



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης  
Πολυτεχνική Σχολή  
Τμήμα Ηλεκτρολόγων Μηχανικών &  
Μηχανικών Υπολογιστών  
Τομέας Ηλεκτρονικής και Υπολογιστών

## Διπλωματική Εργασία

---

# Σύστημα για την Αυτόματη Παρακολούθηση του Χρόνου Λειτουργίας και Απόκρισης ενός Ιστότοπου

---

Εκπόνηση:  
Σεντονάς Σταύρος  
ΑΕΜ: 9386

Επίβλεψη:  
Υπ. Δρ. Καρανικιώτης Θωμάς  
Δρ. Παπαμιχαήλ Μιχαήλ  
Καθ. Συμεωνίδης Ανδρέας

Θεσσαλονίκη, Ιούνιος 2023



---

## Περίληψη

Η εξέλιξη της τεχνολογίας και της πληθώρας εφαρμογών που αναπτύσσονται στα πλαίσια αυτής, καθιστούν επιτακτική την ανάγκη ύπαρξης συστημάτων που θα ελέγχουν την εύρυθμη λειτουργία τους. Πιο συγκεκριμένα μιλάμε για την εξέλιξη στο χώρο του διαδικτύου και των δομών που έχουν υλοποιηθεί πάνω σε αυτό.

Πλέον αναφερόμαστε σε ένα συνεχώς αυξανόμενο και ευρύ δίκτυο web εφαρμογών - λογισμικών ως υπηρεσίας (SaaS - Software as a Service) που ζουν στον Διαδίκτυο (World Wide Web). Η λειτουργία αυτών μπορεί να ελεγχθεί με διάφορους τρόπους. Από Unit Testing, στο πλαίσιο του κύκλου ανάπτυξης του λογισμικού (continuous integration, continuous deployment cycle) προκειμένου να ελεγχθεί λειτουργικά το σύστημα για την αποφυγή bugs, μέχρι και Παρακολούθηση Δικτύου (Network Monitoring), για να επιβεβαιωθεί η σωστή λειτουργία των συστημάτων καθόλη της διάρκεια του κύκλου ζωής τους.

Η παρούσα διπλωματική εστιάζει στην ανάπτυξη ενός συστήματος Παρακολούθησης Δικτύου και κατεπέκταση εφαρμογής που θα δίνει της δυνατότητα στους χρήστες της να παρακολουθούν, εύκολα, την ομαλή λειτουργία των διαδικτυακών σελιδών τους, είτε αυτά είναι εφαρμογές, είτε απλά στατικές σελίδες. Το σύστημα στηρίζεται στη βασική μέθοδο εντοπισμού διαθεσιμότητας μίας ιστοσελίδας, γνωστή και ως ping. Κάνοντας ping μπορούμε να πάρουμε χρήσιμη πληροφορία σχετικά με το αν το υπό μελέτη σύστημα μπορεί να ανταποκριθεί και εφόσον ανταποκριθεί σχετικά με το χρόνο που μεσολάβησε μέχρι να απαντήσει. Συνεχίζοντας την λογική πορεία ενός τέτοιου συστήματος μπορούμε ακόμα στο μήνυμα που στέλνουμε να έχουμε πληροφορία που θα επηρεάζει την απάντηση που θα περιμέναμε να δούμε, έχοντας έτσι έναν ακόμα μηχανισμό για την αναγνώριση και αποφυγή πιθανών bugs, ή λαθών κατά τη διαδικασία ανάπτυξης λογισμικών ως υπηρεσία.

---

# Title

## Development of a System for Uptime Status Monitoring

### Abstract

The evolution of technology and the abundance of applications developed within this framework make it imperative to have systems that will control their smooth operation. More specifically, we are talking about monitoring in the realm of the internet and the structures implemented on it.

