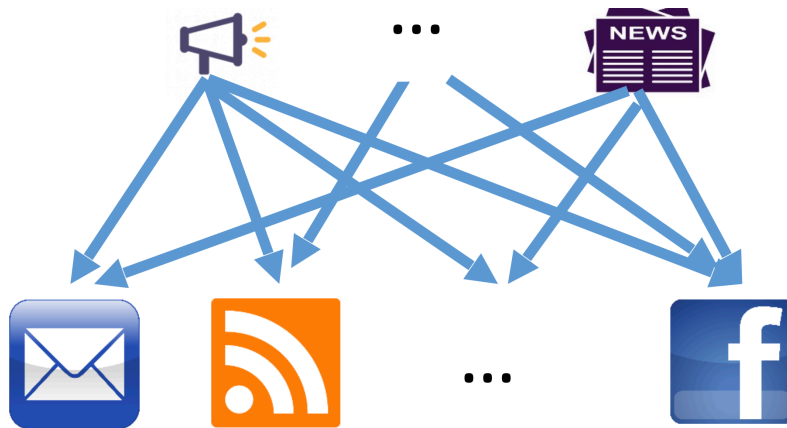


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
«20154 – ΣΥΣΤΗΜΑ ΔΙΑΔΟΣΗΣ ΕΙΔΟΠΟΙΗΣΕΩΝ»



Του φοιτητή
Αντώνιου Μπάρμπα
Αρ. Μητρώου: 154504

Επιβλέπων
Αντώνης Σιδηρόπουλος
Επίκουρος Καθηγητής

Τίτλος Δ.Ε. Σύστημα Διάδοσης Ειδοποιήσεων
Κωδικός Δ.Ε. 20154
Ονοματεπώνυμο φοιτητή/τών Αντώνης Μπάρμπας
Ονοματεπώνυμο εισηγητή Αντώνης Σιδηρόπουλος
Ημερομηνία ανάληψης Δ.Ε. 08/04/2020
Ημερομηνία περάτωσης Δ.Ε. 14/09/2021

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Αντώνιου Μπάρμπα που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Στην οικογένεια μου»

Πρόλογος

Πρώτα απ' όλα η γλώσσα προγραμματισμού που προτάθηκε και χρησιμοποιήθηκε για την υλοποίηση της, την καθιστά αμέσως μια πολύ ενδιαφέρουσα διπλωματική εργασία. Ένας ακόμη λόγος ήταν το γεγονός ότι η εργασία αυτή προορίζεται για εφαρμογή στην σχολή όπου φοίτησα, αυτό αποτελεί από μόνο του μια δοκιμασία για εμένα, για τον λόγο ότι η δουλειά και το project θα πρέπει να γίνουν σε όσο το δυνατόν υψηλότερο επίπεδο, σε επαγγελματικό επίπεδο, σε επίπεδο εταιρίας και το οποίο δημιουργεί ένα τεράστιο όφελος. Επίσης, κατά την διάρκεια υλοποίησης, η επικοινωνία που 'προγραμματιστικά' είχα με το Facebook, το Twitter, το RSS, το Gmail μέσω των APIs καθώς και η αρέσκεια μου για το back-end, συνέθεσαν ένα πολύ βασικό κίνητρο και μια ευχαρίστηση στην επιλογή της. Τα οφέλη της εργασίας ήταν οι γνώσεις που αποκτήθηκαν στο πώς λειτουργούν τα πράγματα στη μεριά που δε βλέπει ο χρήστης, η επικοινωνία των δεδομένων και των διακομιστών, η εξέλιξη σαν προγραμματιστής, η δομή κώδικα με συνοχή, ευανάγνωστο, ελαστικό και συντηρήσιμο.

Περίληψη

Το θέμα της παρούσας διπλωματικής εργασίας είναι ένα σύστημα διάδοσης ειδοποιήσεων ανάμεσα σε διακομιστές και μέσα κοινωνικής δικτύωσης κυρίως. Ουσιαστικά αποτελεί μια λύση στο πρόβλημα της αυτόματης μετάδοσης δεδομένων μέσω της δημιουργίας ενός μηχανισμού που θα εκτελείται κάθε λίγα λεπτά. Με λίγα λόγια ο μηχανισμός αυτός θα μπορεί μέσω ενός αρχείου το οποίο θα διαμορφώνει ο χρήστης, να διαβάζει τις πηγές και τους προορισμούς όπου θα ανταλλάσσονται τα μηνύματα και οι ειδοποιήσεις καθώς και να αποθηκεύει τα στοιχεία αυτά σε μια βάση δεδομένων ώστε να μην υπάρχουν διπλότυπα και να αποτρέπει την συνεχή μετάδοση αυτών. Πρώτα απ' όλα κάθε πηγή και κάθε προορισμός εκπροσωπεί ένα διακομιστή, ο οποίος χρειάζεται διαφορετική μεταχείριση. Το αρχείο του χρήστη επιπρόσθετα θα ορίζει όλα τα απαραίτητα πεδία για την αυθεντικοποίηση στους διακομιστές που θα χρησιμοποιηθούν. Στη συνέχεια το πρόγραμμα, εγκαθιδρύει όλες τις απαραίτητες διασυνδέσεις όπου θα αλληλεπιδράσει, στη συνέχεια θα ελέγχει μέσω των APIs αν υπάρχει κάποια ενημέρωση ή μια νέα ειδοποίηση που θα πρέπει να μεταδοθεί, αν τα δεδομένα αυτά υπάρχουν στη βάση δεδομένων τότε αναλόγως επιτυγχάνεται μετάδοση ή όχι. Από την άλλη πλευρά και ξεχωριστά από τη λειτουργικότητα, γίνεται παρουσίαση των εργαλείων και των βιβλιοθηκών που συνέβαλλαν στην επικοινωνία με τους διακομιστές όπως του Facebook Graph API για το Facebook, το tweepy για το Twitter, endpoints και GET μέθοδοι για το RSS/Atom.

«Automated Notification System»

«Barmpas Antwnhs»

Abstract

Automation is a broad term that can cover many areas of technology where human input is minimized. There are different business-specific types of automation. This article focuses on the main automation areas of Social Media automation. Social Media automation is the process of optimizing social interactions using automated tools. This can include scheduling social posts ahead of time or republishing articles. Automating social media publication engagement and management reduces the hours spent on maintaining and growing accounts. This article presents an independent Notification system that spreads articles and announcements across different platforms and servers, using all the necessary tools and libraries. The system allows users to share their posts and announcements accross servers by setting up a configuration file and by providing the necessary credentials and access keys-tokens. These posts are also stored by the system for preventing duplicates. The system establishes the connection between APIs, works with API-based tools to communicate with the servers like Facebook Graph API for Facebook, twwepey for Twitter, GET methods and requests for RSS/Atoms. Work also done on how the system is collecting the data, checking if there is a new entry for broadcasting and isolating the primary fields for each post-announcement. The Notification system that is been presenting in this article, not only integrates with APIs but it creates a method for automatically organising social media so that users may more efficiently use social media as part of their bussines operations.

Ευχαριστίες

Ένα μεγάλο ευχαριστώ στην οικογένεια μου, που συνέβαλαν ηθικά και οικονομικά, καθώς και σε όλους τους φίλους μου που μου έμαθαν την αξία του να επιμένω σε κάτι.

Περιεχόμενα

Πρόλογος.....	vii
Περίληψη	viii
Abstract	ix
Ευχαριστίες.....	x
Περιεχόμενα	xi
Κατάλογος Σχημάτων.....	xiv
Συνομογραφίες.....	xv
1. Αυτοματοποιημένη μετάδοση.....	1
1.1.Εισαγωγή.....	1
1.2.Το πρόβλημα	1
1.3.Η ροή της αυτοματοποίησης	2
2. Επισκόπηση τεχνολογιών	5
2.1. HTTP	5
2.1.1. Τεχνικά χαρακτηριστικά HTTP.....	5
2.1.2. Μέθοδοι αίτησης HTTP	5
2.1.3. Κωδικοί αίτησης HTTP.....	7
2.2. HTTPS	7
2.2.1. Σημαντικότητα HTTPS	8
2.2.2. Δημιουργία σύνδεσης	8
2.2.3. Κλείσιμο επικοινωνίας.....	8
2.2.4. Διαφορές HTTP και HTTPS	9
2.3. XML.....	9
2.3.1. Ορολογία XML	9
2.3.2. Χαρακτήρες.....	11
2.3.2.1. Unicode κωδικοποίηση.....	11
2.3.2.2. Χαρακτήρες εξόδου	11
2.3.2.3. Σχόλια	11
2.3.2.4. Παγκόσμια χρήση	11
2.3.3. Ιστορία της XML	12
2.3.3.1. Εκδόσεις	12
2.4. JSON	12
2.4.1. Σύνταξη JSON αρχείου.....	13
2.4.2. Βάσεις δεδομένων σε JSON.....	15

2.4.2.1. Διαχείριση περιεχομένου	15
2.4.2.2. Προσωπικό προφίλ	15
2.4.2.3. Κατάλογοι	15
2.4.2.4. Real-time δεδομένα	15
2.4.3. JSON και XML	15
2.5. RSS	16
2.6. ATOM Feed	17
2.6.1. Διαφορές ATOM και RSS	18
3. Το σύστημα	21
3.1. Εισαγωγή	21
3.2. Αυτοματοποίηση σε Python 3	21
3.3. Αλληλεπίδραση με JSON αρχεία	21
3.4. Σύνδεση APIs με Python 3	22
3.5. Διαχείριση χρόνου στην Python 3	22
3.6. Διαχείριση ημερομηνιών στην Python 3	23
3.7. Διαχείριση παλιών ανακοινώσεων στην Python 3	23
3.8. Παράδειγμα χρήσης του Συστήματος	23
3.9. Επίλογος	28
4. Facebook API	29
4.1. Περιγραφή του Facebook Graph API	29
4.2. Χρήση του Facebook Graph API	29
4.3. Πεδία αρχικής σελίδας της Facebook εφαρμογής	30
4.4. Δημιουργία Facebook εφαρμογής	30
4.5. Επίλογος	31
5. Twitter API	33
5.1. Εισαγωγή	33
5.2. Περιγραφή Twitter API	33
5.3. Tweepy	33
5.4. Δημιουργία Twitter Εφαρμογής	34
5.5. Επίλογος	35
6. GMAIL API	37
6.1. Εισαγωγή	37
6.2. Περιγραφή GMAIL API	37
6.3. Χρήση GMAIL API	37
6.4. API vs SMTP	39
6.5. Επίλογος	40

7. Γενικά APIs.....	41
7.1.Εισαγωγή.....	41
7.2.Aboard.iee.ihu	41
7.2.1.Παράδειγμα χρήσης aboard.....	41
7.3.Προσθήκη APIs.....	42
7.4.SMS API.....	44
8. Συμπέρασμα και Μελλοντικές κατευθύνσεις	45
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	47

Κατάλογος Σχημάτων

Σχήμα 1.3: Activitylifecycle.....	3
-----------------------------------	---

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
API	Application Program Interface
JSON	Javascript Object Notation

1. Αυτοματοποιημένη μετάδοση

1.1. Εισαγωγή

Τα κοινωνικά μέσα δικτύωσης όπως το Facebook, Twitter έχουν καρφίτσωθεί στο πρόγραμμα ενός μεγάλου μέρους του πληθυσμού και η επικοινωνία σήμερα εξαρτάται πολύ έντονα από τις διαδικτυακές τεχνολογίες. Παρόλα αυτά όσοι επικοινωνούν μέσω αυτών των τεχνολογιών μπορεί να μην είναι καν άνθρωποι. Πλήρως αυτόματοι ή ημί-αυτόματοι λογαριασμοί χρηστών οι οποίοι συμμετέχουν όλο και περισσότερο σε διαδικτυακές αλληλεπιδράσεις. Στην συγκεκριμένη εργασία παρουσιάζεται ένα πρόβλημα το οποίο λύνεται μέσω των αυτόματων λογαριασμών, όπου συγκεκριμένα βοηθούν χρήστες να μεταδίδουν εκπαιδευτικού περιεχομένου ανακοινώσεις.

1.2. Το πρόβλημα

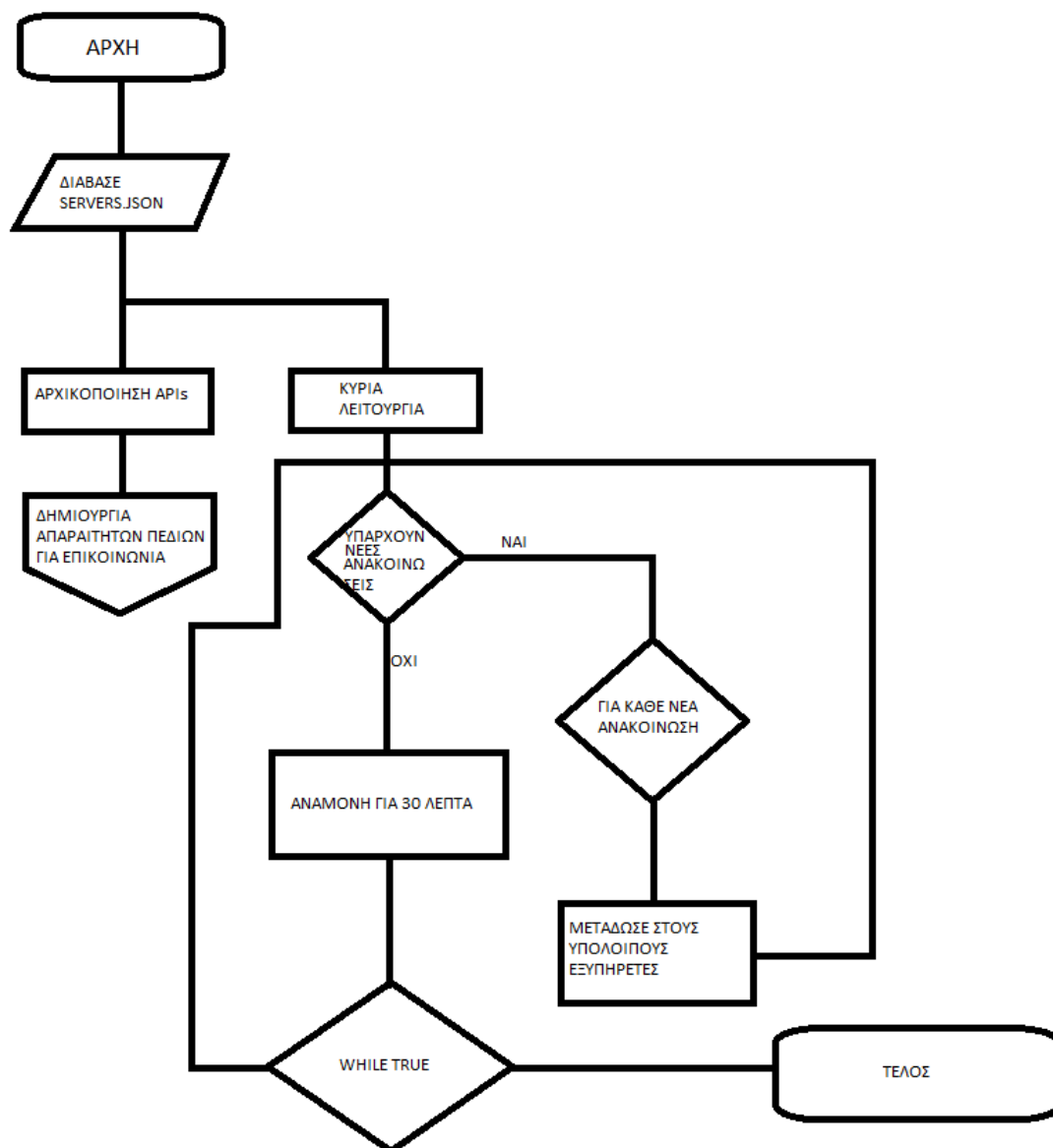
Οι χρήστες ηλεκτρονικών πινάκων ανακοινώσεων, οι χρήστες social media και γενικότερα οι χρήστες που χρησιμοποιούν ηλεκτρονικές πλατφόρμες οι οποίες περιέχουν αναρτήσεις ανακοινώσεων, αποστολές μηνυμάτων και οτι αφορά την ενημέρωση και την επικοινωνία με άλλους ανθρώπους ηλεκτρονικά, συνήθως επιθυμούν να συνδυάζουν τις πλατφόρμες με τις οποίες αλληλεπιδρούν ώστε να μπορούν να διαθέσουν την επικοινωνία οποιασδήποτε μορφής σε ένα μεγαλύτερο μέρος ανθρώπων. Για να γίνει πιο ξεκάθαρο το πρόβλημα, ένας καθηγητής ενός εκπαιδευτικού ιδρύματος θέλει να ενημερώσει τους μαθητές για τους βαθμούς που πήραν στο μάθημά του, αρχικά αναρτά την ανακοίνωση με την βαθμολογία στην κεντρική ιστοσελίδα του τμήματος, έπειτα επειδή θεωρεί ότι δεν θα την λάβουν όλοι οι μαθητές συνδέεται με τον λογαριασμό του στο Facebook και αναρτά την βαθμολογία στην ομάδα του τμήματος, μπορεί μερικοί μαθητές ή συνάδελφοι να χρειαστεί να ενημερωθούν μέσω μόνο GMail γιατί δεν χρησιμοποιούν social media, μπορεί μετά από συνεννόηση με τον καθηγητή κάποιοι μαθητές να μην έχουν σύνδεση στο ίντερνετ και χρειάζεται να ενημερωθούν μέσω sms στο κινητό τηλέφωνο τους. Όλα αυτά δημιουργούν προϋποθέσεις που ο καθηγητής θα πρέπει να πληροί και να μεριμνήσει, όπως να έχει έτοιμο λογαριασμό σε κάθε πλατφόρμα που χρησιμοποιεί και κυρίως να ξέρει με κάποιον τρόπο να την λειτουργεί.

