



**ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΘΕΜΑ: «Ανάπτυξη εργαλείου αυτόματης ενημέρωσης λογαριασμού στο
Facebook»**

Γκέγκα Ευρώπη(08/3329)

egkegka@it.teithe.gr

Κωστοπούλου Ειρήνη(08/3357)

eirkost@it.teithe.gr

Επιβλέπων καθηγητής: Ηλιούδης Χρήστος

iliou@it.teithe.gr

Ιούλιος 2014
Θεσσαλονίκη

ΠΡΟΛΟΓΟΣ

Στόχος της παρούσας πτυχιακής εργασίας αποτελεί η αυτόματη ενημέρωση των χρηστών από μία σελίδα στο κοινωνικό δίκτυο Facebook. Αυτό σαν πρόβλημα εγείρει πολλές σκέψεις για το πως μπορεί να υλοποιηθεί μια τέτοια υπηρεσία. Η μεγαλύτερη δυσκολία έγκειται στο γεγονός ότι πρέπει να ανακτήσουμε την επιθυμητή πληροφορία από μία δεξαμενή πληροφορίας, η οποία δεν παρέχει κατάλληλα μοντελοποιημένο περιεχόμενο. Εφόσον αυτό γίνει εφικτό και το περιεχόμενο διαμορφωθεί κατάλληλα, θα ακολουθήσει η διοχέτευση της ζητούμενης πληροφορίας στο Facebook. Για να επιτευχθεί αυτός ο στόχος, θα μελετηθούν αρκετές τεχνολογίες, άλλες πιο απλές και άλλες πιο περίπλοκες.

ΠΕΡΙΛΗΨΗ

Στα πλαίσια αυτής της εργασίας μελετήθηκαν τρόποι μοντελοποίησης των δημόσιων ανακοινώσεων του τμήματος Μηχανικών Πληροφορικής, οι οποίες αντλούνται από την HTML σελίδα όπου ανακοινώνονται, καθώς και η διεπαφή της πλατφόρμας του Facebook. Σκοπός ήταν να αναρτώνται αυτόματα οι ανακοινώσεις σε μια σελίδα του κοινωνικού δικτύου Facebook με στόχο την ενημέρωση των χρηστών της. Έχοντας εξετάσει υφιστάμενες λύσεις και πως λειτουργούν, καθώς και το πως υλοποιούνται και λειτουργούν οι εφαρμογές που αναπτύσσονται στο Facebook (Facebook apps), προχωρήσαμε στην ανάπτυξη του δικού μας εργαλείου. Το εργαλείο μας, αφού μετατρέψει τα περιεχόμενα του ιστότοπου της ύδρας σε XML μορφή, και αφού απομονώσει τη ζητούμενη μόνο πληροφορία με την χρήση της τεχνολογίας XPath, χρησιμοποιεί την Facebook εφαρμογή *hydranewsfeed* για να δημοσιεύσει το περιεχόμενο. Το περιεχόμενο τελικά αναρτάται στην σελίδα “Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙΘ” η οποία είναι διαθέσιμη στον παρακάτω υπερσύνδεσμο: <https://www.facebook.com/thehydranews>.

ABSTRACT

In this thesis, we studied ways of modeling the public announcements of the Computer Engineering Department, drawn from the HTML page where are announced and the Facebook interface. The purpose was to automatically post announcements on a page of the Facebook social network in order to inform its users. Having studied existing solutions and how they work, and how Facebook apps are implemented and operated, we proceeded with the development of our tool. Our tool, after converting the contents of the site *hydra* in XML format and having isolated the desired information using technology XPath, it uses the Facebook application *hydranewsfeed* to publish the content. The content, finally, is posted on the page " Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙΘ" which is available on the hyperlink below: <https://www.facebook.com/thehydranews>.

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστούμε θερμά τον κ.Ηλιούδη, που μας έδωσε την ευκαιρία να ασχοληθούμε με ένα τόσο ενδιαφέρον θέμα και για την καθοδήγησή του.

Θα θέλαμε, επίσης, να ευχαριστήσουμε τις οικογένειες μας και τους φίλους μας για τη στήριξη τους, καθόλη τη διάρκεια της εκπόνησης αυτής της εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ	12
1.1	Όγκος πληροφορίας που διακινείται στο Facebook.....	12
1.2	Features για την ενημέρωση των χρηστών του Facebook.....	15
1.3	Σκοπός της εργασίας	16
1.4	Επιτεύγματα της εργασίας.....	17
1.5	Η καινοτομία στην υπηρεσία μας.....	17
1.6	Συνοπτική ανάλυση επόμενων κεφαλαίων	18
2	ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΑΣ	19
2.1	Ανάκτηση και ανάρτηση περιεχομένου	19
2.1.1	Τεχνολογία Pull.....	19
2.1.2	Τεχνολογία Push.....	21
2.1.3	Σύγκριση Τεχνολογιών Push και Pull	24
2.2	Τεχνολογίες επικοινωνίας σε πραγματικό χρόνο	25
2.2.1	AJAX Polling	27
2.2.2	AJAX Long-Polling	27
2.2.3	HTML5 Server Sent Events.....	28
2.2.4	HTML5 Websockets.....	30
2.2.5	Σύγκριση τεχνολογιών επικοινωνίας πραγματικού χρόνου	32
2.3	Εφαρμογές αυτόματης ανάρτησης	34
2.3.1	Mail	34
2.3.1.1	Μειονεκτήματα email newsletter	35
2.3.2	RSS Feed	35
2.3.2.1	Πλεονεκτήματα RSS.....	38
2.3.3	Atom	38
2.3.3.1	Σύγκριση Atom και RSS	39
3	ΜΕΛΕΤΗ, ΑΞΙΟΛΟΓΗΣΗ ΥΦΙΣΤΑΜΕΝΩΝ ΛΥΣΕΩΝ (ΥΠΗΡΕΣΙΩΝ) ΠΟΥ ΕΧΟΥΝ ΑΝΑΠΤΥΧΘΕΙ	41

3.1	Argyle Social	41
3.2	Buffer	42
3.3	HootSuite	44
3.4	Sprout Social.....	45
3.5	Graph API	46
3.5.1	Βασικά Στοιχεία.....	46
3.5.2	Ανάκτηση πληροφορίας	47
3.5.3	Δημοσίευση πληροφορίας.....	48
3.6	Αξιολόγηση υφιστάμενων λύσεων	48
4	ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΤΗΣ ΥΠΗΡΕΣΙΑΣ	50
4.1	Μεθοδολογικό πλαίσιο	50
4.2	Ανάλυση απαιτήσεων.....	51
4.2.1	Ανάκτηση και διαμόρφωση πληροφορίας.....	51
4.2.2	Δημοσίευση πληροφορίας στο μέσο κοινωνικής δικτύωσης	52
4.2.2.1	Facebook εφαρμογές.....	52
4.2.2.2	Δημιουργία της Facebook εφαρμογής.....	55
4.2.2.3	Αυθεντικοποίηση της εφαρμογής.....	58
4.2.2.4	Λειτουργικότητα του εργαλείου	62
4.2.3	Διαγράμματα UML και ροές της εφαρμογής	63
5	ΣΥΝΟΠΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΤΕΧΝΟΛΟΓΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΕ.....	66
5.1	Facebook	66
5.1.1	Το facebook είναι μια μονολιθική εφαρμογή	67
5.1.2	Μονολιθική Αρχιτεκτονική.....	68
5.1.2.1	Πλεονεκτήματα Μονολιθικών συστημάτων	68
5.1.2.2	Μειονεκτήματα Μονολιθικών συστημάτων	69
5.2	Τα API του Facebook	70
5.2.1	Open Graph πρωτόκολλο	70

5.2.1.1	Πως δουλεύει το Open Graph.....	71
5.2.2	Graph API.....	72
5.2.2.1	Μέθοδοι δημοσίευσης περιεχομένου	73
5.2.2.2	Μέθοδοι ανάκτησης περιεχομένου.....	74
5.2.3	Άλλα APIs	76
5.3	Facebook SDK για PHP	77
5.3.1	Η Βιβλιοθήκη SDK του Facebook για την PHP	77
5.4	Apache HTTP Server	78
5.5	XAMPP	79
5.6	XML	81
5.6.1	Πλεονεκτήματα XML	81
5.6.2	Ένα XML έγγραφο σαν δενδρική αναπαράσταση.....	83
5.7	Xpath	83
5.8	Μέθοδοι GET και POST του HTTP πρωτοκόλλου	84
5.8.1	Μέθοδος GET	85
5.8.2	Μέθοδος POST.....	85
5.9	Auth 2.0 protocol.....	86
5.9.1	Διαδικασία αυθεντικοποίησης χρηστών.....	86
5.9.2	Αυθεντικοποίηση στο Facebook.....	87
5.10	Τεχνολογίες για τη συγγραφή και τον προγραμματισμό της πτυχιακής εργασίας.....	89
6	ΑΝΑΠΤΥΞΗ-ΑΝΑΛΥΣΗ ΕΝΔΕΙΚΤΙΚΩΝ ΣΥΣΤΑΤΙΚΩΝ ΤΗΣ ΥΠΗΡΕΣΙΑΣ.....	90
6.1	Η κύρια κλάση index.php.....	90
6.2	Βιβλιοθήκη του Facebook.....	97
6.3	Output της υπηρεσίας μας.....	98
7	ΚΑΙΝΟΤΟΜΙΑ ΚΑΙ ΠΡΟΣΤΙΘΕΜΕΝΗ ΑΞΙΑ ΤΗΣ ΥΠΗΡΕΣΙΑΣ ΜΑΣ	100
8	ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	101
9	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	103

10	ΠΑΡΑΡΤΗΜΑ ΜΕ ΤΟΝ ΚΩΔΙΚΑ	107
10.1	Index.php	107
10.2	Βιβλιοθήκες Facebook	108

Ευρετήριο εικόνων

Εικόνα 1 Ποσότητα δεδομένων στο διαδίκτυο σε ένα λεπτό	13
Εικόνα 2 Διάγραμμα κοινωνικών δικτύων που επισκέπτονται οι φοιτητές.....	14
Εικόνα 3 Πρώτο το Facebook σε αναζήτηση θεμάτων που αφορούν την εκπαίδευση	15
Εικόνα 4 Απεικόνιση τεχνολογίας pull.....	20
Εικόνα 5 Απεικόνιση τεχνολογίας push	21
Εικόνα 6 Ένα απλό HTTP αίτημα	26
Εικόνα 7 Πως δουλεύει το Ajax Polling	27
Εικόνα 8 Πως δουλεύει το Ajax Long Polling	28
Εικόνα 9 Πως λειτουργούν τα Server Set Events (SSE)	29
Εικόνα 10 Περιγραφή λειτουργίας της τεχνολογίας WebSockets	31
Εικόνα 11 Διαφορές Ajax long polling, HTTP SSE και HTTP WebSocket.....	33
Εικόνα 12 Σύνδεση μεταξύ RSS feed, XML αρχείων και του προσωπικού μας υπολογιστή.....	37
Εικόνα 13 Αρχική σελίδα Argyle Social.....	41
Εικόνα 14 Παράθυρο διαλόγου Buffer	42
Εικόνα 15 Χρονοδιάγραμμα Buffer	43
Εικόνα 16 Παράθυρο διαλόγου HootSuite	44
Εικόνα 17 Παράθυρο διαλόγου Sprout Social.....	45
Εικόνα 18 Βήματα ανάκτησης & διαμόρφωσης της ζητούμενης πληροφορίας.....	52
Εικόνα 19 Δημιουργία νέας εφαρμογής	55
Εικόνα 20 Παράθυρο διαλόγου δημιουργίας μιας νέας εφαρμογής.....	56
Εικόνα 21 Παράθυρο διαλόγου ρυθμίσεων εφαρμογής hydranewsfeed	57
Εικόνα 22 Ρυθμίσεις σελίδας	58
Εικόνα 23 Αδειοδοτικό πρόσβασης της εφαρμογής μας	59
Εικόνα 24 Script σε php.....	60
Εικόνα 25 Λήψη τιμής του code.....	60

Εικόνα 26 Access tokens όλων των σελίδων με διαχειριστή το Bot Bot.....	61
Εικόνα 27 Πληροφορίες για το access token	62
Εικόνα 28 Διάγραμμα UML του συστήματός μας	64
Εικόνα 29 Περιγραφή ενός μονολιθικού συστήματος.....	67
Εικόνα 30 Αρχική οθόνη XAMPP.....	80
Εικόνα 31 Get και Post μέθοδος HTTP πρωτοκόλλου.....	85
Εικόνα 32 Βήματα αυθεντικοποίησης χρηστών & εξουσιοδότησης εφαρμογών.....	86
Εικόνα 33 Δομή του project hydra_post.....	90
Εικόνα 34 Περιγραφή παραμέτρων της μεθόδου api().....	98
Εικόνα 35 Screenshot της υπηρεσίας μας.....	99

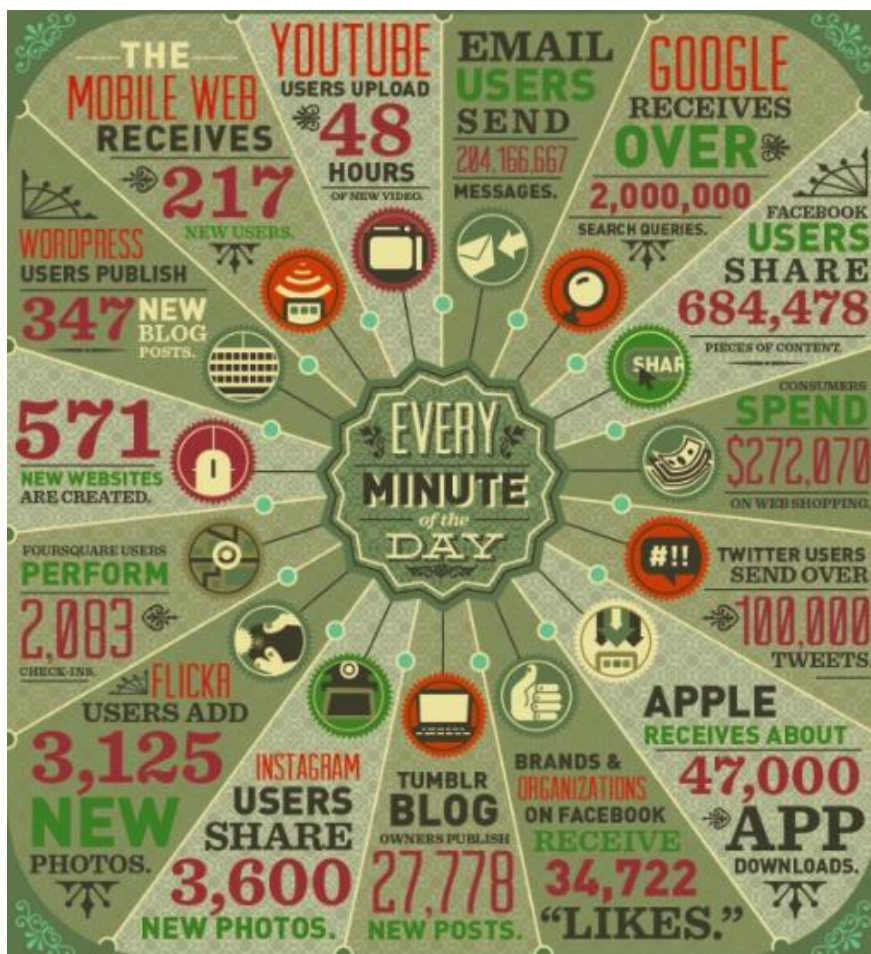
1 Εισαγωγή

Στην παρούσα πτυχιακή εργασία θα μελετήσουμε την ανάπτυξη ενός εργαλείου αυτόματης ενημέρωσης λογαριασμού στο Facebook. Το Facebook είναι ένας δημοφιλής ιστοχώρος κοινωνικής δικτύωσης. Οι χρήστες μπορούν να επικοινωνούν μέσω μηνυμάτων με τις επαφές τους και να τους ειδοποιούν όταν ανανεώνουν τις προσωπικές πληροφορίες τους. Όλοι έχουν ελεύθερη πρόσβαση στο να συμμετάσχουν σε δίκτυα που σχετίζονται μέσω πανεπιστημίου, θέσεων απασχόλησης ή γεωγραφικών περιοχών.

1.1 Όγκος πληροφορίας που διακινείται στο Facebook

Το διαδίκτυο στις μέρες μας είναι ένας τόπος, όπου τεράστιες ποσότητες πληροφορίας και δεδομένων παράγονται κάθε μέρα. Ο τεράστιος αριθμός δεδομένων δεν είναι μία αφηρημένη έννοια που δημιούργησαν οι επιστήμονες της πληροφορικής, αλλά ένα συνεχόμενο αυξανόμενο ρεύμα πληροφορίας που παράγεται καθημερινά ως αποτέλεσμα της ψηφιακής δραστηριότητας όλων των (απλών και μη) χρηστών του διαδικτύου. Τα δεδομένα δεν παύουν να παράγονται και να διακινούνται ποτέ. Κάθε λεπτό παράγονται τεράστια ποσά πληροφορίας στο διαδίκτυο.

Για να έχουμε μία οπτική απεικόνιση των πραγμάτων, η παρακάτω εικόνα μας δείχνει την ποσότητα των δεδομένων, καθώς και την πηγή από την οποία παράγονται τα δεδομένα αυτά στο διαδίκτυο κάθε λεπτό.



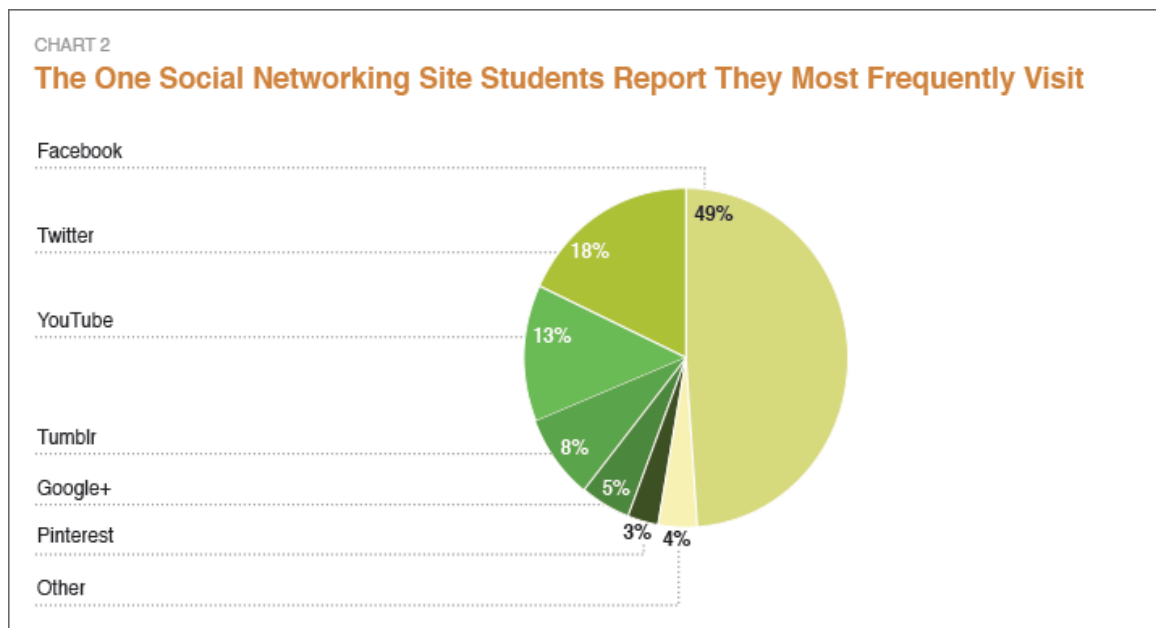
Εικόνα 1 Ποσότητα δεδομένων στο διαδίκτυο σε ένα λεπτό

Βλέπουμε πως το Facebook ευθύνεται για ένα σημαντικό κομμάτι της παραγόμενης πληροφορίας αυτής, καθώς οι χρήστες του μοιράζονται 684,478 δημοσιεύσεις περιεχομένου (από απλές ενημερώσεις- status updates, μέχρι links άρθρων από διάφορες ιστοσελίδες, προσωπικά βίντεο και φωτογραφίες) κάθε λεπτό. Συνολικά, σε μία μόνο μέρα, το σύστημα του Facebook επεξεργάζεται 2,5 δισεκατομμύρια περιεχομένου, συνολικού όγκου άνω των 500 terabytes.

Μετά το Google, το Facebook είναι η σελίδα που επισκέπτονται οι περισσότεροι χρήστες καθημερινά. Είναι το κοινωνικό δίκτυο με τους περισσότερους χρήστες, έχοντας σχεδόν 1,2 δισεκατομμύρια χρήστες σε όλο τον κόσμο.[\[1\]](#)

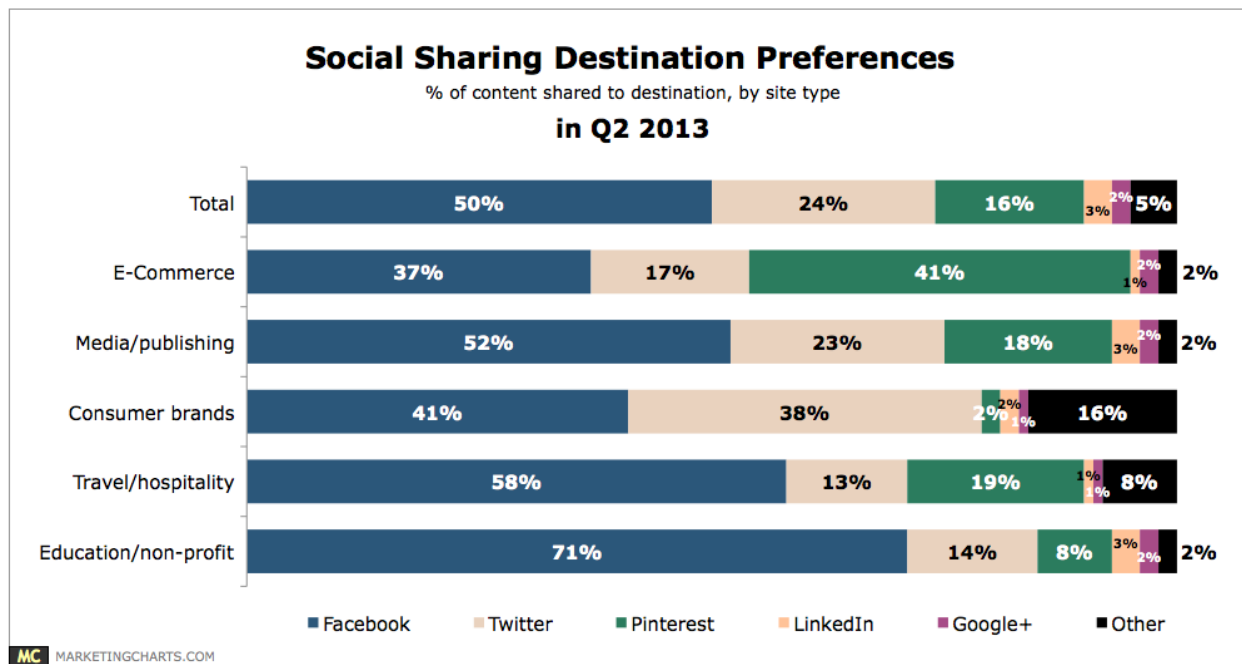
Είναι αξιοσημείωτο, επίσης, να αναφέρουμε ότι ο μέσος χρόνος παραμονής ενός χρήστη ανά επίσκεψη είναι 20 λεπτά. Οι χρήστες που κάνουν τουλάχιστον login μία φορά την ημέρα ξεπερνούν το ποσοστό του 75%.

Όσον αφορά την χρήση του από τους νέους μεταξύ 18-25 ετών, αλλά και φοιτητών συγκεκριμένα, το Facebook έρχεται πρώτο μεταξύ όλων των δικτύων κοινωνικής δικτύωσης, και μάλιστα με ποσοστό που αγγίζει το 49%.^[2]



Εικόνα 2 Διάγραμμα κοινωνικών δικτύων που επισκέπτονται οι φοιτητές

Όχι μόνο οι φοιτητές επισκέπτονται περισσότερο το Facebook από οποιοδήποτε άλλο κοινωνικό δίκτυο, αλλά και όσον αφορά τη δημοσίευση θεμάτων που αφορούν την εκπαίδευση, το Facebook βρίσκεται στην πρώτη θέση προτίμησης με ποσοστό 71%, όπως φαίνεται και στο ακόλουθο διάγραμμα:



Εικόνα 3 Πρώτο το Facebook σε αναζήτηση θεμάτων που αφορούν την εκπαίδευση

Η ευρεία λοιπόν χρήση του Facebook από φοιτητές, αλλά και το γεγονός ότι πολλά ακαδημαϊκά θέματα δημοσιεύονται πλέον στο δίκτυο αυτό, μας οδήγησαν στην απόφασή μας να αναπτύξουμε ένα εργαλείο το οποίο θα συμβάλει στην ενημέρωση των φοιτητών του τμήματος μας για τις δημόσιες ανακοινώσεις του από το Facebook, μια σελίδα την οποία επισκέπτονται και αφιερώνουν χρόνο καθημερινά.

1.2 Features για την ενημέρωση των χρηστών του Facebook

Το Facebook έχει δημιουργήσει πολλές δυνατότητες για την ενημέρωση των χρηστών του.[\[3\]](#) Μερικές από αυτές είναι οι ακόλουθες:

- το graph search, το οποίο επιτρέπει στους χρήστες να αναζητούν μέσα στο δίκτυο των φίλων τους για απαντήσεις σε ερωτήσεις όπως ταινίες που άρεσαν στους φίλους μου και του έχεις αρέσει και η ταινία “The Hobbit”,

- το ticker, στο οποίο ο χρήστης μπορεί να ενημερωθεί για το τι κάνουν οι φίλοι τους και ενημερώσεις σελίδων,
- το facebook live, το οποίο είναι ένα live streaming κανάλι βίντεο και έχει σκοπό να κρατάει τους Facebook χρήστες ενήμερους για το τι συμβαίνει στη σελίδα κοινωνικής δικτύωσης και
- το hash tagging, το οποίο είναι ένα hashtag (#) που μπορεί ο χρήστης να κάνει κλικ πάνω του και μπορεί να αναζητήσει θέματα τα οποία συζητιούνται σε όλο τον κόσμο πιο εύκολα.

Όλα τα παραπάνω features, χρησιμοποιούν το Graph API για να αντλήσουν την ζητούμενη πληροφορία. Αυτή η διαδικασία περιγράφεται αναλυτικά παρακάτω.

Λόγω της αυξανόμενης όμως χρήσης του Facebook, έχουν αναπτυχθεί πολλές τέτοιες εφαρμογές και από τρίτους, χρησιμοποιώντας την πλατφόρμα του Facebook, με σκοπό την ενημέρωση και όχι μόνο.

1.3 Σκοπός της εργασίας

Το γεγονός που καθιστά σημαντική αυτή την υπηρεσία είναι ότι θα παρέχει έναν νέο, πιο απλοποιημένο και γρήγορο τρόπο ενημέρωσης στους φοιτητές του τμήματος Μηχανικών Πληροφορικής

Σκοπός της εργασίας είναι η ανάπτυξη ενός εργαλείου μέσω του οποίου θα μπορούν οι φοιτητές και οι καθηγητές του τμήματος, καθώς και όποιος άλλος επιθυμεί, να ενημερώνονται για τα δημόσια νέα - ανακοινώσεις του τμήματος μέσα από την σελίδα του Facebook. Μέχρι στιγμής, αυτή η δυνατότητα δίνεται μέσω της ιστοσελίδας ύδρα (<https://hydra.it.teithe.gr/s/>). Με την συνεχώς αυξανόμενη χρήση του Facebook από όλο και περισσότερους φοιτητές, κρίναμε ότι ένα τέτοιο εργαλείο θα βοηθούσε πολύ στην αμεσότερη ενημέρωση, κυρίως των φοιτητών, καθώς οι ανακοινώσεις θα δημοσιεύονται

στην υπηρεσία μας, τη στιγμή που ανακοινώνονται και στην ύδρα και θα είναι ορατές στα νέα (news feed) καθενός που έχει κάνει like στη σελίδα “Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙΘ”. Επίσης, για έναν απλό χρήστη είναι πολύ πιο εύκολο να κάνει like σε μια σελίδα στο Facebook από την οποία θα ενημερώνεται για τις δημόσιες ανακοινώσεις του τμήματος, παρά να πρέπει να επισκέπτεται τον ιστότοπο της ύδρας. Σε περίπτωση ακόμη που ξεχάσει να επισκεφτεί τον ιστότοπο της ύδρας, είναι σχεδόν σίγουρο ότι θα ενημερωθεί καθώς το Facebook είναι μία σελίδα που οι περισσότεροι χρήστες επισκέπτονται καθημερινά.