Today we refer to a constantly growing and extensive network of web applications - software as a service (SaaS) - that reside on the Internet (World Wide Web), whose operations can be monitored in various ways. From unit testing, within the software development cycle (continuous integration, continuous deployment) to ensure that the system functions are running properly and avoid bugs, to network monitoring, to verify the correct functioning of the systems throughout their life cycle.

This thesis focuses on the development of a Network Monitoring system and a web application that will enable its users to easily monitor how their systems operate, whether they are applications or simply static pages. The system is based on the basic method of checking the availability of a website, known as ping. By pinging, we can obtain useful information regarding whether the system under study can respond and, if so, the time it takes to respond. Continuing the logical progression of such a system, we can further enhance the message we send with predetermined data to check the response of the system and verify the returned data, thereby identifying and avoiding potential bugs or error during the software's development process

Sentonas Stavros

Electrical & Computer Engineering Department,  
Aristotle University of Thessaloniki, Greece

June 2023



# Περιεχόμενα

Περίληψη . . . . .	3
Abstract . . . . .	4
Ακρωνύμια . . . . .	9
<b>1 Εισαγωγή</b>	<b>10</b>
1.1 Περιγραφή του Προβλήματος . . . . .	11
1.2 Σκοπός - Συνεισφορά της Διπλωματικής Εργασίας . . . . .	12
1.3 Διάρθρωση της Αναφοράς . . . . .	12
<b>2 Θεωρητικό Υπόβαθρο</b>	<b>14</b>
2.1 Hypertext Transfer Protocol . . . . .	14
2.1.1 Μέθοδοι . . . . .	15
2.1.2 HTTP versions . . . . .	15
2.1.3 Κωδικοί Κατάστασης . . . . .	16
<b>Βιβλιογραφία</b>	<b>19</b>

# Κατάλογος Σχημάτων

2.1	Βασική Δομή ενός αιτήματος <a href="#">http</a> . . . . .	15
-----	---	----

# Κατάλογος Πινάκων

1.1	Χαρακτηριστικά Ενεργής και Παθητικής Παρακολούθησης . . . . .	11
-----	---	----



# Ακρωνύμια Εγγράφου

Παρακάτω παρατίθενται ορισμένα από τα πιο συχνά χρησιμοποιούμενα ακρωνύμια της παρούσας διπλωματικής εργασίας:

RUM → Real User Monitoring  
API → Application Programming Interface  
SaaS → Software as a Service  
HTTP → Hypertext Transfer Protocol

# 1

## Εισαγωγή

Τα τελευταία χρόνια, ο κλάδος του Διαδικτύου προσεγγίζει ένα μεγαλύτερο κομμάτι ανθρώπων, τόσο από τη μεριά του καταναλωτή όσο και από τη μεριά του παραγωγού. Όσο αφορά τον καταναλωτή οι δυνατότητες που του προσφέρονται μπορούν να διακριθούν στους εξής τομείς:

- **Επικοινωνία:** το διαδίκτυο παρέχει τη δυνατότητα άμεσης επικοινωνίας μεταξύ μεγάλων αποστάσεων, που δεν περιορίζεται μόνο στο ακουστικό ερέθισμα, αλλά επιτρέπει και την μετάδοση οπτικο-ακουστικής πληροφορίας
- **Πρόσβαση Πληροφορίας:** ίσως το σημαντικότερο αγαθό που προσφέρει το διαδίκτυο είναι η πληθώρα πληροφορίας που στεγάζει. Μηχανές Αναζήτηση (search engines), Online Βάσεις Δεδομένων (online databases), και άλλου είδους εφαρμογών εκπαιδευτικού χαρακτήρα που δίνουν πρόσβαση σε άτομα που το επιθυμούν, να κάνουν έρευνα
- **Ποιότητα ζωής:** σε αυτή την κατηγορία περιλαμβάνονται όλες εκείνες οι υπηρεσίες που διευκολύνουν την καθημερινότητα των χρηστών. Online αγορές (eshops) που γλιτώνουν την αναμονή σε ουρές ή ακόμα επιτρέπουν την εύκολη αγορά προϊόντων από απομακρυσμένες περιοχές του πλανήτη, ψυχαγωγία και πρόσβαση σε υπηρεσίες που επιταχύνουν ενέργειες που υπό άλλες περιπτώσεις θα ήταν χρονοβόρες (online banking, πληρωμή λογαριασμών, κρατήσεις ξενοδοχείων/εισητηρίων)