Παρόλα αυτά, το αληθινό πρόβλημα έγκειται στην επαναληπτική διαδικασία την οποία θα χρησιμοποιήσει ο καθηγητής για να ενημερώνει κάθε φορά το σύνολο των ανθρώπων που επιθυμεί, έγκειται στο ότι θα υπάρξουν περισσότερες από μια ανακοινώσεις που θα πρέπει να μεταδοθούν με αποτέλεσμα ο καθηγητής να σπαταλά πολύ χρόνο στη διαδικασία της δημιουργίας των ανακοινώσεων και στην εύρεση των ατόμων που θα τις στείλει. Για τον λόγο αυτό, ο καθένας που χρησιμοποιεί τέτοιου είδους μετάδοση πληροφορίας θα επιθυμούσε να γίνεται αυτόματα, με τις καθορισμένες πλατφόρμες, για καθορισμένο αριθμό παραληπτών και όπως επίσης για καθορισμένο χρονικό διάστημα.

1.3. Η ροή της αυτοματοποίησης

Η λύση του παραπάνω προβλήματος έγκειται στην εύρεση των πλατφορμών τις οποίες θέλει να χρησιμοποιήσει ο καθηγητής και την αποθήκευση τους σε ένα στατικό αρχείο το οποίο θα περιέχει όλες τις απαραίτητες πληροφορίες για την αυθεντικοποίηση και τους λογαριασμούς του. Όπως επίσης και ένα πρόγραμμα το οποίο θα δημιουργεί με τα στοιχεία του στατικού αρχείου την επικοινωνία ανάμεσα στους εξυπηρετητές και θα επιτυγχάνει την ανταλλαγή όλων των επιθυμητών πληροφοριών.

Παρουσιάζοντας γραφικά και εν μέρει προγραμματιστικά το σύστημα μετάδοσης ειδοποιήσεων, κατά την εκτέλεση του, το πρόγραμμα δέχεται σαν ένα και μοναδικό όρισμα το αρχείο που έχει δημιουργηθεί από τον χρήστη με τις απαραίτητες πληροφορίες. Αρχικοποιεί τα πεδία του αρχείου σε μεταβλητές στην αντίστοιχη γλώσσα προγραμματισμού ώστε να μπορέσει να εγκαθιδρύει την επικοινωνία. Στη συνέχεια, μέσα σε έναν ατέρμων βρόχο, ελέγχει εάν υπάρχουν νέες αναρτήσεις μεταξύ των εξυπηρετητών, εάν υπάρχουν τότε σύμφωνα με τους παραλήπτες που έχουν οριστεί μεταδίδει την πληροφορία, αλλιώς εάν δεν υπάρχουν, υπάρχει μια παραμετροποιήσιμη μεταβλητή που ορίζει τον χρόνο που θα περιμένει το σύστημα μέχρι την επόμενη εκτέλεση του, δηλαδή μέχρι να ξανά γίνει ο επόμενος έλεγχος, η αρχική τιμή της μεταβλητής αυτής είναι τα 30 λεπτά.



Σχήμα 1.3: Διάγραμμα ροής

2. Επισκόπηση τεχνολογιών

Στο παρών κεφάλαιο παρουσιάζονται όλες οι τεχνολογίες που χρησιμοποιήθηκαν στο Σύστημα Διάδοσης Ειδοποιήσεων, όπως η αποθήκευση δεδομένων με την τεχνολογία JSON, όλες οι αιτήσεις προς του διακομιστές με HTTP/HTTPS, η μετάδοση ηλεκτρονικών μηνυμάτων με το πρωτόκολλο SMTP και η λήψη των ανακοινώσεων μέσω RSS Feeds με την μορφή XML/ATOM.

2.1. HTTP

Το Hypertext Transfer Protocol (HTTP) είναι ένα πρωτόκολλο επιπέδου εφαρμογής για καταναμημένα συστήματα πληροφοριών. Το HTTP είναι σε χρήση στο διαδίκτυο απο το 1998. Η πρώτη έκδοση του HTTP αναφερόταν σαν HTTP/0.9, ήταν ένα απλό πρωτόκολλο για την αποστολή μη επεξεργασμένων δεδομένων στο διαδίκτυο. Αρχικά το πρωτόκολλο δεν μετέφερε καμία πληροφορία σχετικά με το πρόγραμμα-πελάτης και η μόνη επιλογή που υπήρχε ήταν η ζήτηση από τον εξυπηρετητή μίας σελίδας κειμένου το οποίο περιείχε μόνο χαρακτήρες ASCII και πιθανόν χαρακτήρες τερματισμού γραμμής. Το HTTP/1.0 όπως ονομάστηκε η επόμενη έκδοση βελτίωσε το πρωτόκολλο, με το να δέχεται μηνύματα σε ευανάγνωστη μορφή που περιέχουν πληροφορίες σχετικές με τα δεδομένα που μεταφέρονται, όπως και την λειτουργία τροποποίησης των response/request μεθόδων.

Σήμερα το πρωτόκολλο αυτό είναι πλέον καθιερωμένο και διαδεδομένο σε σημείο που σχεδόν όλοι οι φυλλομετρητές να το θεωρούν δεδομένο και να το χρησιμοποιούν σε περίπτωση που ο χρήστης δεν καθορίσει ποιο πρωτόκολλο θέλει να χρησιμοποιήσει. Αν δηλαδή ο χρήστης δεν γράψει:

```
http://my.url
```

αλλά γράφει σκέτο το:

```
my.url
```

σχεδόν όλοι οι φυλλομετρητές θεωρούν σαν δεδομένο το πρωτόκολλο http και όχι κάποιο άλλο.

2.1.1. Τεχνικά χαρακτηριστικά HTTP

Η διαδικασία που ακολουθούσε το αρχικό πρωτόκολλο ήταν η εξής:

Σύνδεση στον εξυπηρετητή

Ερώτηση προς τον εξυπηρετητή

Απάντηση από τον εξυπηρετητή

Σήμερα χρησιμοποιεί πολύ περισσότερα χαρακτηριστικά τα οποία παρέχουν ακόμα και τη δυνατότητα στο πρόγραμμα-πελάτη να στέλνει δεδομένα στον εξυπηρετητή.

2.1.2. Μέθοδοι αίτησης HTTP

Αν και το HTTP πρωτόκολλο σχεδιάστηκε για χρήση στον Ιστό, υποστηρίζει λειτουργίες που είναι πιο γενικές απ' ότι απαιτείται. Οι λειτουργίες αυτές ονομάζονται μέθοδοι. Κάθε αίτηση αποτελείται από μία ή περισσότερες γραμμές κειμένου ASCII. Η πρώτη λέξη της πρώτης γραμμής της αίτησης είναι το όνομα της ζητούμενης μεθόδου.

Παρακάτω παρουσιάζονται συνοπτικά οι ενσωματωμένες μέθοδοι αίτησης του πρωτοκόλλου HTTP:

GET

Η μέθοδος GET ζητά από το διακομιστή να στείλει τη σελίδα. Η σελίδα κωδικοποιείται κατάλληλα σε μορφή MIME. Η πιο συνήθης μορφή της μεθόδου GET είναι η εξής:

```
GET όνομα_αρχείου HTTP/1.1
```

όπου το όνομα_αρχείου προσδιορίζει το όνομα του πόρου που πρέπει να προσκομιστεί και το 1.1 είναι η έκδοση του πρωτοκόλλου που χρησιμοποιείται.

HEAD

Η μέθοδος HEAD ζητά μόνο την κεφαλίδα του μηνύματος, χωρίς τη πραγματική σελίδα. Η μέθοδος αυτή μπορεί να χρησιμοποιηθεί για τη συλλογή πληροφοριών για λόγους δεικτοδότησης ή απλώς και μόνο για τον έλεγχο εγκυρότητας μιας διεύθυνσης URL.

POST

Η μέθοδος POST χρησιμοποιείται κατά την υποβολή φορμών. Όπως και η μέθοδος GET, η POST περιέχει μια διεύθυνση URL αλλά αντί να ανακτά απλώς τη σελίδα μεταφέρει δεδομένα στον διακομιστή όπως για παράδειγμα τα περιεχόμενα της φόρμας. Έπειτα ο διακομιστής κάνει κάτι με αυτά τα δεδομένα ανάλογα με το URL. Τέλος, η μέθοδος επιστρέφει μια σελίδα που δείχνει το αποτέλεσμα.

PUT

Η μέθοδος PUT είναι η αντίστροφη της GET, δηλαδή αντί να διαβάζει τη σελίδα, γράφει τη σελίδα. Η μέθοδος αυτή κάνει εφικτή την κατασκευή μιας συλλογής ιστοσελίδων σε έναν απομακρυσμένο διακομιστή. Το σώμα της αίτησης περιέχει τη σελίδα. Μπορεί να κωδικοποιείται μέσω του MIME, οπότε οι γραμμές που ακολουθούν την PUT μπορεί να περιέχουν κεφαλίδες Content-Type και πιστοποίησης ταυτότητας, ώστε να αποδείξουν ότι ο αιτών έχει πραγματικά την άδεια να εκτελέσει τη ζητούμενη λειτουργία.

DELETE

Η μέθοδος DELETE καταργεί τη σελίδα ή τουλάχιστον δηλώνει ότι ο διακομιστής Ιστού έχει συμφωνήσει να καταργήσει τη σελίδα. Όπως και με την PUT και σε αυτή τη μέθοδο παίζουν μεγάλο ρόλο η πιστοποίηση της ταυτότητας και της άδειας εκτέλεσης της λειτουργίας.

TRACE

Η μέθοδος TRACE χρησιμοποιείται για αποσφαλμάτωση. Ζητά από τον διακομιστή να επιστρέψει την αίτηση. Η μέθοδος αυτή είναι χρήσιμη όταν η επεξεργασία των αιτήσεων δεν γίνεται σωστά και ο πελάτης θέλει να δει ποια αίτηση έλαβε πραγματικά ο διακομιστής.

CONNECT

Η μέθοδος CONNECT επιτρέπει στον χρήστη να πραγματοποιήσει σύνδεση με έναν διακομιστή Ιστού μέσω μιας ενδιάμεσης συσκευής, για παράδειγμα μέσω μια κρυφής μνήμης Ιστού.

OPTIONS

Η μέθοδος OPTIONS παρέχει έναν τρόπο ώστε ο πελάτης να στέλνει ερωτήματα στον διακομιστή σχετικά με μια σελίδα και να λαμβάνει τις μεθόδους και τις κεφαλίδες που μπορούν να χρησιμοποιηθούν με αυτή τη σελίδα.

2.1.3. Κωδικοί αίτησης HTTP

Κάθε αίτηση λαμβάνει μια απάντηση η οποία αποτελείται από μια γραμμή κατάστασης και πιθανόν πρόσθετες πληροφορίες. Η γραμμή κατάστασης περιέχει ένα τριψήφιο κωδικό κατάστασης, ο οποίος δηλώνει κατά πόσον εξυπηρετήθηκε η αίτηση και αν δεν εξυπηρετήθηκε, γιατί συνέβη αυτό. Το πρώτο ψηφίο χρησιμοποιείται για την υποδιαίρεση των αιτήσεων σε 5 κατηγορίες.

Οι κωδικοί 1xx χρησιμοποιούνται σπανίως στη πράξη. Παράδειγμα: 100= ο διακομιστής συμφωνεί να χειριστεί την αίτηση του πελάτη.

Οι κωδικοί 2xx σημαίνουν ότι ο χειρισμός της αίτησης έγινε με επιτυχία και ότι επιστρέφεται το περιεχόμενο αν υπήρχε. Παράδειγμα: 200 = η αίτηση πέτυχε, 204 = δεν υπάρχει περιεχόμενο.

Οι κωδικοί 3xx λένε στον πελάτη να ψάξει αλλού, είτε χρησιμοποιώντας μια διαφορετική διεύθυνση URL, είτε εξετάζοντας την κρυφή μνήμη. Παράδειγμα: 301 = η σελίδα μετακινήθηκε.

Οι κωδικοί 4xx σημαίνουν ότι η αίτηση απέτυχε λόγω κάποιου σφάλματος του πελάτη, όπως άκυρη αίτηση ή ανύπαρκτη σελίδα. Παράδειγμα: 403 = απαγορευμένη σελίδα, 404 = η σελίδα δεν βρέθηκε.

2.2. HTTPS

Το Hypertext transfer protocol secure (HTTPS) είναι η μια έκδοση του HTTP που έχει επενδύσει στην ασφάλεια και αποτελεί το κύριο πρωτόκολλο αποστολής δεδομένων. Σχεδόν όλες οι "ασφαλείς" επικοινωνίες που συμβαίνουν στο διαδίκτυο λαμβάνουν χώρα πάνω στο HTTPS, συμπεριλαμβανομένων και συναλλαγών τραπεζών, ηλεκτρονικού ταχυδρομείου και ηλεκτρονικής αγοράς. Το HTTPS βασίζεται στο TLS πρωτόκολλο επιπέδου μεταφοράς και κρυπτογράφησης και σε μία υποδομή δημοσίων κλειδιών για αποκρυπτογράφηση που αποτελείται χιλιάδες πιστοποιητικά ασφαλείας, ώστε όλοι οι browsers να γίνονται εμπιστευτοί από τους χρήστες τους.

Για να χρησιμοποιηθεί το HTTPS σε έναν server, θα πρέπει ο διαχειριστής του να εκδώσει ένα πιστοποιητικό δημοσίου κλειδιού. Σε servers που χρησιμοποιούν το λειτουργικό σύστημα UNIX αυτό μπορεί να γίνει μέσω του προγράμματος OpenSSL. Στην συνέχεια το πιστοποιητικό αυτό θα πρέπει να υπογραφεί από μία αρχή πιστοποίησης (certificate authority), η οποία πιστοποιεί ότι ο εκδότης του πιστοποιητικού είναι νομότυπος και ότι το πιστοποιητικό είναι έγκυρο. Με τον τρόπο αυτό οι χρήστες μπορούν να δουν την υπογραφή της αρχής πιστοποίησης και να βεβαιωθούν ότι το πιστοποιητικό είναι έγκυρο και ότι κανένας κακόβουλος χρήστης δεν το έχει πλαστογραφήσει.

2.2.1. Σημαντικότητα HTTPS

Το HTTPS εμποδίζει τις ιστοσελίδες από την μετάδοση των πληροφοριών με τρόπο που καθιστά εύκολη την παρακολούθηση από τρίτους στο δίκτυο. Όταν οι πληροφορίες αποστέλλονται με την κανονική έκδοση του HTTP, είναι καταμεμημένες σε πακέτα δεδομένων που μπορούν εύκολα να κλαπούν. Αυτό καθιστά την επικοινωνία μέσω ενός μη ασφαλισμένου μέσου πχ. δημόσιο Wi-Fi πολύ ευάλωτη. Στην πραγματικότητα κάθε επικοινωνία που γίνεται μέσω HTTP, όλα τα δεδομένα μεταδίδονται σε απλό κείμενο, με αποτέλεσμα να γίνονται άμεσα ευάλωτες οποιοσδήποτε πληροφορίες υπάρχουν σε διάφορους τρόπους επιθέσεων.

Με το HTTPS, όλη η κυκλοφορία των δεδομένων είναι κρυπτογραφημένη, οπότε ακόμη και αν μερικά πακέτα εισπραχθούν ή παρεμποδιστούν από τρίτους χρήστες θα εμφανίζονται σαν συμβολοσειρές τυχαίων χαρακτήρων. Για παράδειγμα:

Πριν την κρυπτογράφηση:

This is a string of text that is completely readable

Μετά την κρυπτογράφηση:

ITM0IRYiEhVpa6VnKyExMiEgNveroYWBPlgGyfkflYjDaaFf/
Kn3bo3OfghBPDWo6AfSHlNtL8N7ITEwIXc1gU5X73xMsJormzzXlwOyrCs+9XCPk63Y+z0=

Στις ιστοσελίδες χωρίς HTTPS, είναι δυνατόν οι πάροχοι των υπηρεσιών διαδικτύου ή άλλοι μεσάζοντες να εισάγουν περιεχόμενο σε ιστοσελίδες χωρίς την έγκριση του ιδιοκτήτη της ιστοσελίδας. Αυτό το γεγονός συνήθως αναφέρεται σε τρόπους διαφήμισης, όπου διάφοροι πάροχοι θέλουν να αυξήσουν τα έσοδα και την προώθηση προϊόντων. Όπως είναι λογικό με τον τρόπο αυτό, τα κέρδη των διαφημίσεων και η ποιότητα αυτών των διαφημίσεων δεν περιλαμβάνουν τον ιδιοκτήτη της ιστοσελίδας. Το HTTPS εξαλείφει την ικανότητα σε τρίτους χρήστες να μπορούν να εισάγουν περιεχόμενα σε ιστοσελίδες χωρίς την έγκριση των ιδιοκτητών.

2.2.2. Δημιουργία σύνδεσης

Ο χρήστης ο οποίος ενεργεί και σαν HTTP client θα πρέπει να ενεργεί και σαν TLS client. Δηλαδή θα πρέπει να ξεκινήσει μια σύνδεση με τον διακομιστή στην κατάλληλη θύρα και στη συνέχεια να στείλει ένα συγκεκριμένο TLS CLIENT/HELLO μήνυμα ώστε να ξεκινήσει η σύνδεση. Ο χρήστης στη συνέχεια μπορεί να ξεκινήσει το πρώτο HTTP αίτημα. Όλα τα δεδομένα HTTP πρέπει να αποστέλλονται ως "δεδομένα επιπέδου εφαρμογής" και να ακολουθούνται συγκεκριμένα βήματα για την λειτουργία του HTTP πρωτοκόλλου.

2.2.3. Κλείσιμο επικοινωνίας

Το TLS πρωτόκολλο προσφέρει μια λειτουργία επιπλέον όσον αφορά την ασφάλεια στο κλείσιμο της σύνδεσης. Όταν ληφθεί μια έγκυρη ειδοποίηση κλεισίματος, ένα εργαλείο του πρωτοκόλλου εξασφαλίζει ότι δεν θα μεταδοθούν άλλα δεδομένα σε αυτή τη σύνδεση. Οι λειτουργίες του TLS, αφού στείλουν μια ειδοποίηση κλεισίματος, μπορούν να κλείσουν την σύνδεση χωρίς να περιμένουν τον δέκτη να στείλει την δικιά του ειδοποίηση κλεισίματος, δημιουργώντας έτσι ένα "μη-ολοκληρωμένο" κλείσιμο σύνδεσης.

Παρόλα αυτά μια λειτουργία που συμπεριφέρεται έτσι, υπάρχει πιθανότητα να χρησιμοποιήσει ξανά την ίδια σύνδεση. Αυτό μπορεί να γίνει μόνο όταν η εφαρμογή έχει στείλει και έχει λάβει όλα τα δεδομένα που την ενδιαφέρουν. Επειδή το TLS πρωτόκολλο δεν εξαρτάται από τις μεθόδους του

HTTP, είναι απαραίτητο όλα τα δεδομένα να εξετάζονται, συγκεκριμένες κεφαλίδες δηλαδή, για να προσδιοριστεί πότε εμφανίστηκε η "απότομη" διακοπή της σύνδεσης.

