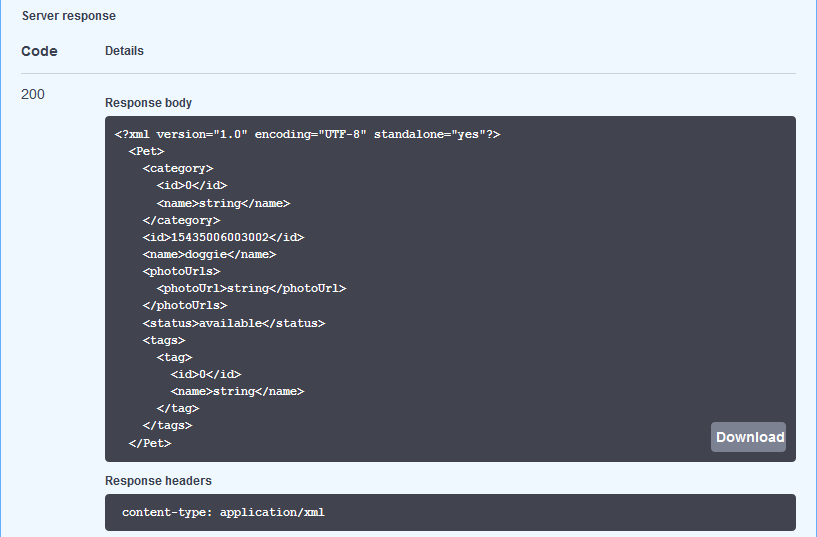
### Πόρος Pet

#### Μοντέλο δεδομένων Pet



#### Endpoint πόρου Pet





*<Συμπληρώνετε τα υπόλοιπα endpoints του πόρου και συνεχίζεται με τους υπόλοιπους πόρους σε επόμενες υποενότητες>*

# Υλοποίηση Συστήματος με Node-RED

## Αντιστοίχιση των REST Υπηρεσιών σε Ροές NodeRed

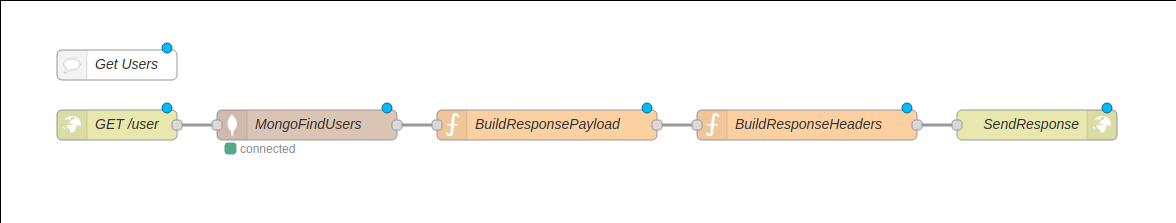
Σε αυτό το κεφάλαιο καλείστε να αντιστοιχίσετε τις Ιστορίες Χρηστών σε αντίστοιχες ροές (flows) στο NodeRED (https://nodered.org/). Για τον λόγο αυτό θα χρειαστεί να αντιστοιχήσετε κάθε REST endpoint σε μία ανεξάρτητη ροή NodeRED και να τις παρουσιάσετε. Στη συνέχεια, θα πρέπει να παρουσιάσετε τα NodeRED flows των Ιστοριών χρηστών.

Θα πρέπει να παραδώσετε τα flows σας (σε αρχείο τύπου .zip) κάνοντας εξαγωγή από το NodeRED περιβάλλον σας, δίνοντας σχετικό ενεργό υπερσύνδεσμο (Dropbox or Gdrive or Onedrive link).

### Ροές πόρου ΧΧ

Για κάθε endpoint υλοποιήστε το αντίστοιχο NodeRED flow και παραθέστε τη σχετική πληροφορία (δείτε οδηγίες στο elearning). Παρακάτω δίνετε παράδειγμα του πόρου endpoint get /user

Ροή endpoint get /user

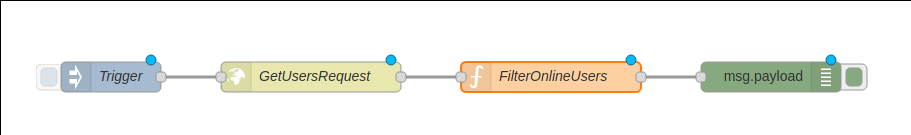
*<Συμπληρώνετε τα υπόλοιπα endpoints του πόρου και συνεχίζεται με τους υπόλοιπους πόρους σε επόμενες υποενότητες>*

## Υλοποίηση Ιστοριών χρήστη

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

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

### Ιστορία Χρήστη ΧΧ



*Εδώ γράφετε μια σύντομη περιγραφή της υλοποίησης.*

*Στη συνέχεια, καταγράψτε τους κόμβους που χρησιμοποιήθηκαν και περιγράψτε τις λειτουργίες τους σε πίνακα όπως παρακάτω.*

| Όνομα κόμβου | Τύπος κόμβου | Περιγραφή |
| --- | --- | --- |
| Trigger | Inject | Χρησιμοποιείται για την ενεργοποίηση της εκτέλεσης της ροής. |
| GetUsersRequest | http-request | Κάνει κλήση προς την υπηρεσία /user η οποία επιστρέφει την λίστα με τους εγγεγραμένους χρήστες στο σύστημα. |
| FilterOnlineUsers | function | Εφαρμόζεται φιλτράρισμα στα δεδομένα (λίστα χρηστών) και επιστρέφει την λίστα με τους ενεργούς χρήστες. |
| Msg.payload | debug | Τυπώνει στην κονσόλα την λίστα με τους ενεργούς χρήστες . |

# Παράρτημα I – Πίνακας Ιχνηλασιμότητας

<Εφόσον χρειάζεται, ορίστε έναν πίνακα Ιχνηλασιμότητας>.

# Παράρτημα IΙ – Ανοιχτά Θέματα

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

1.  Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document. Original template is available at: *http://www.processimpact.com/* [↑](#footnote-ref-1)