Από τη μεριά του παραγωγού, τα μέσα που υπάρχουν για την ανάπτυξη τέτοιων εφαρμογών/υπηρεσιών/συστημάτων μέρα με τη μέρα αυξάνονται. Η ραγδαία εξέλιξη στον χώρο των cloud υποδομών, καθιστά ευκολότερη και επισπεύδει τόσο την δημιουργία διαδικτυακών εφαρμογών, και σελιδών σε ένα γενικότερο πλαίσιο, όσο και την μεγέθυνση και αύξηση αυτών (scale up). Μάλιστα η επιλογή κατάλληλου παρόχου τέτοιων υπηρεσιών αποτελεί ένα αρκετά σημαντικό αντικείμενο μελέτης

[1]. Πέρα από τον οικονομικό παράγοντα θα πρέπει να προσμετρηθούν οι παροχές, τα πλεονεκτήματα αλλά και η αποδοτικότητα που κάθε ένας προσφέρει.

Βλέποντας λοιπόν το πόσο συνυφασμένη είναι η ζωή του σύγχρονου ανθρώπου με το δίκτυο αλλά και τις δυνατότητες και τα μέσα που έχει ο καθένας για να αναπτύξει εφαρμογές σε αυτό, καθίσταται επιτακτική η ανάγκη ύπαρξης μηχανισμών που θα αναγνωρίζουν σφάλματα (bugs) και θα επιβλέπουν την ορθή λειτουργία των υπό μελέτη συστημάτων καθόλη τη διάρκεια ζωής τους.

## 1.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Η Παρακολούθηση (Monitoring) ενός συστήματος που "ζει" στο χώρο του διαδικτύου μπορεί να γίνει κυρίως με δύο τρόπους:

- **Ενεργή Παρακολούθηση (Active Monitoring):** έχει περισσότερο προγνωστικό και προληπτικό χαρακτήρα. Συχνά αναφέρεται και ως **Συνθετική παρακολούθηση (Synthetic Monitoring)**, λόγω της φύσης των ενεργειών της. Ουσιαστικά δημιουργεί πλασματικά api calls και όχι πραγματικά δεδομένα χρηστών προκειμένου να ελεγχθεί η απόκριση του υπό μελέτη συστήματος. Η συχνότητα αποστολής των συνθετικών αιτημάτων συνήθως ρυθμίζεται από το χρήστη.
- **Παθητική Παρακολούθηση (Passive Monitoring):** παρέχει μία πιο πλήρη εικόνα σχετικά με πως χρησιμοποιούνται οι πόροι του δικτύου καταγράφοντας, αποθηκεύοντας και αναλύοντας τα δεδομένα του χρήστη. Για αυτό πολλές φορές αναφέρεται στη βιβλιογραφία ως **Παρακολούθηση Πραγματικών Χρηστών (Real User Monitoring - RUM)**. Έτσι μπορεί κανείς να εντοπίσει τις τάσεις χρήσης του δικτύου για τη βελτίωση και βελτιστοποίησή του συστήματος.