2.2.4. Διαφορές HTTP και HTTPS

Από τεχνική άποψη, το HTTPS δεν αποτελεί ένα ξεχωριστό πρωτόκολλο από το HTTP. Απλώς χρησιμοποιεί κρυπτογράφηση επάνω στις λειτουργίες του HTTP μέσω του TLS πρωτοκόλλου. Το HTTP δομείται με βάση την μετάδοση πιστοποιητικών TLS, τα οποία επαληθεύουν την ταυτότητα των παρόχων.

Επίσης όταν ένας χρήστης συνδέεται σε μια ιστοσελίδα, η ιστοσελίδα θα στείλει το TLS πιστοποιητικό που περιέχει το δημόσιο κλειδί που απαιτείται για να ξεκινήσει ένα ασφαλές session. Οι δυο υπολογιστές, ο πελάτης και ο πάροχος, περνούν από μια διαδικασία στέλνοντας πληροφορίες που θα εγκαθιδρύσουν μια ασφαλής επικοινωνία, ώστε να σταλθεί οποιοδήποτε περιεχόμενο.

2.3. XML

Η Extensible Markup Language (XML), παρουσιάζει μια κλάση απο αντικείμενα δεδομένων τα οποία ονομάζονται XML documents και περιγράφουν την συμπεριφορά των προγραμμάτων που τα χρησιμοποιούν. XML είναι μια μορφή εφαρμογής της SGML (Standard Generalized Markup Language). Όσον αφορά την κατασκευή, τα έγγραφα XML συντάσσονται από έγγραφα της SGML.

Τα XML έγγραφα απαρτίζονται απο τμήματα μνήμης που ονομάζονται entities, τα οποία περιέχουν επεξεργασμένα ή μη δεδομένα. Τα επεξεργασμένα δεδομένα είναι φτιαγμένα από χαρακτήρες, τα οποία δημιουργούν είτε δεδομένα συμβολοσειρών είτε Markup δεδομένα. Η XML περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση των εγγράφων. Ορίζεται, κυρίως στην προδιαγραφή XML 1.0, που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium) αλλά και σε διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων. Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μια μορφοποίηση δεδομένων κειμένου, με ισχυρή κωδικοποίηση Unicode για όλες τις γλώσσες του κόσμου.

Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στο διαδίκτυο. Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελαίνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML.

2.3.1. Ορολογία XML

Χαρακτήρας Unicode

Εξ ορισμού, ένα κείμενο XML είναι μία ακολουθία χαρακτήρων. Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανίζεται σε ένα κείμενο XML.

Επεξεργαστής και Εφαρμογή

Είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML. Είναι αναμενόμενο, ότι ένας επεξεργαστής δουλεύει για μία εφαρμογή. Υπάρχουν μερικές πολύ συγκεκριμένες απαιτήσεις, σχετικά με το τι μπορεί και τι δεν μπορεί να κάνει ένας επεξεργαστής XML, αλλά καμία, όσον αφορά στη συμπεριφορά της εφαρμογής. Ο επεξεργαστής (όπως ονοματίζεται από την προδιαγραφή), αναφέρεται συχνά, με τον αγγλικό όρο XML parser.

Σήμανση και Περιεχόμενο

Οι χαρακτήρες που απαρτίζουν ένα κείμενο XML, αποτελούν είτε τη σήμανση είτε το περιεχόμενό του. Η σήμανση και το περιεχόμενο, μπορούν να επισημανθούν και να διακριθούν, ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων. Όλα τα αλφαριθμητικά που συνιστούν τη σήμανση, είτε ξεκινούν με το χαρακτήρα "<" και καταλήγουν στο χαρακτήρα ">", είτε ξεκινούν με το χαρακτήρα "&" και καταλήγουν στο χαρακτήρα ";" . Ακολουθίες χαρακτήρων που δε συνιστούν τη σήμανση, αποτελούν το περιεχόμενο ενός κειμένου XML.

Ετικέτα

Ένα στοιχείο σήμανσης που ξεκινά με το χαρακτήρα "<" και καταλήγει στο χαρακτήρα ">". Υπάρχουν τρία είδη ετικέτας: ετικέτες-αρχής, για παράδειγμα <section>, ετικέτες-τέλους, για παράδειγμα </section>, και ετικέτες-χωρίς-περιεχόμενο, για παράδειγμα <line-break/>.

Στοιχείο

Ένα λογικό απόσπασμα ενός κειμένου, που είτε ξεκινά με μία ετικέτα-αρχής και καταλήγει σε μία ετικέτα-τέλους, είτε αποτελείται μόνο από μία ετικέτα-χωρίς-περιεχόμενο. Οι χαρακτήρες που υπάρχουν, αν υπάρχουν, μεταξύ μιας ετικέτας-αρχής και μιας ετικέτας-τέλους, συνιστούν το περιεχόμενο του στοιχείου, το οποίο μπορεί να περιέχει σήμανση, συμπεριλαμβανομένων και άλλων στοιχείων, που ονομάζονται στοιχεία-παιδιά. Ένα παράδειγμα ενός στοιχείου είναι το <Greeting>Hello, world.</Greeting>. Ένα άλλο είναι το <line-break/>.

Χαρακτηριστικό

Ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι όνομα/τιμή, το οποίο υπάρχει μέσα σε μία ετικέτα-αρχής ή σε μία ετικέτα-χωρίς-περιεχόμενο. Στο παράδειγμα παρακάτω, το στοιχείο img έχει δύο χαρακτηριστικά, τα src και alt: . Ένα άλλο παράδειγμα θα ήταν το <step number="3">Connect A to B.</step>, όπου το όνομα του χαρακτηριστικού είναι "number" και η τιμή του είναι "3".

Δήλωση XML

Τα κείμενα XML μπορούν να αρχίζουν, με τη δήλωση κάποιων πληροφοριών σχετικών με αυτά, όπως στο ακόλουθο παράδειγμα:

```
<?xml version="1.0" encoding="UTF-8"?>
  Παράδειγμα: <?xml version="1.0" encoding='UTF-8'?>
  <painting>
    
    <caption>This is Raphael's "Foligno" Madonna, painted in
    <date>1511</date>--<date>1512</date>.</caption>
  </painting>
```

Κώδικας 1: Δήλωση κειμένου XML

2.3.2. Χαρακτήρες

2.3.2.1. Unicode κωδικοποίηση

Το σύνολο Unicode χαρακτήρων μπορεί να κωδικοποιηθεί σε bytes για αποθήκευση ή μετάδοση με διάφορους τρόπους ή διάφορα "encodings". Η ίδια Unicode κωδικοποίηση ορίζει τα "encodings" αυτά που καλύπτουν κάθε πιθανή χρήση, τα πιο γνωστά "encodings" είναι το UTF-8 και το UTF-16. Υπάρχουν πολλές άλλες κωδικοποιήσεις κειμένου που προέρχονται απ τη Unicode, όπως η ASCII και η ISO/IEC, όπου το ρεπερτόριο των χαρακτήρων τους είναι συνήθως υποσύνολο της Unicode. Η XML επιτρέπει την χρήση οποιασδήποτε κωδικοποίησης που βασίζεται στη Unicode ή που οι χαρακτήρες της εμφανίζονται στη Unicode. Επίσης, η XML προσφέρει ένα μηχανισμό στον οποίο υπάρχει ένας επεξεργαστής που μπορεί αξιόπιστα να αναγνωρίζει την κωδικοποίηση που υπάρχει.

2.3.2.2. Χαρακτήρες εξόδου

Η XML προσφέρει χαρακτήρες εξόδου ώστε να αντιμετωπίζει "προβληματικά" κείμενα που τυχαίνει να υπάρχουν στα έγγραφα. Για παράδειγμα:

Οι Χαρακτήρες "<" και "&" αποτελούν κλειδιά στην σύνταξη των entities, οπότε δεν προτείνεται να χρησιμοποιούνται έξω από αυτά. Μερικές κωδικοποιήσεις χαρακτήρων λόγω ότι αποτελούν υποσύνολο της Unicode, υστερούν στον συγχρονισμό και στην μετάφραση, όπως η ASCII. Επίσης μερικοί χαρακτήρες δεν μπορούν να διαχωριστούν εικονικά απο άλλους χαρακτήρες, όπως ο χαρακτήρας space " " ή το λατινικό κεφαλαίο A με το κυριλικό κεφαλαίο A.

Όλοι οι επιτρεπόμενοι χαρακτήρες μπορούν να εκφραστούν και με αριθμούς. Για παράδειγμα, ο Κινέζικος χαρακτήρας "中", του οποίου ο αριθμητικός κωδικός είναι ο δεκαεξαδικός αριθμός 4E2D ή ο δεκαδικός 20,013. Οι χρήστες οι οποίοι δεν έχουν όλες τις γλώσσες στο πληκτρολόγιό τους, μπορούν να προσθέσουν χαρακτήρες σαν και αυτούς σε ένα XML έγγραφο γράφοντας των κωδικό του χαρακτήρα πχ. 中 ή 中.

2.3.2.3. Σχόλια

Τα σχόλια μπορούν να εμφανιστούν οπουδήποτε σε ένα XML έγγραφο εκτός απο το εσωτερικό των αντικειμένων και πριν από την αρχικοποίηση του εγγράφου. Τα σχόλια ξεκινάν με τη συμβολοσειρά '<!--' και τελειώνουν με την '-->'. Όσον αφορά τη συμβατότητα με τη SGML, η συμβολοσειρά '--' δεν μπορεί να χρησιμοποιηθεί μέσα στα σχόλια και δεν πρόκειται να εμφανιστεί. Το '&' δεν έχει ιδιαίτερη σπουδαιότητα μέσα στα σχόλια, έτσι ώστε να εμφανίζονται κανονικά και όχι σαν XML κείμενο οι χαρακτήρες και οι οντότητες, και επίσης δεν υπάρχει τρόπος ούτε εντός σχόλιων να τυπωθεί χαρακτήρας που δεν υποστηρίζεται από την προεπιλεγμένη κωδικοποίηση.

2.3.2.4. Παγκόσμια χρήση

Η XML 1.0 και η XML 1.1 υποστηρίζουν απευθείας χρήση σε οποιαδήποτε γλώσσα, σχεδόν όλης της Unicode κωδικοποίησης, όπως όνομα, ιδιότητες μεταβλητές, χαρακτήρες, μεθόδους επεξεργασίας δεδομένων. Για παράδειγμα το παρακάτω έγγραφο XML περιέχει Κινέζικου, Αρμένικους και Κυριλικούς χαρακτήρες:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<俄语 语言="руси́йский">данные</俄语>
```

2.3.3. Ιστορία της XML

Η XML αποτελεί μια μορφή της SGML εφαρμογής. Η ευελιξία της SGML για την απεικόνιση δυναμικά πληροφοριών κατανοήθηκε από τους πρώτους ψηφιακούς εκδότες στα τέλη της δεκαετίας του 1980 πριν από την άνοδο του διαδικτύου. Στα μέσα της δεκαετίας του 90 ορισμένοι επαγγελματίες που είχαν αποκτήσει εμπειρία στην SGML, πίστευαν ότι η γλώσσα αυτή θα πρόσφερε σημαντικές λύσεις σε προβλήματα που ίσως αντιμετώπιζε το νέο διαδίκτυο εκείνη την εποχή. Ο Dan Connolly όντας μέλος της W3C, πρόσθεσε την SGML στον κατάλογο των δραστηριοτήτων της εταιρίας. Οι εργασίες της XML ξεκίνησαν το 1995 απ τον μηχανικό της Sun Microsystems Jon Bosak, ο οποίος δημιούργησε μια μικρή ομάδα τεχνικών με εμπειρία τόσο στην SGML όσο και γενικά στο διαδίκτυο.

Η ομάδα τεχνικών της XML δεν συναντήθηκε ποτέ, ο σχεδιασμός ολοκληρώθηκε χρησιμοποιώντας ένα συνδυασμό ηλεκτρονικών μηνυμάτων και εβδομαδιαίων τηλεδιασκέψεων. Οι κυριότερες αποφάσεις και ο μεγαλύτερος σχεδιασμός, επιτεύχθηκε σε πολύ μικρό χρονικό διάστημα από τον Αύγουστο μέχρι τον Νοέμβριο του 1996 όταν δημοσιεύτηκαν και οι πρώτες προδιαγραφές. Στη συνέχεια οι εργασίες σχεδιασμού συνεχίστηκαν και το 1997, έως ότου τον Φεβρουάριο του 1998 να προταθεί η πρώτη έκδοση XML 1.0 από την W3C.

2.3.3.1. Εκδόσεις

XML 1.0

Η πρώτη έκδοση της XML, δημιουργήθηκε το 1998. Από τότε έχει υποστεί ελάχιστες και πολύ μικρές αλλαγές, ώστε να της χαρίσουν μια νέα έκδοση αλλά τυπικά βρίσκεται στην πέμπτη, όπως δημοσιεύθηκε στις 26 Νοεμβρίου του 2008. Παρόλα αυτά ακόμα χρησιμοποιείται ευρέως και συνιστάται για χρήση.

XML 1.1

Η δεύτερη XML 1.1 δημοσιεύθηκε στις 4 Φεβρουαρίου 2004, την ίδια μέρα με την τρίτη έκδοση της XML 1.0, και σήμερα βρίσκεται στην δεύτερη έκδοση της. Περιέχει χαρακτηριστικά που σκοπό έχουν να κάνουν την XML λίγο πιο εύκολη στην χρήση της σε συγκεκριμένες περιπτώσεις.

Οι κύριες αλλαγές και διαφορές της XML είναι η ενεργοποίηση των χαρακτήρων που περιγράφουν το τέλος γραμμής, την χρήση scripts και τον απεικονισμό χαρακτήρων που απουσιάζουν απ τη Unicode κωδικοποίηση. Η XML 1.1 δεν είναι τόσο διαδεδομένη και η χρήση της συνιστάται κυρίως σε συγκεκριμένες περιπτώσεις.

2.4. JSON

Το JSON (JavaScript Object Notation) αποτελεί ένα μορφότυπο δεδομένων που βασίζεται στο τύπος δεδομένων της JavaScript γλώσσας προγραμματισμού. Τα τελευταία χρόνια η JSON μορφή έχει αποκτήσει μεγάλη δημοτικότητα στο διαδίκτυο και αποτελεί την κύρια μορφή αναπαράστασης των δεδομένων που ανταλλάσσονται. Σήμερα παίζει πολύ μεγάλο ρόλο στην ανάπτυξη διαδικτυακών εφαρμογών. Κάθε λογισμικό που εκτελεί μεθόδους απομακρυσμένα απο κάποιον διακομιστή πρέπει να δημιουργήσει ένα πρωτόκολλο ώστε να μπορεί να στέλνει αιτήσεις για δεδομένα στον διακομιστή αυτό, το οποίο ονομάζεται API (Application Program Interface). Δεδομένου ότι τα δεδομένα σε μορφή JSON μπορούν να διαβαστούν και απ τους ανθρώπους και απ τους υπολογιστές, την κατατάσσει στην πιο δημοφιλή μορφή για την αποστολή αιτημάτων HTTP.

Για παράδειγμα, μια εφαρμογή που προβάλλει και περιέχει πληροφορίες σχετικά με τον καιρό, παρέχει ένα API ώστε να επιτρέψει σε άλλα λογισμικά να έχουν πρόσβαση στις πληροφορίες αυτές. Μια

Κεφάλαιο 2

υποθετική κλήση και αίτηση στο API θα μπορούσε να περιέχει αυτά τα ζευγάρια τιμών και δεδομένων:

```
{"Country": "Chile", "City": "Santiago"}
```

όπου ο πελάτης ζητά τον καιρό στην πόλη Santiago στη Χιλή. Η HTTP απάντηση του API θα έδινε τις απαραίτητες πληροφορίες επίσης σε JSON μορφή:

```
{"timestamp": "14/10/2015 11:59:07",  
  "temperature": 25, "Country": "Chile",  
  "City": "Santiago", "description": "Sunny"}
```

δηλώνοντας ότι η θερμοκρασία είναι στους 25 βαθμούς και η μέρα ηλιόλουστη.

Το παράδειγμα αυτό αντικατοπτρίζει την απλότητα και εύκολη αναγνωσιμότητα της μορφής αυτής, η οποία εξηγεί εν μέρει την γρήγορα υιοθεσία της.

2.4.1. Σύνταξη JSON αρχείου

Όπως περιγράφεται παραπάνω, η μορφή JSON είναι μια συμβολοσειρά που μοιάζει με αντικείμενα και συμβολοσειρές τις γλώσσας JavaScript. Ίδιοι τύποι δεδομένων μπορούν να συμπεριληφθούν σε ένα JSON αρχείο όπως και σε ένα απλό πρόγραμμα σε JavaScript, συμβολοσειρές, αριθμοί, πίνακες, booleans και άλλα αντικείμενα. Αυτό επιτρέπει την κατασκευή μιας ιεραρχίας δεδομένων:

```
{  
  "squadName": "Super hero squad",  
  "homeTown": "Metro City",  
  "formed": 2016,  
  "secretBase": "Super tower",  
  "active": true,  
  "members": [  
    {  
      "name": "Molecule Man",  
      "age": 29,  
      "secretIdentity": "Dan Jukes",  
      "powers": [  
        "Radiation resistance",  
        "Turning tiny",  
        "Radiation blast"  
      ]  
    },  
    {  
      "name": "Madame Uppercut",  
      "age": 39,  
      "secretIdentity": "Jane Wilson",  
      "powers": [  
        "Million tonne punch",  
        "Damage resistance",  
        "Superhuman reflexes"  
      ]  
    },  
    {  
      "name": "Eternal Flame",  
      "age": 1000000,  
      "secretIdentity": "Unknown",
```

Κώδικας 2: Σύνταξη JSON αρχείου

Η ιεραρχία αυτή μπορεί να έχει 2 δομές:

- Μια συλλογή ζεύγων όνομα-τιμή ("name": "Eternal Flame"). Σε διάφορες γλώσσες αυτό ορίζεται σαν αντικείμενο, δομή, dictionary ή και hash πίνακας.
- Μια διατεταγμένη λίστα τιμών. Σε διάφορες γλώσσες αυτό ορίζεται ως πίνακας, διάνυσμα, λίστα ή ακολουθία.

2.4.2. Βάσεις δεδομένων σε JSON

2.4.2.1. Διαχείριση περιεχομένου

Ένα έγγραφο σε μορφή JSON είναι μια εξαιρετική επιλογή για εφαρμογές διαχείρισης περιεχομένου, όπως blogs, πλατφόρμες για βίντεο, διότι κάθε μια οντότητα μπορεί να αποθηκευτεί σε ένα ξεχωριστό αρχείο. Επίσης όταν χρειαστεί κάποια αλλαγή το μοντέλου σχεδιασμού της βάσης, τότε αλλάζουν μόνο τα αρχεία τα οποία επηρεάζουν τις οντότητες αυτές.

2.4.2.2. Προσωπικό προφίλ

Οι βάσεις δεδομένων σε μορφή JSON αποτελούν μια καλή λύση για προσωπικά προφίλ χρηστών στο διαδίκτυο, στα οποία ο καθένας παρέχει διαφορετικές πληροφορίες. Χρησιμοποιώντας μια τέτοια βάση, επιτυγχάνεται η αποθήκευση εγγράφων που το καθένα περιέχει τα χαρακτηριστικά που αφορούν ένα συγκεκριμένο και μόνο χρήστη. Οι βάσεις αυτές με αυτόν τον τρόπο αποκτούν αρκετά απλά την ιδιότητα της ατομικότητας.

2.4.2.3. Κατάλογοι

Τα έγγραφα σε μορφή JSON είναι εξίσου αποτελεσματικά όσον αφορά την αποθήκευση πληροφοριών καταλόγου. Συνήθως πληροφορίες καταλόγου υπάρχουν σε εφαρμογές ηλεκτρονικού εμπορίου, όπου κάθε προϊόν έχει διαφορετικό αριθμό χαρακτηριστικών. Η περιγραφή αυτών των χαρακτηριστικών μπορεί να ενσωματωθεί σε ένα JSON αρχείο για ευκολότερη διαχείριση και γρηγορότερη ανάκτηση απ' ό,τι σε μια σχεσιακή βάση δεδομένων.

2.4.2.4. Real-time δεδομένα

Επίσης, το να μπορεί κάποιος να συλλέγει πληροφορίες και δεδομένα πραγματικού χρόνου είναι μια πολύ σημαντική και κρίσιμη ενέργεια σε ένα πολύ ανταγωνιστικό επιχειρηματικό περιβάλλον. Με την χρήση των συγκεκριμένων βάσεων δεδομένων σε μορφή JSON, μια επιχείρηση μπορεί να αποθηκεύσει και επεξεργαστεί δεδομένα από οποιαδήποτε πηγή και ταυτόχρονα να τροφοδοτήσει τις μεθόδους ανάλυσής της, χωρίς να υπάρχουν ξεχωριστά περιβάλλοντα και σχεδιασμοί.

2.4.3. JSON και XML

Πολλές φορές υπάρχει σύγχυση για τον τρόπο αναπαράστασης κάποιου μοντέλου δεδομένων. Η XML όπως αναφέρθηκε πάνω είναι ένας τρόπος, όσο και η JSON μορφή. Παρακάτω παρουσιάζονται οι διαφορές ανάμεσα στις 2 αυτές μορφές αναπαράστασης:

1. Η JSON δομή συγκεντρώνει όλα τα δεδομένα μαζί είτε είναι σε σειρά είτε όχι. Τα JSON αντικείμενα δεν είναι ταξινομημένα, αλλά όσον αφορά τους πίνακες, μπορούμε να κάνουμε τυχαία προσπέλαση αναλόγως της θέσης του. Η XML απαιτεί μια πιο αυστηρή ιεραρχία μεταξύ των κόμβων και των οντοτήτων. Μετατρέποντας JSON έγγραφα σε XML, θα υπήρχαν πολλά προβλήματα στην διασταύρωση των κόμβων και της συγγένειας μεταξύ τους.

2. Η JSON πίνακες δεν είναι ούτε λίστες ούτε σύνολα. Ναι μὲν μπορούμε να έχουμε τυχαία πρόσβαση αλλά δεν υπάρχει η δυνατότητα για τον προσδιορισμό συγγενών, επόμενων ή προηγούμενων κόμβων όπως στις λίστες. Η προσπάθεια προσαρμογής αυτού του προσδιορισμού σε διάφορες γλώσσες, αποτελεί μια σχετικά σύνθετη διαδικασία.

3. Η δομή ενός JSON αρχείου είναι μονοδιάστατη. Δηλαδή επιβάλλει το κάθε κλειδί οποιουδήποτε αντικειμένου να είναι ένα διακριτό στοιχείο, να υπάρχει μόνο μια φορά, με αποτέλεσμα ένας κόμβος να αναφέρεται μόνο σε αυτό το κλειδί. Αντίθετα στην XML δεν υπάρχει αυτός ο περιορισμός, αφού δεν υπάρχουν ετικέτες για τα αντικείμενα και ένας κόμβος μπορεί να έχει πολλούς συγγενείς.

2.5. RSS

Το RSS είναι ένα σύστημα περιεχομένου ιστοσελίδων, δομημένο από XML έγγραφα που περιέχει σύντομες περιγραφές συνδέσμων του διαδικτύου. Τα XML έγγραφα είναι προσβάσιμα μετά από μια HTTP αίτηση ενός συνδέσμου, και ο σύνδεσμος αυτός για κάθε XML έγγραφο προσδιορίζει το RSS feed. Ένας χρήστης του διαδικτύου μπορεί έτσι να ενημερώνεται αυτομάτως για γεγονότα και νέα από όσες ιστοσελίδες υποστηρίζουν RSS, αρκεί να έχει εγγραφεί ο ίδιος συνδρομητής στην αντίστοιχη υπηρεσία. Οι εν λόγω ενημερώσεις περιέχουν τα πλήρη δεδομένα, σύνοψη των δεδομένων, σχετικά μεταδεδομένα, ημερομηνία έκδοσης ενώ αποστέλλονται στον συνδρομητή μέσω Διαδικτύου.

Οι ιστότοποι συνήθως χρησιμοποιούν RSS ροές για να δημοσιεύουν συχνά πληροφορίες για το περιεχόμενο του ιστοτόπου, όπου αυτός μπορεί να είναι ένα blog, νέα ειδήσεων, οπτικοακουστικό υλικό ή διάφορα podcasts. Ένα έγγραφο RSS περιλαμβάνει ένα πλήρες ή μια σύνοψη κειμένου, μια ημερομηνία και το όνομα του συγγραφέα και ορίζεται χρησιμοποιώντας XML αρχεία.

Η RSS ροή είναι απλό κείμενο που έχει μορφοποιηθεί μέσω της XML. Η ίδια η μορφή RSS είναι σχετικά εύκολα στην ανάγνωση τόσο από μηχανές όσο και από τους ανθρώπους. Ένα παράδειγμα θα μπορούσε να έχει περιεχόμενο όπως το εξής:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <rss version="2.0">
    <channel>
      <title>RSS Title</title>
      <description>This is an example of an RSS feed</description>
      <link>http://www.example.com/main.html</link>
      <copyright>2020 Example.com All rights reserved</copyright>
      <lastBuildDate>Mon, 06 Sep 2010 00:01:00 +0000</lastBuildDate>
      <pubDate>Sun, 06 Sep 2009 16:20:00 +0000</pubDate>
      <ttl>1800</ttl>
```

Κώδικας 3: Σύνταξη RSS ροής

ΑΝΑΚΤΗΣΗ ΔΕΔΟΜΕΝΩΝ

Όταν ανακτώνται RSS δεδομένα, υπάρχουν προγράμματα ανάγνωσης XML που θα μπορούσε κάποιος να χρησιμοποιήσει για να παρουσιάσει μια τελική ροή στον χρήστη της RSS ροής, Υπάρχουν διάφορα εργαλεία και λογισμικά συγκέντρωσης νέων και δεδομένων για τον υπολογιστή ή μια κινητή συσκευή, παρόλα αυτά η τεχνολογία του RSS μπορεί να ενσωματωθεί απευθείας μέσα στους φυλλομετρητές ή σε συστήματα ηλεκτρονικού ταχυδρομείου όπως το Mozilla Thunderbird.

2.6. ATOM Feed

Η ροή ενός ιστότοπου συνήθως ονομάζεται "feed", είναι μια δομή σχετικών μεταξύ τους στοιχείων όπου ένας πάροχος δημοσιεύει στο διαδίκτυο. Ένα ATOM FEED είναι μια ροή ιστότοπου που χρησιμοποιεί την μορφή ATOM και το αντίστοιχο ATOM πρωτόκολλο για την δημοσίευση. Η ATOM οντότητα απαρτίζεται από ένα XML έγγραφο το οποίο περιγράφει ένα ATOM FEED και διάφορες πληροφορίες, και από ένα πρωτόκολλο δημοσίευσης. Σύχνα οι πάροχοι περιεχομένων δημιουργούν feeds χρησιμοποιώντας το RSS όπως παρουσιάστηκε παραπάνω. Παρόλα αυτά υπάρχουν οργανισμοί οι οποίοι υποστηρίζουν είτε το ένα απ τα δύο ή και τα δύο. Τα στοιχεία και οι πληροφορίες που δημιουργούν ένα ATOM FEED ονομάζονται ATOM ENTRIES.

Ένας πάροχος δημοσιεύει ένα ATOM FEED κάνοντας το διαθέσιμο μέσω συνδέσμων στο διαδίκτυο και ανανεώνοντας το με νέα στοιχεία. Οι ιστοσελίδες εμφανίζουν τα στοιχεία στο ATOM FEED και οι χρήστες μπορούν να τα λάβουν χρησιμοποιώντας πρόγραμμα ή φυλλομετρητές που υποστηρίζουν τετοιές ροές. Γενικά, το ATOM FEED μπορεί να χρησιμοποιηθεί σαν ένα μέρος ενός μεγαλύτερου συνόλου πληροφορίας, όπου μαζί με μια ιστοσελίδα συγχωνεύουν ένα μεγάλο κομμάτι δεδομένων και δίνουν στον χρήστη την ευκαιρία να κατανοήσει όλα αυτά τα δεδομένα με ένα πιο εύκολο τρόπο.

Ένα παράδειγμα ενός ATOM εγγράφου:

```
<?xml version="1.0" encoding="utf-8"?>
  <feed xmlns="http://www.w3.org/2005/Atom">

    <title>Example Feed</title>
    <subtitle>A subtitle.</subtitle>
    <link href="http://example.org/feed/" rel="self" />
    <link href="http://example.org/" />
    <id>urn:uuid:60a76c80-d399-11d9-b91c-0003939e0af6</id>
    <updated>2003-12-13T18:30:02Z</updated>

    <entry>
      <title>Atom-Powered Robots Run Amok</title>
      <link href="http://example.org/2003/12/13/atom03" />
      <link rel="alternate" type="text/html" href="http://
example.org/2003/12/13/atom03.html"/>
      <link rel="edit" href="http://example.org/2003/12/13/atom03/
edit"/>
```

```

<updated>2003-12-13T18:30:02Z</updated>

  <summary>Some text.</summary>

  <content type="xhtml">
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p>This is the entry content.</p>
    </div>
  </content>

  <author>
    <name>John Doe</name>
    <email>johndoe@example.com</email>
  </author>
</entry>
</feed>

```

Κώδικας 4: Σύνταξη ATOM εγγράφου

2.6.1. Διαφορές ATOM και RSS

ΜΟΡΦΟΠΟΙΗΣΗ ΔΕΔΟΜΕΝΩΝ

Οι προδιαγραφές του RSS βασίζονται στη χρήση του χρονοδιαγράμματος του RFC 822, ώστε να λαμβάνει πληροφορίες σχετικά με το πότε δημιουργήθηκαν τα στοιχεία του FEED και πότε ενημερώθηκαν τελευταία φορά. Αντίθετα η ομάδα του ATOM αποφάσισε να χρησιμοποιήσει το χρονοδιάγραμμα και τους κανόνες που καθορίζονται απ το RFC 3339.

ΔΙΕΘΝΟΠΟΙΗΣΗ ΤΗΣ ΓΛΩΣΣΑΣ

Ενώ το λεξιλόγιο του RSS διαθέτει μηχανισμό για να υποδείξει την ανθρώπινη γλώσσα, δεν υπάρχει τρόπος να καθοριστεί μια συγκεκριμένη γλώσσα για συγκεκριμένα στοιχεία.

Απ την άλλη πλευρά, το ATOM χρησιμοποιεί την Standard XML έτσι ώστε να υπάρχει η δυνατότητα καθορισμού ενός γλωσσικού πλαισίου για κάθε κομμάτι που υπάρχει στο feed και είναι σε ανθρώπινη γλώσσα.

Το ATOM επίσης διαφέρει απ το RSS γιατί υποστηρίζει την χρήση πόρων που αναγνωρίζουν την γλώσσα, οι οποίοι μπορούν με συνδέσμους ή identifiers να παράξουν χαρακτήρες και εκτός του συνόλου χαρακτήρων US ASCII.

ΔΟΜΗ

Τα στοιχεία του λεξιλογίου του RSS δεν είναι γενικά επαναχρησιμοποιήσιμα σε άλλα λεξιλόγια. Από την άλλη πλευρά, η σύνταξη και το λεξιλόγιο του ATOM σχεδιάστηκε ειδικά έτσι ώστε να επιτρέπεται η επαναχρησιμοποίηση και εκτός του ATOM FEED. Για παράδειγμα, υπάρχουν περιπτώσεις που συναντιούνται ATOM σύνδεσμοι σε RSS FEEDs.

3. Το σύστημα

3.1. Εισαγωγή

Η βασική ιδέα είναι ότι με την εκτέλεση μια εντολής σε ένα command line, ο χρήστης θα μπορεί να αναμεταδίδει οποιαδήποτε ενημέρωση έχει υπάρξει ανάμεσα στους εξυπηρέτες σε διάστημα 30 λεπτών. Το προτεινόμενο σύστημα αναφέρεται σε ένα shell script αρχείο το οποίο είναι υλοποιημένο σε Python 3 και δύο αρχεία σε μορφή JSON που βοηθούν την αρχικοποίηση και την αποθήκευση.

3.2. Αυτοματοποίηση σε Python 3

Η γλώσσα προγραμματισμού Python αποτελεί ένα ισχυρό εργαλείο που διαθέτει πολλές ιδιότητες για αυτοματοποιημένο έλεγχο. Είναι ανοιχτού κώδικα, διατίθεται δωρεάν και συνεχώς αυξάνεται μέσω της συμβολής μιας διεθνούς κοινότητας προγραμματιστών. Η σύνταξη είναι ιδιαίτερα αναγνώσιμη και σχεδιάστηκε με σαφήνεια. Είναι ευέλικτη και επεκτάσιμη, ανταγωνίζεται σε ταχύτητα υπολογισμού τις ταχύτερες γλώσσες και απαιτεί ελάχιστες γραμμές κώδικα για υψηλή παραγωγικότητα. Επιπλέον τα πακέτα αυτοματισμού όπως και της επικοινωνίας με διάφορους εξυπηρέτες που είναι γραμμένα σε Python, είναι σχεδόν ανεξάρτητα από την πλατφόρμα, καθώς υποστηρίζονται από τα περισσότερα λειτουργικά συστήματα συμπεριλαμβανομένων των Linux, Windows, Max OS X.

3.3. Αλληλεπίδραση με JSON αρχεία

Η JavaScript Object Notation είναι μια τυπική μορφή αρχείου και ανταλλαγής δεδομένων, που χρησιμοποιεί κείμενο αναγνώσιμο από τον άνθρωπο για την αποθήκευση και την μετάδοση και αποτελείται από ζεύγη key-value και αντικείμενα πινάκων. Συγκεκριμένα η χρήση αρχείων JSON στην Python επιτυγχάνεται με την υπάρχουσα βιβλιοθήκη json που υπάρχει εγκατεστημένη στη γλώσσα προγραμματισμού. Η χρήση στο επιτυγχάνεται με την παρακάτω γραμμή κώδικα:

```
import json
```

Η αλληλεπίδραση, το διάβασμα αρχείων JSON και η εγγραφή σε αρχεία JSON, υλοποιείται με την χρήση συνήθως των μεθόδων json.dumps(), json.dump() για εγγραφή και json.loads(), json.load() για διάβασμα. Χρησιμοποιούνται δυο αρχεία σε μορφή JSON στο σύστημα, το servers.json που περνάει σαν όρισμα και το database.json, μια "πρόχειρη" βάση δεδομένων που αποθηκεύει τις αναρτήσεις και τις ανακοινώσεις. Μερικά ζεύγη key-value στα δύο αρχεία:

```
Servers.json :
"servers_in": {
    "facebook": {
        "url": "facebook.com",
        "last_facebook_update": "2020-09-23T10:56:16+0000",
        "last_feed_update": "2020-09-18T18:19:19.545516"
    },
```

```

"twitter": {
    "url": "www.twitter.com",
    "last_twitter_update": "Wed Sep 23 10:56:19 +0000 2020"
}

Database.json: {
    "server_in": "Twitter",
    "rss_post_id": "null",
    "fb_post_id": "110315817302887_173477600986708",
    "twitter_post_id": "1309493308943630336",
    "gmail_id": "174c591f9dd0f05e",
    "post_link": "Send this to fb gmail only",
    "deletion_date": "2021-07-25"
}

```

Κώδικας 5: Key-values ζεύγη JSON αρχείων

3.4. Σύνδεση APIs με Python 3

Καθώς το σύστημα αλληλεπιδρά με διακομιστές(Facebook, Twitter κτλπ) και άρα με λειτουργικά περιβάλλοντα, είναι απαραίτητη η σύνδεση με τα αντίστοιχα APIs των παραπάνω εφαρμογών. Όπως παρουσιάζεται και παρακάτω, μια διεπαφή προγράμματος (Application Program Interface - API) είναι ένα σύνολο ρουτινών, πρωτοκόλλων και εργαλείων για τη δημιουργία εφαρμογών λογισμικού. Η Python προσφέρει αντίστοιχες βιβλιοθήκες για κάθε ένα λειτουργικό περιβάλλον που χρησιμοποιήθηκε ώστε να υπάρχει συνέπεια στην επικοινωνία, όπως και στην υλοποίηση του συστήματος. Κάθε ένα API προσφέρει documentation στο πώς μπορεί να χρησιμοποιηθεί και τους περιορισμούς που αντιμετωπίζει.