1.4 Επιτεύγματα της εργασίας

Αυτό που επιτεύχθηκε, τελικά, με την ανάπτυξη της εφαρμογής μας είναι η άμεση δημοσίευση των νέων του τμήματος μας στα νέα (news feed) των λογαριασμών των φοιτητών στο Facebook. Κάθε μέσος φοιτητής θα μπει στο λογαριασμό του στο Facebook τουλάχιστον μία φορά την ημέρα. Έτσι, αν υπάρχει νέα ανακοίνωση θα ενημερωθεί χωρίς να χρειαστεί να κάνει κάποια άλλη διαδικασία. Συμπερασματικά λοιπόν, αυτοματοποιήθηκε η διαδικασία ενημέρωσης τους και έγινε πιο γρήγορη και πιο εύκολη.

1.5 Η καινοτομία στην υπηρεσία μας

Η καινοτομία που έχει να προσφέρει το εργαλείο που αναπτύξαμε είναι ότι τραβάει ακατέργαστη πληροφορία από μια πηγή (στατική ιστοσελίδα) και έπειτα, με τις κατάλληλες διαδικασίες που δημιουργήσαμε, την διαμορφώνει σε χρήσιμο περιεχόμενο, το οποίο τελικά δημοσιοποιείται αυτόματα σε μία σελίδα στο Facebook.

1.6 Συνοπτική ανάλυση επόμενων κεφαλαίων

Σε αυτό εδώ το υποκεφάλαιο, θα σας κατευθύνουμε για το τι θα διαβάσετε παρακάτω στην εργασία μας. Στο επόμενο κεφάλαιο, θα δούμε λίγο πιο αναλυτικά το θεωρητικό υπόβαθρο της εφαρμογής μας και θα μελετήσουμε την αναγκαιότητα της υπηρεσίας μας.

Στο 3ο κεφάλαιο, μελετάμε υφιστάμενες λύσεις του προβλήματός μας και προσπαθούμε να εντοπίσουμε τα πλεονεκτήματα και τα μειονεκτήματα τους και τέλος, τις συγκρίνουμε με βάση κάποια κριτήρια που έχουμε θέσει εξ αρχής.

Στο 4ο κεφάλαιο, περιγράφουμε το μεθοδολογικό πλαίσιο της εφαρμογής μας. Χρησιμοποιούμε σχήματα και διαγράμματα UML για να δώσουμε στον αναγνώστη να καταλάβει την ανάλυση απαιτήσεων και τις ροές της εφαρμογής μας. Τέλος, αναφέρουμε τα βασικά συστατικά της υπηρεσίας μας.

Στο 5ο κεφάλαιο, αναφέρουμε τις τεχνολογίες που χρησιμοποιήσαμε και τις περιγράφουμε συνοπτικά. Εξηγούμε γιατί διαλέξαμε την εκάστοτε τεχνολογία και σε μερικές από αυτές, γίνεται μια σύγκριση με μια αντίστοιχη τεχνολογία για να καταλήξουμε σε ένα συμπέρασμα.

Στο 6ο κεφάλαιο, αναλύουμε τα βασικά συστατικά της υπηρεσίας μας. Αυτό σημαίνει ότι έχουμε παραθέσει τις κλάσεις που χρησιμοποιήσαμε και τις αναλύουμε γραμμή γραμμή. Σε αυτό το κεφάλαιο επίσης, δείχνουμε πως έγινε η μοντελοποίηση της πληροφορίας μας.

Στο 7ο κεφάλαιο, προτείνουμε μελλοντικές επεκτάσεις, καθώς και που θα μπορούσε να ενσωματωθεί το εργαλείο αυτό έτσι ώστε να παρέχεται στο χρήστη μια πιο ολοκληρωμένη εφαρμογή με περισσότερες δυνατότητες.

Ακολουθούν η βιβλιογραφία και οι πηγές, που συμβουλευθήκαμε για την συγγραφή αυτής της πτυχιακής εργασίας.

2 Θεωρητικό υπόβαθρο της εφαρμογής μας

Σε αυτό το κεφάλαιο, θα ασχοληθούμε με τον τρόπο που μπορεί να γίνει η ανάκτηση και η ανάρτηση περιεχομένου στον ιστό. Υπάρχουν δύο τεχνολογίες οι οποίες υποστηρίζουν την λειτουργία αυτή, η τεχνολογία push και η τεχνολογία pull. Θα αναλύσουμε τον τρόπο με τον οποίο λειτουργούν, και θα αναφερθούμε σε τεχνολογίες επικοινωνίας σε πραγματικό χρόνο (Real-time Web Applications) που βασίζονται στις τεχνολογίες αυτές. Στην συνέχεια, θα μελετήσουμε κάποιες υπηρεσίες και εφαρμογές που διατίθενται για αυτόματη ανάρτηση περιεχομένου στους χρήστες. Θα περιγράψουμε το πώς λειτουργούν και θα προσπαθήσουμε να εντοπίσουμε τα πλεονεκτήματα και τα μειονεκτήματα της κάθε μίας.

2.1 Ανάκτηση και ανάρτηση περιεχομένου

Όπως αναφέρθηκε και παραπάνω, υπάρχουν δύο τεχνολογίες από τις οποίες υποστηρίζεται η προώθηση περιεχομένου σε μία ιστοσελίδα. Η τεχνολογία Push ή αλλιώς server push περιγράφει έναν τρόπο επικοινωνίας βασισμένης στο διαδίκτυο, όπου το αίτημα για μια δεδομένη συναλλαγή ενεργοποιείται από τον εκδότη ή το κεντρικό διακομιστή (server), ενώ αντίθετα η τεχνολογία Pull, αφορά τον τόπο επικοινωνίας κατά τον οποίο το αίτημα για τη διαβίβαση των πληροφοριών ξεκινά από τον δέκτη ή αλλιώς client.[\[4\]](#)

2.1.1 Τεχνολογία Pull

Η τεχνολογία Pull, η αλλιώς client pull, είναι το μοντέλο που ακολουθείται κατά κύριο λόγο στο διαδίκτυο (client-server ή request/response). Είναι μια από τις μεθόδους για διαδικτυακή επικοινωνία, όπου το αρχικό αίτημα για τα δεδομένα προέρχεται από τον χρήστη (client), και στη συνέχεια η απάντηση από το διακομιστή.



Εικόνα 4 Απεικόνιση τεχνολογίας pull

Στην Pull δηλαδή, ο χρήστης κάνει αίτηση για να λάβει κάποιο περιεχόμενο και το δίκτυο του στέλνει σαν απάντηση το περιεχόμενο αυτό ή ένα μήνυμα αποτυχίας.

Η τεχνολογία pull θεωρείται πρωτοπόρα των υπολογιστικών δικτύων. Έθεσε την βάση για την επικοινωνία πελάτη - εξυπηρετητή καθώς εισήγαγε μια αξιόπιστη τεχνική που υποστήριζε την διαδικασία αυτή. Το θεμέλιο των υπολογιστικών δικτύων βασίζεται στις αιτήσεις από πολλούς πελάτες - χρήστες σε έναν κεντρικό διακομιστή, ο οποίος επεξεργάζεται όλα τα αποτελέσματα ταυτόχρονα, ή ένα ένα με την σειρά. Η τεχνολογία pull χρησιμοποιείται ευρέως στο διαδίκτυο για HTTP αιτήματα σελίδας από ιστότοπους και για την περιήγηση γενικότερα των χρηστών σε ιστοσελίδες και ιστότοπους. Την χρησιμοποιούν οι μεγαλύτερες εταιρίες όπως η Google, η Hotmail και η Yahoo για να διαχειριστούν την αίτηση πρόσβασης σε αυτές από τους χρήστες.

Δεν υπάρχει συγκεκριμένο εύρος της δυνατότητας εφαρμογής της τεχνολογίας pull, όμως οι κύριες χρήσεις της μπορούν να κατηγοριοποιηθούν σε 4 ευρείες λειτουργίες:

1. Φόρτωση ιστοσελίδας
2. Mail αιτήματα
3. Λήψη αρχείων μέσω του διαδικτύου
4. Κοινοπραξίες διαδικτύου

Η φόρτωση ιστοσελίδας πραγματοποιείται με την αποστολή ενός αιτήματος pull στο διακομιστή, ο οποίος το λαμβάνει και επιστρέφει την σελίδα στον browser του χρήστη. Η αίτηση για mail είναι καθημερινή δραστηριότητα των χρηστών του ίντερνετ που ελέγχουν αν έχουν λάβει νέα πληροφορία στο mail

box τους. Το αίτημα αυτό μεταφέρεται στον διακομιστή μέσω της τεχνολογίας pull. Ομοίως, η λήψη αρχείων αιτείται τα επιθυμητά δεδομένα κάνοντας pull το αίτημα αυτό στον διακομιστή. Τα αρχεία μπορεί να είναι οποιοδήποτε τύπου. Κοινοπραξία διαδικτύου μπορεί να είναι η περιοδική ανανέωση ενός εγγράφου για τους καθημερινούς αναγνώστες ειδήσεων. Η ανανέωση αυτή πραγματοποιείται από τον διακομιστή μετά από αίτηση του χρήστη.

Συνήθως, οι χρήστες ψάχνουν για μία σελίδα και την επισκέπτονται μόνο εάν παρέχει χρήσιμο και ελκυστικό περιεχόμενο. Ο κώδικας pull είναι περισσότερο αποτελεσματικός για τον χρήστη που αναζητά ένα συγκεκριμένο θέμα.

Η απλή αναζήτηση σε μία μηχανή αναζήτησης, ή ακόμη και η εισαγωγή ενός συγκεκριμένου URL στη γραμμή διευθύνσεων, ισοδυναμούν με ένα client pull στο διακομιστή. Οι περισσότερες τροφοδοσίες στο διαδίκτυο (Web feeds) όπως το RSS , τεχνικά πραγματοποιούνται με client pull. Με το RSS, ο RSS reader του χρήστη ελέγχει τον διακομιστή περιοδικά για νέο περιεχόμενο· ο διακομιστής δεν αποστέλλει πληροφορίες στον πελάτη χωρίς να έχουν ζητηθεί. Αυτός ο συνεχής έλεγχος είναι αναποτελεσματικός και συνέβαλε στη μείωση ή ακόμη και την διακοπή λειτουργίας πολλών δημοφιλών RSS feeds που δεν μπορούσαν να χειριστούν το εύρος ζώνης. Για την επίλυση αυτού του προβλήματος, χρησιμοποιήθηκε η τεχνολογία push.[\[5\]\[6\]](#)

2.1.2 Τεχνολογία Push

Η τεχνολογία Push, ή server push, περιγράφει μια επικοινωνία βασισμένη στο διαδίκτυο, όπου το αίτημα για μία συναλλαγή ξεκινά από τον κεντρικό διακομιστή. (Σε αντίθεση με την τεχνολογία pull, όπου η αίτηση για τη διαβίβαση των πληροφοριών ξεκινά από τον πελάτη).



Εικόνα 5 Απεικόνιση τεχνολογίας push

Με τον όρο Push εννοούμε την αποστολή περιεχομένου σε ένα χρήστη δικτύου, χωρίς ο ίδιος να την έχει ζητήσει ρητά την στιγμή που την λαμβάνει. Ουσιαστικά είναι σαν να «σπρώχνει» το δίκτυο την πληροφορία στον χρήστη, εξ' ου και το όνομα Push. Μια άλλη ονομασία του είναι και publish / subscribe. Το μοντέλο αυτό είναι ακριβώς το αντίθετο από το Pull και βασίζεται σε προτιμήσεις, όσον αφορά τις πληροφορίες, που εκφράζονται εκ των προτέρων. Μια Push υπηρεσία είναι μια εφαρμογή λογισμικού που γνωρίζει τι είδους περιεχόμενο ενδιαφέρει τον κάθε χρήστη της και του το παραδίδει όταν αυτό είναι διαθέσιμο ή τον πληροφορεί από που μπορεί να το πάρει. Ο πελάτης δηλαδή, γίνεται “συνδρομητής” σε διάφορα “κανάλια” πληροφορίας που παρέχονται από ένα διακομιστή, και όταν νέο περιεχόμενο είναι διαθέσιμο σε ένα από αυτά τα κανάλια, ο διακομιστής ωθεί την πληροφορία αυτή στον πελάτη. Οι συνδιασκέψεις και η αποστολή άμεσων μηνυμάτων είναι κάποια τυπικά παραδείγματα των υπηρεσιών που προσφέρει η push. Μηνύματα συνομιλίας, και μερικές φορές αρχεία, γίνονται push στο χρήστη τη στιγμή που λαμβάνονται από την υπηρεσία άμεσων μηνυμάτων. Τόσο τα αποκεντρωτικά, όσο και τα συγκεντρωτικά προγράμματα (όπως το IRC και το XMPP) επιτρέπουν να προωθούνται αρχεία, πράγμα που σημαίνει ότι ο αποστολέας ενεργοποιεί τη μεταφορά δεδομένων και όχι ο παραλήπτης.

Το email μπορεί να είναι επίσης ένα σύστημα προώθησης (push system). Το πρωτόκολλο SMTP είναι ένα πρωτόκολλο push. Ωστόσο, στο τελευταίο βήμα, από το διακομιστή αλληλογραφίας στην επιφάνεια εργασίας του υπολογιστή, συνήθως χρησιμοποιείται ένα πρωτόκολλο pull, όπως το POP3 ή IMAP.

Σύγχρονοι e-mail clients κάνουν αυτό το βήμα να φαίνεται στιγμιαίο, ελέγχοντας επανειλημμένα το διακομιστή αλληλογραφίας, για τον εντοπισμό νέας αλληλογραφίας. Το πρωτόκολλο IMAP περιλαμβάνει την εντολή IDLE, η οποία επιτρέπει στο διακομιστή να ειδοποιεί τον πελάτη - χρήστη για το αν φτάνουν νέα μηνύματα.

Άλλες χρήσεις των διαδικτυακών εφαρμογών που υποστηρίζουν την τεχνολογία push είναι τα συστήματα online συνομιλιών, στοιχημάτων και

παιχνιδιών, καθώς επίσης και αποτελέσματα αθλημάτων, κονσόλες παρακολούθησης και αισθητήρες παρακολούθησης δικτύου.

HTTP Server Push

Το HTTP Server Push, γνωστό και ως HTTP streaming, είναι ένας μηχανισμός για την αποστολή αυτόκλητων (ασύγχρονων) δεδομένων από ένα διακομιστή σε web browser και μπορεί να επιτευχθεί μέσα από διάφορους μηχανισμούς. Γενικά, ένας διακομιστής δεν τερματίζει μια σύνδεση αμέσως μόλις τα δεδομένα απόκρισης έχουν παραδοθεί σε έναν πελάτη. Αφήνει ανοικτή τη σύνδεση, έτσι ώστε αν συμβεί ένα γεγονός (π.χ.: αλλαγή στα εσωτερικά δεδομένα τα οποία πρέπει να αναφερθούν σε έναν ή πολλούς πελάτες), να μπορεί να σταλεί αμέσως. Διαφορετικά, το γεγονός θα πρέπει να μπει στην ουρά μέχρι την επόμενη φορά που θα ληφθεί αίτημα από τον πελάτη. Οι περισσότεροι διακομιστές προσφέρουν αυτή τη λειτουργία μέσω CGI (π.χ. μέσω non-parsed header σεναρίων για Apache).[\[7\]\[8\]](#)

Για να λειτουργήσει όμως σωστά και αποτελεσματικά το μοντέλο αυτόματης προώθησης πρέπει να πληρούνται κάποιες προϋποθέσεις. Η πρώτη και πιο σημαντική είναι να υποστηρίζεται από το δίκτυο ποιότητας υπηρεσίας (QoS). Οι περισσότερες υπηρεσίες Push είναι άχρηστες, αν δεν παραληφθούν άμεσα από τον χρήστη (πχ ειδοποίηση για κυκλοφοριακή κίνηση). Βέβαια, η ποιότητα υπηρεσίας είναι πιο δύσκολο να εξασφαλιστεί στα δίκτυα που υποστηρίζουν υπηρεσίες Push απ' ότι σε αυτά που προσφέρουν μόνο τις κλασικές υπηρεσίες Pull. Αυτό συμβαίνει επειδή η δικτυακή «συμπεριφορά» του μοντέλου Push έχει κάποιες ιδιαιτερότητες που δεν την καθιστούν προβλέψιμη (και όπως είναι γνωστό η πρόβλεψη παίζει μεγάλο ρόλο στην εξασφάλιση ποιότητας υπηρεσίας). Μία από αυτές είναι το ότι η δικτυακή κίνηση που προκαλείται είναι απρόβλεπτη, αφού δεν εξαρτάται τόσο από τον ίδιο τον χρήστη, όσο από την στατιστική συμπεριφορά (στο πεδίο του χρόνου) του συμβάντος που σκανδαλίζει την Push υπηρεσία. Το δίκτυο μπορεί να υποπέσει σε ακόμα χειρότερη κατάσταση αν οι Push υπηρεσίες αιτηθούν ξαφνικά από μεγάλο πλήθος συνδρομητών. Επίσης, οι Push υπηρεσίες

συνήθως μεταφέρουν περιεχόμενο. Και εφόσον αυτό μπορεί να είναι όσο ογκώδες θέλει ο χρήστης, το δίκτυο διατρέχει μεγαλύτερο κίνδυνο να υπερφορτωθεί (αυτή είναι μια ιδιαιτερότητα γενικά των υπηρεσιών προστιθέμενης αξίας).[\[6\]\[8\]](#)

2.1.3 Σύγκριση Τεχνολογιών *Push* και *Pull*

Όπως αναφέραμε και παραπάνω, η τεχνολογία *push* είναι το αντίθετο της τεχνολογίας *pull*. Η κύρια διαφορά τους έγκειται στο γεγονός ότι ο όρος “*pull*” αφορά την επιλογή των πληροφοριών με πρωτοβουλία του αιτούντα, ενώ αντίθετα ο όρος “*push*” αναφέρεται σε μια αυτοματοποιημένη διαδικασία παράδοσης πληροφορίας, η οποία υποκινείται από τον διακομιστή, χωρίς να την έχει αιτηθεί.

Υπάρχουν πολλά πλεονεκτήματα της τεχνολογίας *pull*. Κατ’ αρχήν ο διακομιστής δεν χρειάζεται να κάνει κάποια ενέργεια, εκτός και αν υπάρχει κάποιο αίτημα προς αυτόν που εκκρεμεί. Διαφορετικά, όσο δεν του αποστέλλονται αιτήματα, παραμένει αδρανής. Ο πελάτης λαμβάνει μόνο την πληροφορία που χρειάζεται, όταν αυτός την χρειάζεται. Επίσης, ο πελάτης είναι αυτός που ασκεί τον έλεγχο της αλληλεπίδρασης, έτσι ώστε η αλληλεπίδραση (θεωρητικά) να συμβαίνει όταν και όπου ο πελάτης είναι σε καλύτερη θέση να κάνει χρήση της υπηρεσίας.[\[9\]](#)

Από την άλλη πλευρά όμως, η τεχνολογία *pull* έχει και αυτή τα μειονεκτήματά της. Η επιβάρυνση είναι του πελάτη. Ο διακομιστής μπορεί να κατακλύζεται με αιτήσεις και να μην μπορέσει να ανταπεξέλθει. Το δίκτυο λοιπόν μπορεί να είναι γεμάτο από αιτήματα, δεδομένου ότι κάθε πελάτης στέλνει τα αιτήματα αυτά ξεχωριστά. Γενικά, θα δαπανηθεί περισσότερη ενέργεια. Για παράδειγμα, είναι πιθανό να υπάρχει πολύ μεγαλύτερη κίνηση (*traffic*) στο δίκτυο. Ο φόρτος του διακομιστή μπορεί να είναι αποσπασματικός και απρόβλεπτος, δηλαδή μπορεί να υπάρξουν στιγμές που ο διακομιστής θα είναι πολύ απασχολημένος, και άλλες στιγμές κατά τις οποίες δεν θα χρησιμοποιείται καθόλου. Το γεγονός αυτό καθιστά αναποτελεσματική τη

χρήση του μηχανήματος και των πόρων του. Η Client pull είναι αποτελεσματικότερη, όταν τα αιτήματα των πελατών ποικίλλουν, αλλά συνολικά δεν αποφέρουν πολύ μεγάλο φορτίο στον διακομιστή.

Αντίστοιχα, το μοντέλο push έχει αρκετά πλεονεκτήματα. Αρχικά, ο διακομιστής ελέγχει ποιος παίρνει τι και πότε. Αυτό σημαίνει ότι μπορεί να διαχειρίζεται τους πόρους του και να ελέγξει το φόρτο του. Αν ο διακομιστής τροφοδοτεί πολλούς πελάτες με την ίδια ή παρόμοια πληροφορία, μπορεί να υπάρξει οικονομία πόρων. Για παράδειγμα, ο διακομιστής μπορεί να φτιάξει το πακέτο με τις ζητούμενες πληροφορίες μία φορά μόνο και να τις στείλει σε πολλαπλούς πελάτες. Οι πελάτες δεν χρειάζεται να κάνουν κάτι. Η υπηρεσία εκτελείται κάθε φορά που ο διακομιστής το κρίνει σκόπιμο. Και αυτό το μοντέλο όμως δεν δουλεύει τέλεια. Έχει και κάποια μειονεκτήματα. Οι πληροφορίες θα μπορούσαν να παραδοθούν σε μια κακή για τον πελάτη στιγμή. Αυτό συμβαίνει για παράδειγμα σε υπολογιστικά συστήματα, όπου ο διακομιστής έχει πολύ μεγαλύτερη επεξεργαστική ισχύ από αυτή του πελάτη και ο πελάτης πρέπει να δαπανήσει το σύνολο του χρόνου του χειριζόμενος τις αποστολές του διακομιστή. Ακόμα κι αν ο διακομιστής διατηρεί τον έλεγχο, υπάρχει η πιθανότητα κάποια στιγμή να είναι πολύ απασχολημένος και έτσι, η παράδοση των πληροφοριών να καθυστερήσει. Ορισμένοι πελάτες ενδέχεται να χρειάζονται συχνότερη προσοχή. Ένας τέτοιος πελάτης μπορεί να μην μπορεί να πάρει ό, τι χρειάζεται αρκετά συχνά, ή εγκαίρως. Μερικές φορές, ο διακομιστής δεν γνωρίζει ότι (ή πότε) ο πελάτης θέλει ή χρειάζεται κάτι.

Όπως αναφέραμε λοιπόν παραπάνω, η κάθε τεχνολογία έχει τα θετικά και τα αρνητικά της. Οι απαιτήσεις της κάθε εφαρμογής είναι ο παράγοντας βάση του οποίου η μια τεχνολογία θα κριθεί καταλληλότερη σε σχέση με την άλλη.

2.2 Τεχνολογίες επικοινωνίας σε πραγματικό χρόνο

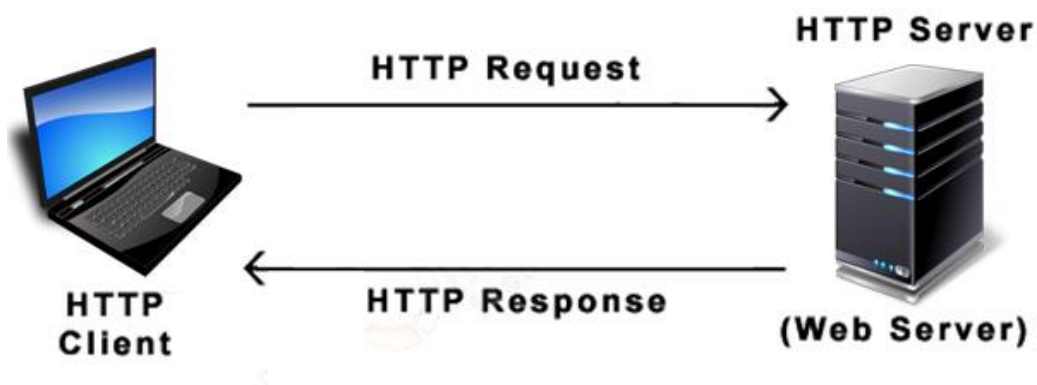
Η εφαρμογή που θα δημιουργήσουμε θα είναι μία διαδικτυακή εφαρμογή η οποία θα εκτελείται σε πραγματικό χρόνο. Μία τέτοια εφαρμογή ιστού ορίζεται

ως “ένα σύνολο τεχνολογιών και πρακτικών που επιτρέπουν στους χρήστες να λαμβάνουν πληροφορίες, τη στιγμή που δημοσιεύονται”. Το Facebook είναι και αυτό μία από τις υπηρεσίες που χρησιμοποιεί τέτοιου είδους εφαρμογές.

Υπάρχουν αρκετά πλαίσια (frameworks) και υπηρεσίες, οι οποίες μπορούν να βοηθήσουν στο να δημιουργήσουμε μία εφαρμογή ιστού σε πραγματικό χρόνο. Όμως, όλες αυτές οι υπηρεσίες χρησιμοποιούν μία ή ένα συνδυασμό των ακόλουθων τεχνολογιών:

- AJAX Polling
- AJAX Long-Polling
- HTML5 Server Sent Events
- HTML5 Websockets.

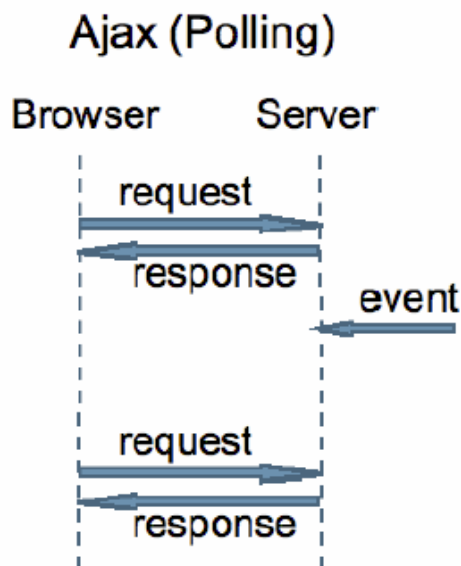
Προτού αναλύσουμε αυτές τις τεχνολογίες ξεχωριστά, θα αναφερθούμε εν συντομία στο πως λειτουργεί ένα απλό http αίτημα. Η διαδικασία αυτή είναι απλή. Ο πελάτης αιτείται μία ιστοσελίδα από τον διακομιστή στέλνοντας του ένα αίτημα HTTP Request) . Ο διακομιστής υπολογίζει την απόκριση και τη στέλνει στον πελάτη (HTTP Response).[\[10\]](#)



Εικόνα 6 Ένα απλό HTTP αίτημα

2.2.1 AJAX Polling

Η πρώτη τεχνολογία είναι το AJAX Polling. Ο πελάτης αιτείται μία ιστοσελίδα από τον διακομιστή, όπως κάνει τακτικά, και ο διακομιστής απαντά. Στη συνέχεια, μέσω περιοδικών AJAX αιτημάτων, ο πελάτης αιτείται δεδομένα από το διακομιστή. Τέλος, ο διακομιστής απαντά και μπορεί να διοχετεύσει νέα δεδομένα στο χρήστη.[\[10\]](#)



Εικόνα 7 Πως δουλεύει το Ajax Polling

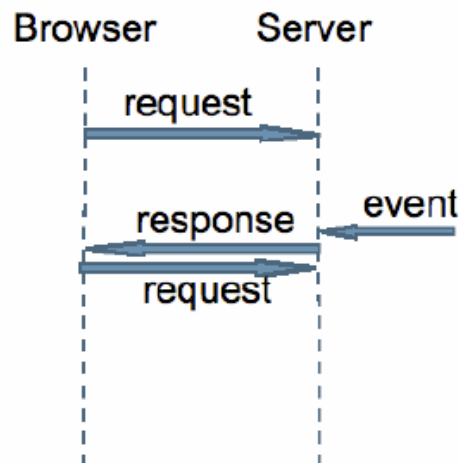
2.2.2 AJAX Long-Polling

Η δεύτερη τεχνολογία που χρησιμοποιείται ονομάζεται AJAX Long-Polling. Η τεχνολογία αυτή κρατά την σύνδεση πελάτη - διακομιστή ανοιχτή μέχρι ο πελάτης να λάβει απάντηση στο αίτημα του από τον διακομιστή, και στη συνέχεια, κάνει ξανά poll και περιμένει για την επόμενη απόκριση.

Η Long-polling δεν είναι αυστηρά ένα client-side μοτίβο, αλλά απαιτεί από τον web server να κρατήσει ανοικτή τη σύνδεση.