Πίνακας 1.1: Χαρακτηριστικά Ενεργής και Παθητικής Παρακολούθησης

Ενεργή Παρακολούθηση (Active Monitoring)	Παθητική Παρακολούθηση (Passive Monitoring)
<ul style="list-style-type: none"> <li>• Στηρίζεται σε συνθετικά API calls</li> <li>• Παράγει δεδομένα για συγκεκριμένες πτυχές του δικτύου</li> <li>• Μπορεί να μετρήσει την κίνηση εντός και εκτός του δικτύου</li> <li>• Μπορεί να εντοπίσει προβλήματα πριν ακόμα μπορέσουν να τα εντοπίσουν οι χρήστες</li> </ul>	<ul style="list-style-type: none"> <li>• Αναλύει δεδομένα πραγματικών χρηστών</li> <li>• Πλήρης εικόνα της απόδοσης του δικτύου</li> <li>• Μετράει κίνηση μόνο εντός του δικτύου</li> <li>• Εντοπίζει προβλήματα που εμφανίζονται εκείνη τη στιγμή</li> </ul>

Και οι δύο μέθοδοι έχουν πλεονεκτήματα και μειονεκτήματα, τα οποία φαίνονται και στον παραπάνω πίνακα 1.1. Όπως είναι εμφανές η παθητική παρακολούθηση γίνεται πάνω στο σύστημα που θέλουμε να μελετήσουμε, πράγμα το οποίο σήμαινει ότι σαν εξωτερικοί παράγοντες στο σύστημα δεν θα μπορέσουμε να προσφέρουμε ανάλογες υπηρεσίες. Για το λόγο αυτό συνεχίζουμε την ανάλυση στο πλαίσιο της Ενεργής Παρακολούθησης Δικτύων.

Οι βασικοί λόγοι που χρειάζονται τέτοιου είδους υπηρεσίες όπως αναφέρεται και στα [2], [3] είναι οι εξής:

- Βελτίωση προβλημάτων που σχετίζονται με την απόδοση του συστήματος πρώτου τα βιώσουν οι πραγματικοί χρήστες του συστήματος
- Ύπαρξη κάποιας μονάδας αξιολόγησης της απόδοσης του
- Αξιολόγηση του συστήματος υπό μεγαλύτερο φορτίο
- Διασφάλιση της Συμφωνίας Επιπέδου Υπηρεσιών (Service Level Agreement - SLA), μεταξύ του παρόχου υπηρεσιών και των χρηστών
- Παρέχει χρήσιμα δεδομένα ακόμα και σε καινούργια συστήματα που ακόμα μπορεί να μην έχουν χρήστες

### 1.2 ΣΚΟΠΟΣ - ΣΥΝΕΙΣΦΟΡΑ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

---

Η παρούσα διπλωματική εργασία μελετά τη χρήση σύγχρονων τεχνολογιών για τη δημιουργία ενός συστήματος Ενεργής Παρακολούθησης (Active Monitoring) σε συνδυασμό με μία SaaS εφαρμογή που θα παρουσιάζει μέσα από διαγράμματα τα αποτελέσματα της ανάλυσης της πληροφορίας που εξάγεται.

Εξετάζονται διάφοροι τρόποι και υλοποιήσεις που δοκιμάστηκαν κατά τη διάρκεια εκπόνησης της διπλωματικής αυτής εργασίας, και τέλος θα αναλύσουμε τα αποτελέσματα που παράξαμε καθόλη της διάρκεια των πειραμάτων που διενεργήθηκαν.

### 1.3 ΔΙΑΡΘΡΩΣΗ ΤΗΣ ΑΝΑΦΟΡΑΣ

---

Η διάρθρωση της παρούσας διπλωματικής εργασίας είναι η εξής:

- **Κεφάλαιο 2:** Περιγράφονται τα βασικά εργαλεία και θεωρητικά στοιχεία στα οποία βασίστηκαν οι υλοποιήσεις
- **Κεφάλαιο 3** Αναφορά συστημάτων που ήδη χρησιμοποιούνται και παράθεση διαφορών με την υλοποίησή μας
- **Κεφάλαιο 4** Περιγραφή των υλοποιήσεων και πλήρης περιγραφή του τελικού συστήματος
- **Κεφάλαιο 5** Παρουσιάζονται τα τελικά συμπεράσματα.