3.5. Διαχείριση χρόνου στην Python 3

Το Σύστημα μετάδοσης ειδοποιήσεων που παρουσιάζεται, έχει ρυθμιστεί στο να εκτελείται κάθε 30 λεπτά. Ο όρος "εκτελείται" αναφέρεται ουσιαστικά στην λειτουργία όλων των μεθόδων που έχουν οριστεί στο πρόγραμμα και στο αποτέλεσμα το οποίο θα επιφέρουν, δηλαδή στην επικοινωνία και την μετάδοση όλων των ειδοποιήσεων και ανακοινώσεων. Η διαχείριση του χρόνου στην Python και

συγκεκριμένα σε αυτό το σύστημα επιτυγχάνεται με την χρήση της βιβλιοθήκης `time()` η οποία προσφέρει μεθόδους σχετικές με τον χρόνο, την ώρα και την ημερομηνία. Η μέθοδοι που έχουν χρησιμοποιηθεί από την βιβλιοθήκη `sleep()` είναι αρχικά η `time.sleep()` οι οποία σταματάει οποιαδήποτε εκτέλεση στο πρόγραμμα για τον ορισμένο αριθμό δευτερολέπτων, δηλαδή το πέρασμα των 30 λεπτών γίνεται ουσιαστικά με το κάλεσμα της `time.sleep(1800)`.

3.6. Διαχείριση ημερομηνιών στην Python 3

Μιλώντας για χρόνο, η λογική του συστήματος ειδοποιήσεων όπως και η υλοποίηση συναντάται άμεσα με τον διαχωρισμό όλων των πληροφοριών ανάλογα με την ημερομηνία εγγραφής, δηλαδή θα πρέπει οι πιο πρόσφατες ειδοποιήσεις να είναι έτοιμες να μεταδοθούν, όπως επίσης μια ειδοποίηση να μην υπάρχει για πάντα αναρτημένη. Όλα αυτά προϋποθέτουν ελέγχους αντικείμενων της Python που δεν είναι ίδια μεταξύ τους, συγκεκριμένα `Date` και `String` αντικείμενα, διότι η επικοινωνία με διακομιστές ορίζεται με συγκεκριμένους κανόνες όσον αφορά την λήψη πληροφοριών. Η βασική βιβλιοθήκη για τις ημερομηνίες είναι η `datetime()` και θα παρουσιαστεί μέσω ενός παραδείγματος. Όταν ένας προγραμματιστής επιθυμεί να χρησιμοποιήσει μια ημερομηνία από μια ανάρτηση του Facebook, δεν έχει άλλο τρόπο απ το να εκλάβει ένα `Date` αντικείμενο με την ημερομηνία, καθώς αυτό το αντικείμενο χρειαστεί να συγκριθεί με κάποιο άλλο τύπου `String` θα πρέπει να γίνει μια μετατροπή ενός από τα δύο αυτά αντικείμενα. Η μέθοδος `datetime.strftime()` μετατρέπει τα αντικείμενα `Date` σε `string`, η μέθοδος `datetime.strptime()` μετατρέπει τα `String` αντικείμενα σε `Date` με οποιοδήποτε `format` επιθυμούμε πχ. "Thu Oct 01 09:48:27" ή "2020-09-25T16:21:38".

3.7. Διαχείριση παλιών ανακοινώσεων στην Python 3

Κάθε φορά που γίνεται μια εγγραφή στην "βάση" δεδομένων, αυτόματα στην νέα ανακοίνωση δημιουργείται μια ημερομηνία διαγραφής, και κάθε φορά συγκρίνεται με την ημερομηνία που εκτελείται το Σύστημα, αν τυχόν έχει περάσει το διάστημα των 10 μηνών τότε η ανακοίνωση διαγράφεται απ την κύρια βάση δεδομένων. Οι ανακοινώσεις που διαγράφηκαν μετακινούνται κάθε φορά σε ένα δεύτερο αρχείο που αποτελεί μια μικρότερη βάση δεδομένων που παίρνει το όνομα της ανάλογα με την ημέρα διαγραφής και στη συνέχεια όλες αυτές οι μικρές βάσεις συμπίεζονται σε ένα συνολικό αρχείο.

3.8. Παράδειγμα χρήσης του Συστήματος

Παρακάτω γίνεται μια παρουσίαση στο πως λειτουργεί το Σύστημα Διάδοσης Ειδοποιήσεων, όταν χρειάζεται να μεταδοθεί μια ανακοίνωση από ένα RSS Feed στο facebook, στο Twitter και στο GMAIL. Όλα ξεκινούν με το `servers.json` αρχείο. Αρχικά για να μπορέσει κάποιος να βάλει σε λειτουργία το σύστημα, χρειάζεται να έχει εγκατεστημένη την γλώσσα προγραμματισμού Python 3, μέσα απ την γραμμή εντολών να κατευθυνθεί στον φάκελο του συστήματος και με την εντολή:

```
python3 notify_sys.py servers.json
```

Όστε να μπορέσει να ξεκινήσει το σύστημα, δίνοντας σαν όρισμα στο σύστημα το `servers.json` αρχείο.

Στο αρχείο αυτό υπάρχουν οι πληροφορίες που χρειάζεται το σύστημα για να μπορέσει να επικοινωνήσει με τα APIs των παραπάνω διακομιστών. Γίνεται μια αντιστοιχία των πεδίων του `servers.json` με το πρόγραμμα ώστε να οριστικοποιηθούν οι σταθερές του συστήματος, υπάρχουν πεδία όπως:

```

"url": "https://api.iee.ihu.gr/announcements/feed/atom",
  "consumer_key": "NhacJJopaloBj9zEAnXUC0Tk1",
  "page_id": "110315817302887",
  "emails": [
    "antwnhs.barbas@gmail.com"
  ],

```

Κώδικας 6: Πεδία servers.json

που αρχικοποιούνται αντίστοιχα στις σταθερές:

```

tw_consumer_key = servers['servers_out']['twitter']['consumer_key']
rss_feed_url = (servers['servers_in']['RSS Feed']['url'])
fb_pageID = servers['servers_out']['facebook']['page_id']
emails = servers['servers_out']['emails']

```

Κώδικας 7: Σταθερές του συστήματος

Το γεγονός αυτό βοηθάει στην διαχείριση των πόρων του συστήματος και στην απόδοση, γιατί εφόσον και το servers.json παραμένει σταθερό δεν χρειάζεται να καλείται συνέχεια στο σύστημα ώστε αυτό να λαμβάνει τις απαραίτητες τιμές.

Στη συνέχεια καλείται η αρχική μέθοδος του συστήματος whereToSend(), όπου στον κορμό της υπάρχουν οι βασικές μέθοδοι.

DEFINE_DB(): Ορίζει το αρχείο που θα χρησιμοποιείται σαν βάση δεδομένων για τις ανακοινώσεις. Το όνομα της βάσεις είναι μεταβλητό ανάλογα πως έχει δηλωθεί στο servers.json, γιατί μπορεί να υπάρχουν πολλές βάσεις για διαφορετική θεματολογία ανακοινώσεων. Σε περίπτωση που το αρχείο υπάρχει τότε δεν ξανά δημιουργείται.

delete_posts(): Ελέγχει αν υπάρχει στην βάση κάποια ανακοίνωση η οποία πρόκειται να διαγραφεί επειδή έχει ξεπεράσει το χρονικό όριο παραμονής στη βάση δεδομένων. Κάθε ανακοίνωση στην βάση έχει το πεδίο deletion_date, το οποίο συγκρίνεται κάθε φορά με την ημερομηνία που τρέχει το σύστημα.

where_to_send(): Είναι η μέθοδος που ξεκινάει την διαδικασία ελέγχου και μέσω αυτής καλούνται και οι μέθοδοι αποστολής όλων των ανακοινώσεων μεταξύ των διαδομιστών. Αρχικά

ελέγχονται οι πλατφόρμες με τις ανακοινώσεις, αν υπάρχει κάποια καινούργια. Αυτό επιτυγχάνεται με το `last_feed_update` πεδίο που περιέχει την τελευταία ενημέρωση και συγκρίνεται με την ημερομηνία της τελευταίας ανακοίνωσης της πλατφόρμας, αν είναι μικρότερη τότε ξεκινάει η διαδικασία της μετάδοσης.

ΜΕΤΑΔΟΣΗ

Έστω ότι υπάρχει μια νέα ανακοίνωση απ το RSS Feed. Η ημερομηνία δημιουργίας της ανακοίνωσης είναι μεγαλύτερη απ το πεδίο του συστήματος. Τότε με την βοήθεια της μεθόδου `post_from_RSS()` γίνεται η μετάδοσή της.

Αρχικά γίνεται έλεγχος της ταυτότητα της ανακοίνωσης στην βάση, ώστε αν υπάρχει να μην ξανά μεταδοθεί, παρόλο που το περιεχόμενο ή ο τίτλος είναι ίδιος γίνεται έλεγχος μόνο για την ταυτότητα. Αυτό επιτυγχάνεται με τις εντολές:

```
for posts in jdata['database']['posts']:
    if rss_post.id in posts.values():
        print(rss_post.id + "Already exists")
        rss_post_flag = False
    break
```

Κώδικας 8: Έλεγχος ταυτότητας για διπλότυπες ανακοινώσεις

Όπου με την βοήθεια της εντολής `for`, γίνεται μια επαναληπτική διαδικασία για κάθε ανακοίνωση στη βάση, το οποίο ελέγχεται με την ταυτότητα της ανακοίνωσης που πρόκειται να μεταδοθεί. Η μέθοδος `values()` έχει αυτόν το σκοπό, με το να προσφέρει τις τιμές των πεδίων της βάσης για σύγκριση. Στη συνέχεια εάν υπάρχει η ίδια ταυτότητα στη βάση, τότε υπάρχει η μεταβλητή `rss_post_flag` που με βάση αυτή ορίζεται η μετάδοση.

Ένα επίπεδο πάνω στη μέθοδο, υπάρχει η σύγκριση της μεταβλητής `rss_post_flag`, η οποία ξεκινάει με την τιμή `TRUE`. Αν υπάρχει διπλότυπο μέσα στη βάση όπως αναφέρθηκε, στον εσωτερικό βρόγχο η τιμή της μετατρέπεται σε `FALSE`, έτσι διακόπτεται και η μετάδοση της ανακοίνωσης. Σε περίπτωση που συνεχίζει να είναι `TRUE` τότε, πρώτη πλατφόρμα ανακοίνωσης είναι το Facebook.

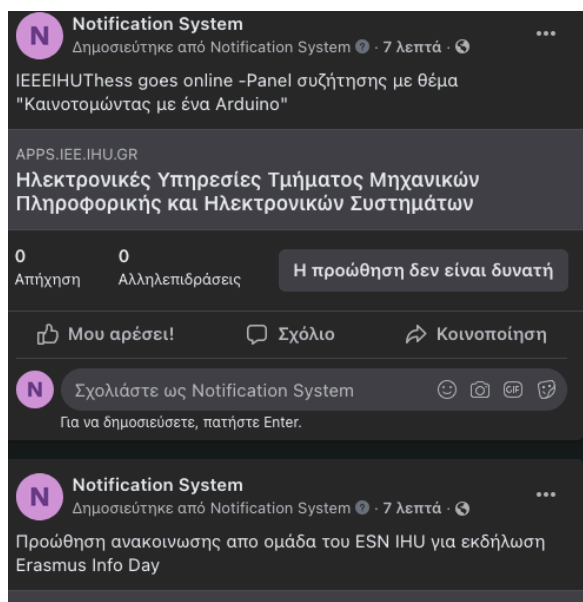
Το Facebook API μας προσφέρει την μέθοδο `put_object()`. Η μέθοδος αυτή, δεδομένων των στοιχείων του API που την έχουν δοθεί και τον προορισμό μετάδοσης αναρτεί την ανακοίνωση στο Facebook. Συγκεκριμένα τα ορίσματα της μεθόδου του Συστήματος είναι τα εξής:

parent_object=fb_pageID: Είναι ο προορισμός που θέλουμε να αναρτηθεί η ανακοίνωση. Συνήθως είναι μια ομάδα του Facebook που έχει δημιουργηθεί για αυτό το σκοπό

connection_name='feed': Είναι σύνδεση μεταξύ των προορισμών, αποτελεί το παιδί της ομάδας που θα μεταδοθεί η ανακοίνωση. Για παράδειγμα, το σχόλιο της ανακοίνωσης αποτελεί "παιδί" της ανακοίνωσης και θα έπρεπε να δηλωθεί με αυτό το όρισμα.

message=rss_post.title: Συνήθως κάθε ανακοίνωση στο Facebook έχει ένα τίτλο, το όρισμα αυτό δημιουργεί ακριβώς αυτό το αντικείμενο

link=rss_post.link: Επίσης κάθε ανακοίνωση, μπορεί να έχει υπερσυνδέσμους προς άλλες ιστοσελίδες εκτός από λόγια, με το όρισμα αυτό μεταδίδεται ο σύνδεσμος της RSS ανακοίνωσης.

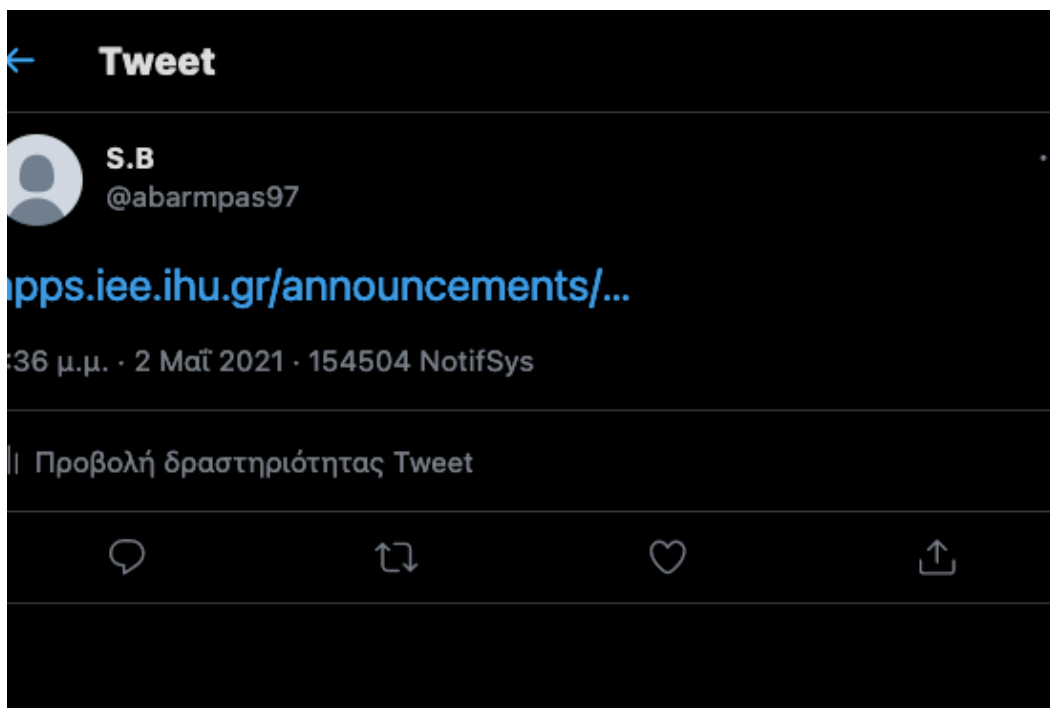


Εικόνα 1: Δημοσίευση ανακοίνωσης στο Facebook

Το Twitter API επιτυγχάνει την μετάδοση με την μέθοδο `update_status()`. Η μέθοδος αυτή είναι πίο απλή απ την αντίστοιχη του Facebook, για τον λόγο ότι χρησιμοποιείται απ το Tweepy το οποίο είναι ένα εργαλείο που απαλείφει της χαμηλού επιπέδου λεπτομέρειες και χρειάζεται μόνο ένα όρισμα του συνδέσμου της ανακοίνωσης:

```
api.update_status(rss_post.link)
```

Καθώς μπορεί να αποκωδικοποιεί την XML μορφή κειμένου και να λαμβάνει όλα τα πεδία τα οποία πρόκειται να προβληθούν στη σελίδα του Twitter.



Εικόνα 2: Δημοσίευση ανακοίνωσης στο Twitter

Όσον αφορά το GMAIL, υπάρχει μια διαφορετική διαδικασία. Πρώτα γίνεται η δημιουργία του μηνύματος που επιθυμούμε να υπάρχει στο ηλεκτρονικό μήνυμα Για παράδειγμα:

```
message = "Title: " + rss_post.title + "\nLink: " + rss_post.link +  
"\nPublished: " + rss_post.updated
```

```
for profile in emails:  
    reciever = profile  
    emailBd = create_message(sender, reciever, "New  
post from RSS Feed", message, "text")  
    email = send_message(service, 'me', emailBd)
```

Κώδικας 9: Εντολή επανάληψης δημιουργίας και αποστολής μηνύματος

Και στη συνέχεια όπως αναφέρθηκε, υπάρχουν πεδία μέσα στο `servers.json` που ορίζουν τον προορισμό, δηλαδή τις ηλεκτρονικές διευθύνσεις που θα μεταδοθεί η ανακοίνωση. Για κάθε ηλεκτρονική διεύθυνση που υπάρχει, θα δημιουργηθεί ένα μήνυμα με την μέθοδο `create_message()` και σταλεί με την μέθοδο `send_message()`.

Η μέθοδος `create_message()` δέχεται σαν ορίσματα, την διεύθυνση του παραλήπτη, την διεύθυνση του αποστολέα, το θέμα του μηνύματος, το μήνυμα και το είδος γιατί υπάρχει πιθανότητα, ο κορμός του μηνύματος να περιέχει φωτογραφίες ή δυναμικά καταχωρημένο HTML κώδικα. Η μέθοδος αυτή προσφέρεται απ το API του GMAIL, είναι τροποποιήσιμη αλλά επειδή επιστρέφει το μήνυμα σε μορφή που συμφωνεί με τους κανόνες του API, πρέπει να αποθηκεύεται σε μεταβλητή.