Τα AJAX-αιτήματα έχουν μία ιδιότητα, που ονομάζεται “timeout”. Έτσι, όταν αρχίζουν να στέλνονται αιτήματα στο διακομιστή, οι πελάτες μπορούν να περιμένουν την απάντηση του διακομιστή για αρκετό χρονικό διάστημα. Στην τεχνολογία αυτή πραγματοποιούνται μεγαλύτερα αιτήματα στον διακομιστή, έτσι ώστε όταν ο διακομιστής έχει μια ενημέρωση, να την στείλει στον πελάτη. Στη συνέχεια, μπορεί να γίνει ένα νέο αίτημα το οποίο ομοίως θα κρατήσει την σύνδεση ανοιχτή για αρκετό χρονικό διάστημα. Αυτή η τεχνολογία χρησιμοποιείται σε υπηρεσίες email, όπου ο χρήστης δεν χρειάζεται να κάνει απολύτως τίποτα.[\[10\]](#)

Ajax Push (Long Poll)



Εικόνα 8 Πως δουλεύει το Ajax Long Polling

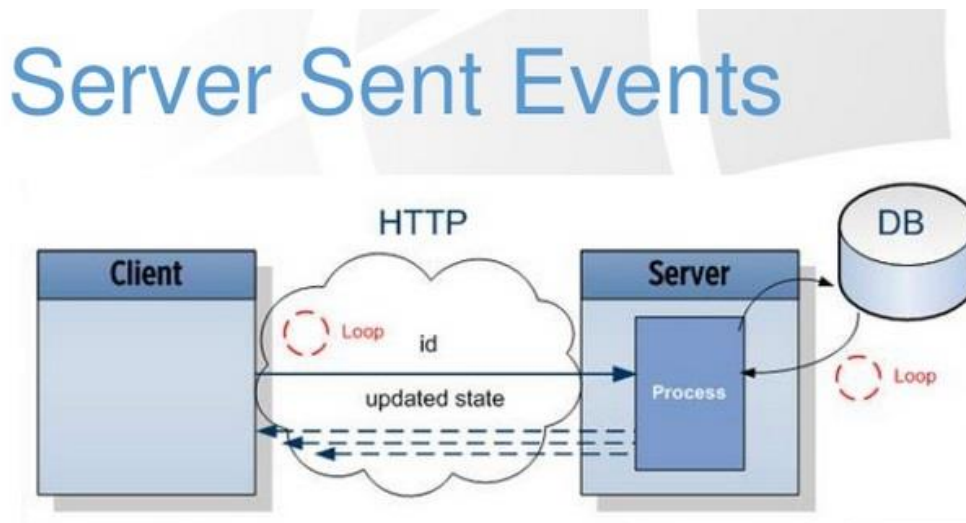
2.2.3 HTML5 Server Sent Events

Η τρίτη τεχνολογία ονομάζεται HTML5 Server Sent Events. Ως server-sent event ορίζεται η διαδικασία κατά την οποία μια ιστοσελίδα ανανεώνεται αυτόματα από τον διακομιστή και ουσιαστικά τυποποιεί τον τρόπο με τον οποίο γίνεται η ροή δεδομένων από τον διακομιστή στον πελάτη.

Οι αυτόματες ενημερώσεις μιας ιστοσελίδας από ένα διακομιστή είναι κάτι που υποστηριζόταν και παλαιότερα, όμως η ιστοσελίδα έπρεπε να “ρωτάει” αν

ύπηρχαν νέες ενημερώσεις διαθέσιμες. Με τα server-sent events, η ενημέρωση γίνεται αυτόματα.[\[11\]](#)

Το Server-sent events (SSE) είναι ένα πρότυπο που περιγράφει πως οι διακομιστές μπορούν να ξεκινήσουν την μετάδοση δεδομένων προς τους πελάτες, εφόσον η αρχική σύνδεση μεταξύ διακομιστή και πελάτη έχει συσταθεί. Η σύνδεση αυτή είναι μόνιμη και καταργεί έτσι την ανάγκη για συνεχές polling στον διακομιστή. Συνήθως, χρησιμοποιούνται για την αποστολή μηνυμάτων ενημέρωσης ή συνεχών ροών δεδομένων στον browser ενός πελάτη. Ακόμη, με την χρήση των SSE είναι δυνατό να ωθούνται (γίνονται push) DOM events συνεχόμενα από τον διακομιστή στον browser του πελάτη.[\[12\]](#) Έχουν σχεδιαστεί για την ενίσχυση της φυσικής, cross-browser ροής μέσω ενός JavaScript API που ονομάζεται EventSource, μέσω του οποίου ένας πελάτης ζητά μια συγκεκριμένη διεύθυνση URL για να λάβει μία ροή γεγονότος (event stream).



Εικόνα 9 Πως λειτουργούν τα Server Set Events (SSE)

Το αντικείμενο EventSource χρησιμοποιείται για την λήψη SSE ειδοποιήσεων. Το αντικείμενο δημιουργείται με σκοπό να ανοίξει μία HTTP σύνδεση, για να λαμβάνει μέσω αυτής push ενημερώσεις από έναν διακομιστή με την μορφή DOM events.

Το EventSource API του SSE έχει τυποποιηθεί σαν μέρος της HTML5 από το W3C[\[13\]](#) και έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι επεκτάσιμο και

να μπορεί να δουλέψει παράλληλα και με άλλα συστήματα push ενημερώσεων

Μερικά παραδείγματα εφαρμογών και υπηρεσιών που χρησιμοποιούν SSE είναι οι ενημερώσεις στα κοινωνικά δίκτυα Facebook και Twitter, ενημερώσεις των τιμών των μετοχών, τροφοδοσίες ειδήσεων, αθλητικών αποτελεσμάτων, κ.ά.

Στα αρνητικά στοιχεία αυτής τη τεχνολογίας συγκαταλέγεται το γεγονός ότι δεν υποστηρίζεται από όλους τους web browsers.

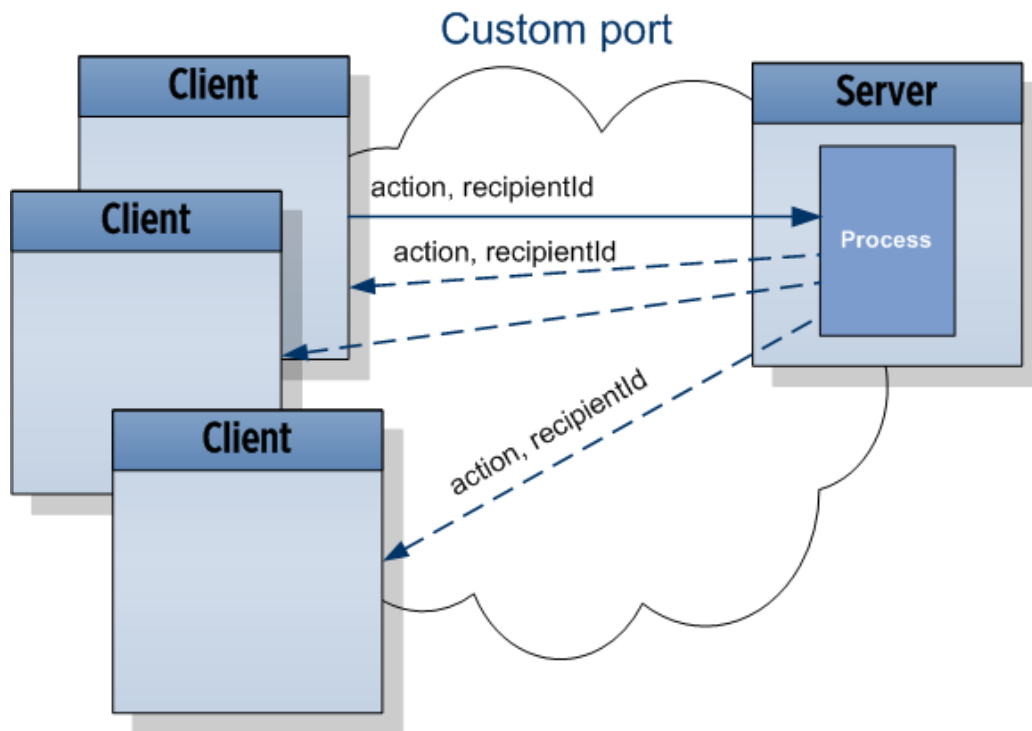
2.2.4 HTML5 Websockets

Η τελευταία τεχνολογία ονομάζεται HTML5 Websockets και λειτουργεί μέσα από ένα και μόνο socket και εμφανίζει τα αποτελέσματα της μέσω μιας διεπαφής JavaScript σε όλους τους HTML 5 συμβατούς browsers.[\[14\]](#) Με την χρήση αυτής της τεχνολογίας ο χρήστης μπορεί να ανοίξει μία αμφίδρομη σύνδεση με το διακομιστή. Το πρωτόκολλο WebSocket τυποποιήθηκε από το IETF ως RFC 6455 το 2011, και το WebSocket API στο Web IDL έχει τυποποιηθεί από το W3C.[\[15\]](#)

Τα WebSocket είναι σχεδιασμένα έτσι ώστε να μπορούν να υλοποιηθούν σε προγράμματα περιήγησης (web browsers) και διακομιστές, έχουν όμως την δυνατότητα να χρησιμοποιηθούν σε οποιαδήποτε εφαρμογή πελάτη ή εφαρμογή διακομιστή. Το πρωτόκολλο WebSocket είναι ένα ανεξάρτητο πρωτόκολλο, το οποίο βασίζεται στο TCP. Η μόνη του συσχέτιση με το HTTP είναι ότι η χειραψία του ερμηνεύεται από HTTP διακομιστές ως αίτημα αναβάθμισης.

Τα WebSockets δημιουργούν μία TCP σύνδεση με τον διακομιστή, την οποία κρατούν ανοιχτή για όσο χρονικό διάστημα χρειάζεται. Ο διακομιστής ή ο πελάτης μπορούν εύκολα να καταργήσουν την σύνδεση. Το κύριο χαρακτηριστικό αυτής της τεχνολογίας είναι ότι προσφέρει αμφίδρομη επικοινωνία. Έτσι καθιστά δυνατή, μεγαλύτερη αλληλεπίδραση μεταξύ ενός

προγράμματος περιήγησης και μια ιστοσελίδα, δίνοντας τη δυνατότητα στον διακομιστή, αλλά και στον πελάτη να ανταλλάσσουν δεδομένα και προς τις δύο κατευθύνσεις, οποιαδήποτε χρονική στιγμή. Είναι πολύ αποτελεσματική για εφαρμογές που απαιτούν συχνά μηνύματα, όπως η μετάδοση live περιεχόμενου και οι εφαρμογές πραγματικού χρόνου όπως τα παιχνίδια. Αυτό το επιτυγχάνει με την παροχή ενός τυποποιημένου τρόπου στο διακομιστή, έτσι ώστε να αποστέλλει περιεχόμενο στο browser χωρίς να έχει ζητηθεί από τον πελάτη, και να επιτρέπει στα μηνύματα να διακινούνται κρατώντας την σύνδεση ανοιχτή. Με τον τρόπο αυτό, μια αμφίδρομη συζήτηση μπορεί να λάβει χώρα μεταξύ ενός browser και ενός διακομιστή.



Εικόνα 10 Περιγραφή λειτουργίας της τεχνολογίας WebSockets

Ακόμη, τα WebSockets προσφέρουν διαμόρφωση των δεδομένων, η οποία περιλαμβάνει συγκάλυψη για κάθε μήνυμα που αποστέλλεται από τον πελάτη στον διακομιστή. Με αυτό τον τρόπο, τα στοιχεία που αποστέλλονται είναι απλά κρυπτογραφημένα.

2.2.5 Σύγκριση τεχνολογιών επικοινωνίας πραγματικού χρόνου

Σε αυτό το υποκεφάλαιο, θα συγκρίνουμε τις τεχνολογίες που περιγράψαμε παραπάνω και θα επιλέξουμε πάνω σε ποια τεχνική θα στηριχτεί η δική μας εφαρμογή.

Αρχικά, θα επισημάνουμε τη διαφορά μεταξύ της τεχνολογίας Ajax rolling και της Ajax long rolling. Στην πρώτη, ο πελάτης δημιουργεί μία σύνδεση με το διακομιστή για κάθε αίτημα, στέλνει το αίτημα και παίρνει την απάντηση από το διακομιστή. Έπειτα, η σύνδεση τερματίζεται. Είναι μια απλή διαδικασία αιτήματος - απάντησης για κάθε Ajax κλήση. Στη δεύτερη, ο πελάτης δημιουργεί τη σύνδεση με τον διακομιστή, αλλά κρατάει ανοιχτή τη σύνδεση για κάποιο χρονικό διάστημα. Στη διάρκεια της ανοιχτής αυτής σύνδεσης, ο πελάτης μπορεί να λαμβάνει δεδομένα από το διακομιστή. Ο πελάτης θα πρέπει να επανασυνδέεται περιοδικά, όταν η σύνδεση έχει τερματιστεί λόγω των timeouts.

Οι τεχνολογίες Server Sent Events και WebSockets είναι και οι δύο ικανές να αποστείλουν δεδομένα στους browsers, παρόλα αυτά δεν είναι ανταγωνιστικές τεχνολογίες. Οι WebSockets συνδέσεις μπορούν και να στέλνουν δεδομένα στον browser, αλλά και να λαμβάνουν δεδομένα από αυτόν. Ένα καλό παράδειγμα μιας τέτοιας εφαρμογής είναι μια εφαρμογή chat. Οι SSE συνδέσεις, από την άλλη, μπορούν μόνο να στείλουν δεδομένα στον browser. Πρακτικά, αφού όλα όσα μπορούν να γίνουν με την SSE, μπορούν να γίνουν και με την WebSockets, η WebSockets αποκτά όλο και πιο πολλή προσοχή και πολλοί browsers στηρίζουν την WebSockets από την SSE.[\[16\]](#)[\[17\]](#)

Ένας λόγος που η τεχνολογία SSE δεν είναι τόσο διαδεδομένη είναι γιατί μεταγενέστερα APIs, όπως WebSockets, παρέχουν ένα πιο πλήρες πρωτόκολλο για την εκτέλεση αμφίδρομης, full-duplex επικοινωνίας. Η ύπαρξη ενός αμφίδρομου δίαυλου επικοινωνίας είναι πιο ελκυστική για υπηρεσίες όπως παιχνίδια, εφαρμογές ανταλλαγής μηνυμάτων, καθώς και για τις περιπτώσεις όπου είναι απαραίτητες ενημερώσεις σε πραγματικό χρόνο

και προς τις δύο κατευθύνσεις. Ωστόσο, σε ορισμένες περιπτώσεις εφαρμογών δεν είναι απαραίτητη η αποστολή δεδομένων από τον πελάτη. Χρειάζονται απλά ενημερώσεις για κάποιες ενέργειες που πραγματοποιεί ο διακομιστής.

Παρακάτω παρουσιάζουμε συγκεντρωτικά τις διαφορές ανάμεσα στις τεχνολογίες Ajax Long Polling, Server-Sent Events, Websockets.[\[18\]](#)

	Long-polling	Server-Sent Events	WebSockets
Υποστήριξη από Browser	Υποστηρίζεται από τους περισσότερους browsers που χρησιμοποιούνται σήμερα	Υποστηρίζεται από τους Chrome 9+, Firefox 6+, Opera 11+, Safari 5+	Το πιο πρόσφατο hybi-10 πρωτόκολλο υποστηρίζεται από τον Chrome 14 και τον Firefox 7, ενώ το hybi-07 υποστηρίζεται από τον Firefox 6
Server-loading	Καταλαμβάνει λίγους πόρους της CPU, αλλά δημιουργεί αδράνεις διαδικασίες ανά χρήστη, δαπανώντας έτσι μνήμη στο διακομιστή	Δουλεύει κατά ένα ποσοστό σαν long polling, εκτός και αν το SSE απαιτεί να μην κλείνει η σύνδεση κάθε φορά που μια απάντηση αποστέλλεται	Η καλύτερη δυνατή λύση. Ο server έχει τη μοναδική μέθοδο που εξυπηρετεί οποιοδήποτε αίτημα. Χωρίς βρόχους επανάληψης, δαπάνη μνήμης / CPU ανά πελάτη, αλλά και ανά συναλλαγή του πελάτη
Client-loading	Εξαρτάται από την υλοποίηση, ωστόσο πάντα είναι ασύγχρονη διαδικασία.	Υλοποιείται εγγενώς στον browser, καταλαμβάνει ελάχιστους πόρους	Υλοποιείται εγγενώς στον browser, καταλαμβάνει ελάχιστους πόρους
Χρονικό πλαίσιο	Σχεδόν σε πραγματικό χρόνο. Υπάρχει ένα κενό διάστημα μεταξύ του αιτήματος και της απάντησης που μπορεί να προσθέσει μία μικρή καθυστέρηση στην εκτέλεση του επόμενου αιτήματος.	Υπάρχει μία προκαθορισμένη καθυστέρηση 3 δευτερόλεπτα, μπορεί όμως να μεταβληθεί	Σε πραγματικό χρόνο
Πολυπλοκότητα υλοποίησης	Υλοποιείται πολύ εύκολα	Υλοποιείται ακόμη ευκολότερα	Απαιτεί EventMachine διακομιστή και μία ειδική θύρα να είναι ανοιχτή.

Εικόνα 11 Διαφορές Ajax long polling, HTTP SSE και HTTP WebSocket

Τα SSE αποστέλλονται μέσω του απλού HTTP. Αυτό σημαίνει πως δεν απαιτούν κάποιο ειδικό πρωτόκολλο ή υλοποίηση του εξυπηρετητή για να λειτουργήσουν. Τα WebSockets αντίθετα, απαιτούν full-duplex συνδέσεις και νέους Web Socket διακομιστές για να χειριστούν το πρωτόκολλο. Επιπλέον, τα SSE έχουν μια ποικιλία χαρακτηριστικών τα οποία λείπουν από τα

WebSockets εκ σχεδιασμού, όπως η αυτόματη επανασύνδεση, αναγνωριστικά συμβάντων (event IDs), καθώς και τη δυνατότητα για αποστολή αυθαίρετων γεγονότων.

Σύμφωνα με τα παραπάνω, λοιπόν, και ειδικά λόγω της μεταφοράς με απλό HTTP αντί ενός προσαρμοσμένου πρωτοκόλλου και της ενσωματωμένης υποστήριξης για επανασύνδεση και event-id, επιλέξαμε η εφαρμογή μας να βασιστεί πάνω στην τεχνική της SSE τεχνολογίας.

2.3 Εφαρμογές αυτόματης ανάρτησης

Το διαδίκτυο στις μέρες μας παρέχει πολλές εφαρμογές μέσω των οποίων οι χρήστες μπορούν να ενημερώνονται αυτόματα, χωρίς να χρειάζεται να ακολουθήσουν ή να κάνουν κάποια διαδικασία. Σε αυτό το υποκεφάλαιο, θα αναφέρουμε μερικές από αυτές και θα δούμε πώς δουλεύουν, καθώς και μερικά πλεονεκτήματα ή μειονεκτήματά τους.

2.3.1 Mail

Πολλές ιστοσελίδες δίνουν τη δυνατότητα στους χρήστες τους να εγγραφούν σε ειδοποιήσεις μέσω ηλεκτρονικών μηνυμάτων (email newsletter). Είναι μια καλή τακτική, έτσι ώστε ο χρήστης να θυμάται την ιστοσελίδα και να μπορεί να ενημερώνεται για τυχόν αλλαγές. Ένας χρήστης μπορεί να κάνει subscribe σε μια ιστοσελίδα και αυτό πρακτικά σημαίνει ότι θα λαμβάνει ένα ηλεκτρονικό μήνυμα κάθε φορά που δημοσιεύεται ένα νέο άρθρο ή αν μια ιστοσελίδα παρέχει πληροφορίες για πολλά θέματα, μπορεί ο χρήστης να διαλέξει τα θέματα που τον ενδιαφέρουν και θα ενημερώνεται για αυτά μέσω ενός email.

Το μοντέλο που μπορεί να χρησιμοποιεί αυτός ο τρόπος αυτόματης ενημέρωσης είναι είτε push messaging είτε pull messaging. Η διαφορά τους έγκειται στο πώς ανακτούν το μήνυμα. Στο push μοντέλο, τα νέα μηνύματα παραδίδονται αυτόματα στον πελάτη από τον διακομιστή, ενώ στο pull μοντέλο τα νέα μηνύματα παραδίδονται μόνο όταν το αιτηθεί ο πελάτης.[\[19\]](#)

2.3.1.1 *Μειονεκτήματα email newsletter*

Αυτός ο τρόπος αυτόματης ενημέρωσης, όμως έχει μειονεκτήματα. Αρχικά, όταν ένας χρήστης έχει κάνει πολλά subscribe, λαμβάνει πολλά ηλεκτρονικά μηνύματα με αποτέλεσμα να μην είναι καλά οργανωμένα και διακριτά. Υπάρχει η πιθανότητα, λοιπόν, να διαγράψει κατά λάθος κάποιο email. Μπορεί, επίσης, μερικά email να θεωρηθούν ως spam (κακόβουλα ηλεκτρονικά μηνύματα) με αποτέλεσμα ο χρήστης να μην τα δει.[\[20\]](#)

2.3.2 *RSS Feed*

Το RSS (Really Simple Syndication -*Πολύ Απλή Διανομή*) είναι ένας τρόπος να εγγραφεί κανείς ως “συνδρομητής” σε μία πηγή πληροφόρησης, όπως για παράδειγμα σε κάποιον ιστότοπο, ώστε να του παραδίδονται σύντομες ενημερώσεις. Οι πηγές αυτές ονομάζονται feeds. Όταν λοιπόν κάποιος γίνει συνδρομητής σε μία σελίδα, θα λαμβάνει ένα feed, δηλαδή μια σειρά από τίτλους ακολοθούμενους από σύντομες περιγραφές, από όλα τα άρθρα που δημοσιεύτηκαν στην συγκεκριμένη σελίδα. Με αυτόν τον τρόπο, δίνεται η δυνατότητα στο χρήστη - συνδρομητή να διατρέχει τα άρθρα μίας σελίδας με μεγαλύτερη αποτελεσματικότητα.[\[21\]](#)

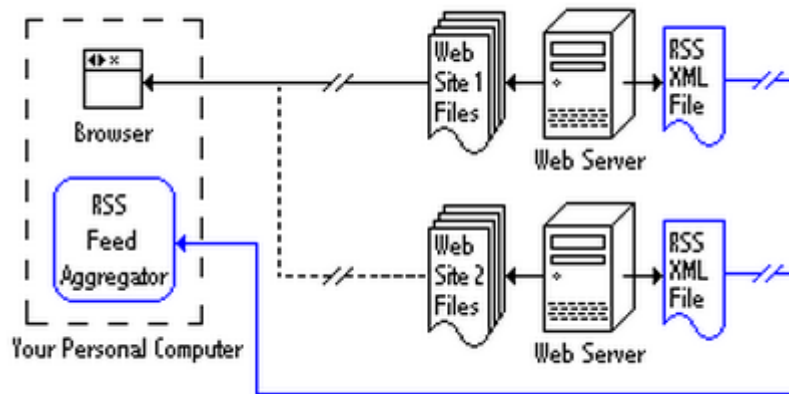
Οι περισσότεροι άνθρωποι ενδιαφέρονται για πολλές ιστοσελίδες, το περιεχόμενο των οποίων αλλάζει απρόβλεπτα. Παραδείγματα τέτοιων ιστοσελίδων είναι σελίδες με ειδήσεις, σελίδες ενημέρωσης της κοινότητας και θρησκευτικών οργανώσεων, σελίδες με πληροφορίες για διάφορα προϊόντα, ιατρικές σελίδες και ιστολόγια. Ο επαναλαμβανόμενος έλεγχος στη κάθε σελίδα για να δούνε αν έχει αλλάξει το περιεχόμενο της μπορεί να καταστήσει πολύ κουραστικός.

Το RSS είναι ο καλύτερος τρόπος για να ειδοποιείται κανείς για νέα ή αλλαγή περιεχομένου. Οι κοινοποιήσεις των αλλαγών σε πολλαπλές ιστοσελίδες

αντιμετωπίζονται εύκολα και τα αποτελέσματα παρουσιάζονται καλά οργανωμένα και πιο διακριτά από ότι σε ένα email.

Το RSS λειτουργεί έχοντας τον συγγραφέα της ιστοσελίδας να διατηρεί έναν κατάλογο των κοινοποιήσεων της ιστοσελίδας του με έναν τυποποιημένο τρόπο. Αυτή η λίστα των ειδοποιήσεων ονομάζεται "RSS Feed". Οποιοσδήποτε επιθυμεί να ανακαλύψει τους τελευταίους τίτλους ή ενημερώσεις μπορεί να ελέγξει αυτή τη λίστα. Έχουν μάλιστα αναπτυχθεί ειδικά προγράμματα, που λέγονται "RSS aggregators", τα οποία με αυτοματοποιημένο τρόπο έχουν πρόσβαση στα RSS feeds ιστοσελίδων για της οποίες ενδιαφέρεται ο χρήστης τους, και οργανώνουν τα αποτελέσματα των ενημερώσεων κάθε σελίδας αντί για αυτόν. Συχνά τα προγράμματα αυτά καλούνται και "RSS Channels" ή "RSS Readers".

Η παραγωγή ενός RSS feed είναι ιδιαίτερα απλή, και όλο και περισσότερες ιστοσελίδες παρέχουν την δυνατότητα αυτή, συμπεριλαμβανομένων φυσικά και των ιστοτόπων μεγάλων εφημερίδων όπως η New York Times και το BBC, καθώς επίσης και πολλών άλλων ειδησεογραφικών ιστολογίων (blogs). Το RSS παρέχει πολύ βασικές πληροφορίες για να δημιουργήσει τις ειδοποιήσεις του. Αποτελείται από μια λίστα των στοιχείων που παρουσιάζονται κατά σειρά από το νεότερο στο παλαιότερο. Κάθε στοιχείο αποτελείται συνήθως από ένα απλό τίτλο που περιγράφει το στοιχείο σε συνδυασμό με μια πιο ολοκληρωμένη περιγραφή και ένα σύνδεσμο προς την ιστοσελίδα όπου το πραγματικό άρθρο περιγράφεται. Ορισμένες φορές αυτή η περιγραφή είναι το πλήρες άρθρο (όπως βρίσκεται σε ένα blog), και μερικές φορές είναι απλά μια περίληψη.



Εικόνα 12 Σύνδεση μεταξύ RSS feed, XML αρχείων και του προσωπικού μας υπολογιστή

Εκτός από το να ειδοποιεί τον χρήστη για νέους τίτλους ειδήσεων και αλλαγές στις ιστοσελίδες, το RSS μπορεί να χρησιμοποιηθεί και για άλλους σκοπούς.