- **Κεφάλαιο 6** Προτείνονται θέματα για μελλοντική μελέτη, αλλαγές και επεκτάσεις.

# 2

## Θεωρητικό Υπόβαθρο

Στο κεφάλαιο αυτό θα παρουσιαστούν εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του Συστήματος Ενεργής Παρακολούθησης, καθώς και έννοιες και τεχνολογίες που αξιοποιήθηκαν για το σκοπό αυτό.

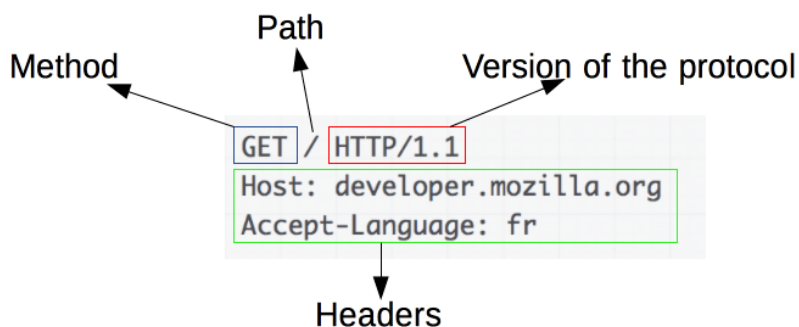
### 2.1 HYPERTEXT TRANSFER PROTOCOL

---

Το πρωτόκολλο επικοινωνίας HTTP (Hypertext Transfer Protocol) αποτελεί το πιο διαδεδομένο και ευρέως γνωστό πρωτόκολλο στο χώρο του διαδικτύου. Αναπτύχθηκε από τους Tim Berners-Lee και την ομάδα του το 1990 και από τότε έχει περάσει πολλές αλλαγές προκειμένου να μπορεί να ανταπεξέλθει στις ολοένα και συνεχώς αυξανόμενες ανάγκες του σήμερα.

Αποτελεί τη βάση κάθε μετάδοσης πληροφορίας στο διαδίκτυο. Στηρίζεται στην επικοινωνία δύο υπολογιστών, ενός που κάνει τα αιτήματα (client) και ενός που απαντά σε αυτά (server). Στο τέλος της επικοινωνίας στην μεριά του παραλήπτη θα υπάρχει ανακατασκευασμένο ένα ολοκληρωμένο αρχείο, από τα διάφορα υποαρχεία που μαζεύτηκαν, που μπορεί να είναι αρχεία ήχου, εικόνας, video. Τα αιτήματα αυτού που ξεκινάει την επικοινωνία ονομάζονται requests, ενώ οι απαντήσεις του αποστολέα responses.

Η βασική δομή ενός http αιτήματος [σχήμα 2.1](#) περιληπτικά περιλαμβάνει τη μέθοδο (method) του αιτήματος, που περιγράφει τη βασική λειτουργία του, το μονοπάτι (path) στο οποίο θα επικοινωνήσει με τον server, την έκδοση του πρωτοκόλλου που θα χρησιμοποιηθεί και τέλος headers προκειμένου να κρίνει ο server αν πρέπει να απαντήσει ή όχι πίσω στον client



Σχήμα 2.1: Βασική Δομή ενός αιτήματος http

### 2.1.1 Μέθοδοι

Πιο συγκεκριμένα οι βασικές μέθοδοι που παρέχει το http και οι συνήθεις λειτουργίες τους είναι οι εξής:

- **GET**: παίρνει πληροφορία από τον server
- **POST**: υποβάλλει πληροφορία, προκαλώντας αλλαγές στον τρόπο λειτουργίας του server. Σχετίζεται συχνά με τη δημιουργία πληροφορίας που προηγουμένως δεν υφίστατο
- **PUT**: όπως και πριν στέλνει πληροφορία στον παραλήπτη υπολογιστή, αλλά αυτή τη φορά επηρεάζει πόρους που ήδη υπήρχαν στο σύστημα. Σχετίζεται συχνά με την τροποποίηση ήδη υπάρχουσας πληροφορίας
- **DELETE**: διαγράφει από το σύστημα του server το συγκεκριμένο πόρο.