Η μέθοδος `send_message()` δέχεται σαν όρισμα, την υπηρεσία που χρησιμοποιεί ο χρήστης και έχει επαληθεύσει με τα στοιχεία που του έχει προσφέρει το API, την ταυτότητα του χρήστη και το μήνυμα που έχει δημιουργηθεί προηγουμένως.

Στη συνέχεια γίνεται μια παύση του συστήματος για μερικά δευτερόλεπτα ώστε, να μπορέσουν να ανανεωθούν όλες οι πλατφόρμες. Αυτό βοηθάει στο γεγονός ότι το σύστημα χρειάζεται να αποθηκεύσει την τελευταία ημερομηνία που υπήρξε μετάδοση, ώστε να ξεκινήσει ξανά τον έλεγχο, δηλαδή την ημερομηνία της πιο πρόσφατης ανακοίνωσης. Καθώς αποκτούνται οι ημερομηνίες, γίνεται και αποθήκευση της νέας ανακοίνωσης στη βάση, πρώτα γράφονται όλα τα στοιχεία που χρειάζονται απ την ανακοίνωση σε ένα JSON αντικείμενο και έπειτα η μέθοδος `append()` ενώνει το json αρχείο της βάσης με την νέα json εγγραφή.

Τέλος, καθ' όλη τη διάρκεια, επειδή το Σύστημα εξαρτάται προφανώς απ την λειτουργία των διακομιστών, οι υπηρεσίες των APIs μπορεί να μην είναι διαθέσιμες με αποτέλεσμα να δημιουργηθούν προβλήματα στη μετάδοση. Προς το παρόν το σύστημα περιέχει ενέργειες, οι οποίες δεν θα σταματήσουν την λειτουργία του, που αφορούν την διπλότυπη μετάδοση και απαγορεύονται από την πολιτική του διακομιστή. Για παράδειγμα:

```
except tweepy.TweepError as error:
    if error.api_code == 187:
        print(" ")
except Facebook.GraphAPIError as dupli:
    if dupli.code == 506:
        print(" ")
```

Κώδικας 10: Κώδικας πρόβλεψης σφαλμάτων από τους διακομιστές

3.9. Επίλογος

Στο συγκεκριμένο κεφάλαιο παρουσιάστηκε η δομή και η υλοποίηση του συστήματος μέσω όλων των μεθόδων και βιβλιοθηκών που υπάρχουν στην Python 3 και πώς κάθε στοιχείο του συστήματος δημιουργείται και επεξεργάζεται μέσω του κώδικα.

4. Facebook API

Όπως έχει αναφερθεί, το Σύστημα μετάδοσης ειδοποιήσεων πρέπει να αλληλεπιδρά με διάφορους διακομιστές και χρήση της Python 3 προσφέρει πολλά πλεονεκτήματα και παρέχει πολύ μεγάλο χώρο για βελτίωση όσον αφορά την επικοινωνία αυτή. Τα πλεονεκτήματα αυτά αναφέρονται στην διαχείριση πληροφοριών ανάμεσα σε πλατφόρμες όπως είναι το Facebook, το Twitter. Συγκεκριμένα όπως περιγράφεται παρακάτω ένα εύχρηστο API για την πλατφόρμα του Facebook είναι το Facebook's Graph API.

4.1. Περιγραφή του Facebook Graph API

Το Graph API του Facebook είναι ο βασικός τρόπος για να πάρουμε δεδομένα από την πλατφόρμα του Facebook όπως επίσης και να προσθέσουμε και να αναρτήσουμε δεδομένα σε αυτό. Είναι ένα API βασισμένο σε HTTP όπου μπορούν οι διάφορες εφαρμογές όπως και το Σύστημα ειδοποιήσεων να χρησιμοποιούν ώστε φιλτράρουν δεδομένα, να αναρτούν νέες φωτογραφίες και ανακοινώσεις, να διαχειρίζονται τις διαφημίσεις και να εκτελούν ένα μεγάλο εύρος λειτουργιών με την χρήση shell script αρχείων και προγραμμάτων.

Μέχρι πρόσφατα, το Graph API του Facebook περνούσε δεδομένα σε εφαρμογές με την μορφή JSON. Οπότε κάθε δεδομένο μπορεί να χρησιμοποιηθεί και να αξιοποιηθεί με την μορφή αυτή. Οι κύριες πληροφορίες που μας ενδιαφέρουν σε αυτό το πρόγραμμα είναι το ID της ανακοίνωσης, η ημερομηνία της ανακοίνωσης, όπως επίσης και τα πεδία για την αυθεντικοποίηση του API, παρακάτω περιγράφεται αναλυτικά η χρήση του Graph API του Facebook.

4.2. Χρήση του Facebook Graph API

Γενικά, το Graph API αναφέρεται στην ιδέα ενός γραφήματος κοινωνικής δικτύωσης. Ένα γράφημα κοινωνικής δικτύωσης ουσιαστικά αποτελείται από ένα γράφημα το οποίο απεικονίζει σχέσης και διασυνδέσεις ανάμεσα σε ανθρώπους, ομάδων, οργανισμών σε ένα μέσο κοινωνικής δικτύωσης , συγκεκριμένα του Facebook.

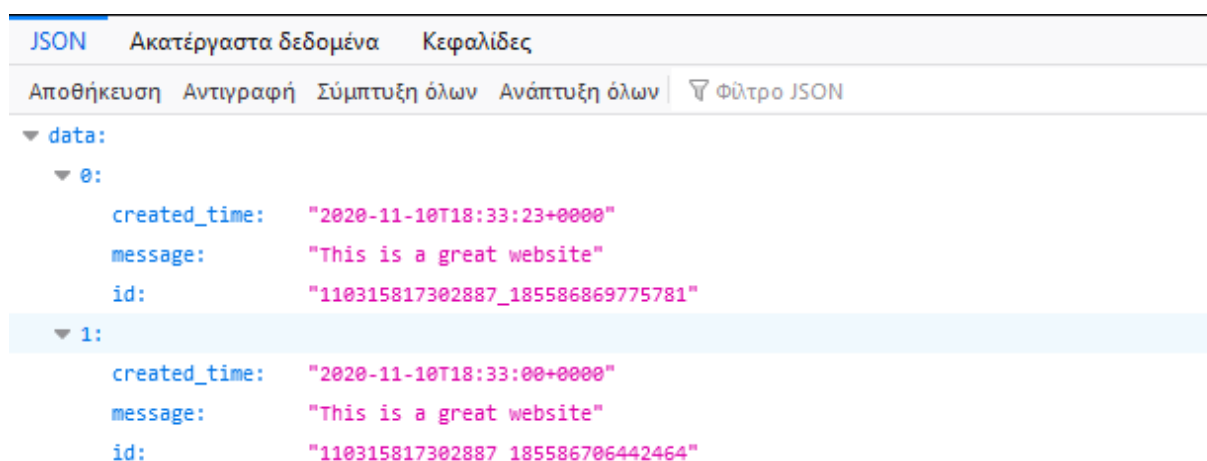
Από την άλλη πλευρά, ο όρος «γράφημα κοινωνικής δικτύωσης» χρησιμοποιείται επίσης για την περιγραφή του μέσου κοινωνικής δικτύωσης ενός και μόνο ατόμου. Παρουσιάζοντας το Graph API σαν χάρτη, ο οποίος θα αποτελείται από κόμβους ατόμων ενωμένα με γραμμές. Οι κόμβοι αναφέρονται σε αντικείμενα ατόμων, ομάδων, σελίδων και κάθε ένα από αυτά έχει τα δικά του πεδία δεδομένων όπως το όνομα, ημερομηνία γενεθλίων, ημερομηνία δημιουργίας κοκ.

Το συγκεκριμένο API είναι βασισμένο στο HTTP πρωτόκολλο, το οποίο σημαίνει ότι λειτουργεί με οποιαδήποτε γλώσσα προγραμματισμού που έχει μια HTTP βιβλιοθήκη, συγκεκριμένα στην Python έγινε η χρήση της request βιβλιοθήκης. Επιπλέον, το API είναι προσβάσιμο και από έναν browser με την μορφή ενός συνδέσμου. Ο σύνδεσμος που χρησιμοποιήθηκε κυρίως είναι η αρχική σελίδα της

εφαρμογής και του συστήματος από την οποία μεταδίδονται αυτόματα οι ανακοινώσεις σε άλλους διακομιστές.

4.3. Πεδία αρχικής σελίδας της Facebook εφαρμογής

Σύμφωνα με της προηγούμενη υποενότητα, αν κάποιος πληκτρολογήσει το https://graph.facebook.com/110315817302887/feed?access_token=xxxxxxx σε έναν browser θα πάρει όλη την αρχική σελίδα της εφαρμογής σε μορφή JSON με τα αντίστοιχα πεδία όπως παρουσιάζονται παρακάτω. Το “created_time” αναφέρεται στην ημερομηνία που δημιουργήθηκε η ανακοίνωση, το “message” η περιγραφή της και το “id” ο μοναδικός κωδικός της. Πως μπορεί όμως κάποιος να δημιουργήσει μια Facebook εφαρμογή ώστε να την συνδέσει με μια κοινόχρηστη Facebook σελίδα?



Εικόνα 2: Πεδία της εφαρμογής σε μορφή JSON

4.4. Δημιουργία Facebook εφαρμογής

Πρώτα απ'όλα, το Facebook όπως και τα περισσότερα social media, προσφέρουν ιστοσελίδες που απευθύνονται σε προγραμματιστές και δημιουργούς λογισμικών. Για να δημιουργήσει κάποιος μια εφαρμογή ώστε να την αλληλεπιδρά με την σελίδα του στο Facebook, θα πρέπει να δημιουργήσει ένα λογαριασμό στο developers.facebook.com. Έπειτα να ακολουθήσει τα βήματα τα οποία θα του δώσουν αρχικά το access_token και στη συνέχεια όλες τις άδειες που χρειάζονται για την χρήση των δεδομένων του Facebook.

Απο την ιστοσελίδα Facebook for Developers, μπορεί κάποιος, αφότου δημιουργήσει την εφαρμογή και παραλάβει τις απαραίτητες άδειες χρήσης, να χρησιμοποιήσει τα εργαλείο που του προσφέρονται και με τις κατάλληλες εντολές να πάρει τα δεδομένα που χρειάζεται. Για να μπορέσει όμως να τα εξάγει στο πρόγραμμα του θα πρέπει μέσω βιβλιοθηκών να στήσει μια γέφυρα επικοινωνίας ανάμεσα στο Facebook και το πρόγραμμα που θα το χρησιμοποιήσει, στην προκειμένη περίπτωση το Σύστημα Διάδοσης ειδοποιήσεων.

Στην Python 3, θα πρέπει κάποιος να εγκαταστήσει το facebook-sdk και να προσθέσει τις βιβλιοθήκες urllib3, facebook, requests. Ένα παράδειγμα εντολής που υπάρχει στο Σύστημα της παρούσας εργασίας είναι:

```
fb_api.put_object(parent_object=fb_pageID,  
connection_name='feed',message=status.text, link=status.source_url)
```

Η εντολή αυτή προσθέτει μια ανακοίνωση στον "τοίχο" του parent_object που στην προκειμένη είναι μια συγκεκριμένη ομάδα στο Facebook, ακριβώς στη διεύθυνση που θέλουμε μέσω του connection_name και σύμφωνα με τη παράμετρο message και link, η ανακοίνωση παίρνει την μορφή μιας δημοσίευσης του Facebook.

4.5. Επίλογος

Όπως παρουσιάζεται στην συγκεκριμένη εργασία, τα κοινωνικά μέσα δικτύωσης παίζουν μεγάλο ρόλο όσον αφορά τον στόχο και το αποτέλεσμα που θα επέλθει μέσω του Συστήματος Διάδοσης Ειδοποιήσεων. Πλαισιώνουν τον σκοπό του έργου που είναι η επικοινωνία των χρηστών μαζί τους για οποιοδήποτε λόγο χρειαστεί. Όπως παρουσιάστηκε ο συνδυασμός της ανάπτυξης λογισμικού και του Facebook API προσφέρουν εργαλεία και υπηρεσίες για το Σύστημα Διάδοσης Ειδοποιήσεων ώστε να μπορέσει να δημιουργήσει ένα σύνολο αυτοματοποιημένων διαδικασιών με στόχο προφανώς την διάδοση των ειδοποιήσεων. Οι απαιτήσεις για την χρήση του Facebook API είναι η δημιουργία μια εφαρμογής στην αντίστοιχη ιστοσελίδα και η ενσωμάτωση μέσω βιβλιοθηκών στον κώδικα του προγράμματος. Το Facebook είναι απ τα μέσα κοινωνικής δικτύωσης που προσφέρει και υποστηρίζει μια πληθώρα εργαλείων και υπηρεσιών για οποιοδήποτε πρόγραμμα.

5. Twitter API

5.1. Εισαγωγή

Το Twitter, είναι αποτελεί ένα παγκόσμιο δίκτυο που χρησιμοποιείται από εκατομμύρια χρήστες κατά κύριο λόγο για να έρχονται σε επαφή με τους φίλους, την οικογένεια ή τους συνεργάτες τους μέσω ενός υπολογιστή ή ενός κινητού. Έχει τη δυνατότητα να προσφέρει πληροφορίες και δεδομένα για οποιαδήποτε εφαρμογή, αυτό σημαίνει ότι το δίκτυο του Twitter μπορεί να ενσωματωθεί σε συστήματα τα οποία θέλουν να έχουν μια αλληλεπίδραση με τα μέσα κοινωνικής δικτύωσης. Όπως και με το Facebook έτσι και με το Twitter το Σύστημα Διάδοσης Ειδοποιήσεων χρησιμοποιεί το API του μέσου αυτού ώστε να μπορέσει να ενσωματώσει όλες τις απαραίτητες υπηρεσίες για την υλοποίηση της διάδοσης.

5.2. Περιγραφή Twitter API

Τα δεδομένα του Twitter όπως και των υπόλοιπων μέσων κοινωνικής δικτύωσης είναι μοναδικά, δεν σχετίζονται με κάποια άλλη πλατφόρμα, και αντανακλούν πληροφορίες τις οποίες οι χρήστες έχουν συμφωνήσει να δημοσιεύσουν. Η εφαρμογή του αντίστοιχου API προσφέρει πρόσβαση σε δημόσια δεδομένα του Twitter τα οποία οι χρήστες έχουν δεχτεί να δημοσιεύσουν και επίσης υποστηρίζει υπηρεσίες για την διαχείριση των προσωπικών τους δεδομένων όπως μηνύματα, στοιχεία και προσφέρει τις ανάλογες πληροφορίες και όποια χρήση δικαιούνται πάνω σε αυτά, στους προγραμματιστές.

Συγκεκριμένα, το Twitter API μπορεί να χρησιμοποιηθεί για οποιαδήποτε εγγραφή ή διάβασμα πληροφοριών που σχετίζονται με οντότητες του Twitter, καθώς και πολλές HTTP διευθύνσεις αναφερόμενες (όπως ονομάζονται στον "χώρο" αυτό):

- *Tweets*
- *Retweets*
- *Likes*
- *Messages*
- *Favorites*
- *Trends*

Το Tweepy όπως φαίνεται παρακάτω, δημιουργεί ένα τρόπο να επικαλείται αυτές τις HTTP διευθύνσεις χωρίς χρειάζονται περαιτέρω πληροφορίες για το πως είναι δυνατή η πρόσβαση σε αυτές.

Επίσης το Twitter API χρησιμοποιεί το OAuth σαν πρωτόκολλο αυθεντικοποίησης για όλες τις υπηρεσίες καθώς και όριο χρήσης για το πόσο συχνά καλούνται οι μέθοδοι της εφαρμογής. Σε περίπτωση που κάποιος περάσει αυτά τα όρια, θα πρέπει να περιμένει 5 με 15 λεπτά ώστε να μπορέσει να χρησιμοποιήσει το API ξανά.

5.3. Tweepy

Το Tweepy είναι ένα ανοιχτό προς όλους πακέτο της Python το οποίο σου προσφέρει ένα τρόπο για να επιτευχθεί η πρόσβαση στο API του Twitter με την Python. Το πακέτο αυτό περιέχει ένα σετ από κλάσεις και μεθόδους το οποίο παρουσιάζει οντότητες του Twitter όπως και τις διευθύνσεις του API,

επιπλέον χειρίζεται διάφορες τεχνικές λεπτομέρειες όπως:

- Την κωδικοποίηση και την αποκωδικοποίηση των δεδομένων
- HTTP ενέργειες
- Την ομαδοποίηση των δεδομένων
- Την OAuth αυθεντικοποίηση
- Τα όρια χρήσης

Η χρήση του Tweepy βοηθάει στο γεγονός ότι αντιμετωπίζει κάθε "χαμηλού επιπέδου" λεπτομέρεια, όπως τις HTTP ενέργειες, την σειριοποίηση των δεδομένων την αυθεντικοποίηση και τα χρονικά όρια. Όλα αυτά μπορεί να καταναλώσουν πολύ χρόνο και να δημιουργήσουν πόλλα λάθη. Σχεδόν όλη η λειτουργικότητα επιτυγχάνεται μέσω του Tweepy.