Μερικές γνωστές χρήσεις είναι οι ακόλουθες:

- Ειδοποίηση άφιξης νέων προϊόντων σε ένα κατάστημα
- Απογραφή και ειδοποίηση για θέματα που αφορούν newsletter, περιλαμβανομένων και των email newsletters
- Καιρικές συνθήκες και άλλες καταχωρήσεις μεταβαλλόμενων συνθηκών
- Ειδοποίηση για τις προσθήκες των νέων στοιχείων σε μια βάση δεδομένων, ή τα νέα μέλη σε μια ομάδα

Ένας RSS aggregator είναι το μόνο που χρειάζεται κανείς για να διαβάσει όλα τα RSS feeds, είτε αυτά θα είναι απλά τίτλοι των άρθρων, είτε ειδοποιήσεις και

ενημερώσεις. Η τεχνολογία του RSS εξελίσσεται σε έναν πολύ δημοφιλή και χρήσιμο τρόπο επικοινωνίας.[\[22\]](#)

2.3.2.1 Πλεονεκτήματα RSS

Τα πλεονεκτήματα του να δημιουργεί κανείς RSS Feeds είναι πολύ σημαντικά. Αρχικά, φέρνει περισσότερη κίνηση σε μία ιστοσελίδα. Αντί να βασιζόμαστε στο αν ο επισκέπτης θα προσθέσει μία ιστοσελίδα στους σελιδοδείκτες του για να την επισκεφτεί αργότερα, του δίνεται η δυνατότητα να προσθέσει το rss feed της ιστοσελίδας στον rss aggregator, ή στον browser του έτσι ώστε κάθε νέα ενημέρωση να λαμβάνεται από τον rss reader και ο χρήστης να μπορεί να δει τις νέες αναρτήσεις της σελίδας κάθε φορά που είναι διαθέσιμες. Επιπλέον, το RSS μπορεί να χρησιμοποιηθεί ως εναλλακτική λύση των newsletters emails. Όταν ένα newsletter email αποστέλλεται, περισσότεροι από 40-50% των παραληπτών δεν θα το λάβουν καν είτε λόγω των φίλτρων ηλεκτρονικής αλληλογραφίας για spam filters είτε επειδή χάνεται ανάμεσα σε όλα τα άχρηστα emails που λαμβάνουν οι συνδρομητές του newsletter. Με τα RSS feeds, δεν υπάρχει λόγος ανησυχίας, καθώς δεν αποστέλλεται κάποιο email, αλλά ένα προς συλλογή feed προς όλους τους rss readers.[\[23\]](#)

2.3.3 Atom

Το Atom είναι μια μορφή αρχείου που βασίζεται στη XML και χρησιμοποιείται για να ενώνει περιεχομένο. Το Atom σχεδιάστηκε για να είναι ένα παγκόσμιο πρότυπο εκδόσεων για τα blogs και άλλες ιστοσελίδες, όπου το περιεχόμενο ενημερώνεται συχνά. Οι χρήστες που επισκέπτονται μια ιστοσελίδα που περιέχει τροφοδοσία Atom, μπορούν να βρουν ένα αρχείο που περιγράφεται ως "atom.xml" στη διεύθυνση URL, το οποίο μπορεί να αντιγραφεί και να επικολληθεί σε ένα aggregator, έτσι ώστε να μπορεί να εγγραφεί ο χρήστης στην ροή (feed).[\[24\]](#)

Οι μεταδόσεις ιστού επιτρέπουν σε προγράμματα λογισμικού να ελέγχουν για ενημερώσεις που δημοσιεύονται σε μια ιστοσελίδα. Για να παρεχθεί μια μετάδοση ιστού, ένας ιδιοκτήτης ιστοσελίδας μπορεί να χρησιμοποιήσει ένα εξειδικευμένο λογισμικό (όπως ένα σύστημα διαχείρισης περιεχομένου) το οποίο δημοσιεύει μια λίστα (ή ροές) από τα πιο πρόσφατα άρθρα ή το περιεχόμενο σε μια τυποποιημένη και αναγνώσιμη-από-μηχανή μορφή. Η ροή μπορεί στη συνέχεια να ληφθεί από τα προγράμματα που τη χρησιμοποιούν, όπως ιστοσελίδες που ενώνουν περιεχόμενο από τη ροή ή από feed reader προγράμματα που επιτρέπουν στους χρήστες του διαδικτύου να εγγραφούν σε ροές και να δουν το περιεχόμενό τους.

Μια ροή περιέχει καταχωρήσεις, οι οποίες μπορεί να είναι πρωτοσέλιδα, άρθρα, αποσπάσματα, περιλήψεις, ή / και σύνδεσμοι για ένα περιεχόμενο σε ένα δικτυακό τόπο, μαζί με διάφορα μεταδεδομένα.[\[25\]](#)

2.3.3.1 Σύγκριση Atom και RSS

Η τυποποίηση Atom αναπτύχθηκε ως μία εναλλακτική για την τεχνολογία RSS. Ο Ben Trott, υπέρμαχος του Atom, πίστευε ότι το RSS είχε περιορισμούς και αδυναμίες, όπως την έλλειψη της εν εξελίξει καινοτομίας και την αναγκαιότητα του να παραμείνει συμβατό προς τα πίσω, και ότι υπήρχαν πλεονεκτήματα στο νέο σχεδιασμό.

Το Atom διαφέρει από το RSS σε πολλά σημεία:

- Το Atom είναι ένα ανοιχτό πρότυπο.
- Το Atom βρίσκεται μέσα σε ένα XML-namespace.
- Το Atom περιλαμβάνει "autodiscovery," επιτρέποντας στους feed aggregators να ανιχνεύουν αυτόματα την παρουσία μιας ροής.
- Το Atom έχει ξεχωριστά στοιχεία "περίληψης" και "περιεχομένου".

- Κάθε καταχώρηση έχει ένα παγκόσμιο μοναδικό αναγνωριστικό.

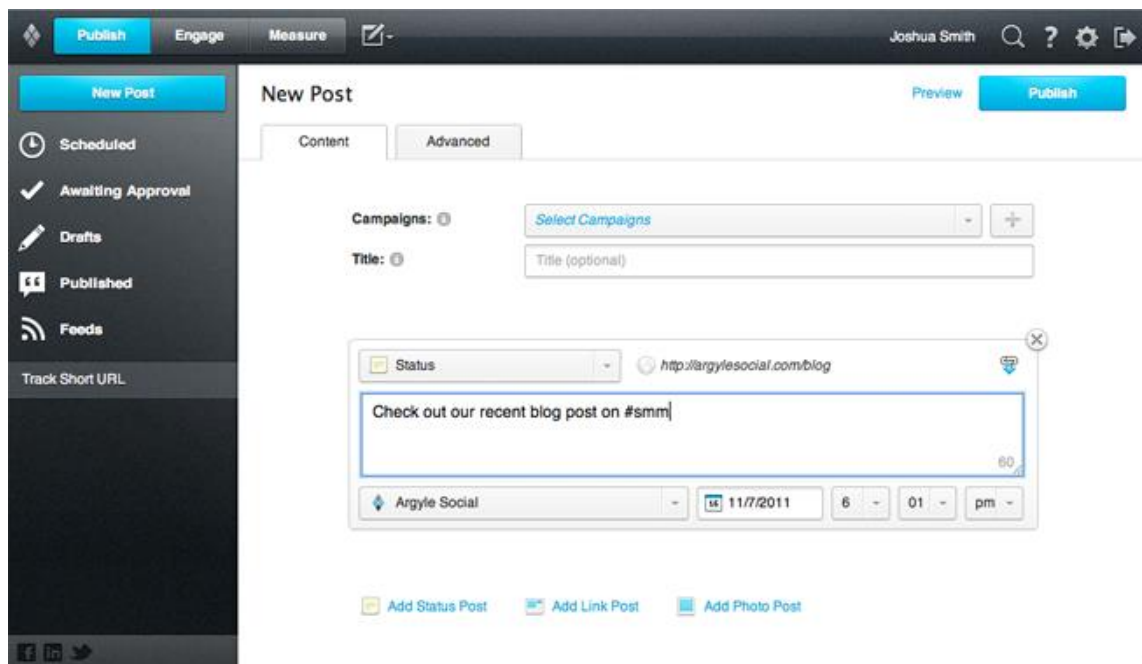
Στο κεφάλαιο αυτό περιγράψαμε τις τεχνολογίες push και pull που χρησιμοποιούνται στο διαδίκτυο για ανάκτηση και ανάρτηση περιεχομένου, εντοπίσαμε τα πλεονεκτήματα και τα μειονεκτήματα τους, καθώς και τις διαφορές τους. Στην συνέχεια, ασχοληθήκαμε με τις τεχνολογίες που είναι κατάλληλες για την ανάπτυξη εφαρμογών επικοινωνίας που εκτελούνται σε πραγματικό χρόνο, αφού και η εφαρμογή που θα αναπτύξουμε ανήκει στην κατηγορία αυτή. Περιγράψαμε την λειτουργία των τεχνολογιών AJAX Polling, AJAX Long-Polling, HTML5 Server Sent Events, HTML5 Websockets και έπειτα κάναμε μια σύγκριση ανάμεσα στις τεχνολογίες αυτές. Μέσα από αυτή τη σύγκριση καταλήξαμε στο συμπέρασμα, πως η τεχνολογία Server Sent Events(SSE) είναι η καταλληλότερη για την ανάπτυξη της εφαρμογής μας. Τέλος, αναφερθήκαμε στο email newsletter και τις τεχνολογίες RSS και Atom, υπηρεσίες και εφαρμογές που διατίθενται για αυτόματη ανάρτηση περιεχομένου, και τις συγκρίναμε.

3 Μελέτη, αξιολόγηση υφιστάμενων λύσεων (υπηρεσιών) που έχουν αναπτυχθεί

Στο κεφάλαιο αυτό θα αναφερθούμε σε κάποιες ήδη υπάρχουσες εφαρμογές, οι οποίες μπορούν να θεωρηθούν υφιστάμενες λύσεις στο πρόβλημα που θέλουμε να επιλύσουμε και θα τις αξιολογήσουμε σύμφωνα με την λειτουργικότητα αλλά και την επεκτασιμότητα τους.

Τα εργαλεία που κρίναμε ότι θα μπορούσαν να είναι λύσεις του προβλήματος μας είναι τα ακόλουθα:

3.1 Argyle Social

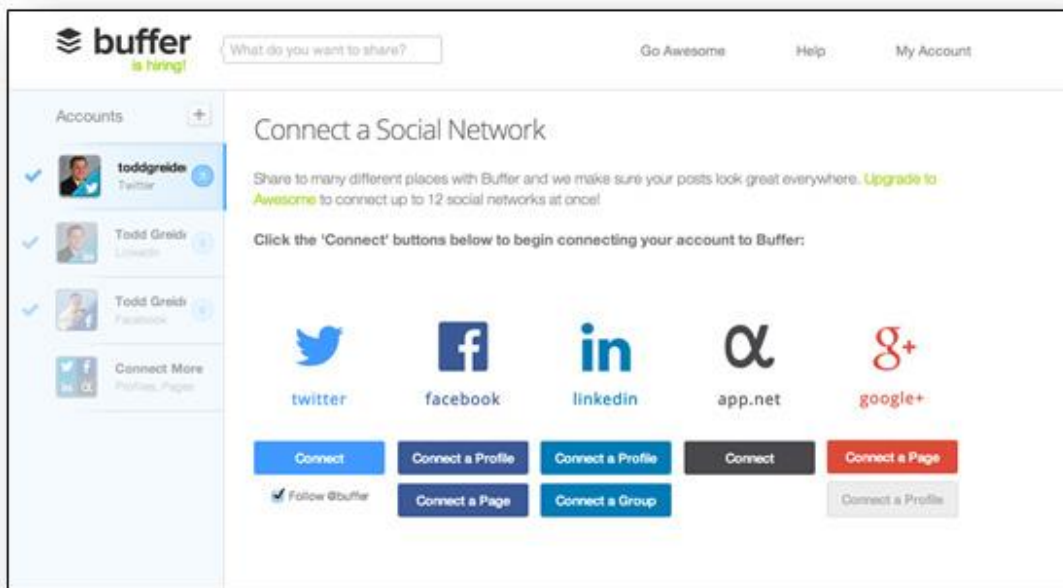


Εικόνα 13 Αρχική σελίδα Argyle Social

Το εργαλείο αυτό επιτρέπει στο χρήστη να δρομολογήσει και να δημοσιεύσει τις ενημερώσεις που επιθυμεί σε κορυφαία κοινωνικά δίκτυα όπως το

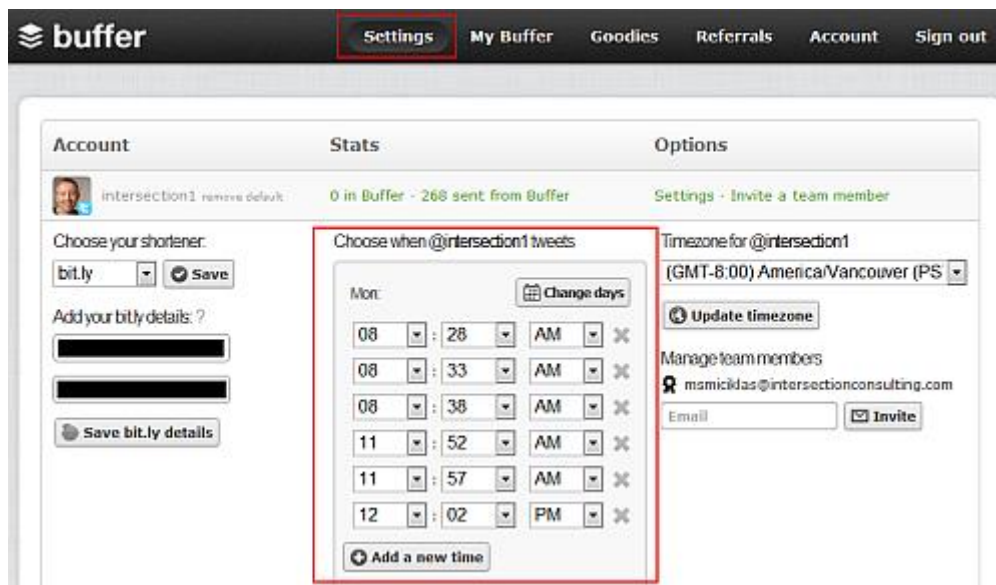
Facebook, το Twitter και το LinkedIn. Μπορεί, επίσης, ο χρήστης να δημοσιεύσει αυτόματα το τελευταίο κείμενο του blog του χρησιμοποιώντας το RSS. Το Argyle Social περιλαμβάνει πλήθος επιπρόσθετων χαρακτηριστικών για επιχειρήσεις, περιλαμβανομένων μεθόδων λογικής ανάλυσης, κοινωνικών διαστάσεων ROI και επιλογές συνεργασίας. Το εργαλείο αυτό διατίθεται επί πληρωμή.[\[26\]](#)

3.2 Buffer



Εικόνα 14 Παράθυρο διαλόγου Buffer

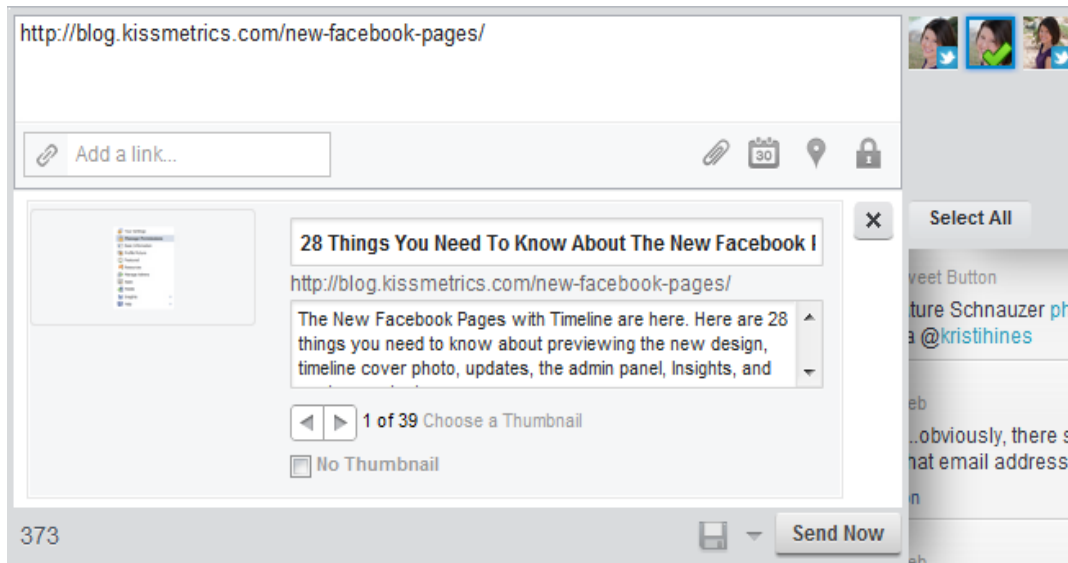
Επιτρέπει στο χρήστη να προγραμματίσει και να δημοσιεύσει ενημερώσεις. Σε αντίθεση όμως με άλλες πλατφόρμες που πρέπει ο χρήστης να προγραμματίσει κάθε ενημέρωση να δημοσιευτεί σε ένα συγκεκριμένο χρόνο, το Buffer επιτρέπει να ορίσει ο χρήστης ένα συγκεκριμένο χρονοδιάγραμμα. Όταν προστεθεί μία νέα ενημέρωση στο Buffer, θα δημοσιευτεί στην επόμενη διαθέσιμη μονάδα χρόνου.



Εικόνα 15 Χρονοδιάγραμμα Buffer

Οι ενημερώσεις μπορούν να στέλνονται στους λογαριασμούς του Facebook, του Twitter και του LinkedIn. Η χρήση του Buffer είναι δωρεάν για μέχρι και 10 προγραμματισμένες ενημερώσεις τη φορά. Για δρομολόγηση παραπάνω ενημερώσεων είναι διαθέσιμο επί πληρωμή. [\[26\]\[27\]](#)

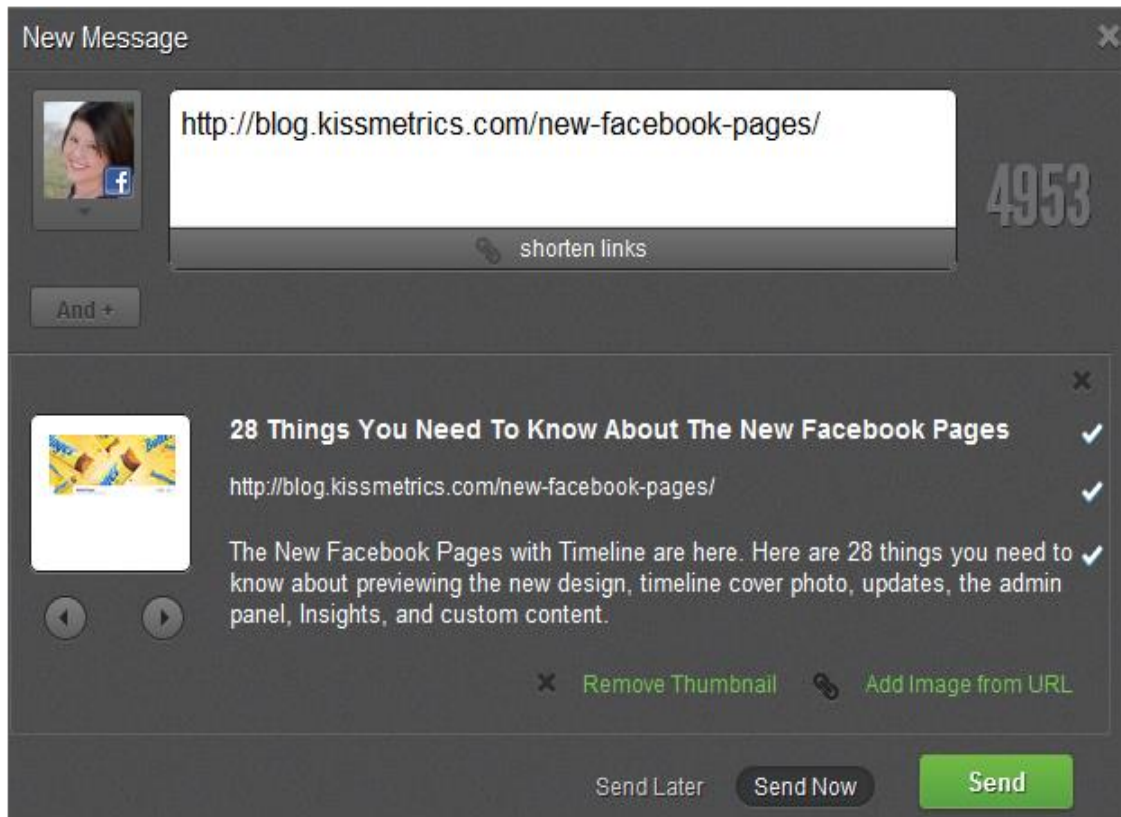
3.3 HootSuite



Εικόνα 16 Παράθυρο διαλόγου HootSuite

Το εργαλείο HootSuite δίνει τη δυνατότητα στο χρήστη να δρομολογήσει και να δημοσιεύσει ενημερώσεις σε πολλά κοινωνικά δίκτυα συμπεριλαμβανομένου και του Facebook. Επίσης, επιτρέπει στο χρήστη να προσθέσει RSS feeds στο λογαριασμό του για να ενημερώνει τα κοινωνικά δίκτυα όταν νέα άρθρα είναι διαθέσιμα στο blog, καθώς και σε όποια άλλα blogs επιθυμεί. Είναι δωρεάν για χρήση σε μέχρι και 5 προφίλ. [26][28]

3.4 Sprout Social



Εικόνα 17 Παράθυρο διαλόγου Sprout Social

Το Sprout Social δίνει τη δυνατότητα στο χρήστη να δρομολογήσει και να δημοσιεύσει ενημερώσεις στους Facebook, Twitter, και LinkedIn λογαριασμούς του. Ακόμη, ο χρήστης μπορεί να συνδέσει RSS feeds στο λογαριασμό του που θα ενημερώνουν τακτικά τα κοινωνικά δίκτυα όταν νέο περιεχόμενο θα είναι διαθέσιμο. Επίσης, είναι διαθέσιμη η δυνατότητα για παρακολούθηση της επιρροής των δημοσιεύσεων στα κοινωνικά δίκτυα. Το εργαλείο αυτό δεν διατίθεται για δωρεάν χρήση.[\[26\]](#)

3.5 Graph API

Και τα τέσσερα αυτά εργαλεία, για να αντλήσουν αλλά και να δημοσιεύσουν νέες ενημερώσεις σε ένα προφίλ ή μια σελίδα στο Facebook χρησιμοποιούν το Graph API του Facebook.

Το Graph API είναι ο κύριος τρόπος για εξαγωγή και εισαγωγή δεδομένων από και προς το social graph του Facebook. Είναι ένα χαμηλού επιπέδου (low-level) βασισμένο σε HTTP API το οποίο μπορεί να χρησιμοποιηθεί σε δεδομένα ερωτημάτων (query data), δημοσίευση νέων ενημερώσεων, ανέβασμα φωτογραφιών και πολλές άλλες λειτουργίες που μία εφαρμογή (app) χρειάζεται. [\[29\]](#)

3.5.1 Βασικά Στοιχεία

Το Graph API έχει πάρει το όνομα του από την ιδέα ενός 'social graph' - μιας αναπαράστασης της πληροφορίας από την οποία αποτελείται το Facebook :

- nodes (βασικά "αντικείμενα" όπως ένας Χρήστης (User), μια Φωτογραφία (Photo), μία Σελίδα (Page), ένα Σχόλιο (Comment))
- edges (οι συνδέσεις μεταξύ αυτών των "αντικειμένων", όπως μία Φωτογραφία μιας Σελίδας ή τα Σχόλια μίας Φωτογραφίας)
- fields (πληροφορίες σχετικά με αυτά τα "αντικείμενα", όπως για παράδειγμα η ημερομηνία γέννησης ενός Χρήστη ή το όνομα μίας Σελίδας).

Το Graph API βασίζεται στο πρωτόκολλο HTTP, επομένως συνεργάζεται με οποιαδήποτε γλώσσα προγραμματισμού έχει μία HTTP βιβλιοθήκη, όπως η cURL και η urllib. [\[29\]](#)

Τα περισσότερα από τα αιτήματα του Graph API απαιτούν τη χρήση access tokens τα οποία μπορεί να παράγει μία εφαρμογή (app) με την εκτέλεση της Facebook Login.

3.5.2 Ανάκτηση πληροφορίας

Μπορεί να γίνει ανάκτηση πληροφορίας από το API κάνοντας HTTP GET αιτήματα σε κόμβους ή συνδέσεις μεταξύ των κόμβων αυτών.

Κάθε κόμβος έχει ένα μοναδικό ID το οποίο χρησιμοποιείται για την πρόσβαση του μέσω του Graph API.

Οι περισσότερες κλήσεις στο API πρέπει να υπογράφονται με ένα διακριτικό πρόσβασης (access token). Το ποιες άδειες πρόσβασης είναι απαραίτητες, εξαρτάται από τον κόμβο ή τη σύνδεση στην οποία θα γίνει προσπέλαση.

Ο κόμβος /me είναι ένα ειδικό endpoint το οποίο μεταφράζει τη μεταβλητή user_id του χρήστη (ή τη μεταβλητή page_id της Facebook σελίδας) της οποίας το access token χρησιμοποιείται για να γίνουν οι κλήσεις στο API. Έχοντας το access token ενός χρήστη, μπορούν να αντληθούν πολλές πληροφορίες σχετικά με αυτόν.

Σχεδόν όλα τα αιτήματα στέλνονται στο API στη διεύθυνση graph.facebook.com - η μοναδική εξαίρεση είναι οι μεταφορτώσεις βίντεο, οι οποίες χρησιμοποιούν την graph-video.facebook.com.

3.5.3 Δημοσίευση πληροφορίας

Η δημοσίευση πληροφορίας μέσω του Graph API πραγματοποιείται μέσω HTTP POST αιτημάτων στη σχετική σύνδεση (edge), που συμπεριλαμβάνει τις απαιτούμενες παραμέτρους, προς τον επιθυμητό κόμβο (node).

Οι εφαρμογές (Apps) μπορούν να δημιουργήσουν νέες δημοσιεύσεις (status updates) εκ μέρους κάποιου προφίλ χρήστη ή κάποιας Facebook Σελίδας. (Χρήση /{user-id}/feed για χρήστες του Facebook και χρήση /{page-id}/feed για σελίδες του Facebook.)

Όλες οι κλήσεις για δημοσίευση πληροφορίας στο API πρέπει να υπογράφονται με ένα διακριτικό πρόσβασης (access token).

3.6 Αξιολόγηση υφιστάμενων λύσεων

Οι υφιστάμενες λύσεις που παραθέσαμε, χρησιμοποιούν το Graph API και στηρίζονται στις τεχνολογίες push και pull για να ενημερώνουν αυτόματα το λογαριασμό ενός χρήστη ή μιας σελίδας, επιτρέποντας στο χρήστη να προγραμματίσει ενημερώσεις. Ο χρήστης έχει τη δυνατότητα να εισάγει δεδομένα προς δημοσίευση και μέσω των εφαρμογών, οι ενημερώσεις να δημοσιεύονται σε όποια από τα δίκτυα που οι εφαρμογές υποστηρίζουν. Όμως πέρα από αυτή την λειτουργία, οι ίδιες οι εφαρμογές μπορούν να τραβήξουν δεδομένα μέσω RSS και να τα δημοσιεύσουν. Η Argyle Social και η Sprout Social είναι διαθέσιμες μόνο για το Facebook, το Twitter και το LinkedIn, σε αντίθεση με τη Buffer και τη HootSuite, οι οποίες καλύπτουν μεγαλύτερο φάσμα κοινωνικών δικτύων. Πρέπει να αναφέρουμε επίσης, ότι η εταιρία Argyle δυστυχώς κλείνει οπότε σταματάει και η υποστήριξη της εφαρμογής της. Η Buffer είναι η μόνη εφαρμογή, η οποία επιτρέπει στο χρήστη να ορίσει πότε θα δημοσιευτεί το περιεχόμενο που επιθυμεί μέσω

ενός χρονοδιαγράμματος.[\[30\]](#) Γενικά, όλες οι εφαρμογές που εξετάσαμε είναι επί πληρωμή. Δεν είναι ανοιχτού κώδικα και δεν παρέχουν στο χρήστη τη δυνατότητα επέκτασης, επομένως δεν θεωρούνται επεκτάσιμες.

Το Graph API είναι ο κυρίαρχος τρόπος για την εισαγωγή και την εξαγωγή δεδομένων από το social graph του facebook και είναι η βάση για κάθε Facebook εφαρμογή. Δίνει ελευθερία στον προγραμματιστή να χειριστεί όπως θέλει ο ίδιος, σχεδόν όλες τις οντότητες που υπάρχουν στο Facebook. Χρησιμοποιείται κατά κόρον για την δημοσίευση πληροφορίας, ενώ πιο σύνθετες εφαρμογές, οι οποίες αναπαριστούν αντικείμενα του πραγματικού κόσμου, απαιτούν τη χρήση Open Graph (θα το αναλύσουμε στο πέμπτο κεφάλαιο). Η εφαρμογή που θα αναπτύξουμε λοιπόν, θα βασίζεται στο Graph API του Facebook μιας και αφορά την ανάκτηση των ανακοινώσεων από ένα δικτυακό τόπο και τη δημοσίευση τους σε μία Facebook σελίδα με αυτόματο τρόπο.

4 Περιγραφή του σχεδιασμού της υπηρεσίας

Σε αυτό το κεφάλαιο, θα περιγράψουμε το σχεδιασμό της εφαρμογής μας και θα μελετήσουμε το μεθοδολογικό πλαίσιο της, καθώς και την ανάλυση απαιτήσεών της, τις ροές και τα βασικά συστατικά της.

4.1 Μεθοδολογικό πλαίσιο

Σκοπός της εργασίας μας είναι να δημιουργήσουμε ένα εργαλείο, το οποίο θα αυτοματοποιήσει τη διαδικασία της ενημέρωσης των φοιτητών για τις δημόσιες ανακοινώσεις του τμήματος. Αυτή η διαδικασία θα απλοποιηθεί μέσω της δημοσίευσης τους στο Facebook.