Αξίζει να σημειωθεί ότι πέρα από τις τέσσερις αυτές βασικές μεθόδους υπάρχουν και άλλες όπως είναι η **PATCH** που αποτελεί ειδική περίπτωση της **PUT**, η **HEAD** που αποτελεί ειδική περίπτωση της **GET**, καθώς και άλλες που σχετίζονται με τη σύνδεση μεταξύ server και client. Αυτές είναι οι **CONNECT**, **OPTIONS** και **TRACE**

### 2.1.2 HTTP versions

Η πρώτη έκδοση του HTTP παρόλο που δεν είχε κάποια συγκεκριμένο τίτλο, εκ των υστέρων το αποκάλεσαν HTTP/0.9. Αποτελεί την πιο απλή έκδοση του πρωτοκόλλου. Δεν υποστηρίζονταν headers, κωδικοί κατάστασης (status codes). Εξυπηρετούσε μόνο GET αιτήματα και η μοναδική απάντηση που μπορούσε να επιστρέψει ήταν hypertext αρχεία. Κάθε φορά που ο server ανταποκρινόταν και έστελνε απάντηση η επικοινωνία με τον client έκλεινε κατευθείαν.

Στη συνέχεια και με την ανάπτυξη του διαδικτύου προστέθηκαν και άλλες λειτουργίες. Πέριξ του 1996, με τη επόμενη έκδοση του πρωτοκόλλου (HTTP/1.0) τα αιτήματα πλέον συνοδεύονταν από headers, μεταπληροφορία σχετικά με τη

κατάσταση του αιτήματος, τον τύπο της πληροφορίας που περιμένουμε να έρθει (stylesheets, media, hypertext) καθώς και την έκδοση του HTTP που χρησιμοποιήθηκε στη συγκεκριμένη επικοινωνία. Επιπλέον πέρα από τη GET μέθοδο υπάρχει η δυνατότητα για POST και PUT, δημιουργία και τροποποίηση πληροφορίας δηλαδή.

Στη συνέχεια το HTTP/1.1 προσπαθεί να βελτιώσει τις ήδη υπάρχουσες δυνατότητες κάνοντας την επικοινωνία μεταξύ server και client πιο αποδοτική. Αντί να κλείνει η επικοινωνία μετά από κάθε μήνυμα, η σύνδεση παραμένει ανοιχτή γλιτώνοντας έτσι μία σταθερή καθυστέρηση που υπήρχε σε κάθε αίτημα

Φτάνοντας στο σήμερα, μιλάμε για το HTTP/2.0 [4]. Αξιοποιώντας το πρωτόκολλο SPDY (Speedy) που αναπτύχθηκε κάποια χρόνια πριν την κυκλοφορία του, και κτίζοντας πάνω σε αυτό, κατάφερε να μειώσει τους χρόνους επικοινωνίας server-client. Μερικοί από τους τρόπους που επιτυγχάνεται αυτό είναι η μετατροπή του http από text πρωτόκολλο, σε δυαδικό (binary protocol), επιτρέποντας έτσι χρήση καλύτερων και αποδοτικότερων τεχνικών επικοινωνίας. Επιπλέον συμπιέζει τους headers (header compression) καθώς αποτελούν πληροφορία που επαναλαμβάνεται όταν τα αιτήματα στον server είναι συνεχή. Ο server ακόμα, αποκτά έναν μηχανισμό (server-push) που του επιτρέπει να προωθεί πληροφορία στον client (στην cache του client συγκεκριμένα), που δεν έχει ζητήσει ακόμα, αλλά βάση αυτού που αιτήται, μάλλον θα ζητήσει εντός του ίδιας συνεδρίας.