Μερικές εντολές του Tweepy στην Python που χρησιμοποιήθηκαν στο Σύστημα Διάδοσης ειδοποιήσεων και αντίστοιχα χωρισμένες σε κατηγορίες είναι:

OAuth αυθεντικοποίηση

```
auth = tweepy.OAuthHandler(tw_consumer_key, tw_consumer_secret)
auth.set_access_token(tw_access_token, tw_access_token_secret)
api = tweepy.API(auth)
```

Εξαγωγή όλων των ανακοινώσεων που υπάρχει στον Twitter λογαριασμό (screen_name) abarmpas97

```
twitter_timeline = api.user_timeline(screen_name='abarmpas97')
```

Ενημέρωση ή δημιουργία ανακοίνωσης στον λογαριασμό που έχει οριστεί για εκπρόσωπος της Twitter εφαρμογής

```
api.update_status(fb_post['message'])
```

Εντοπισμός λαθών μέσω της βιβλιοθήκης TweepError

```
except tweepy.TweepError as error:
    if error.api_code == 187:
        print(" ")
```

5.4. Δημιουργία Twitter Εφαρμογής

Το Twitter API όπως περιγράφεται και παραπάνω, για κάθε διεύθυνση του κάνει χρήση του OAuth πρωτοκόλλου, οπότε χρειάζεται η δημιουργία κωδικών αυθεντικοποίησης ώστε να γίνει χρήση του. Κάποια απ τα απαραίτητα πεδία είναι:

1. Consumer key
2. Consumer secret
3. Access Token

4. Access secret

Όπως και στο Facebook, έτσι και το Twitter προσφέρει μια ιστοσελίδα για την ανάπτυξη λογισμικού και γενικά για προγραμματιστές που είναι υπεύθυνα για την δημιουργία της εφαρμογής και όλων των απαραίτητων πεδίων για αυθεντικοποίηση. Κατά την δημιουργία της εφαρμογής ο χρήστης θα πρέπει να δώσει κάποιες πληροφορίες για την εφαρμογή όπως το όνομα, την περιγραφή, την ιστοσελίδα που θα γίνει χρήση του API και ο τρόπος που θα χρησιμοποιηθεί όπου στην προκειμένη περίπτωση το όνομα είναι Σύστημα Διάδοσης Ειδοποιήσεων και ο τρόπος χρήσης είναι η μετάδοση ανακοινώσεων στο Twitter μέσω αυτοματοποιημένης διαδικασίας. Στη συνέχεια για να δημιουργηθούν τα απαραίτητα πεδία αυθεντικοποίησης γίνεται η χρήση ενός κοινόχρηστου κώδικα στην Python και μέσω βιβλιοθήκης του Tweepy:

```
import tweepy

# Authenticate to Twitter
auth = tweepy.OAuthHandler("Consumer key", "Consumer secret")
auth.set_access_token("Access Token", "Access secret")

api = tweepy.API(auth)
```

Κώδικας 11: Εντολές αυθεντικοποίησης της Twitter εφαρμογής

5.5. Επίλογος

Όπως παρουσιάστηκε στην εργασία αυτή αλλά και όπως χρησιμοποιήθηκε στον προγραμματισμό του Συστήματος, το Twitter πρόσφερε ένα πολύ αποτελεσματικό εργαλείο ώστε να επιτευχθούν όλοι οι στόχοι της εργασίας. Το Tweepy διαχειρίζεται με ευκολία το API του Twitter χωρίς να χρειάζεται καμία περαιτέρω ενέργεια όσον αφορά τις "χαμηλού επιπέδου" λεπτομέρειες και πηγαίνει την λειτουργικότητα στο επόμενο επίπεδο. Όσον αφορά το όνομα του Twitter σαν μέσο κοινωνικής δικτύωσης αποτελεί επίσης μαζί με το Facebook ένα απ τα μεγαλύτερο και πιο διαδεδομένα μέσα, οπότε εξυπηρετεί σε μεγάλο βαθμό και παίζει σημαντικό ρόλο στον στόχο που έχει το Σύστημα Διάδοσης Ειδοποιήσεων.

6. GMAIL API

6.1. Εισαγωγή

Απ τις πιο δύσκολες ερωτήσεις στην ανάπτυξη ενός συστήματος και γενικά ενός προϊόντος, είναι το πως αυτό το προϊόν θα προσελκύσει ακόμη περισσότερους χρήστες. Όπως έχουμε δει, τα καλά νέα είναι ότι οι μεγάλες επιχειρήσεις που απασχολούν ένα μεγάλο εύρος των χρηστών στην αγορά της τεχνολογίας όπως το Facebook, Twitter, Google αφήνουν τους developers να δημιουργούν εφαρμογές τα οποία χρησιμοποιούν δεδομένα χρηστών απ τα ελεύθερα προσβάσιμα APIs τους. Όλο το σύστημα που παρουσιάζεται εδώ αποτελεί ένα τέτοιο παράδειγμα χρήσης τέτοιων APIs.

6.2. Περιγραφή GMAIL API

Ένα απο τα πιο πρόσφατα και αρκετά υποτιμημένο κοινόχρηστο API είναι το GMAIL API, το οποίο προσφέρει μια πρόσβαση τύπου RESTful στα e-mail χρηστών, στη διαμόρφωση της αλληλογραφίας του χρήστη, στις ετικέτες των μηνυμάτων και όπως επίσης απ τις πιο βασικές λειτουργίες ή αποστολή ή αποθήκευση μηνυμάτων εκ μέρος των χρηστών. Σε συνδυασμό με το Σύστημα Ειδοποιήσεων, το GMAIL API υλοποιεί τον τρόπο που θα λαμβάνουν οι χρήστες, οι οποίοι έχουν δηλώσει το email τους, οποιαδήποτε νέα ανακοίνωση υπάρχει απευθείας στην ηλεκτρονική αλληλογραφία τους.

Παρόλα αυτά υπάρχουν και κάποιες το ίδιο ενδιαφέρουσες λειτουργίες που θα μπορούσε να υλοποιήσει κάποιος, όπως:

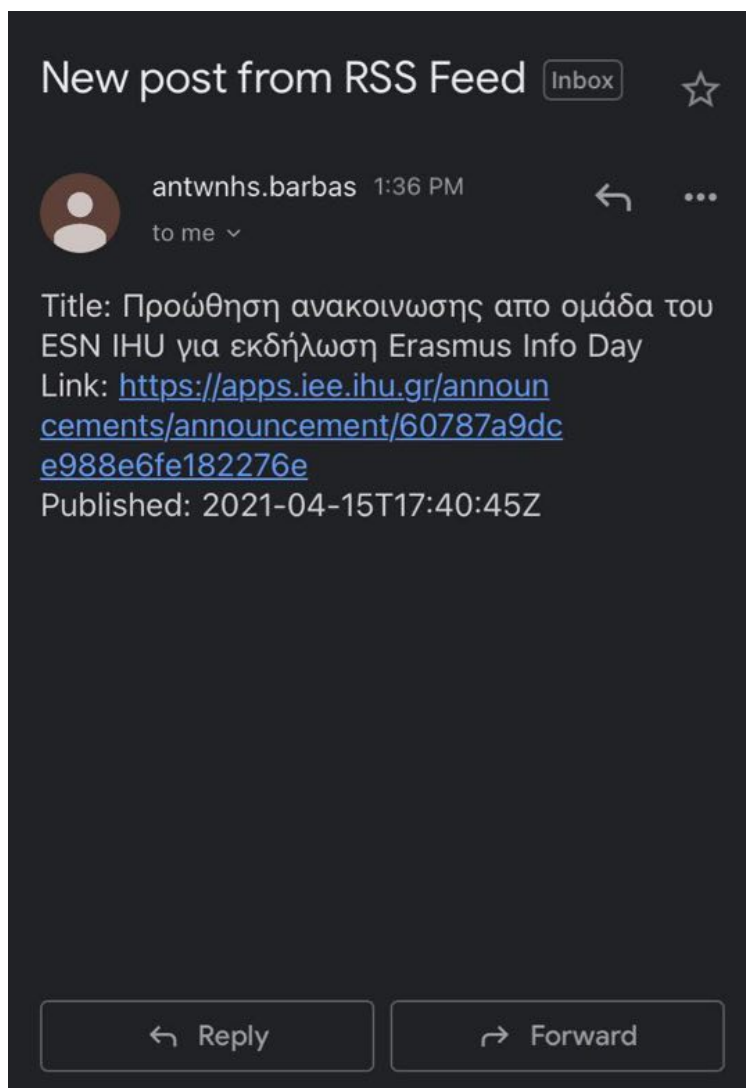
- η δημιουργία γραφικών στην διεπαφή χρήστη
- η βοήθεια να μειωθεί ο φόρτος των μη σημαντικών μηνυμάτων
- η παρουσίαση αναλυτικών δεδομένων για την αλληλεπίδραση ανάμεσα σε χρήστες
- η εξόρυξη σημαντικών πληροφοριών
- η δημιουργία υπενθυμίσεων

Επίσης πριν γίνει οποιαδήποτε πρόσβαση στο API, όπως είναι γνωστό χρειάζεται αυθεντικοποίηση του λογαριασμού και της εφαρμογής ώστε να δικαιολογείται και η χρήση που θα γίνει. Όλα τα APIs της Google χρησιμοποιούν το OAuth 2.0 πρωτόκολλο με το οποίο η εφαρμογή του χρήστη διαπραγματεύεται όλες τις ενέργειες με την πλατφόρμα της Google ώστε να λάβει ένα access_token. Το token αυτό προσφέρει περιορισμένα δικαιώματα στην χρήση των πόρων, αναλόγως με τις άδειες που έχουν οριστεί κατά τη διάρκεια της αυθεντικοποίησης.

6.3. Χρήση GMAIL API

Η ενσωμάτωση του GMAIL API στο Σύστημα Διάδοσης ειδοποιήσεων διαφέρει λίγο στον τρόπο που γίνεται η αυθεντικοποίηση των στοιχείων της εφαρμογής, προσφέροντας ένα συγκεκριμένο αρχείο που θα πρέπει να συμπεριλαμβάνεται μέσα στον φάκελο του προγράμματος και με τις σωστές διαδρομές αρχείων να μπορεί να διαβαστεί ώστε να μπορέσει να δώσει πρόσβαση στις μεθόδους του API. Επίσης στην Python όπως και στις περισσότερες γλώσσες προγραμματισμού θα πρέπει να εγκατασταθούν οι απαραίτητες βιβλιοθήκες.

Με την ολοκλήρωση της εγκατάστασης, υπάρχουν μερικές υπηρεσίες που χρησιμοποιήθηκαν πιο πολύ στο Σύστημα Διάδοσης Ειδοποιήσεων όπως και κάποιες που θα μπορούσαν να χρησιμοποιηθούν:



Εικόνα 3: Παραλαβή ανακοίνωσης από RSS μέσω GMAIL

ΜΗΝΥΜΑΤΑ: είναι οντότητες οι οποίες δημιουργούν το ηλεκτρονικό ταχυδρομείο, με το GMAIL API μπορείς να διαγράψεις και να δημιουργήσεις μηνύματα και να βάλεις ετικέτες. Τα μηνύματα μπορούν να ανήκουν σε 2 κατηγορίες:

DRAFT: Ένα μήνυμα το οποίο δεν έχει σταλθεί παίρνει αυτή την ετικέτα. Όταν μπορέσει να σταλθεί τότε αυτόματα αντικαθιστάται ετικέτα του σε SEND.

THREAD: Μια συλλογή απο μηνύματα τα οποία παρουσιάζουν την εξέλιξη μιας συνομιλίας. Το GMAIL αυτόματα δημιουργεί Threads όταν 2 χρήστες ανταλλάσουν μηνύματα. Με το GMAIL API μπορεί κάποιος να διαγράψει Threads ή να προσθέσει μηνύματα μέσα σε ήδη υπάρχοντα.

ΕΤΙΚΕΤΕΣ: χρησιμοποιούνται για να κατηγοριοποιήσουν και να οργανώσουν τα μηνύματα και τα threads σε έναν GMAIL λογαριασμό. Αντιθέτως με την οργάνωση που προσφέρουν οι φάκελοι, πολλές ετικέτες μπορούν να έχουν σχέση με πολλά μηνύματα. Ονοματικά, μερικά μηνύματα μπορεί να έχουν την ετικέτα INBOX, SPAM, DRAFT, STARRED, IMPORTANT αλλά επίσης μπορεί να υπάρχουν ετικέτες που δημιουργήθηκαν απ τον χρήστη.

ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΑΠΟΣΤΟΛΗ ΜΗΝΥΜΑΤΩΝ: Πριν την αποστολή των μηνυμάτων, γίνεται δημιουργία μια προσωρινή κατηγοριοποίηση του μηνύματος με την ετικέτα DRAFT, η ετικέτα αυτή προσφέρει μια μοναδική διεύθυνση για κάθε ένα μήνυμα που την κατέχει. Όσον αφορά τα μηνύματα με την συγκεκριμένη ετικέτα, έχουν όλες τις ιδιότητες όπως τα υπόλοιπα μηνύματα αλλά δεν μπορούν να πάρουν οποιαδήποτε άλλη ετικέτα. Όταν τα DRAFT μηνύματα στέλνονται, το GMAIL αυτόματα αντικαθιστά την ετικέτα DRAFT με την ετικέτα SENT.

Το Σύστημα Διάδοσης ειδοποιήσεων περιέχει δύο μεθόδους (create_message(), send_message()) οι οποίες λειτουργούν με συγκεκριμένα βήματα για την δημιουργία και την αποστολή των μηνυμάτων του GMAIL:

1. Δημιουργία ενός MIME μηνύματος με την μέθοδο MIMEText

```
message = MIMEText(message_text)
```

2. Κωδικοποίηση του μηνύματος σε αναγνώσιμη μορφή

```
raw_message = base64.urlsafe_b64encode(message.as_string().encode("utf-8"))  
return {  
    'raw': raw_message.decode("utf-8")  
}
```

3. Ορισμός του αποστολέα, του παραλήπτη και του θέματος

```
message['to'] = to  
message['from'] = sender  
message['subject'] = subject
```

4. Αποστολή του μηνύματος

```
message = service.users().messages().send(userId=user_id,  
body=message).execute()
```

6.4. API vs SMTP

Ένα ενδιαφέρον θέμα που προκύπτει από την προσπάθεια δημιουργίας μιας αυτοματοποιημένης διαδικασίας για την αποστολή μηνυμάτων μέσω του GMAIL, αναφέρεται στις διαφορές ανάμεσα στα γνωστά πρωτόκολλα αποστολής μηνυμάτων και την χρήση του API. Αρχικά το SMTP πρωτόκολλο είναι ένα σύνολο κανόνων για την αποστολή μηνυμάτων από κάποιον αποστολέα σε ένα server ή ανάμεσα σε servers. Οι περισσότερες υπηρεσίες ηλεκτρονικού ταχυδρομείου χρησιμοποιούν SMTP για την αποστολή και POP3/IMAP4 για την λήψη.

Από την άλλη πλευρά το API είναι μια δίοδος αλληλεπίδρασης που χρησιμοποιείται από εφαρμογές, πλατφόρμες και σύστημα όπως έχει αναφερθεί. Συγκεκριμένα με το GMAIL API μπορεί κάποιος να στείλει μηνύματα χρησιμοποιώντας μόνο πληροφορίες του HTTP πρωτοκόλλου, ένα σύνολο κανόνων δηλαδή που καθορίζει το πώς είναι δομημένα τα μηνύματα και πως μεταφέρονται.

Και οι 2 επιλογές έχουν τα θετικά και τα αρνητικά τους, αναφέρονται σε διαφορετικές τεχνολογίες και αλλάζει ο τρόπος που διαχειρίζονται τα μηνύματα. Το SMTP είναι πολύ διαδεδομένο και εύκολο στην εγκατάσταση και χρήση. Επίσης δεν χρειάζεται κάποιο δεξιότητα στον προγραμματισμό και στην δημιουργία κώδικα. Από την άλλη πλευρά το API, προσφέρει προγραμματιστικά μια ποικιλία λειτουργιών για την εφαρμογή, όπως στο συγκεκριμένο σύστημα που το χρησιμοποιεί για μια

αυτόματη διαδικασία. Και επίσης το API προσφέρει μια επιπλέον ασφάλεια, το οποίο είναι σημαντικό όταν υπάρχουν ευαίσθητα δεδομένα.

6.5. Επίλογος

Χρησιμοποιώντας το GMAIL, αμέσως αναπληρώνονται οποιαδήποτε κενά μπορεί να αφήσουν τα μέσα κοινωνικής δικτύωσης, καθώς τα ηλεκτρονικά μηνύματα και γενικά το ηλεκτρονικό ταχυδρομείο αποτελεί τον πιο σημαντικό τρόπο στην επικοινωνία μεταξύ των ανθρώπων. Το ηλεκτρονικό μήνυμα (e-mail) είναι ευρέως γνωστό γιατί διαφέρει στον τρόπο που κάποιος τον χρησιμοποιεί, δηλαδή από την επικοινωνία αυτή καθ' αυτή, πολλοί χρησιμοποιούν το ηλεκτρονικό μήνυμα για να έρθουν σε επαφή με δουλειές, παρουσιάζουν διαφορετικά τον εαυτό τους και μπορούν να οργανώσουν τις υποχρεώσεις τους. Στο Σύστημα Διάδοσης ειδοποιήσεων τα ηλεκτρονικά μηνύματα είναι σημαντικά γιατί, αναφέρονται σε ανακοινώσεις και σε εκπαιδευτικό υλικό και αυτομάτως καλύπτουν το ποσοστό των χρηστών που δεν χρησιμοποιούν κάποιο μέσο κοινωνικής δικτύωσης αλλά σίγουρα έχουν στην κατοχή τους μια ηλεκτρονική διεύθυνση.

7. Γενικά APIs

7.1. Εισαγωγή

Το Σύστημα Διάδοσης Ειδοποιήσεων έχει δημιουργηθεί ως μια υπηρεσία διάδοσης ειδοποιήσεων εκπαιδευτικού ή μη περιεχομένου, και προσπαθεί να καλύψει όσο γίνεται μεγαλύτερο εύρος πομπού και δέκτη. Δηλαδή σε ένα τέτοιο σύστημα δεν γινόταν να μην υπάρχει μια ιστοσελίδα που θα αναφέρεται κυριώς σε εκπαιδευτικές ανακοινώσεις, όπως είναι το `aboard.iee.ihu`, όσον αφορά τον πομπό και ενδεχομένως αποστολέα ανακοινώσεων. Απο την άλλη πλευρά, όσον αφορά τον δέκτη, γίνεται μια προσπάθεια για την αποστολή ειδοποιήσεων και όχι την ανακοίνωση αυτή καθ'αυτή με την μορφή μηνύματος στο κινητό (SMS).

7.2. Aboard.iee.ihu

Το `aboard` αποτελεί μια ιστοσελίδα της σχολής των Μηχανικών πληροφορικής και Ηλεκτρονικών συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, η οποία λειτουργεί σαν πίνακας ανακοινώσεων σχετικά με θέματα της σχολής. Η προσέγγιση που γίνεται για το συγκεκριμένο API είναι πιο απλή απ ότι τα υπόλοιπα, καθώς δεν υπάρχει προς το παρόν κάποια αυθεντικοποίηση που πρέπει να γίνει ώστε να επιτευχθεί η πρόσβαση και επίσης με την χρήση της `requests` βιβλιοθήκης της Python και της GET μεθόδου, μπορεί κάποιος να λάβει τα επιθυμητά δεδομένα απο τον σύνδεσμο που έχει ορισθεί απο το `aboard` API.