Αρχικά λοιπόν, μελετήσαμε τις τεχνολογίες push και pull για να κατανοήσουμε πως γίνεται η ανάκτηση και ανάρτηση περιεχομένου, καθώς και τις τεχνικές στις οποίες βασίζονται οι εφαρμογές επικοινωνίας που λειτουργούν σε πραγματικό χρόνο και καταλήξαμε σε ποιες τεχνολογίες θα χρησιμοποιήσουμε. Στη συνέχεια, αναζητήσαμε υφιστάμενες λύσεις και ερευνήσαμε το πώς η υλοποίησή τους θα μπορούσε να μας βοηθήσει. Έπειτα, προγραμματίσαμε την εφαρμογή μας στο Facebook (app) και παράλληλα, ξεκινήσαμε τη συγγραφή της εφαρμογής.

Για την δημιουργία της εφαρμογής διασύνδεσης με το κοινωνικό μέσο δικτύωσης Facebook υλοποιήθηκε μία εφαρμογή σε γλώσσα προγραμματισμού PHP σε συνδυασμό με τον Apache εξυπηρετητή. Για την ανάκτηση της πληροφορίας χρησιμοποιήθηκε η μέθοδος Get του HTTP πρωτοκόλλου, που στη συνέχεια μετατράπηκε σε ένα XML έγγραφο και με τη βοήθεια της XPath, ανακτήσαμε την ζητούμενη μόνο πληροφορία, η οποία είναι η τελευταία (ή τελευταίες) ανακοίνωση. Η πληροφορία αυτή, διαμορφώνεται κατάλληλα με την XPath και περνιέται μέσα στον PHP κώδικά μας και μέσω της μεθόδου Post του HTTP πρωτοκόλλου δημοσιεύεται στο Facebook.

4.2 Ανάλυση απαιτήσεων

Για την ανάπτυξη του εργαλείου μας, είχαμε να ασχοληθούμε με δύο κύρια μέρη, τα οποία είναι

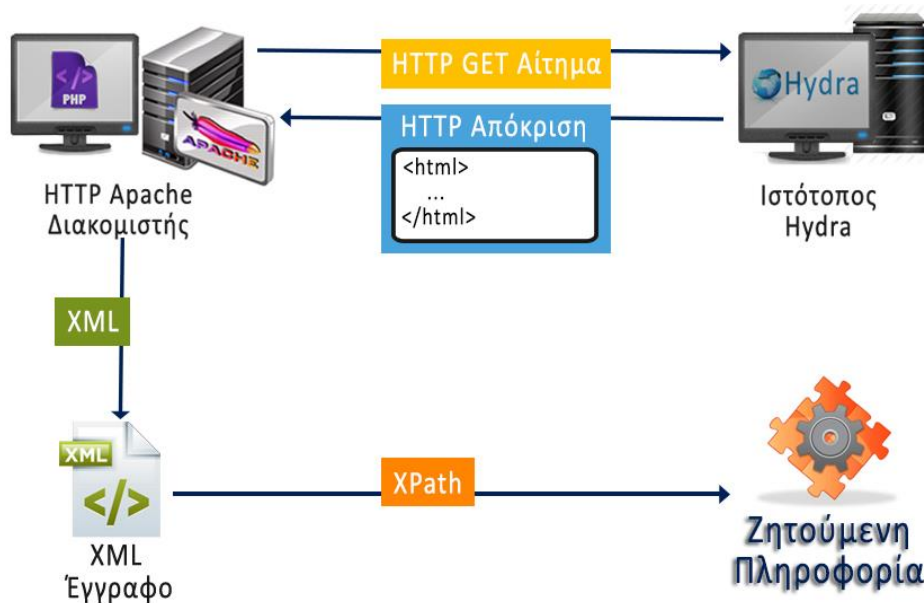
- πρώτον, η ανάκτηση και διαμόρφωση της ζητούμενης πληροφορίας και
- δεύτερον, η δημοσίευση της στη σελίδα του Facebook.

4.2.1 Ανάκτηση και διαμόρφωση πληροφορίας

Το ζητούμενο σε αυτό το μέρος της ανάπτυξης είναι να απομονώνουμε κάθε φορά την κάθε καινούργια ανακοίνωση που δημοσιεύεται στον ιστότοπο του τμήματος για τις δημόσιες ανακοινώσεις.

Για την ανάκτηση της πληροφορίας από τον διακομιστή του ιστότοπου των δημόσιων ανακοινώσεων, χρησιμοποιήθηκε η μέθοδος Get του HTTP πρωτοκόλλου. Με αυτό τον τρόπο, παίρνουμε όλο το περιεχόμενο της σελίδας <http://hydra.it.teithe.gr/s/index.php?m=itdep-bbpublic&bid=news> με ένα μόνο GET αίτημα. Όμως, προκειμένου να επεξεργαστούμε και να διαμορφώσουμε το περιεχόμενο στην τελική μορφή που εμείς θέλουμε να έχει, επιλέξαμε αρχικά να χρησιμοποιήσουμε την τεχνολογία XML για να αποθηκεύσουμε το περιεχόμενο της σελίδας σε ένα XML έγγραφο. Η XML όμως, δεν παρέχει τη δυνατότητα να εντοπίσουμε συγκεκριμένα κομμάτια δεδομένων μέσα στο XML έγγραφο. Για το λόγο αυτό, από τη στιγμή που έχουμε όλη την πληροφορία αποθηκευμένη στο XML έγγραφο, επιλέξαμε να χρησιμοποιήσουμε την XPath, η οποία μας παρέχει τη δυνατότητα διαμόρφωσης της πληροφορίας στη μορφή που εμείς επιθυμούμε.

Τα παραπάνω βήματα φαίνονται στο ακόλουθο διάγραμμα:



Εικόνα 18 Βήματα ανάκτησης & διαμόρφωσης της ζητούμενης πληροφορίας

4.2.2 Δημοσίευση πληροφορίας στο μέσο κοινωνικής δικτύωσης

Στο δεύτερο μέρος, πρέπει να δημοσιεύσουμε την διαμορφωμένη πλέον πληροφορία στο Facebook. Για να συμβεί αυτό, πέρα από τον κώδικα μας θα πρέπει πρώτα να έχουμε δημιουργήσει μία εφαρμογή (app) στο Facebook, μέσω της οποίας θα καθίσταται η επικοινωνία του κώδικα μας με το Graph API του Facebook, ενέργεια απαραίτητη για να δημοσιευτεί η πληροφορία.

4.2.2.1 Facebook εφαρμογές

Προτού περιγράψουμε πως δημιουργήσαμε την εφαρμογή μας στο Facebook, θα προσπαθήσουμε να αναλύσουμε σε θεωρητικό επίπεδο τον τρόπο με τον οποίο λειτουργούν οι εφαρμογές που είναι αναπτυγμένες στην πλατφόρμα του Facebook (facebook apps), έτσι ώστε να κατανοήσουμε τον τρόπο αυτό

και να μπορέσουμε να αναπτύξουμε το δικό μας εργαλείο, το οποίο θα βασίζεται σε μία τέτοια εφαρμογή.

Γενικά, οι εφαρμογές στο Facebook (facebook apps) λειτουργούν όπως και οι υπηρεσίες ιστού (web services). Το Facebook στέλνει πληροφορίες σε αυτόν που παρέχει την εφαρμογή, και από εκεί επιστρέφονται δεδομένα (π.χ. σε HTML) πίσω στο Facebook, το οποίο με την σειρά του τα παρουσιάζει στον τελικό χρήστη. Επομένως, μπορούμε να πούμε πως η πλατφόρμα του Facebook λειτουργεί σαν “ενδιάμεσος” μεταξύ του τελικού χρήστη και του παρόχου της εφαρμογής.

Για την δημιουργία εφαρμογών στο Facebook θα πρέπει να υπάρχει πρόσβαση σε κάποιον διακομιστή όπου να μπορεί να αποθηκευτεί η εφαρμογή, καθώς το Facebook δεν φιλοξενεί προγράμματα φτιαγμένα από τρίτους. Οι εφαρμογές μπορούν να είναι διαδικτυακές, επιτραπέζιες ή για κινητά τηλέφωνα. Με άλλα λόγια, μπορούν να δημιουργηθούν εφαρμογές που εκμεταλλεύονται την πλατφόρμα του Facebook αλλά δεν είναι ενσωματωμένες στα προφίλ των χρηστών. Για παράδειγμα, είναι δυνατό να προγραμματιστεί μια εφαρμογή που θα δημιουργεί ένα παράθυρο στην επιφάνεια εργασίας του χρήστη, το οποίο θα είναι συνδεδεμένο και θα παρουσιάζει το news feed του λογαριασμού του.

Η διεπαφή προγραμματισμού εφαρμογών του Facebook (API) είναι βασισμένη στη REST (Representational State Transfer) διεπαφή. Το δίκτυο REST υπήρξε η εξιδανικευμένη εκδοχή του παγκόσμιου ιστού (World Wide Web) και περιέχει ανεξάρτητα συστατικά τα οποία συνεργάζονται για την μεγιστοποίηση της αποτελεσματικότητας της μεταφοράς δεδομένων.

Όσον αφορά το API του Facebook, η REST διεπαφή σημαίνει ότι οι εφαρμογές διασύνδεσης με το Facebook αποστέλλουν κλήσεις μεθόδων χρησιμοποιώντας HTTP Get ή HTTP Post αιτήματα. Το HTTP είναι το πρωτόκολλο επικοινωνίας που χρησιμοποιείται από τον παγκόσμιο ιστό (www). Το αίτημα GET ανακτά πληροφορία, ενώ το αίτημα POST προσθέτει πληροφορία σε μια υπάρχουσα ιστοσελίδα. Αυτό πρακτικά σημαίνει πως οι

Facebook εφαρμογές μπορούν να ανακτήσουν πληροφορίες από τα προφίλ των χρηστών, να δημοσιεύουν μηνύματα στα προφίλ αυτά ή και τα δύο.

Οι προγραμματιστές για την συγγραφή των εφαρμογών τους μπορούν να χρησιμοποιήσουν την Facebook Query Language (FQL), η οποία είναι παρόμοια με την Structured Query Language (SQL). Αυτές οι γλώσσες προγραμματισμού έχουν σχεδιαστεί για την ανάκτηση πληροφοριών από βάσεις δεδομένων. Με την χρήση της FQL, ένας προγραμματιστής μπορεί να λάβει πληροφορίες σχετικά με κάθε χρήστη που κατεβάζει την εφαρμογή του. Ουσιαστικά, αυτό είναι που πραγματικά οι εφαρμογές του Facebook αφορούν, τη συλλογή πληροφοριών σχετικά με τους χρήστες. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν αυτές τις πληροφορίες ως ένα τρόπο για να δημιουργήσουν ένα κοινό στο οποίο θα απευθύνεται κάποια διαφήμιση ή για την οικοδόμηση μιας βάσης δεδομένων των καταναλωτών ενός συγκεκριμένου προϊόντος. Το Facebook έχει λάβει αρνητικές κριτικές από ανθρώπους που πιστεύουν ότι επιτρέπει στους προγραμματιστές να συλλέγουν δεδομένα από τους χρήστες των εφαρμογών τους, συμπεριλαμβανομένων των προσωπικών δεδομένων τους καθώς και τις συνήθειες τους όταν περιηγούνται στο διαδίκτυο .

Για τους προγραμματιστές που θέλουν να δημιουργήσουν εφαρμογές που ενσωματώνονται στην πλατφόρμα του Facebook, υπάρχει η γλώσσα FBML (Facebook Markup Language). Το Facebook προέκυψε από την γλώσσα HTML, την οποία επέκτεινε προσθέτοντας κάποιες ειδικές ετικέτες που αφορούσαν τα δικά του μοναδικά χαρακτηριστικά. Χρησιμοποιώντας αυτήν την γλώσσα, οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές οι οποίες καθίστανται αναπόσπαστο μέρος της εμπειρίας των χρηστών στο Facebook, επηρεάζοντας την εμφάνιση και την λειτουργικότητα των προφίλ τους.

Κάθε εφαρμογή έχει ένα χώρο στο Facebook που ονομάζεται σελίδα καμβάς (canvas page), τον οποίο οι προγραμματιστές μπορούν να χρησιμοποιήσουν όπως αυτοί επιθυμούν. Όταν ένας χρήστης κάνει κλικ στο εικονίδιο μιας εφαρμογής, ο web browser του τον μεταφέρει στη σελίδα καμβά της εν λόγω

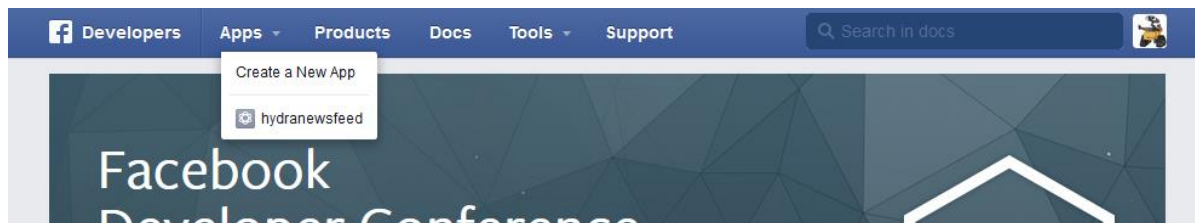
εφαρμογής, όπου και τελικά εμφανίζεται στον τελικό χρήστη το αποτέλεσμα της εφαρμογής που ο προγραμματιστής δημιούργησε.[\[31\]](#)

4.2.2.2 Δημιουργία της Facebook εφαρμογής

Για να δημιουργήσουμε την εφαρμογή μας στο Facebook, θα πρέπει να έχουμε δημιουργήσει πρώτα την σελίδα, η οποία θα αντιπροσωπεύει το τμήμα μας στον ιστότοπο κοινωνικής δικτύωσης, και στην οποία θα προστεθεί η εφαρμογή αυτή. Σε αυτό το σημείο, είναι απαραίτητο να δημιουργηθεί πρώτα ένα προφίλ χρήστη στο Facebook, ο οποίος θα είναι διαχειριστής για τη σελίδα μας.

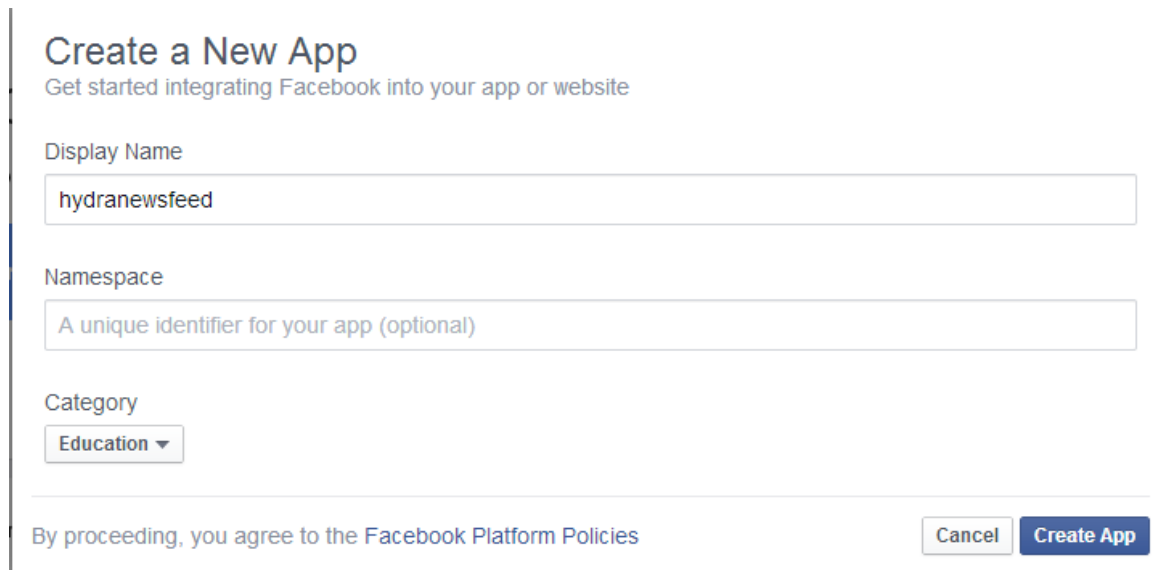
Επομένως, δημιουργήσαμε πρώτα ένα λογαριασμό χρήστη τον οποίο ονομάσαμε *Bot Bot*, και μέσω αυτού δημιουργήσαμε τη σελίδα “**Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙΘ**” της οποίας είναι διαχειριστής. Στη σελίδα αυτή προσθέσαμε την εφαρμογή hydranewsfeed. Αυτή η εφαρμογή δημιουργήθηκε ως εξής:

Αρχικά, συνδεόμαστε ως Bot Bot στο <https://developers.facebook.com/> και επιλέγουμε να δημιουργήσουμε μία νέα εφαρμογή.



Εικόνα 19 Δημιουργία νέας εφαρμογής

Στην οθόνη μας εμφανίστηκε η παρακάτω εικόνα:



Create a New App
Get started integrating Facebook into your app or website

Display Name

Namespace

Category

By proceeding, you agree to the Facebook Platform Policies

Εικόνα 20 Παράθυρο διαλόγου δημιουργίας μιας νέας εφαρμογής

Στην εφαρμογή μας δώσαμε το όνομα hydranewsfeed και το πεδίο namespace το αφήσαμε κενό. Σαν κατηγορία στην οποία ανήκει η εφαρμογή επιλέξαμε “εκπαίδευση”.

Πατήσαμε “Create App” και αφού στην συνέχεια έγινε έλεγχος ασφαλείας με το σύστημα Captcha, η εφαρμογή δημιουργήθηκε.

Στη συνέχεια, λοιπόν, παρουσιάστηκε η παρακάτω οθόνη όπου φαίνονται τα μοναδικά αναγνωριστικά της εφαρμογής (App ID και App Secret), καθώς και όλες οι ρυθμίσεις που μπορούμε να κάνουμε στην εφαρμογή μας.

Συμπληρώνουμε ένα mail επικοινωνίας. (Το οποίο είναι το e-mail που χρησιμοποιήσαμε για να δημιουργήσουμε το λογαριασμό χρήστη του Bot Bot).

hydranewsfeed	Basic	Advanced	Migrations
Dashboard	App ID 1377208439213066	App Secret Show	
Settings	Display Name hydranewsfeed	Namespace	
Status & Review	App Domains	Contact Email thehydranews@gmail.com	
App Details	Website x		
Roles	Site URL http://localhost/hydra	Mobile Site URL URL of your mobile site	
Open Graph	+ Add Platform		
Alerts			
Localize			
Payments			
Audience Network			

Εικόνα 21 Παράθυρο διαλόγου ρυθμίσεων εφαρμογής hydranewsfeed

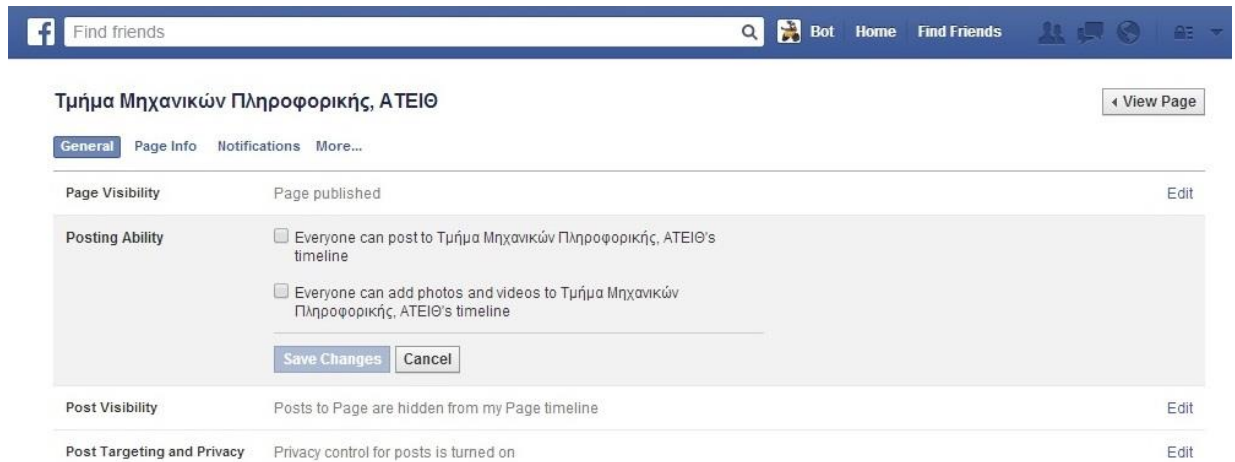
Σε αυτές τις ρυθμίσεις, σημαντικό είναι να καθορίσουμε τα εξής:

- το callback url, το οποίο είναι το url στο οποίο έχουμε αποθηκεύσει την εφαρμογή μας,
- το canvas page url, το οποίο είναι το url που θα φαίνεται ότι είναι η εφαρμογή μας στο facebook (<http://apps.facebook.com/hydranewsfeed/>),
- το developer mode, το οποίο είναι καλό να είναι ενεργοποιημένο κατά την διαδικασία της ανάπτυξης, έτσι ώστε να έχουν πρόσβαση στην εφαρμογή μόνο οι developers.[\[32\]](#)

Τα υπόλοιπα τα αφήνουμε ως έχουν.[\[33\]](#)

Τέλος, ελέγχουμε τις ρυθμίσεις της σελίδας μας, και βεβαιωνόμαστε ότι οι επιλογές “*Everyone can post to Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙΘ’s timeline*” και “*Everyone can add photos and videos to Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙΘ’s timeline*” δεν είναι επιλεγμένες, καθώς δεν θέλουμε ο απλός χρήστης που έχει κάνει like στην σελίδα να μπορεί να δημοσιεύει

περιεχόμενο. Η σελίδα πρέπει να περιέχει μόνο τις δημοσιεύσεις που αναρτώνται αυτόματα από την εφαρμογή hydranewsfeed.



Εικόνα 22 Ρυθμίσεις σελίδας

Με το προφίλ χρήστη αλλά και με την σελίδα δεν θα ασχοληθούμε στο μέλλον, ούτε χρειάζεται κάποιος να έχει το ρόλο του διαχειριστή, καθώς η πληροφορία θα δημοσιεύεται αυτόματα στην σελίδα.

Αφού δημιουργήσαμε λοιπόν την εφαρμογή, πήραμε τα απαραίτητα στοιχεία που χρειαζόμασταν, δηλαδή το appid και το appsecret και τα ενσωματώσαμε στον κώδικά μας.

4.2.2.3 Αυθεντικοποίηση της εφαρμογής

Ένα βασικό κομμάτι της λειτουργίας της εφαρμογής είναι η αυθεντικοποίηση της. Το Graph API επιτρέπει την εύκολη πρόσβαση σε όλη τη δημόσια πληροφορία ενός αντικειμένου. Για να αποκτήσουμε επιπλέον πληροφορία για ένα χρήστη πρέπει πρώτα να πάρουμε την άδεια του. Σε ένα υψηλό επίπεδο, απαιτείται η απόκτηση ενός αδειοδοτικού πρόσβασης για το χρήστη του Facebook. Μετά την απόκτηση του αδειοδοτικού, είναι δυνατή η αποστολή εξουσιοδοτημένων αιτήσεων εκ μέρους του χρήστη περιλαμβάνοντας το αδειοδοτικό πρόσβασης στις Graph API αιτήσεις. Για την εξουσιοδότηση, το Graph API χρησιμοποιεί το OAuth 2.0. Έχοντας αποκτήσει

ένα αδειοδοτικό πρόσβασης για ένα χρήστη χωρίς επεκτεταμένες άδειες επιτρέπεται η πρόσβαση στην πληροφορία που ο χρήστης έχει διαθέσει σε όλους στο Facebook. Σε περίπτωση που απαιτείται συγκεκριμένη πληροφορία για ένα χρήστη, όπως η διεύθυνση του email του ή το ιστορικό του, θα πρέπει να ζητηθούν οι συγκεκριμένες επεκτεταμένες άδειες. Για την δική μας εφαρμογή μας, δεν απαιτείται πρόσβαση σε καμία επιπλέον πληροφορία του χρήστη. Επομένως, το αδειοδοτικό πρόσβασης της ζητά τις βασικές άδειες, οι οποίες επιτρέπουν την πρόσβαση στα δημόσια στοιχεία του χρήστη. Στην παρακάτω εικόνα, φαίνονται τα παράθυρα διαλόγου με τα οποία ζητά πρόσβαση η εφαρμογή μας:



Εικόνα 23 Αδειοδοτικό πρόσβασης της εφαρμογής μας

Πέρα από την αδειοδότηση της εφαρμογής από τον χρήστη, είναι απαιτούμενο το access token της σελίδας μας για να μπορέσουμε να δημοσιεύουμε αυτόματα περιεχόμενο μέσω της εφαρμογής στην σελίδα. Για να βρούμε το access token της σελίδας μας ακολουθήσαμε μία διαδικασία, την οποία θα περιγράψουμε εν συντομία παρακάτω.

Αρχικά, πρέπει να δημιουργήσουμε ένα απλό script σε php μέσα σε έναν υποφάκελο στο project μας, του οποίου το URL θα εμπεριέχεται στις διευθύνσεις που ορίσαμε στο Canvas URL (redirect_uri) όταν δημιουργήσαμε την εφαρμογή Facebook.

```
1 | <?php
2 | print_r($_GET);
3 | ?>
```

Εικόνα 24 Script σε php

Στη συνέχεια πρέπει να εισάγουμε στον περιηγητή (browser) μας την εξής διεύθυνση:

https://www.facebook.com/dialog/oauth?client_id=1377208439213066&redirect_uri=http://hydra.local/fb-tokens/&scope=manage_pages,publish_stream

όπου client_id είναι το App ID της εφαρμογής μας και redirect_uri είναι το URL που ορίζεται ως Canvas URL. Το κομμάτι &scope=manage_pages,publish_stream στο τέλος της διεύθυνσης δηλώνει τα αδειοδοτικά τα οποία απαιτεί η εφαρμογή.

Η παραπάνω διεύθυνση μας οδηγεί στο ακόλουθο αποτέλεσμα:

```
Array
(
    [code] => 10c46fe9c7d6abb79a8497b5aa62a962
)
```

Εικόνα 25 Λήψη τιμής του code

Αφού έχουμε λάβει την τιμή του code εισάγουμε στον browser μας την διεύθυνση:

https://graph.facebook.com/oauth/access_token?client_id=1377208439213066&redirect_uri=http://hydra.local/fb-tokens/&client_secret=10c46fe9c7d6abb79a8497b5aa62a962&code=code

όπου client_id είναι το App ID της εφαρμογής μας και redirect_uri είναι το URL που ορίζεται ως Canvas URL και code η τιμή που λάβαμε.

Αυτό μας επιστρέφει ένα Access Token το οποίο ανήκει στο προφίλ του διαχειριστή της σελίδας μας, δηλαδή στο Bot Bot. Για να βρούμε το Access Token της σελίδας μας, πρέπει να εκτελέσουμε και το παρακάτω βήμα. Εισάγουμε την διεύθυνση:

https://graph.facebook.com/me/accounts?access_token=Access_Token

όπου Access_Token είναι το Access Token που μας επιστράφηκε παραπάνω.

The screenshot shows the Facebook Access Token Debugger interface. At the top, there are navigation tabs: Developers, Apps, Products, Docs, Tools, and Support. A search bar is present with the text 'Search in docs'. The main section is titled 'Access Token Debugger' and includes an 'API Version' dropdown set to 'v2.0'. Below this is a text input field for the 'Access Token, or Open Graph Action ID', containing the token: 'CAATkkFPNDaBafDZCx44V7p7hQQPofFDvCeOEWxP08q4sekBBag1WDdVn5SdpPZCSGVj06AmIQAwKU6lGogpZCudc5Q2dOA0kr4D7GU1PasgAVSHbeUVQ'. A 'Debug' button is located below the input field. The 'Access Token Info' section displays the following details:

Access Token Info	
App ID	1377208439213066 : hydranewsfeed
Profile ID	████████████████████
User ID	100006494020799 : Bot Bot
Issued	██████████ (about an hour ago)
Expires	Never
Valid	True
Origin	Web
Scopes	public_profile, basic_info, manage_pages, publish_actions, user_photos, user_friends, user_status

Εικόνα 27 Πληροφορίες για το access token

Διαπιστώσαμε ότι τα access tokens που αφορούν σελίδες, σε αντίθεση με τα access tokens των χρηστών που διαρκούν μόλις για δύο μήνες, δεν έχουν ημερομηνία λήξης.