Τέλος, πρέπει να αναφερθούμε στην τελευταία, αν και όχι ακόμα ευρέως διαδεδομένη, έκδοση HTTP/3.0. Η βασική διαφορά με τους προκάτοχους του είναι ότι αλλάζει το πρωτόκολλο επικοινωνίας που χρησιμοποιεί όλα αυτά τα χρόνια, από TCP (Transfer Communication Protocol) σε έναν συνδυασμό UDP (User Datagram Protocol) και QUIC, μίας νέας τεχνολογίας που λύνει το πρόβλημα και βελτιστοποιεί τόσο το πρόβλημα της ασφάλειας των επικοινωνιών (TLS handshakes), όσο και της απώλειας πληροφορίας που μπορεί να υπήρχε λόγω UDP, πρωτοκόλλου που είναι γνωστό για την ταχύτερη απόδοσή του σε σχέση με το TCP, αλλά και το γεγονός ότι είναι πιο επιρρεπές σε σφάλματα. Η νέα αυτή έκδοση από τα αποτελέσματα του [5] φαίνεται να έχει ήδη καλύτερους χρόνους σε σχέση με τις παλαιότερες εκδόσεις και ήδη το 28% του διαδικτύου αξιοποιεί τις δυνατότητές του.

### 2.1.3 Κωδικοί Κατάστασης

Οι κωδικοί κατάστασης (status codes) αποτελούν μέρος της απάντησης του server. Σου επιτρέπουν με μία ματιά να καταλάβεις αν το αίτημα που έχεις κάνει έχει επιστρέψει σωστά, ή έχει γίνει κάποιο λάθος στη μεριά του server. Υπάρχουν πέντε μεγαλύτερες κατηγορίες που στεγάζουν όλες τις υποπεριπτώσεις αυτών. Πιο συγκεκριμένα:

- **Εύρος 100-199:** Υποδηλώνουν ενημερωτική απάντηση σχετικά με τη λειτουργία του server
- **Εύρος 200-299:** Επιτυχή αιτήματα.
- **Εύρος 300-399:** Υποδηλώνουν την ανακατεύθυνση του μηνύματος του client. Συνήθως συνοδεύονται από το νέο url στο οποίο πρέπει να αποστείλουν το αίτημα



- **Εύρος 400-499:** Ανεπιτυχές αίτημα, που οφείλεται στον client. Ένα σύννηθες παράδειγμα είναι αίτησης πρόσβασης σε προστατευόμενους πόρους χωρίς κάποιου είδους αυθεντικοποίησης, ή χωρίς τα σωστά στοιχεία για αυθεντικοποίηση
- **Εύρος 500-599:** Ανεπιτυχές αίτημα, που οφείλεται στον server.



## Βιβλιογραφία

- [1] Santhosh S and Narayana Swamy Ramaiah. “*Cloud-Based Software Development Lifecycle: A Simplified Algorithm for Cloud Service Provider Evaluation with Metric Analysis*“. Big Data Mining and Analytics, 6(2):127–138, june 2023.
- [2] Sorin POPA. “*WEB Server monitoring*“. Annals of University of Craiova - Economic Sciences Series, 2(36):710–715, may 2008.
- [3] Siddhesh Vaidya and Prabhat Padhy. “*View towards Synthetic Monitoring using HTTP Archive*“. International Journal of Engineering Research and Technology, august 2022.
- [4] Daniel Stenberg. “*HTTP2 explained*“. ACM SIGCOMM Computer Communication Review, 44:120–128, 07 2014.
- [5] Martino Trevisan, Danilo Giordano, Idilio Drago, and Ali Safari Khatouni. “*Measuring HTTP/3: Adoption and Performance*“, 02 2021.