Τέλος γίνεται μια επεξεργασία των κεφαλίδων που επιστρέφονται σε μορφή JSON και έπειτα γίνεται κανονικά η μετάδοση.

```
aboard_search_tag = requests.get(search_tag)
aboard_data = aboard_search_tag.json()
```

7.2.1. Παράδειγμα χρήσης `aboard`

Το `aboard` API προσφέρει υπηρεσίες για την λήψη ανακοινώσεων της σχολής, οι οποίες ομαδοποιούνται με διαφορετικές ετικέτες και υπάρχουν διαφορετικοί τρόποι που το σύστημα μπορεί να έχει πρόσβαση στα δεδομένα, για παράδειγμα:

MEMONOMENH ANAKOINΩΣH

```
/api/announcements/{id}
```

Όπου επιστρέφονται όλες οι πληροφορίες για μια συγκεκριμένη ανακοίνωση με το συγκεκριμένο `{id}`, όπως ο συγγραφέας, η ημερομηνία δημιουργίας αλλά και σε ποια κατηγορία ανακοινώσεων ανήκει.

ΟΛΕΣ ΟΙ ΑΝΑΚΟΙΝΩΣΕΙΣ`/api/announcements`

Επιστρέφονται όλες οι ανακοινώσεις που υπάρχουν στην ιστοσελίδα με όλα τα δεδομένα που αναφέρθηκαν παραπάνω.

ΟΛΕΣ ΟΙ ΕΤΙΚΕΤΕΣ`/api/tags`

Επιστρέφονται όλες οι κατηγορίες με τις οποίες διαχωρίζονται οι ανακοινώσεις. Οι ετικέτες αποτελούν ένα πολύ σημαντικό κομμάτι για λειτουργία του Συστήματος γιατί το σενάριο είναι ότι κάθε ετικέτα έχει έναν αντίστοιχο παραλήπτη, επομένως κάθε ομάδα ανακοινώσεων θα στέλνεται σε διαφορετική ηλεκτρονική διεύθυνση.

```

{
    "id": 2,
    "title": "Εξάμηνο Α",
    "parent_id": 1,
    "is_public": 0
}

```

Κώδικας 12: Απόκριση αίτησης στο API του aboard

ΟΛΕΣ ΟΙ ΑΝΑΚΟΙΝΩΣΕΙΣ ΓΙΑ ΜΙΑ ΕΤΙΚΕΤΑ`/api/search/tag/{id}`

Η υπηρεσία που χρησιμοποιεί το Σύστημα Διάδοσης Ειδοποιήσεων, επιστρέφει όλες τις ανακοινώσεις που ανήκουν σε μια ετικέτα. Για κάθε μια ανακοίνωση το σύστημα ελέγχει την ημερομηνία δημιουργίας και εφόσον περάσει τους ελέγχους, ακολουθεί την διαδικασία μετάδοσης προς όλους τους διακομιστές με την ιδιαιτερότητα ότι θα πρέπει να δημιουργήσουμε από τα πεδία JSON ένα συνολικό μήνυμα.

```

aboard_message = "{} \n \n {} \n \n {}".format(aboard_post["title"],
aboard_post["body"], aboard_post["author"]["name"])

```

7.3. Προσθήκη APIs

Ένα βασικό στοιχείο στο Σύστημα Διάδοσης Ειδοποιήσεων είναι η επαναχρησιμοποίηση των μεθόδων όσον αφορά την μετάδοση της πληροφορίας. Δηλαδή υπάρχουν πολύ μικρές διαφορές στο πως θα δημιουργηθεί το μήνυμα και καμία διαφορά στο τρόπο που θα σταλθεί και με ποιες μεθόδους. Για παράδειγμα τα βήματα που θα μπορούσε να κάνει κάποιος για να προσθέσει ένα API στο σύστημα, που είτε θα λειτουργεί σαν πομπός είτε σαν δέκτης είτε και τα δύο όπως το Facebook και το Twitter, που δέχονται και στέλνουν ανακοινώσεις ταυτόχρονα, είναι:

ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ

Πρακτικά για να μπορέσει κάποιος να χρησιμοποιήσει τις υπηρεσίες ενός εξωτερικού διακομιστή, θα πρέπει να δημιουργήσει μια εφαρμογή στην δικιά του πλευρά ώστε να χειρίζεται όλες τις αιτήσεις και γενικά να μπορεί να αλληλεπιδράσει με το API το οποίο πρόκειται να χρησιμοποιήσει.

ΔΗΜΙΟΥΡΓΙΑ SERVER ΠΕΔΙΟΥ

Στο αρχείο servers.json όπως για όλους τους διακομιστές, υπάρχουν τα στοιχεία για οποιαδήποτε αυθεντικοποίηση ζητηθεί στο σύστημα, ώστε να μπορέσει να ανταποκριθεί.

Για παράδειγμα, έστω ότι το LinkedIn προσφέρει ένα κλειδί αυθεντικοποίησης για το API κάθε φορά που δέχεται μια αίτηση. Για να μπορέσει το σύστημα να κάνει χρήση των υπηρεσιών θα πρέπει να δημιουργηθεί ένα JSON πεδίο:

```
linked_in_auth_key: "31232331"
```

Σε περίπτωση που ο διακομιστής λειτουργεί και σαν πομπός και σαν δέκτης, θα πρέπει να οριστεί και ο σύνδεσμος της σελίδας του διακομιστή, ώστε να μπορούν να ληφθούν οι πληροφορίες που βλέπει ο τελικός χρήστης.

```
linked_in_url: "www.linkedin.com"
```

ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΕΔΙΩΝ

Στη συνέχεια πρέπει να γίνει η αρχικοποίηση των σταθερών που θα χρησιμοποιήσει το Σύστημα στις μεθόδους. Στο συγκεκριμένο παράδειγμα η σταθερά είναι η ιστοσελίδα και το κλειδί του διακομιστή.

ΔΗΜΙΟΥΡΓΙΑ ΜΕΘΟΔΟΥ ΣΥΓΚΡΙΣΗΣ ΗΜΕΡΟΜΗΝΙΩΝ

Εφόσον γίνει η λήψη των ανακοινώσεων που είτε πρόκειται να μεταδοθούν είτε όχι, πρέπει να δημιουργηθεί η μέθοδος που βασίζεται στις ημερομηνίες των ανακοινώσεων ώστε να γίνει κατανοητό απ το σύστημα αν πρόκειται να γίνει μετάδοση.

ΔΗΜΙΟΥΡΓΙΑ ΜΕΘΟΔΟΥ ΜΕΤΑΔΟΣΗΣ

Εν τέλει, γίνεται η δημιουργία της μεθόδου μετάδοσης, όπου παίρνει όλες τις υπηρεσίες που προσφέρει το API που πρόκειται να προστεθεί και δημιουργεί ένα σενάριο μετάδοσης της ανακοίνωσης. Το API αυτό με κάποιο τρόπο θα προσφέρει τα δεδομένα, τις ανακοινώσεις και γενικά την ροή που υπάρχει σε αυτό και με αντίστοιχες βιβλιοθήκες της Python 3 θα μπορεί κάποιος να τα χρησιμοποιήσει και έπειτα όπως είναι ορισμένες και οι λειτουργίες για κάθε διακομιστή όπως το Facebook και το Twitter θα μεταδοθούν οι επιθυμητές ανακοινώσεις. Από την άλλη πλευρά θα υπάρχουν μέθοδοι που ίσως προσφέρονται απ το API, το οποίο θα λειτουργεί σαν δέκτης αυτή τη φορά, ώστε να μπορεί το σύστημα να της χρησιμοποιεί για να στέλνει τα επιθυμητά δεδομένα.

7.4.SMS API

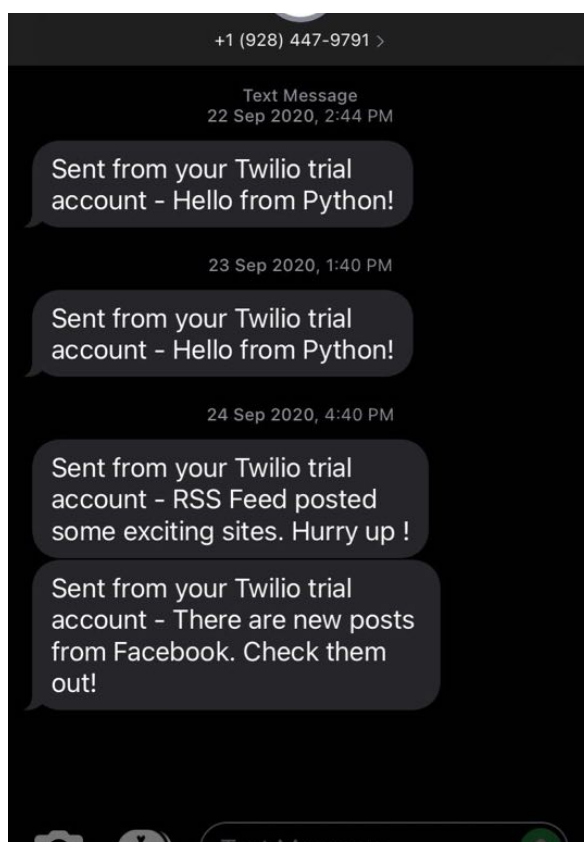
Η λογική της χρήσης ενός API για μηνύματα στο κινητό, είναι ότι θα υπάρχει ένας αριθμός από την υπηρεσία και θα λειτουργεί όπως όλα τα κινητά τηλέφωνα και με μερικές εντολές θα μπορεί κάποιος να ενσωματώσει την διαδικασία αποστολής στο πρόγραμμά του. Στο Σύστημα Διάδοσης Ειδοποιήσεων έγινε χρήση του Twilio API το οποίο προσφέρει μια δωρεάν συνδρομή και έναν αριθμό τηλεφώνου με δωρεάν μονάδες ώστε να μπορέσουν να γίνουν μερικές δοκιμές. Παρόλα αυτά η συγκεκριμένη υπηρεσία δεν μπορεί να συνεχιστεί, καθώς είναι υπό πληρωμή όπως και τα περισσότερα API για αριθμούς κινητής τηλεφωνίας.

Παρακάτω παρουσιάζονται οι εντολές του συστήματος που χρησιμοποιήθηκαν σύμφωνα με το Twilio API ώστε να σταλθεί ειδοποίηση σε κινητό τηλέφωνο όπως και το αποτέλεσμα αυτού:

μέθοδος `send_sms(server_in)`

```
if server_in == "Facebook":
    client.messages.create(to=to_number,
                          from_=from_number,
                          body="There are new posts from Facebook. Check them out!")
```

Κώδικας 13: Αποστολή μηνύματος στο κινητό



Εικόνα 4: Λήψη ειδοποιήσεων μέσω SMS

8. Συμπέρασμα και Μελλοντικές κατευθύνσεις

Στόχος αυτής της εργασίας ήταν η ανάπτυξη ενός συστήματος αυτόματης διάδοσης ειδοποιήσεων και ανακοινώσεων σε διαφορετικά περιβάλλοντα και μέσα κοινωνικής δικτύωσης. Το αποτέλεσμα ήταν το Σύστημα Διάδοσης Ειδοποιήσεων (Notification_System), όπου αποτελείται από το πρόγραμμα το οποίο εκτελείται σε γραμμή εντολών και από ένα αρχείο που κατευθύνει το πρόγραμμα στο πώς θα λειτουργήσει και με ποιές πλατφόρμες θα αλληλεπιδράσει. Η λειτουργικότητα του συστήματος και το πως συμπεριφέρεται έχει δοκιμαστεί με την ανάπτυξη της ιδέας ενός εκπαιδευτικού οργανισμού να μπορεί να στέλνει ανακοινώσεις εκτός του "χώρου" της αυτόματα.

Η φύση του συστήματος είναι να προσφέρει ένα πεδίο εφαρμογών για περαιτέρω εξελίξεις. Στο Σύστημα Διάδοσης Ειδοποιήσεων, θα μπορούσε να γίνει μια ακόμα πιο ελεγχόμενη ανίχνευση λαθών και μια πιο εκτενής έρευνα στον τομέα αυτό όσον αφορά το τι προσφέρει το κάθε API. Επίσης, μέτρα ασφαλείας μπορούν να ενσωματωθούν στην εφαρμογή, όχι τόσο στο επίπεδο των APIs όσο στο επίπεδο του προγράμματος, καθώς υπάρχουν σημαντικές πληροφορίες δυναμικά καταχωρημένες χωρίς κάποια απόκρυψη. Η λειτουργικότητα του Συστήματος Διάδοσης Ειδοποιήσεων θα μπορούσε να ελεγχθεί σε ένα μεγαλύτερο δίκτυο, χρησιμοποιώντας περισσότερα APIs και να εκπληρώνει ένα μεγαλύτερο φάσμα σκοπών και στόχων. Για το πρόγραμμα αυτό, χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python 3 καθώς και μια πληθώρα JSON αρχείων και δεδομένων. Σύμφωνα με την καθοδήγηση των APIs προσφέρει κάθε πλατφόρμα και με σχετικές βιβλιοθήκες της Python, έγινε η σύνδεση, η αλληλεπίδραση και η αποστολή όλων των δεδομένων στα επιθυμητά περιβάλλοντα. Τα JSON αρχεία βοήθησαν στην οργάνωση και την αρχικοποίηση του συστήματος, ορίζοντας όλες τις απαραίτητες πληροφορίες και δημιουργώντας βάσεις δεδομένων για την αποθήκευση των ανακοινώσεων. Ωστόσο, καθώς τα δεδομένα αυξάνονται και η πολυπλοκότητα υπάρχει στην φύση των APIs, μια ιδανική κατάσταση θα ήταν να υπήρχαν προεγκατεστημένες βάσεις δεδομένων με αντίστοιχες γλώσσες προγραμματισμού όπως και μια πιο φιλική παραμετροποίηση του προγράμματος από τον χρήστη.

Εν τέλει, με την εξάπλωση του ίντερνετ, των μέσων κοινωνικής δικτύωσης και των σχετικών τεχνολογιών, το Σύστημα Διάδοσης Ειδοποιήσεων θα μπορούσε να σταθεί επάξια σε αυτό τον χώρο. Η αυτοματοποιημένη διαδικασία μετάδοσης ανακοινώσεων καθώς και η μετάδοση δουλειάς καθώς και η αυτόματη επικοινωνία με πελάτες επίσης θα μπορούσαν να υπάρξουν σαν υπηρεσία του συστήματος. Η επικοινωνία όλων των υλικών μαζί μπορεί να αποδειχθεί δύσκολη παρουσιάζοντας περιορισμούς, προβλήματα και αποτυχίες. Παρόλα αυτά οι προσπάθειες που γίνονται ή θα γίνουν προς την κατεύθυνση αυτή, θα βοηθήσουν να γίνει κατανοητή όλη η ιδέα της αυτοματοποίησης, από ποιόν και που θα είναι απαραίτητη.

BIBΛIOΓPAΦIA

Application Note

- [1] Tutorials Point, Python - Reading RSS Feed. [Online]. Available: https://www.tutorialspoint.com/python_text_processing/python_reading_rss_feed.htm, 18/09/2021.
- [2] Python 3.9.7, Documentation, Time access and conversions. [Online]. Available: <https://docs.python.org/3/library/time.html>, 20/08/2021.
- [3] HTTPS, Specification, rfc2818. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc2818>, 18/08/2021.
- [4] RSS feed, Specification , Wikipedia. [Online]. Available: <https://en.wikipedia.org/wiki/RSS>, 19/09/2021.
- [5] Miguel Garcia, “How to Make a Twitter Bot in Python With Tweepy”, Real Python, 2019.
- [6] Shivam Dutt Sharma, “Facebook Graph API | Python”, Medium, 2020.
- [7] Strftime, “Python strftime cheatsheet”, 2021.
- [8] George Edwards, “Introduction to the Gmail API”, Quandary Peak Research, 2014.
- [9] Ben Lloyd Pearson, “How to Integrate the Gmail API Into Your App”, Nylas, 2020.
- [10] Dmitriy Shcherbakan, “Send Emails with Gmail API”, Mailtrap, 2020.
- [11] Twitter, “About Twitter’s APIs”, Platform Use Guidelines, 2021. [Online]. Available: <https://help.twitter.com/en/rules-and-policies/twitter-api>.
- [12] Dataquest, “Python API Tutorial: Getting Started with APIs”, Python API, 2020. [Online]. Available: <https://www.dataquest.io/blog/python-api-tutorial/> .
- [13] IBM, “What is automation?”, Automation, 2020. [Online]. Available: <https://www.ibm.com/topics/automation>
- [14] AMAZON, “JSON for Databases”, 2018. [Online]. Available: <https://aws.amazon.com/documentdb/what-is-json/>
- [15] Gabriela Grosseck and Carmen Holotescu, “Can we use Twitter for educational activities”, 4th international scientific conference, eLearning and software for education, Bucharest, Romania, 2008.
- [16] Eiji Aramaki, Sachiko Maskawa, Mizuki Morita, “Twitter Catches The Flu: Detecting Influenza Epidemics using Twitter”, Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK. 2011.
- [17] Frank Müller, Frank Thiesing, “Social networking APIs for companies”, University of Applied Sciences, International Conference on Computational Aspects of Social Networks, 2011.
- [18] R.Piyare, M.Tazil, “BLUETOOTH BASED HOME AUTOMATION SYSTEM USING CELL PHONE”, Department of Electrical & Electronics Engineering Fiji National University, IEEE 15th International Symposium, 2011.
- [19] Tim Bray, “(XML) 1.0 W3C Recommendation”, University of Illinois at Chicago, Jean Paoli (Microsoft), 1998

- [20] rhyssharp, “Defining automation, bots & artificial intelligence”, rackspace, 2019.
- [21] Assenmacher D, Clever L, Frischlich L, Quandt T, Trautmann H, Grimme C. “Demystifying Social Bots: On the Intelligence of Automated Social Media Actors.” *Social Media + Society*, July 2020, doi:10.1177/2056305120939264.
- [22] Sriskanthan, Nadarajah & Tan, F. & Karande, A.. (2002). Bluetooth based home automation system. *Microprocessors and Microsystems*. 26. 281-289. 10.1016/S0141-9331(02)00039-X.
- [23] Johnson JL, tom Wörden H, van Wijk K. “PLACE: An Open-Source Python Package for Laboratory Automation, Control, and Experimentation.” *Journal of Laboratory Automation*, vol. 20, no. 1, Feb. 2015, pp. 10–16, doi:10.1177/2211068214553022.