4.2.2.4 Λειτουργικότητα του εργαλείου

Έχοντας προγραμματίσει το πρώτο μέρος που αφορά την ανάκτηση και την διαμόρφωση της πληροφορίας, έπειτα από τις κατάλληλες διαδικασίες έχουμε πλέον απομονώσει τις ανακοινώσεις σε ένα πίνακα. Αφού λοιπόν, δημιουργήσαμε την Facebook εφαρμογή και πραγματοποιήθηκε και η σύνδεση με αυτήν χρησιμοποιώντας τα appid και appsecret της, φτάνουμε στο ουσιαστικό κομμάτι της ανάπτυξης του εργαλείου μας, το οποίο είναι να δώσουμε λειτουργικότητα σε αυτή την εφαρμογή του Facebook, έτσι ώστε να παραχθεί το τελικό επιθυμητό αποτέλεσμα. Ελέγχουμε για νέες ανακοινώσεις και αφού κάνουμε σύγκριση με τις ήδη δημοσιευμένες ανακοινώσεις, εφόσον

είναι διαφορετικές, προχωράμε στην κλήση του Facebook API για να τις δημοσιεύσουμε.

4.2.3 Διαγράμματα UML και ροές της εφαρμογής

Σε αυτό το υποκεφάλαιο, θα αναφέρουμε βήμα βήμα τις ροές της εφαρμογής μας και θα τις αναπαραστήσουμε με τη βοήθεια των διαγραμμάτων UML. Η ροή του προγράμματος βοηθά στην περιγραφή της διαδικασίας που ακολουθείται και χρησιμοποιείται στην ανάλυση και στην τεκμηρίωση του.

Ας ξεκινήσουμε περιγράφοντας τη ροή του προγράμματος. Το πρόγραμμά μας ακολουθεί τα παρακάτω βήματα:

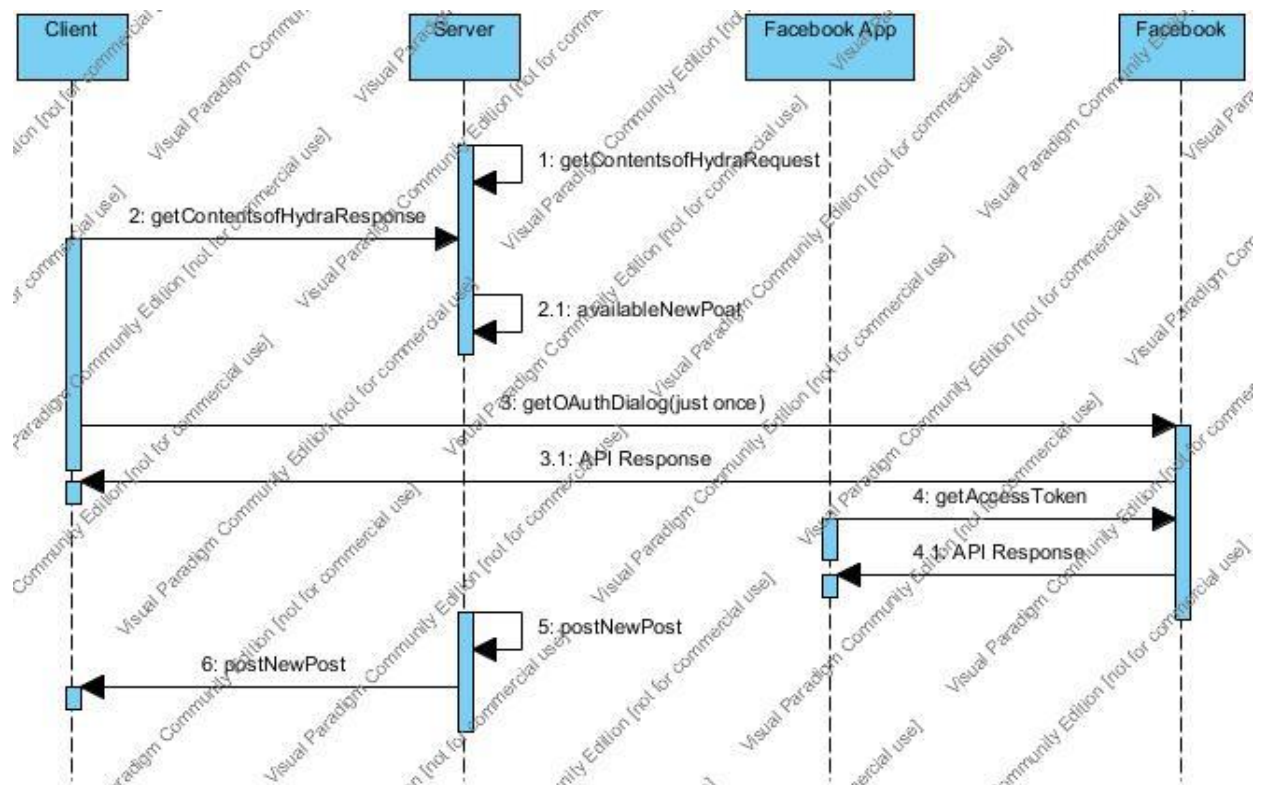
1. Ξεκινάει το session και ανάγνωση μεταβλητής από προηγούμενη εκτέλεση.
2. Ανάκτηση περιεχομένων της ύδρας υπό μορφή HTML.
3. Αποθήκευση στο αρχείο data.xml τα περιεχόμενα της ύδρας σε μορφή XML.
4. Φόρτωμα των δεδομένων του αρχείου data.xml υπό μορφή SimpleXMLElement.
5. Χρήση της XPath έτσι ώστε να απομονώσουμε τα κομμάτια του αρχείου data.xml που επιθυμούμε. Όλα τα divs με γνώρισμα class='new'.
6. Αντιστροφή του αποτελέσματος και αποθήκευση σε ένα πίνακα news.
7. Ανάγνωση κάθε στοιχείου του πίνακα και αποθήκευση του αριθμού των παιδιών του. Αν είναι τρία, η ανακοίνωση δεν περιέχει συνημμένο και αποθηκεύεται σε μια μεταβλητή. Αν έχει τέσσερα, η ανακοίνωση έχει συνημμένο και αποθηκεύεται η επιθυμητή πληροφορία σε μια μεταβλητή.
8. Έλεγχος για το αν η ανακοίνωση που είναι για δημοσίευση, είναι ήδη δημοσιευμένη.
9. Σύνδεση στο Facebook με τα απαραίτητα ζεύγη αναγνωριστικών.
10. Δημιουργία πίνακα με το μήνυμα που θα δημοσιευτεί (τελευταία/τελευταίες ανακοίνωση) και με το access token.

11.Κλήση στο API του Facebook και δημοσίευση του μηνύματος στο κοινωνικό δίκτυο.

12.Αποθήκευση μεταβλητής στο session για επόμενη χρήση.

Τα βήματα εφτά με έντεκα επαναλαμβάνονται για κάθε αδημοσίευτη ανακοίνωση που υπάρχει. Όταν όλες έχουν δημοσιευτεί, η ροή του προγράμματος θα διακοπεί.

Στη συνέχεια, θα περιγράψουμε τη ροή του προγράμματος μας με τη χρήση διαγραμμάτων UML. Στην παρακάτω εικόνα, φαίνεται η λογική του συστήματός μας και πως οι οντότητες επικοινωνούν μεταξύ τους.



Εικόνα 28 Διάγραμμα UML του συστήματός μας

Παρατηρούμε ότι το μεγαλύτερο κομμάτι του κώδικά μας εκτελείται στο server. Αξίζει να αναφέρουμε ότι το πρόγραμμα εκτελείται στο server αυτόματα ανά τακτά χρονικά διαστήματα που έχουμε ορίσει εμείς (ως παράμετρο μίας εντολής). Δεν χρειάζεται δηλαδή να κάνουμε κάτι εμείς χειροκίνητα. Ο server πρέπει να αποκτήσει τα περιεχόμενα της ύδρας με σκοπό να τα επεξεργαστεί κατάλληλα. Ο client του στέλνει την απάντηση στο αίτημά του (απλή

επικοινωνία client - server). Στη μεριά του server γίνονται τα βήματα που περιγράψαμε παραπάνω, δηλαδή αποθηκεύει υπό μορφή HTML τα περιεχόμενα της ύδρας, τα μετατρέπει σε μορφή XML και στο τέλος τα μετατρέπει σε μορφή SimpleXMLElement. Το αποτέλεσμα αποθηκεύεται σε μια μεταβλητή, αφού πρώτα αντιστραφεί. Εφαρμόζουμε XPath πάνω σε αυτή τη μεταβλητή και απομονώνουμε τα κομμάτια που επιθυμούμε. Επεξεργαζόμαστε τα κομμάτια αυτά κατάλληλα, ώστε να αποκτήσουμε την επιθυμητή πληροφορία (ανακοίνωση). Γίνεται έλεγχος για το αν αυτή η πληροφορία είναι δημοσιευμένη. Αν δεν είναι και πρέπει να δημοσιευτεί, βλέπουμε ότι ο server έχει ένα μήνυμα availableNewPost (διαθέσιμη για δημοσίευση μία νέα ανακοίνωση). Για να δημοσιευτεί, όμως, αυτή η πληροφορία, πρέπει ο client να αυθεντικοποιηθεί από το Facebook. Έτσι, βλέπουμε μία επικοινωνία μεταξύ τους, η οποία γίνεται **μόνο μία φορά** και ο client αυθεντικοποιείται. Η Facebook App (εφαρμογή) για να δημοσιεύσει στη σελίδα μας, πρέπει να λάβει ένα access token. Για αυτό το λόγο, λοιπόν, επικοινωνεί με το Facebook και αυτό με τη σειρά του, της επιστρέφει το access token. Ο server τώρα μπορεί να δημοσιεύσει στον client. Είναι σκόπιμο να αναφέρουμε ότι επικεντρωθήκαμε περισσότερο στην επικοινωνία οντοτήτων στην παραπάνω εικόνα, μιας και το τι γίνεται στο server περιγράφεται με βήματα στη ροή του προγράμματος. Στο επόμενο κεφάλαιο, θα αναλύσουμε τις τεχνολογίες που χρησιμοποιήσαμε για να υλοποιηθεί αυτή η υπηρεσία.

5 Συνοπτική περιγραφή του τεχνολογικού περιβάλλοντος που χρησιμοποιήθηκε

Το κύριο συστατικό της εφαρμογής μας είναι μια facebook εφαρμογή (app) η οποία μέσω ενός Apache server εκτελεί PHP κώδικα. Μέσω αυτού του PHP κώδικα και με την χρήση επιπλέον του HTTP πρωτοκόλλου, της XML και της XPath ανακτώνται, διαμορφώνονται και δημοσιεύονται οι ζητούμενες πληροφορίες.

Η πληροφορία ανακτάται από τον διακομιστή (server) στον οποίο βρίσκεται ο ιστότοπος των ανακοινώσεων με την χρήση της μεθόδου Get του HTTP πρωτοκόλλου. Στη συνέχεια η πληροφορία μετατρέπεται σε XML έγγραφο και με την χρήση της Xpath απομονώνονται τα κομμάτια εκείνα που μας ενδιαφέρουν ώστε να δομήσουμε την πληροφορία που επιθυμούμε να δημοσιεύσουμε.

Έπειτα, μέσω του OAuth 2.0 πρωτοκόλλου πραγματοποιείται η εξουσιοδότηση της εφαρμογής μας στο Facebook και στη συνέχεια, με τη χρήση της μεθόδου POST του πρωτοκόλλου HTTP η πληροφορία αποστέλλεται στον Apache server έτσι ώστε να γίνει η δημοσίευσή της ως νέα ενημέρωση στη σελίδα μας στο Facebook.

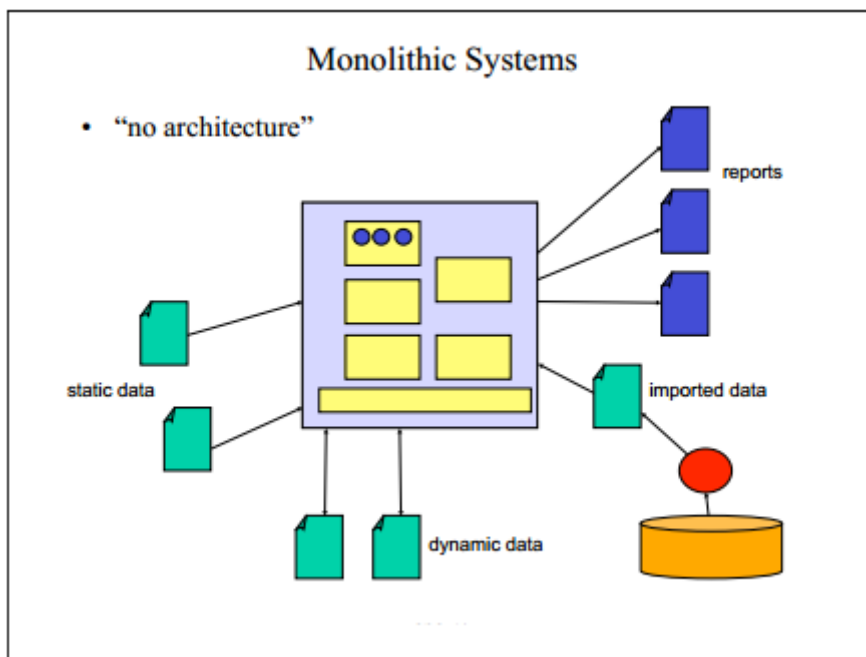
Παρακάτω, στο κεφάλαιο αυτό, θα περιγράψουμε συνοπτικά τις τεχνολογίες αυτές.

5.1 Facebook

Πριν αναφέρουμε τις άλλες τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του εργαλείου μας, κρίναμε σκόπιμο πρώτα να αναλύσουμε το περιβάλλον του Facebook στο οποίο αναπτύξαμε την εφαρμογή μας(app) στην οποία βασίζεται το εργαλείο αυτό.

5.1.1 Το facebook είναι μια μονολιθική εφαρμογή

Ο όρος μονολιθική εφαρμογή στην μηχανική λογισμικού περιγράφει μία εφαρμογή single-tiered λογισμικού στην οποία ο κώδικας του περιβάλλοντος χρήστη και ο κώδικας που αφορά την πρόσβαση στη βάση δεδομένων έχει συνδυαστεί σε ένα και μοναδικό πρόγραμμα. Είναι ουσιαστικά μια προσέγγιση χωρίς καμία δομή.



Εικόνα 29 Περιγραφή ενός μονολιθικού συστήματος

Το πρόγραμμα είναι μια συλλογή από διαδικασίες, κάθε μια από τις οποίες μπορεί να καλέσει οποιαδήποτε άλλη, όταν τη χρειαστεί. Η επικοινωνία μεταξύ των διαδικασιών γίνεται με παραμέτρους. Κάθε μία από τις διαδικασίες είναι ορατή σε οποιαδήποτε άλλη.

Τα χαρακτηριστικά μιας μονολιθικής εφαρμογής είναι τα εξής:

- Είναι συνήθως γραμμένη σε μία και μοναδική γλώσσα προγραμματισμού
- Τα πάντα μεταγλωττίζονται και συνδέονται σε μία μοναδική εφαρμογή, πολύ μεγάλη σε μέγεθος
- Μπορεί να λειτουργήσει και σε batch και σε GUI mode.

- Τα δεδομένα φορτώνονται στη μνήμη, και γράφονται πάλι πίσω με "ρητή" αποθήκευση (explicit save). Δεν πραγματοποιείται ταυτόχρονος διαμοιρασμός των δεδομένων.
- Μπορεί να συγχρονιστεί είτε πολυνηματικά, είτε με multi-processing αλλά υπάρχει μόνο ένα εκτελέσιμο.

5.1.2 Μονολιθική Αρχιτεκτονική

Ένα μονολιθικό σύστημα μπορεί να χαρακτηριστεί από έναν πηγαίο κώδικα, ένα παραγόμενο πρόγραμμα, αλλά, μπορεί να περιέχει συγχρονισμό.

Τα δεδομένα διαβάζονται στη μνήμη στην εφαρμογή, επεξεργάζονται, μπορούν να παράγουν αποτελέσματα και να αποθηκευτούν πίσω στην ίδια βάση ή και σε διαφορετική. Η πρόσβαση και τροποποίηση των δεδομένων από πολλούς χρήστες είναι πολύ δύσκολο να επιτευχθεί και αποτελεί μεγάλο επιπλέον κόστος για την εφαρμογή.

5.1.2.1 Πλεονεκτήματα Μονολιθικών συστημάτων

- Αυτάρκεια
- Ανεξαρτησία από άλλες εφαρμογές
- Επίδοση
 - ✓ Η ανάγνωση και εγγραφή των δεδομένων μπορεί να βελτιστοποιηθεί για την απόδοση χωρίς να λαμβάνονται υπόψη θέματα όπως η multi-user ανταλλαγή δεδομένων.
 - ✓ Ανάγνωση δεδομένων απευθείας από το δίσκο μέσω συστήματος αρχείων.
 - ✓ Ανάγνωση δεδομένων από το δίσκο μέσω λογισμικού RDBMS, OODBMS ή κατανεμημένου διακομιστή δεδομένων (data server).
 - ✓ Η τροποποίηση των δεδομένων in-memory είναι φοβερά γρηγορότερη ενώ η αποθήκευσή τους στη μνήμη cache μπορεί να παρουσιάσει ανεπαίσθητα θέματα για τα συστήματα που μοιράζονται κοινά δεδομένα.

- ✓ Δεν επιβαρύνει το IPC.
- Απλότητα
 - ✓ Λιγότερος κώδικας προς συγγραφή
 - ✓ Λιγότερα προβλήματα προς αντιμετώπιση
- Υποστηρίζει επαναχρησιμοποίηση κομματιών της λογικής της εφαρμογής
- Η φιλοσοφία του σχεδιασμού είναι ότι η εφαρμογή δεν είναι υπεύθυνη μόνο για ένα συγκεκριμένο κομμάτι, αλλά μπορεί να εκτελέσει οποιαδήποτε ενέργεια χρειάζεται για να ολοκληρώσει μία λειτουργία.

5.1.2.2 Μειονεκτήματα Μονολιθικών συστημάτων

- Έλλειψη υποστήριξης για κοινή πρόσβαση πολλών χρηστών
 - ✓ Επιβάλλει one-at-a-time πρόσβαση
 - ✓ Δεν επιτρέπει σχήματα βάσεων δεδομένων τα οποία συγχωνεύουν πολλαπλά αρχεία
 - ✓ Δεν επιτρέπει υβριδικές προσεγγίσεις
- Ποσότητα δεδομένων
 - ✓ Όταν η ποσότητα των δεδομένων είναι πολύ μεγάλη για να φορτωθεί στη μνήμη απαιτείται πολύς χρόνος και πολύ εικονική μνήμη.
- Η πραγματοποίηση αλλαγών είναι δύσκολο να επιτευχθεί
- Ο έλεγχος παίρνει περισσότερη ώρα να πραγματοποιηθεί λόγω του μεγέθους του προγράμματος
- Η προσθήκη νέων λειτουργιών απαιτεί την επαναδόμηση ολόκληρου του πηγαίου κώδικα.[\[34\]](#)

5.2 Τα API του Facebook

Η πλατφόρμα του Facebook επιτρέπει σε οποιονδήποτε να "αναπτύξει social εφαρμογές στο Facebook και το διαδίκτυο". Το Facebook, για να δώσει στους προγραμματιστές τη δυνατότητα να αναπτύξουν τέτοιες εφαρμογές, προσφέρει μια εκτενή συλλογή από βασικά αλλά και προχωρημένα API και SDK. Ένα από αυτά είναι το Graph API.

Το Graph API παρέχει το API το οποίο αποτελεί τον πυρήνα κάθε Facebook εφαρμογής, ενώ το Open Graph Protocol παρέχεται έτσι ώστε οι εφαρμογές να μπορούν να δημιουργήσουν προσαρμοσμένα αντικείμενα (όπως Βιβλία, Μουσική και άλλα αντικείμενα του πραγματικού κόσμου, ανάλογα με τις απαιτήσεις) και τις ενέργειες που μπορούν να εκτελεστούν σε αυτά (όπως το διάβασμα, η ακρόαση). Η δυνατότητα αυτή καθιστά το Facebook και την εκάστοτε εφαρμογή πιο ελκυστικά.

Η σχέση που συνδέει το Graph API με το Open Graph Protocol είναι η εξής: Το Open Graph Protocol επιτρέπει στον προγραμματιστή να ορίσει νέα αντικείμενα και ενέργειες στο social graph ενός χρήστη, και ο τρόπος με τον οποίο θα δημιουργηθούν νέες απεικονίσεις αυτών των αντικειμένων και των ενεργειών ορίζεται από το Graph API.[\[35\]](#)

Παρακάτω θα περιγράψουμε το Open Graph πρωτόκολλο και το Graph API.

5.2.1 Open Graph πρωτόκολλο

Το Open Graph πρωτόκολλο επιτρέπει στους προγραμματιστές να ενσωματώσουν τις σελίδες τους στο κοινωνικό γράφημα (social graph). Αυτές οι σελίδες αποκτούν τη λειτουργικότητα άλλων γραφικών αντικειμένων, περιλαμβανομένων profile συνδέσμων και ρεύματα ενημερώσεων για συνδεδεμένους χρήστες.

Το Open Graph πρωτόκολλο επιτρέπει σε κάθε ιστοσελίδα να γίνει ένα πλούσιο αντικείμενο στο κοινωνικό γράφημα. Για παράδειγμα, η δυνατότητα αυτή χρησιμοποιείται στο Facebook για να επιτρέπει σε κάθε ιστοσελίδα να έχει την ίδια λειτουργικότητα, όπως κάθε άλλο αντικείμενο στο Facebook. Παρόλο που υπάρχουν πολλές διαφορετικές τεχνολογίες και σχήματα

(schemas) και μπορούν να συνδυαστούν μεταξύ τους, δεν υπάρχει μία ενιαία τεχνολογία που να παρέχει αρκετές πληροφορίες που να αντιπροσωπεύουν επαρκώς οποιαδήποτε ιστοσελίδα μέσα στο κοινωνικό γράφημα. Το Open Graph πρωτόκολλο βασίζεται σε αυτές τις υπάρχουσες τεχνολογίες και δίνει στους προγραμματιστές τη δυνατότητα να εφαρμόσουν ένα πράγμα, την προγραμματιστική απλότητα η οποία αποτελεί το βασικό στόχο του πρωτοκόλλου αυτού.[36]

5.2.1.1 Πως δουλεύει το Open Graph

Η ενσωμάτωση του Open Graph ξεκινά με την εφαρμογή. Το Facebook παρέχει SDKs για iOS, Android, JavaScript και PHP. Οι εφαρμογές χρειάζονται να χρησιμοποιήσουν το Facebook Login, το οποίο επιτρέπει στους χρήστες να συνδεθούν στην εκάστοτε εφαρμογή χρησιμοποιώντας τα Facebook διαπιστευτήρια τους. Από τη στιγμή που υπάρχει συσχέτιση μεταξύ της εφαρμογής του Facebook και του ατόμου, το άτομο μπορεί να δημοσιεύει περιεχόμενο από την εφαρμογή.

Τέσσερα είναι τα στοιχεία που υπάρχουν για να δημιουργήσεις κανείς περιεχόμενο(stories). Ας χρησιμοποιήσουμε το παράδειγμα του “ο Christopher τελείωσε να διαβάζει το The Name of the Wind στο Goodreads”:

The actor: Αυτό είναι το άτομο που δημοσιεύει το περιεχόμενο, στην δική μας περίπτωση ο “Christopher”.

The app: Αυτή είναι η εφαρμογή που δημοσιεύει το περιεχόμενο για λογαριασμό του actor. Κάθε περιεχόμενο παράγεται από μία εφαρμογή και περιλαμβάνει την εφαρμογή που χρησιμοποιήθηκε για να δημιουργηθεί.

The action: Αυτή είναι η δραστηριότητα που πραγματοποιεί ο actor, στην δική μας περίπτωση “τελείωσε να διαβάζει”. Το Facebook παρέχει ένα σύνολο από συνήθειες ενέργειες που μπορούν να χρησιμοποιηθούν για την

δημιουργία περιεχομένου, όπως *παρακολουθώ* ή *διαβάζω*, ο καθένας όμως μπορεί να δημιουργήσει δικό του περιεχόμενο και ενέργειες.

The object: Αυτό είναι το αντικείμενο με το οποίο αλληλεπιδρά ο actor, 'The Name of the Wind', ένα βιβλίο. Τα αντικείμενα μπορούν να δημιουργηθούν με δύο διαφορετικούς τρόπους. Οι τρόποι αυτοί είναι:

- Προσθέτοντας μεταδεδομένα σε μία ελεύθερα προσβάσιμη από το κοινό ιστοσελίδα που φιλοξενείται στο διαδίκτυο. Σχεδόν κάθε ιστοσελίδα μπορεί να γίνει αντικείμενο. Σε αυτή τη μορφή, τα αντικείμενα είναι δημόσια πληροφορία.
- Χρησιμοποιώντας το Object API του Facebook για την δημιουργία αντικειμένων που φιλοξενούνται σε διακομιστές του Facebook. Τα αντικείμενα αυτά μπορεί να έχουν την ιδιότητα *private* σε ορισμένους χρήστες, ή να έχουν την ιδιότητα *global* μέσα στα πλαίσια μίας εφαρμογής.[\[37\]](#)

5.2.2 Graph API

Για τη δημιουργία του εργαλείου μας έπρεπε να έχουμε τη δυνατότητα να εισάγουμε δεδομένα στο social graph του Facebook. Αυτή τη δυνατότητα μας τη δίνει το Graph API, που επιτρέπει στον προγραμματιστή να διαβάζει και να γράφει δεδομένα από και προς το Facebook.

Το Graph API δημιουργήθηκε τον Μάρτιο του 2010 με την πρόθεση να αντικαταστήσει τα παλαιότερα REST API. Είναι η επόμενη γενιά του Facebook API, και στόχος του είναι να παρέχει πρόσβαση σε διάφορα κομμάτια δεδομένων του Facebook social graph.. Το νέο αυτό API είναι παντελώς RESTful, και εξ ορισμού τα αποτελέσματα επιστρέφονται σε JSON. Ωστόσο, υπάρχει ακόμα λειτουργικότητα στο REST API το οποίο δεν έχει καμία παρόμοια μέθοδο με το νέο Graph API. Το Facebook έχει δηλώσει την πρόθεσή του να αντικαταστήσει το REST API με το Graph API, γι αυτό πλέον είναι προτιμότερη η χρήση του Graph API.

Ο κοινωνικός γράφος (social graph) βρίσκεται τον πυρήνα του Facebook. Πρόκειται για τα άτομα και τις συνδέσεις τους με οντότητες, για τις οποίες τα άτομα νοιάζονται και ενδιαφέρονται. Το Graph API παρουσιάζει μια απλή, συνεκτική εικόνα του κοινωνικού γράφου του Facebook, αναπαριστώντας ομοιόμορφα τα αντικείμενα στο γράφημα και τις συνδέσεις μεταξύ τους. Σύμφωνα λοιπόν με αυτό, κάθε αντικείμενο ή οντότητα στο κοινωνικό γράφο έχει ένα μοναδικό αναγνωριστικό (id), μόνο μέσω του οποίου μπορεί κάποιος να έχει πρόσβαση στις ιδιότητες του συγκεκριμένου αντικειμένου ή της συγκεκριμένης οντότητας.

Ουσιαστικά, υπάρχουν τρεις μέθοδοι που χρησιμοποιούν το Graph API: ανάκτηση δεδομένων, δημοσίευση δεδομένων και διαγραφή δεδομένων. Ορισμένα δεδομένα μπορούν να αιτηθούν χωρίς να έχει γίνει έλεγχος ταυτότητας, όμως τα περισσότερα δεδομένα εξαρτώνται από την αυθεντικοποίηση, ποιος χρήστης έχει αυθεντικοποιηθεί, τα δικαιώματα που έχουν δοθεί από αυτόν τον χρήστη, και τα δικαιώματα που έχουν δοθεί από το αντικείμενο το οποίο ζητάει. Η δημοσίευση περιεχομένου έχει παρόμοιους περιορισμούς. Η διαγραφή δεδομένων είναι πολύ περιορισμένη και συνήθως μπορεί να γίνει μόνο σε περιεχόμενο που έχει δημοσιευτεί από τον τρέχοντα χρήστη της εφαρμογής.

Όλα τα δεδομένα τα οποία λαμβάνονται είναι κωδικοποιημένα με JSON. Τα δεδομένα που αποστέλλονται μπορούν είτε να είναι περασμένα ως παράμετροι είτε σε GET είτε σε POST μέθοδο (και αυτά είναι κωδικοποιημένα με JSON αν δεν έχουν την μορφή απλής συμβολοσειράς).[\[38\]](#)

Παρακάτω θα περιγράψουμε τις βασικές λειτουργίες του Graph API.

5.2.2.1 Μέθοδοι δημοσίευσης περιεχομένου

Publish

Για τη δημοσίευση στο Facebook graph απαιτούνται HTTP Post αιτήματα στα κατάλληλα URLs σύνδεσης, χρησιμοποιώντας ένα αδειοδοτικό πρόσβασης εκ μέρους του χρήστη ή ένα αδειοδοτικό πρόσβασης εφαρμογής. Οι

περισσότερες λειτουργίες εγγραφής απαιτούν επεκτεταμένες άδειες για τον ενεργό χρήστη.

Η δημοσίευση υποστηρίζεται από τους ακόλουθους τύπους αντικειμένων:

- Profiles
- Posts
- Events
- Albums

Delete

Για τη διαγραφή αντικειμένων στο Facebook graph απαιτούνται HTTP Delete αιτήματα στα URLs των προς διαγραφή αντικειμένων. Για την υποστήριξη των πελατών οι οποίοι δεν υποστηρίζουν όλες τις HTTP μεθόδους (όπως για παράδειγμα οι πελάτες της JavaScript), εναλλακτικά εκδίδεται ένα Post αίτημα σε ένα URL αντικείμενο με την επιπλέον παράμετρο method - delete να αντιπαρέλθει την HTTP μέθοδο.

Analytics

Κατά την δημιουργία μιας εφαρμογής, μπορεί να ζητηθεί να λαμβάνονται λεπτομερή δεδομένα ανάλυσης που αφορούν τα δημογραφικά στοιχεία των χρηστών της εφαρμογής και τι οι χρήστες αυτοί διαμοιράζονται με την εφαρμογή. Αυτό είναι δυνατό με τη χρήση του αντικειμένου insights. Το Graph API παρέχει πρόσβαση με χρήση προγραμματισμού σε όλα αυτά τα δεδομένα, έτσι ώστε να μπορούν οι προγραμματιστές να χρησιμοποιούν τα δεδομένα της πλατφόρμας μέσα σε analytics συστήματα που έχουν αναπτύξει. Για το κατέβασμα δεδομένων του insights αντικειμένου χρειάζεται πρώτα η απόκτηση ενός αδειοδοτικού πρόσβασης εφαρμογής.

5.2.2.2 Μέθοδοι ανάκτησης περιεχομένου

Reading

Το Graph API επιτρέπει την ανάγνωση ιδιοτήτων και συνδέσεων του κοινωνικού γράφου του Facebook. Η χρήση του Grap API επιτρέπει:

-την ανάγνωση συγκεκριμένων πεδίων: οι περισσότερες ιδιότητες του αντικειμένου επιστρέφονται όταν γίνεται ένα αίτημα. Το Grap API, με τη χρήση της παραμέτρου “fields” στο αίτημα, επιτρέπει την επιλογή των πεδίων ,ή συνδέσεων, που επιθυμούνται να επιστραφούν . Επιπλέον, επιτρέπεται η αίτηση πολλαπλών αντικειμένων μέσω ενός απλού αιτήματος (query) χρησιμοποιώντας τη παράμετρο αιτήματος “id”. Η παράμετρος αυτή δέχεται URLs, γεγονός χρήσιμο για την ανεύρεση των id των URLs στο Open Graph. Τέλος, διαθέτει ένα ειδικό αναγνωριστικό με το οποίο αναφέρεται στο τρέχων χρήστη της εφαρμογής,

-την ανάκτηση εικόνων κάθε αντικειμένου: επιτρέπει την επεξεργασία του τρέχοντος προφίλ της φωτογραφίας για ένα αντικείμενο προσθέτοντας τη κατάληξη /picture στο URL του αντικειμένου,

-την σελιδοποίηση: όταν ζητάμε συνδέσεις, υπάρχουν αρκετοί χρήσιμοι παράμετροι που επιτρέπουν το φιλτράρισμα και τη σελιδοποίηση διαμέσου των δεδομένων των διασυνδέσεων,

-την ενδοσκόπηση ενός αντικειμένου για μεταδεδομένα: το Graph API υποστηρίζει την ενδοσκόπηση των αντικειμένων, η οποία επιτρέπει την εμφάνιση όλων των διασυνδέσεων ενός αντικειμένου χωρίς να γνωρίζουμε τον τύπο του εκ των προτέρων. Για την ανάκτηση αυτής της πληροφορίας, προστίθεται το στοιχείο “metadata=1” στο URL του αντικειμένου και το JSON αποτέλεσμα θα περιλαμβάνει μια metadata ιδιότητα που περιλαμβάνει όλες τις υποστηριζόμενες διασυνδέσεις για το συγκεκριμένο αντικείμενο,

-την ανάκτηση πραγματικού χρόνου ανανεώσεων σε κάθε αλλαγή: οι πραγματικού χρόνου ανανεώσεις παρέχουν την ικανότητα λήψης ανανεώσεων για όλους τους χρήστες της εφαρμογής, όπως οι αλλαγές στα δεδομένα τους. Με τέτοιους είδους συνδρομές, είναι σίγουρο ότι τα αποθηκευμένα δεδομένα είναι σωστά χωρίς να χρειάζεται η βοήθεια από τους εξυπηρετητές του Facebook, αυξάνοντας έτσι την αξιοπιστία της εφαρμογής και την ανταπόκριση της εμπειρίας του χρήστη.[\[39\]](#)

Αναζήτηση

Η αναζήτηση όλων των δημόσιων αντικειμένων στο κοινωνικό γράφο γίνεται με τον ακόλουθο τρόπο: <https://graph.facebook.com/search>.

Η αναζήτηση υποστηρίζεται από τους ακόλουθους τύπους αντικειμένων:

Όλα τα δημόσια posts

- Users
- Pages
- Events
- Groups
- Places
- Checkins

5.2.3 Άλλα APIs

Το Facebook, πέρα από το Graph API, παρέχει και κάποια άλλα APIs για πιο εξεζητημένες λειτουργίες.^[40] Τα APIs αυτά είναι τα ακόλουθα:

Public Feed API

Το Public Feed API παρέχει μια ροή από ενημερώσεις κατάστασης (status updates) χρηστών και ενημερώσεις κατάστασης σελίδων, όπως δημοσιεύονται στο Facebook.

Keyword Insights API

Το Keyword Insights API παρέχει ένα επίπεδο ανάλυσης για όλες τις δημοσιεύσεις στο Facebook. Δίνεται έτσι η δυνατότητα συγκεντρωτικής αναζήτησης ανώνυμων στατιστικών στοιχείων για τους ανθρώπους οι οποίοι αναφέρονται σε έναν συγκεκριμένο όρο.

FQL

Η Facebook Query γλώσσα, (ή αλλιώς FQL), επιτρέπει στο χρήστη να χρησιμοποιήσει μία διεπαφή της μορφής SQL-style για να εκτελέσει

ερωτήματα πάνω στα δεδομένα τα οποία είναι εκτεθειμένα από το Graph API. Παρέχει επίσης κάποιες σύνθετες λειτουργίες που δεν είναι διαθέσιμες στο Graph API, όπως για παράδειγμα η χρήση των αποτελεσμάτων ενός ερωτήματος σε ένα άλλο ερώτημα.

Chat API

Προσφέρει την δυνατότητα ενσωμάτωσης του Facebook Chat σε διαδικτυακές, επιτραπέζιες ή για κινητά εφαρμογές άμεσων μηνυμάτων. Ο πελάτης των άμεσων μηνυμάτων συνδέεται με το Facebook Chat μέσω της υπηρεσίας Jabber XMPP.

5.3 Facebook SDK για PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML. Το εργαλείο μας αναπτύχθηκε στην γλώσσα PHP, και ειδικότερα με την χρήση της βιβλιοθήκης SDK του Facebook για την PHP. [\[41\]](#)

5.3.1 Η Βιβλιοθήκη SDK του Facebook για την PHP

Το SDK του Facebook για την PHP επιτρέπει στους προγραμματιστές να εκτελέσουν ένα πλούσιο σύνολο από server-side λειτουργικότητα για την πρόσβαση στο API του Facebook. Περιλαμβάνει πρόσβαση σε όλες τις λειτουργίες του Graph API και της FQL. Ακόμη, λειτουργεί σε συνδυασμό με

το SDK του Facebook για την Javascript για να παρέχει τα μέσα για την ανάπτυξη της λειτουργίας Facebook Login.

Το SDK τυπικά χρησιμοποιείται για την εκτέλεση εργασιών ως διαχειριστής της εφαρμογής(app), όμως μπορεί να χρησιμοποιηθεί και για εκτέλεση εργασιών εκ μέρους του χρήστη της τρέχουσας συνεδρίας(session). Αφαιρώντας την ανάγκη για τη διαχείριση των access tokens με μη αυτόματο τρόπο, το SDK απλοποιεί σημαντικά τη διαδικασία της αυθεντικοποίησης και αδειοδότησης των χρηστών για την εφαρμογή μας (app).[\[42\]](#)

5.4 Apache HTTP Server

Ο Apache HTTP εξυπηρετητής γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε ένας χρήστης επισκέπτεται έναν ιστότοπο, το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους εξυπηρετητές ιστού, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το Mac OS X.

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιούργησε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και τη πλατφόρμα .NET

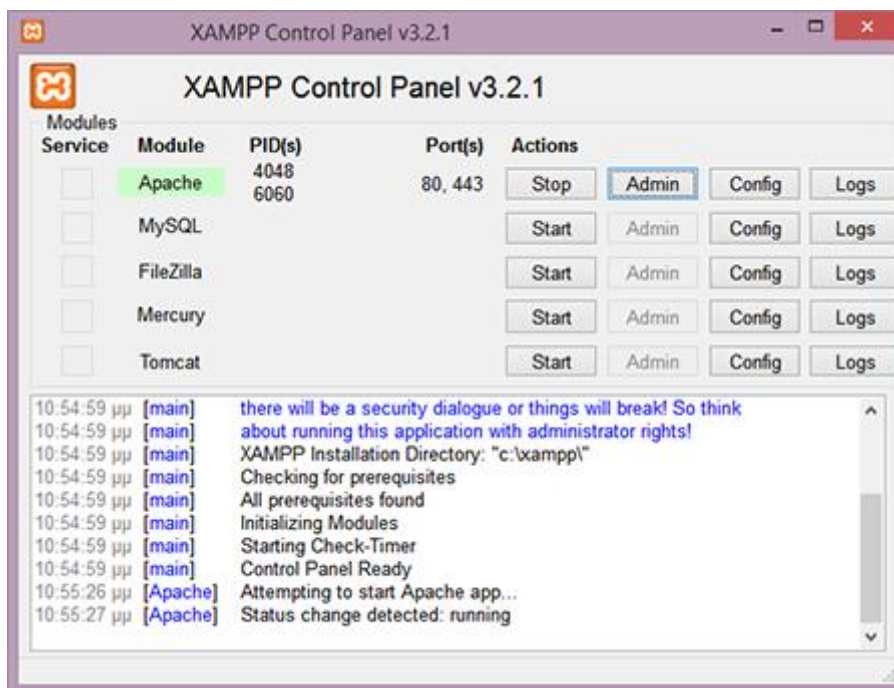
Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους. Ο Apache είναι ένα ελεύθερο ανοικτού κώδικα λογισμικό. Εκτός του κόστους, είναι ιδιαίτερα διαμορφώσιμο και είναι πλήρες των προηγμένων χαρακτηριστικών γνωρισμάτων, όπου αυτός είναι και ο κύριος λόγος για τον οποίο οι περισσότεροι πάροχοι web hosting το χρησιμοποιούν.[\[43\]](#)

5.5 XAMPP

Τον Apache HTTP εξυπηρετητή χρησιμοποιούμε μέσα από το πακέτο του XAMPP. Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl.

Το XAMPP είναι ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:

- X (Αναφέρεται στο «cross-platform» που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας. Τρέχει σε Microsoft Windows, Linux, Solaris, and Mac OS X)
- Apache HTTP εξυπηρετητής
- MySQL
- PHP
- Perl



Εικόνα 30 Αρχική οθόνη XAMPP

Το XAMPP είναι ένα χρήσιμο βοήθημα που μας δίνει την δυνατότητα να κάνουμε τον υπολογιστή μας ένα web server και όχι μόνο. Για να είναι δυνατή η χρήση του, πολλές σημαντικές λειτουργίες ασφάλειας έχουν απενεργοποιηθεί. Προορίζεται κυρίως για όσους θέλουν εύκολα και γρήγορα να εγκαταστήσουν όλα τα απαραίτητα εργαλεία για να αναπτύξουν και να δοκιμάσουν, τοπικά στον υπολογιστή τους, εφαρμογές για το διαδίκτυο. Υπάρχει ειδικό εργαλείο το οποίο περιέχεται στο XAMPP για την προστασία με κωδικό των σημαντικών μερών. Το XAMPP υποστηρίζει την δημιουργία και διαχείριση βάσεων δεδομένων τύπου MySQL και SQLite.

Όταν το XAMPP εγκατασταθεί στον τοπικό υπολογιστή διαχειρίζεται τον localhost ως ένα απομακρυσμένο κόμβο, ο οποίος συνδέεται με το πρωτόκολλο μεταφοράς αρχείων FTP. Η σύνδεση στον localhost μέσω του FTP μπορεί να γίνει με το όνομα χρήστη «newuser» και το κωδικό «xampp». Για την βάση δεδομένων MySQL υπάρχει ο χρήστης «root» χωρίς κωδικό πρόσβασης.[\[44\]](#)

5.6 XML

Η XML (eXtensible Markup Language) μας παρέχει τον τρόπο να δομήσουμε, να ανταλλάξουμε και να περιγράψουμε δεδομένα με ένα πλούσιο, ευέλικτο και αποτελεσματικό τρόπο. Αυτό το επιτυγχάνει με την χρήση περιγραφικών tags (ετικέτες) για τα δεδομένα. Σημαντικό είναι ακόμα και το γεγονός πως η δομή δεδομένων που υπόκειται ενός XML εγγράφου και είναι γνωστή ως δέντρο, είναι ιδανική για να συγκρατήσει την δομή διακριτών πηγών δεδομένων και αρκετά απλή για την επεξεργασία των δεδομένων που συγκρατεί, από αποδοτικούς και κομψούς αλγορίθμους (π.χ. αναδρομικούς).[\[45\]](#)

5.6.1 Πλεονεκτήματα XML

Υπάρχουν πολλοί λόγοι για τους οποίους μπορεί κανείς σήμερα να χρησιμοποιήσει την XML. Η XML είναι ένας από τους πιο δημοφιλείς τρόπους για την αναπαράσταση και την περιγραφή δεδομένων και θα δούμε μερικές ιδιότητες, οι οποίες είναι πολύ χρήσιμες.

- Η XML είναι εύκολα αναγνώσιμη από ανθρώπους και μηχανές: Οι περισσότερες μορφές αποθήκευσης δεδομένων ήταν είτε κατάλληλες για μετάφραση από προγράμματα λογισμικού (π.χ. dBase, GIF), ή αναγνώσιμα από ανθρώπους (text ή CSV αρχεία). Η XML ορίζει ένα σύνολο από κανόνες που κάνουν την μετάφραση από υπολογιστή πολύ απλή. Έτσι ικανοποιούνται και οι δύο πλευρές, αφού τα XML έγγραφα διατηρούν ως βάση τους το κείμενο κι έτσι μπορεί εύκολα να τα χειριστεί ένας άνθρωπος.
- Η XML είναι φιλική στα αντικείμενα (object-friendly): Ενώ το σχεσιακό μοντέλο δεδομένων εμφανίζει μεγάλη επιτυχία για την επεξεργασία μεγάλων ποσοτήτων δεδομένων αποθηκευμένων σε πίνακες, ο χειρισμός άλλων τύπων δεδομένων - όπως είναι το hypertext (κείμενο με hyperlinks), πολυμέσα, γραφικά, μαθηματικές ή χημικές φόρμουλες, ιεραρχική πληροφορία – δεν είναι τόσο απλός. Η XML από την άλλη πλευρά είναι φιλική στα αντικείμενα, υπό την έννοια ότι είναι κατάλληλη για την περιγραφή

αντικειμένων του πραγματικού κόσμου ή οποιουδήποτε αφαιρετικού προβλήματος μοντελοποιώντας τις ιδιότητες όπως ακριβώς είναι, αντί να χρειάζεται μια κανονικοποιημένη διάσπαση σε διάφορους πίνακες, με τους οποίους συνδέονται διάφορες σχέσεις. Αυτό κάνει τα XML έγγραφα περισσότερο κατανοητά κι έτσι μειώνεται ο χρόνος που απαιτείται για την σχεδίαση και υλοποίηση υπολογιστικών συστημάτων που βασίζονται στην XML.

- Η XML έχει ευρέως υιοθετηθεί από την βιομηχανία υπολογιστών: Η XML είναι ευρέως αποδεκτή και υλοποιείται από πολλές εταιρίες. Το γεγονός αυτό έχει ως αποτέλεσμα χαμηλότερο κόστος για όλα τα συστατικά του λογισμικού.
- Η XML είναι παγκόσμια: Για να γίνει ευκολότερα κατανοητός ο λόγος για τον οποίο η XML έγινε τόσο αποδεκτή, θα ήταν χρήσιμο να αναφερθούμε στον ASCII κώδικα (American Standard Code for Information Interchange), ο οποίος είναι επίσης ιδιαίτερα αποδεκτός. Παρόλο που ο ASCII κώδικας διαθέτει ένα συγκεκριμένο αλφάβητο και σύστημα γραφής, ήταν απαραίτητο να επιτρέψει την ελεύθερη ανταλλαγή δεδομένων μεταξύ διαφορετικών τύπων υπολογιστών και λειτουργικών συστημάτων. Η ιδέα του ASCII επεκτάθηκε σε τέτοιο βαθμό ώστε να συμπεριλάβει όλες τις γλώσσες και όλα τα συστήματα γραφής του κόσμου. Σήμερα θεωρούμε ως δεδομένο ότι οι υπολογιστές μπορούν να διαβάζουν και να επεξεργάζονται έγγραφα κειμένου που βασίζονται στον ASCII κώδικα. Η XML επεκτείνει αυτή τη προσέγγιση χρησιμοποιώντας το Unicode και ορίζοντας ένα καθολικό τρόπο για την περιγραφή δομημένων δεδομένων για κάθε διαφορετικό σκοπό. Όλα τα XML έγγραφα είναι εξ' ορισμού βασισμένα σε Unicode, αλλά μπορούν να αποθηκευτούν στο δίσκο ή να μεταδοθούν σε ένα δίκτυο με διάφορες κωδικοποιήσεις όπως ISO-8859-1 ή UTF-8. Αυτός είναι και ένας λόγος που μερικοί σήμερα καλούν την XML "ASCII του μέλλοντος".

5.6.2 Ένα XML έγγραφο σαν δενδρική αναπαράσταση

Ένα XML έγγραφο μπορεί να αναπαρασταθεί σαν ένα δέντρο, του οποίου οι κόμβοι περιέχουν τα δομικά συστατικά του εγγράφου, δηλαδή τα elements, το κείμενο, τα attributes, τα σχόλια και τις processing instruction. Η δενδρική αναπαράσταση ενός XML εγγράφου θυμίζει την δομή των αρχείων και των φακέλων στον σκληρό δίσκο ενός υπολογιστή. Οι κόμβοι είναι πανομοιότυποι με τα αρχεία και τους φακέλους. Έτσι όπως οι φάκελοι μπορεί να περιέχουν άλλα αρχεία και φακέλους, έτσι και ένας κόμβος μπορεί να περιέχει άλλους κόμβους.

5.7 Xpath

Εντούτοις, παρά την ευελιξία που μας παρέχει για την περιγραφή και την ανταλλαγή δεδομένων και πληροφορίας, η XML δεν μας παρέχει τον τρόπο να εντοπίσουμε συγκεκριμένα κομμάτια δομημένων δεδομένων μέσα σε ένα έγγραφο. Έτσι για παράδειγμα, ένα XML έγγραφο που περιέχει δεδομένα σχετικά με διάφορα βιβλία κάποιου εκδοτικού οίκου, θα χρειαστεί να το ψάξουμε στοιχείο προς στοιχείο, προκειμένου να εντοπίσουμε κάποιο συγκεκριμένο βιβλίο που επιθυμούμε να βρούμε. Η διαδικασία αυτή όπως γίνεται εύκολα αντιληπτό, είναι χρονοβόρα και αναποτελεσματική, ιδίως όταν πρόκειται για μεγάλα XML έγγραφα.

Έχοντας μεγάλες ποσότητες δεδομένων να αναπαρίστανται ως XML έγγραφα, η ανάγκη να υπάρξει κάποιος αποτελεσματικός τρόπος αναζήτησης και εύρεσης των δεδομένων ήταν ιδιαίτερως επιτακτική. Για τον λόγο αυτό το W3C (World Wide Web Consortium) πρότεινε την XPath (XML Path Language).

Η XPath παρέχει την σύνταξη για τον αποτελεσματικό εντοπισμό συγκεκριμένων τμημάτων μέσα στο XML έγγραφο (π.χ. τις τιμές των attributes). Η XPath δεν είναι μια δομική γλώσσα όπως είναι η XML. Βασικά

πρόκειται για μία γλώσσα από εκφράσεις με συγκεκριμένη σύνταξη που χρησιμοποιείται από άλλες τεχνολογίες της XML, όπως η XSLT (Extendible Stylesheet Language Transformations) η οποία μετασχηματίζει ή μετατρέπει τα XML έγγραφα σε άλλες μορφές για παράδειγμα σε HTML και τον XPointer (XML Pointer Language), ο οποίος περιγράφει μηχανισμούς για τον εντοπισμό συγκεκριμένων τμημάτων μέσα σε ένα XML έγγραφο.

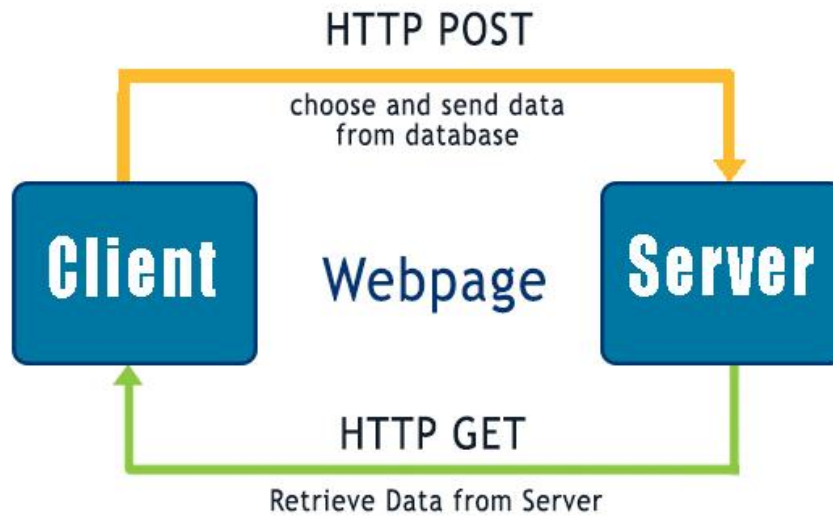
Για την υλοποίηση των εκφράσεων της XPath εκμεταλλευόμαστε τη δενδρική δομή των XML εγγράφων, καθώς στην XPath αντιμετωπίζουμε τα XML έγγραφα σαν να είναι δέντρα στα οποία το κάθε τμήμα του εγγράφου παριστάνεται ως κόμβος.[\[46\]](#)

5.8 Μέθοδοι GET και POST του HTTP πρωτοκόλλου

Το πρωτόκολλο HTTP (Hypertext Transfer Protocol - Πρωτόκολλο Μεταφοράς Υπερκειμένου) είναι σχεδιασμένο να επιτρέπει την επικοινωνία μεταξύ πελατών (clients) και εξυπηρετητών (servers).

Το HTTP λειτουργεί σαν ένα πρωτόκολλο αιτήματος-απόκρισης μεταξύ ενός πελάτη και ενός εξυπηρετητή. Ο πελάτης μπορεί να είναι ένας φυλλομετρητής σελίδων (web browser), ενώ ο εξυπηρετητής μπορεί να είναι μία εφαρμογή σε έναν υπολογιστή που φιλοξενεί έναν ιστότοπο.

Δύο ευραίως χρησιμοποιούμενες μέθοδοι του HTTP πρωτοκόλλου για τη διαδικασία αιτήματος-απόκρισης μεταξύ ενός πελάτη και ενός εξυπηρετητή είναι οι GET and POST.[\[47\]](#)



Εικόνα 31 Get και Post μέθοδος HTTP πρωτοκόλλου

Οι μέθοδοι αυτές μας είναι απαραίτητες στην εφαρμογή μας για την ανάκτηση και δημοσίευση πληροφοριών.

5.8.1 Μέθοδος GET

Η μέθοδος GET χρησιμοποιείται για ανάκτηση πληροφορίας από ένα συγκεκριμένο server χρησιμοποιώντας ένα συγκεκριμένο URI. Οι αιτήσεις που χρησιμοποιούν GET μπορούν μόνο να ανακτούν δεδομένα και δεν μπορούν να έχουν καμία επίδραση στα δεδομένα. Μία GET μέθοδος ανακτά δεδομένα από ένα διακομιστή ιστού καθορίζοντας τις παραμέτρους στο τμήμα URL της αίτησης. Αυτή είναι η κύρια μέθοδος που χρησιμοποιείται για ανάκτηση εγγράφων.

5.8.2 Μέθοδος POST

Ένα POST αίτημα χρησιμοποιείται για να αποστείλει δεδομένα στο διακομιστή, για παράδειγμα πληροφορίες για τους πελάτες, αποστολής αρχείων κ.ά. χρησιμοποιώντας HTML φόρμες. Η μέθοδος POST

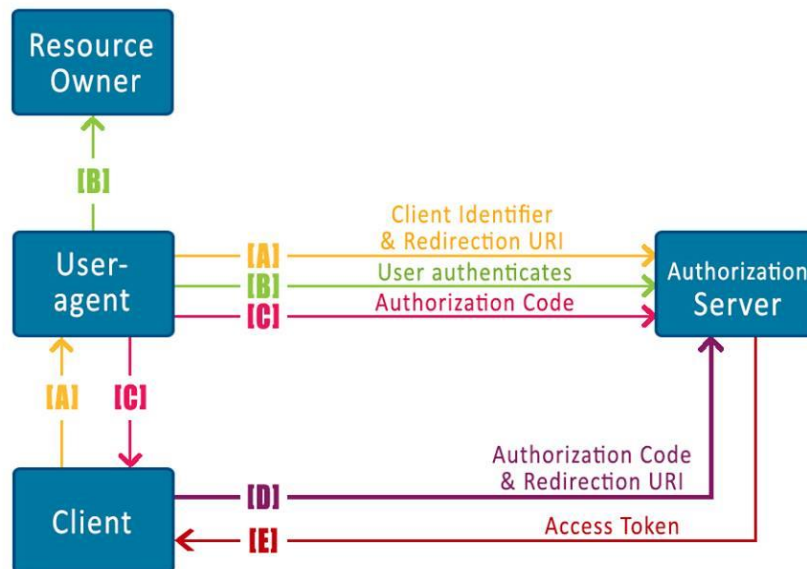
χρησιμοποιείται όταν θέλουμε να στείλουμε μερικά δεδομένα στο διακομιστή, για παράδειγμα ενημέρωση ενός αρχείου, μορφή δεδομένων κ.ά..

5.9 Auth 2.0 protocol

Για την επικοινωνία μεταξύ της εφαρμογής που αναπτύξαμε στο Facebook με τους χρήστες της εφαρμογής αυτής χρησιμοποιήθηκε το OAuth 2.0 πρωτόκολλο. Το πρωτόκολλο αυτό είναι υπεύθυνο για την αυθεντικοποίηση των χρηστών και την εξουσιοδότηση της εφαρμογής.

5.9.1 Διαδικασία αυθεντικοποίησης χρηστών

Σύμφωνα με την παραπάνω τεκμηρίωση του πρωτοκόλλου Auth 2.0, για την υλοποίηση της αυθεντικοποίησης των χρηστών και την εξουσιοδότηση των εφαρμογών ακολουθούνται τα εξής βήματα:



Εικόνα 32 Βήματα αυθεντικοποίησης χρηστών & εξουσιοδότησης εφαρμογών

A. Ο πελάτης (client) ξεκινάει τη ροή με τη διοχέτευση του χρήστη-πράκτορα (user agent) του ιδιοκτήτη των πόρων (resource owner) στο τελικό σημείο της εξουσιοδότησης. Ο πελάτης συγκεντρώνει το αναγνωριστικό πελάτη, το ζητούμενο πεδίο εφαρμογής, την τοπική κατάσταση, και το URI ανακατεύθυνσης, τα οποία στέλνει στο διακομιστή εξουσιοδότησης (authorization server) για να αποφασιστεί αν θα χορηγηθεί ή όχι πρόσβαση στο χρήστη-πράκτορα (user agent).

B. Ο διακομιστής εξουσιοδότησης ελέγχει την ταυτότητα του ιδιοκτήτη των πόρων (μέσω του χρήστη-πράκτορα) και καθορίζει αν ο ιδιοκτήτης των πόρων εγκρίνει ή απορρίπτει την αίτηση πρόσβασης του πελάτη.

Γ. Υποθέτοντας ότι ο ιδιοκτήτης πόρων επιτρέπει την πρόσβαση, ο διακομιστής εξουσιοδότησης ανακατευθύνει το χρήστη-πράκτορα πίσω στον πελάτη χρησιμοποιώντας το URI ανακατεύθυνσης που δόθηκε νωρίτερα. Το URI ανακατεύθυνσης περιλαμβάνει και έναν κωδικό εξουσιοδότησης.

Δ. Ο πελάτης ζητά ένα access token από το τελικό σημείο token του διακομιστή εξουσιοδότησης, ελέγχοντας την ταυτότητα με τη χρήση των διαπιστευτηρίων του πελάτη, και το οποίο περιλαμβάνει τον κωδικό εξουσιοδότησης που λήφθηκε στο προηγούμενο βήμα.

E. Ο διακομιστής εξουσιοδότησης επικυρώνει τα διαπιστευτήρια των πελατών και τον κωδικό εξουσιοδότησης και αν ισχύουν, απαντά με ένα access token.[\[48\]](#)

5.9.2 Αυθεντικοποίηση στο Facebook

Το Facebook χρησιμοποιεί το OAuth 2.0 πρωτόκολλο για την αυθεντικοποίηση και την εξουσιοδότηση του χρήστη. Υποστηρίζονται μια σειρά από διαφορετικές ροές OAuth που μπορούν να χρησιμοποιηθούν για την ανάπτυξη μίας ιστοσελίδας ή εφαρμογής. Οι μηχανισμοί που χρησιμοποιούνται για την υποστήριξη κάθε μίας από αυτές τις ροές είναι διαφορετικοί. Οι δύο κύριες ροές που χρησιμοποιούνται για τη σύνδεση (login) του χρήστη στη πλατφόρμα του Facebook είναι η server-side ροή (σύμφωνα

με τις προδιαγραφές γνωστή και ως ροή αυθεντικοποίησης ή ελέγχου ταυτότητας), η οποία βασίζεται στη γλώσσα PHP και η client-side ροή (γνωστή και ως έμμεση ροή), η οποία βασίζεται σε HTML / Javascript. Οι ροές αυτές είναι απλές και μετασχηματίζονται εύκολα σε άλλες γλώσσες προγραμματισμού. Η server - side ροή χρησιμοποιείται κάθε φορά που απαιτείται να γίνει κλήση στο Graph API από τον web server. Η client-side ροή χρησιμοποιείται όταν απαιτείται να γίνει κλήση στο Graph API απο έναν client, όπως για παράδειγμα JavaScript που εκτελείται σε έναν Web browser ή από ένα κινητό ή μια desktop εφαρμογή.[\[49\]](#)

Ανεξάρτητα από τη ροή που χρησιμοποιείται, η υλοποίηση του OAuth 2.0 περιλαμβάνει τρία διαφορετικά στάδια:

- την αυθεντικοποίηση του χρήστη,
- την εξουσιοδότηση της εφαρμογής και
- την αυθεντικοποίηση της εφαρμογής.

Η αυθεντικοποίηση του χρήστη εξασφαλίζει ότι ο χρήστης είναι αυτός που ισχυρίζεται ότι είναι. Η εξουσιοδότηση της εφαρμογής χορηγεί άδεια στο χρήστη και εξασφαλίζει πως ο χρήστης γνωρίζει ακριβώς ποια δεδομένα, και ποιες δυνατότητες πάνω στα δεδομένα, του παρέχει η εφαρμογή. Η αυθεντικοποίηση της εφαρμογής εξασφαλίζει ότι ο ίδιος ο χρήστης δίνει τις πληροφορίες του στην εφαρμογή και όχι κάποιος άλλος.

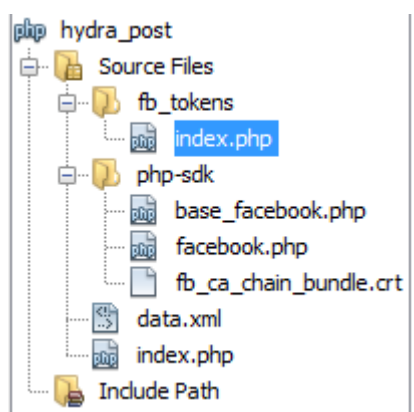
Μόλις αυτά τα βήματα ολοκληρωθούν, στην εφαρμογή εκδίδεται ένα πιστοποιητικό (user access token) που επιτρέπει στον δημιουργό της εφαρμογής να έχει πρόσβαση σε πληροφορίες του χρήστη και να αναλαμβάνει δράση για λογαριασμό του.

5.10 Τεχνολογίες για τη συγγραφή και τον προγραμματισμό της πτυχιακής εργασίας

Τέλος, θα θέλαμε να αναφερθούμε στις τεχνολογίες που χρησιμοποιήσαμε για τη συγγραφή αυτής της εργασίας. Για την συγγραφή του κειμένου, χρησιμοποιήσαμε το Google Drive. Το Google Drive είναι ένα ασφαλές μέρος για να αποθηκεύει κανείς φωτογραφίες, βίντεο, έγγραφα και άλλα αρχεία. Επίσης, δίνει τη δυνατότητα να μοιράζεται κανείς αυτά τα αρχεία με φίλους, οικογένεια και συναδέλφους. Είναι ένας καλός τρόπος για να συγγράφουν δύο άτομα παράλληλα. Και μας βοήθησε ιδιαίτερα, γιατί δεν είχαμε την δυνατότητα να εργαζόμαστε μαζί στον ίδιο χώρο. Για το προγραμματιστικό κομμάτι της εργασίας, χρησιμοποιήσαμε το ολοκληρωμένο περιβάλλον ανάπτυξης NetBeans σε συνδυασμό με το λογισμικό συστήματος ελέγχου GIT. Το Git επιτρέπει σε μία ομάδα ατόμων να δουλεύουν σε συνεργασία, χρησιμοποιώντας τα ίδια αρχεία. Επίσης, επιτρέπει να υπάρχουν “εκδοχές” του προγράμματος, όπου φαίνονται οι αλλαγές που έχουν γίνει στον κώδικα στο πέρασμα του χρόνου και επιτρέπουν την επιστροφή σε προηγούμενη εκδοχή αν κριθεί απαραίτητο για να αναιρεθούν κάποιες αλλαγές.

6 Ανάπτυξη-Ανάλυση ενδεικτικών συστατικών της υπηρεσίας

Σε αυτό το κεφάλαιο, θα αναλύσουμε τα βασικά συστατικά της εφαρμογής μας. Θα περιγράψουμε βήμα βήμα την εκτέλεση του προγράμματός μας. Στην παρακάτω εικόνα, βλέπουμε την δομή του project μας, που ονομάζεται `hydra_post`:



Εικόνα 33 Δομή του project `hydra_post`

Όπως παρατηρούμε και στην εικόνα, υπάρχει η `index.php`, η οποία είναι η κεντρική μας κλάση και μέσα στο φάκελο `php-sdk`, υπάρχουν οι βιβλιοθήκες του Facebook που χρησιμοποιήσαμε. Αρχικά, θα αναλύσουμε και θα περιγράψουμε την κύρια κλάση μας και μετέπειτα, θα επικεντρωθούμε στα κομμάτια κώδικα που μας φάνηκαν χρήσιμα από τις βιβλιοθήκες.

6.1 Η κύρια κλάση `index.php`

Ο σκοπός της εφαρμογής μας ήταν να δημοσιεύουμε αυτόματα στο Facebook τις δημόσιες ανακοινώσεις του Τμήματος Μηχανικών Πληροφορικής. Αρχικά, λοιπόν, πρέπει να αντλήσουμε την πληροφορία που θέλουμε και να την επεξεργαστούμε, έτσι ώστε να πάρει την μορφή που εμείς επιθυμούμε. Έπειτα, ακολουθεί η δημοσίευση αυτής της πληροφορίας στο κοινωνικό δίκτυο.

Αρχικά, απαιτούμε το αρχείο `facebook.php` να αντιγραφεί στο τρέχον αρχείο, όπως φαίνεται παρακάτω.

```
<?php  
require_once 'php-sdk/facebook.php';
```

Όταν κάνουμε `include` ή `require` δήλωση, σημαίνει ότι παίρνει όλο το κείμενο / κώδικα που υπάρχει στο συγκεκριμένο αρχείο και το αντιγράφει μέσα στο αρχείο που χρησιμοποιεί την `include` ή `require` δήλωση. Αυτές οι δύο δηλώσεις είναι πανομοιότυπες, με εξαίρεση τι συμβαίνει όταν αποτυγχάνουν:

- η `require` θα παράγει ένα μοιραίο λάθος (`fatal error`) και θα σταματήσει το script
- η `include` θα παράγει μόνο μία προειδοποίηση (`warning`) και το script θα συνεχίσει να εκτελείται.

Εμείς χρησιμοποιούμε την `require` καθώς η βιβλιοθήκη αυτή απαιτείται από την εφαρμογή μας και χωρίς αυτή, το πρόγραμμά μας δεν θα λειτουργούσε.

Στη συνέχεια, δημιουργούμε τρεις μεταβλητές και τις αρχικοποιούμε. Στην πρώτη, που την ονομάσαμε `hydraBaseLink`, σώζουμε το `base url` της πηγής της πληροφορίας μας. Οι άλλες δύο μεταβλητές χρησιμοποιούνται από εμάς για να τεστάρουμε τον κώδικα και τις αρχικοποιούμε δίνοντας τους μηδενική τιμή.

```
$hydraBaseLink = "http://hydra.it.teithe.gr";  
$s = 0;  
$f = 0;
```

Στη συνέχεια, ξεκινάμε μία σύνοδο (`session`) και χρησιμοποιούμε μια μεταβλητή συνόδου, την οποία την θέλουμε για να αποθηκεύσουμε πληροφορίες σχετικά με τις λειτουργίες μιας συγκεκριμένης μεταβλητής. Ο πίνακας `_SESSION`, λοιπόν, χρησιμοποιείται για να διατηρήσουμε την τιμή της μεταβλητής `lastPost` από εκτέλεση σε εκτέλεση του προγράμματός μας. Αν έχει οριστεί αυτή η μεταβλητή συνόδου, αποθηκεύουμε στην μεταβλητή `lastPost` την τιμή της μεταβλητής συνόδου, αλλιώς η μεταβλητή `lastPost` μένει κενή. Η γραμμή που είναι σχόλιο χρησιμοποιείται από εμάς για να τρέξει το πρόγραμμα για πρώτη φορά.

```
session_start();  
  
if (isset($_SESSION['lastPost'])) {  
    $lastPost = $_SESSION['lastPost'];  
} else {  
    $lastPost = "";  
}  
  
//$lastPost = "";
```

Έπειτα, δημιουργούμε το *hydraContents*, το οποίο είναι ένα αντικείμενο dom document και θα το χρησιμοποιήσουμε για να αποθηκεύσουμε τα περιεχόμενα της ύδρας υπό μορφή HTML.

```
$hydraContents = new DOMDocument;
```

Φορτώνουμε το περιεχόμενο της ύδρας σε μορφή HTML χρησιμοποιώντας την μέθοδο *file_get_contents* και το αποθηκεύουμε στο αντικείμενο *hydraContents*. Η μέθοδος αυτή μετατρέπει ένα αρχείο σε ένα string. Μετά, σώζουμε στο *data.xml* τα περιεχόμενα του *hydraContents* υπό μορφή XML, χρησιμοποιώντας τη μέθοδο *file_put_contents*, η οποία γράφει ένα string σε ένα αρχείο. Στη μεταβλητή *xml* φορτώνουμε τα περιεχόμενα του *data.xml* υπό μορφή SimpleXMLElement. Αυτό το κάνουμε χρησιμοποιώντας την μέθοδο *simplexml_load_file*, η οποία ερμηνεύει (interprets) ένα αρχείο XML σε ένα αντικείμενο. Η παράμετρος *LIBXML_NOCDATA* της μεθόδου αυτής θέτει τα CDATA (Character data) σαν κόμβους κειμένου. Ο όρος CDATA χρησιμοποιείται για δεδομένα κειμένου, τα οποία δεν μπορούν να αναλυθούν από τον XML parser (π.χ. "<", "&").

```
$hydraContents->loadHTML(file_get_contents($hydraBaseLink . "/s/index.php?m=itdep-bbpublic&bid=news"));  
file_put_contents('data.xml', $hydraContents->saveXml());  
$xml = simplexml_load_file('data.xml', 'SimpleXMLElement', LIBXML_NOCDATA);
```

Χρησιμοποιούμε XPath πάνω στη μεταβλητή *xml* και σώζουμε το αποτέλεσμα, αφού πρώτα το αντιστρέψουμε στη μεταβλητή *news*. Ας εξηγήσουμε λίγο παραπάνω τη σύνταξη που βλέπουμε στην XPath. Οι // σημαίνουν ότι θα επιλεχθούν οι κόμβοι στο έγγραφο από τον τρέχοντα κόμβο που ταιριάζουν στην επιλογή ανεξαρτήτως της θέσης τους μέσα στο έγγραφο. Στη συγκεκριμένη περίπτωση δηλαδή, επιλέγει όλα τα *div* στοιχεία ανεξαρτήτως που βρίσκονται μέσα στο έγγραφο. Το @ επιλέγει γνωρίσματα, στην περίπτωσή μας, επιλέγει τα *div* στοιχεία με γνώρισμα κλάσης που είναι *new*.

Δημιουργούμε μία μεταβλητή *flag* τη *foundLast* και τη θέτουμε *false* (δεν έχει βρεθεί η τελευταία ανακοίνωση). Στη μεταβλητή *lastSuccess* σώζουμε την τιμή της μεταβλητής *lastPost*.

```
$path = "//div[@class='new']";  
$news = array_reverse($xml->xpath($path));  
  
$foundLast = false;  
$lastSuccess = $lastPost;
```

Για κάθε *news* ξεχωριστά, μετράμε πόσα παιδιά έχει και σώζουμε τον αριθμό αυτόν στη μεταβλητή *countChildren*. Αν έχει τρία παιδιά, σημαίνει ότι στην ανακοίνωση δεν υπάρχει συνημμένο αρχείο και αποθηκεύουμε τον τίτλο, το κείμενο και την υπογραφή της ανακοίνωσης, ενώ αν έχει τέσσερα παιδιά υπάρχει συνημμένο αρχείο και πρέπει να αποθηκεύσουμε και το συνημμένο. Αυτά τα πεδία που είτε είναι τρία είτε τέσσερα, αποθηκεύονται στη μεταβλητή *post* και γίνεται ένωσή τους (*concatenate*).

```
foreach ($news as $currentNew) {  
    $countChildren = count($currentNew);  
  
    $post = $currentNew->div[0];  
    $post .= $currentNew->div[1];  
  
    if ($countChildren == 3) {  
        $post .= $currentNew->div[2];  
    } else {  
        $post .= $hydraBaseLink . $currentNew->div[2]->a["href"];  
        $post .= $currentNew->div[3];  
    }  
}
```

Αφού έχουμε αποθηκευμένες όλες τις δημοσιεύσεις με αντεστραμμένη σειρά μέσα στον πίνακα *new* (η πιο πρόσφατη ανακοίνωση βρίσκεται στην τελευταία θέση του πίνακα), αυτό που κάνουμε στην συνέχεια είναι να διατρέξουμε τον πίνακα μέχρι να βρούμε κάποια δημοσίευση που δεν έχει δημοσιευτεί και να την δημοσιεύσουμε. Στον πρώτο έλεγχο τις *if*, ελέγχουμε αν υπάρχει το *session*, αν είναι ορισμένο το *lastPost* (όταν τρέξαμε το πρόγραμμά μας για πρώτη φορά κράτησε στη *lastPost* την τελευταία ανακοίνωση που είχε δημοσιευτεί) και επιπλέον, αν η μεταβλητή *foundLast* έχει την τιμή *false*, δηλαδή αν δεν έχουμε βρει ακόμη την τελευταία ανακοίνωση. Εφόσον υπάρχει το *session* και ισχύουν και οι άλλες δύο συνθήκες, διατρέχουμε τον πίνακα ελέγχοντας αν η τιμή της μεταβλητής *post* είναι ίδια με την τιμή της *lastPost* ($\$post == lastPost$). Αν βρεθεί στον πίνακα η τελευταία δημοσιευμένη ανακοίνωση και δεν βρίσκεται στην τελευταία θέση του, σημαίνει πως υπάρχουν νέες ανακοινώσεις που δεν έχουν δημοσιευτεί. Αλλάζουμε την τιμή της *foundLast* σε *true* και προχωράμε στην δημοσίευσή τους. (Εφόσον η μεταβλητή *foundLast* έχει τώρα την τιμή *true*, την επόμενη φορά που θα μπει στο *loop* και θα κάνει τον έλεγχο της *if*, δεν θα ισχύει η συνθήκη, οπότε θα εκτελέσει την *else*, μέσα στην οποία κάνουμε την δημοσίευση στο Facebook).

```
if($_SESSION["lastPost"] != null && $post != $lastPost && $foundLast == false) {  
    continue;  
} elseif ($post == $lastPost) {  
    $foundLast = true;  
    continue;  
} else {
```

Στη συνέχεια λοιπόν για να δημοσιεύσουμε στο Facebook, πραγματοποιούμε την σύνδεση με την διεπαφή του Facebook.

Αποθηκεύουμε στην μεταβλητή *pageid* το page Id της σελίδας μας και δημιουργούμε έναν πίνακα με το όνομα *config*, μέσα στον οποίο αποθηκεύουμε τις τιμές των APP ID και APP SECRET (*'appId'* και *'secret'* αντίστοιχα), όπως ορίστηκαν όταν δημιουργήσαμε την facebook εφαρμογή μας. Έπειτα, δημιουργούμε ένα αντικείμενο το οποίο ονομάζουμε *fb* και στου οποίου το δομητή περνάμε ως παράμετρο τον πίνακα *config*. Με τον τρόπο αυτό, αρχικοποιούμε την facebook κλάση με τα δικά μας διαπιστευτήρια.

```
$pageid = '542273795822262';  
  
// initialize Facebook class using our own Facebook App credentials  
$config = array();  
$config['appId'] = '1377208439213066';  
$config['secret'] = 'xxxxxxxxxxxxxxxxxx';  
  
$fb = new Facebook($config);
```

Ακολουθεί η δημιουργία του πίνακα *params* που περιέχει το μήνυμα που θέλουμε να δημοσιεύσουμε και το *access token* για να μπορέσει η εφαρμογή να δημοσιεύσει το μήνυμα στην σελίδα. Στη μεταβλητή *access token* εναποθέτουμε το access token της σελίδας μας (το οποίο έχουμε περιγράψει σε παραπάνω κεφάλαιο πως αποκτήσαμε) και στη μεταβλητή *message* περνάμε την τιμή που περιέχει η μεταβλητή *post*.

Στη συνέχεια, μέσα στο μπλοκ εντολής `try{}` πραγματοποιούμε την κλήση στο Graph API του Facebook. Η κλήση του API πραγματοποιείται με την εντολή `$ret = $fb->api('/. $pageid . '/feed', 'POST', $params);`

Το `/pageid /feed` είναι το `feed` της σελίδας μας, όπου με την μέθοδο 'POST' δημοσιεύεται το περιεχόμενο του πίνακα `params`.

Επιπλέον, κάνουμε `echo` ένα μήνυμα για να ελέγξουμε αν η ανακοίνωση δημοσιεύτηκε επιτυχώς και βάζουμε την τιμή του `post` στην μεταβλητή `lastSuccess`. Ακόμη, η μεταβλητή `s` την οποία χρησιμοποιούμε ως μετρητή των επιτυχημένων δημοσιεύσεων, αυξάνεται κατά 1.

Στην περίπτωση που αποτύχει το μπλοκ `try`, η εκτέλεση του προγράμματος οδηγείται στο μπλοκ `catch`, το οποίο παράγει `exception` με το οποίο μας επιστρέφεται ένα μήνυμα λάθους και επίσης, η μεταβλητή `f` που χρησιμοποιείται ως μετρητής των αποτυχημένων δημοσιεύσεων αυξάνεται κατά 1.

```
// define your POST parameters
$params = array(
    "access_token" => "CAATkkFPNDaoBAFDZCx44V7p7hQQPotFDvCeOEWxP08q4sekBBAg1WDdVn",
    "message" => $post,
);

try {
    $ret = $fb->api('/'. $pageid .'/feed', 'POST', $params);
    echo 'Successfully posted to Facebook';
    $lastSuccess = $post;
    $s++;
} catch(Exception $e) {
    echo $e->getMessage();
    $f++;
}
}
```

Τέλος, ενημερώνουμε την μεταβλητή `lastPost` στο `session`, δίνοντας της την τιμή της `lastSuccess`, έτσι ώστε να περιέχει την τελευταία ανακοίνωση που δημοσιεύτηκε την επόμενη φορά που θα τρέξει το πρόγραμμα. Χρησιμοποιούμε, επίσης, μία `if` για να ελέγξουμε τον κώδικα μας με χρήση των μεταβλητών `s` και `f`, που τις έχουμε ως μετρητές επιτυχημένων και αποτυχημένων δημοσιεύσεων αντίστοιχα. Λέμε πώς αν `$s+$f` ισούται με μηδέν, δηλαδή αν δεν υπάρχουν ούτε επιτυχημένες ούτε αποτυχημένες δημοσιευμένες ανακοινώσεις (που πρακτικά σημαίνει πως δεν υπάρχουν στην σελίδα της ύδρας νέες ανακοινώσεις) να μας εμφανίσει ένα μήνυμα που

να λέει ότι δεν υπάρχουν νέες ανακοινώσεις. Διαφορετικά, μπαίνει στην else και μας εμφανίζει πόσες επιτυχημένες και πόσες αποτυχημένες δημοσιεύσεις πραγματοποιήθηκαν.

```
$_SESSION["lastPost"] = $lastSuccess;
if ($s+$f==0){
    echo "There are no new announcements!";
} else {
    echo "Succeeded posting: ".$s."<br>Failed Posting: ".$f;
}
```

6.2 Βιβλιοθήκη του Facebook

Από τη βιβλιοθήκη του Facebook, χρησιμοποιήσαμε ουσιαστικά τον πίνακα config και μία μέθοδο για την κλήση στο API.

Στον πίνακα config δηλώνουμε τα διαπιστευτήρια της εφαρμογής μας, δηλαδή το app id και το secret id της facebook app. Στη συνέχεια, αυτός ο πίνακας περνιέται ως παράμετρος στο δομητή της κλάσης Facebook.

```
public function __construct($config) {
    if ((function_exists('session_status')
        && session_status() !== PHP_SESSION_ACTIVE) || !session_id()) {
        session_start();
    }
    parent::__construct($config);
    if (!empty($config['sharedSession'])) {
        $this->initSharedSession();

        // re-load the persisted state, since parent
        // attempted to read out of non-shared cookie
        $state = $this->getPersistentData('state');
        if (!empty($state)) {
            $this->state = $state;
        } else {
            $this->state = null;
        }
    }
}
```

Στη συνέχεια, κάναμε κλήση στο API χρησιμοποιώντας τη μέθοδο *api()*. Η κλήση γίνεται με την εξής εντολή:

```
$ret = $facebook->api($path, $method, $params);
```

Η πρώτη παράμετρος είναι απαραίτητη, ενώ οι άλλες δύο είναι προαιρετικές. Στην παρακάτω εικόνα, φαίνεται η περιγραφή της κάθε παραμέτρου.

ΟΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ
path	Το Graph API μονοπάτι του αιτήματος.
method	(προαιρετικό) Καθορίζει την HTTP μέθοδο που χρησιμοποιείται για το αίτημα: GET, POST, DELETE.
params	(προαιρετικό) Παράμετροι ειδικά για τη συγκεκριμένη μέθοδο του Graph API που καλούμε.

Εικόνα 34 Περιγραφή παραμέτρων της μεθόδου *api()*

```
public function api( /* polymorphic */ ) {
    $args = func_get_args();
    if (is_array($args[0])) {
        return $this->_restserver($args[0]);
    } else {
        return call_user_func_array(array($this, '_graph'), $args);
    }
}
```

6.3 Output της υπηρεσίας μας

Τέλος, παραθέτουμε μία εικόνα του τελικού αποτελέσματος της εκτέλεσης της υπηρεσίας μας, όπου φαίνονται δημοσιευμένες οι πιο πρόσφατες ανακοινώσεις.

Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙΘ
Community

7 likes
Bot Bot, Iwannoula Theodoridou and 2 others like this.

Invite your friends to like Τμήμα Μηχανικών Πληρο...

PEOPLE

- Panagiotis Sfetsos
- Victoria Rizouli

See All Friends

ABOUT

- Τα δημόσια νέα του τμήματος Μηχανικών Πληροφορικής του ΑΤΕΙΘ
- <http://www.it.teithe.gr/>
- Suggest Edits

PHOTOS

Timeline | About | Photos | Likes

Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙΘ
23 June

Μικρή διόρθωση στο πρόγραμμα εξετάσεων εργαστηρίων

Η εξέταση των μαθημάτων της Τετάρτης 28 Μαΐου "Αντικειμενοστρεφής Προγραμματισμός" (Επώνυμα από Α μέχρι Λ) και "Οργάνωση και Αρχιτεκτονική Υπολογιστικών Συστημάτων" θα πραγματοποιηθεί στις 12.00 (αντί στις 11.00 που είχε ανακοινωθεί). Όλα τα άλλα ισχύουν όπως είχαν ανακοινωθεί.
bbfile&build=5585&what=d
Wed 21 May 2014 09:39:37 PM EEST - by Kostoglou Basileios

Like · Comment · Share

Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙΘ
23 June

Αλλαγές στο ωράριο διδασκαλίας θεωριών από 27/5 έως 6/6

Κατά τη διάρκεια της εξεταστικής περιόδου των εργαστηριακών μαθημάτων θα υπάρξουν οι ακόλουθες αλλαγές στο ωράριο διεξαγωγής ορισμένων θεωρητικών μαθημάτων.
Μεταβολές στο ωράριο διδασκαλίας θεωρητικών μαθημάτων κατά τη διάρκεια της περιόδου διεξαγωγής των εξετάσεων των εργαστηρίων (... See More

Like · Comment · Share

Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙΘ
23 June

Πρόγραμμα Α' εξεταστικής θεωρητικών μαθημάτων - 11/6-4/7/2014

Η Α' εξεταστική περίοδος των θεωρητικών μαθημάτων θα πραγματοποιηθεί στο χρονικό διάστημα 11 Ιουνίου 2014 έως 4 Ιουλίου 2014 σύμφωνα με το συνημμένο ωρολόγιο πρόγραμμα.
Σύμφωνα με απόφαση της Συνέλευσης του Τμήματος :
- Θα εξετασθούν οι θεωρίες όλων των μαθημάτων του εαρινού κ... See More

Like · Comment · Share

Τμήμα Μηχανικών Πληροφορικής, ΑΤΕΙΘ
23 June

Προς τους φοιτητές και φοιτήτριες του Τμήματος (συνέχεια προηγούμενης)

2ο συνημμένο αρχείο
<http://hydra.it.teithe.gr/s/index.php?nocache=1403480267&m=itdep-bbfile&build=5542&what=d>
Tue 13 May 2014 07:53:13 PM EEST - by Kostoglou Basileios

Like · Comment · Share

Εικόνα 35 Screenshot της υπηρεσίας μας

7 Καινοτομία και προστιθέμενη αξία της υπηρεσίας μας

Σε αυτό το κεφάλαιο, θα αναζητήσουμε την καινοτομία της υπηρεσίας μας και τι καινούριο προσφέρει. Στο τρίτο κεφάλαιο, μελετήσαμε και αξιολογήσαμε τις υφιστάμενες λύσεις που υπάρχουν. Είδαμε ότι οι περισσότερες από αυτές παρέχουν τις υπηρεσίες τους επί πληρωμή και δεν είναι επεκτάσιμες. Επίσης, παρέχουν ενημέρωση προφίλ σε παραπάνω από ένα κοινωνικά δίκτυα. Η πηγή της πληροφορίας τους μπορεί να είναι περιεχόμενο ενός ιστολογίου (blog), RSS ή ένα μήνυμα ηλεκτρονικού ταχυδρομείου.

Στη δική μας υπηρεσία, αντιθέτως, παρατηρούμε κάτι καινοτόμο. Η πηγή της πληροφορίας μας είναι μια ιστοσελίδα (ύδρα), η οποία περιέχει στατική HTML. Με αυτόν τον τρόπο, προσθέτουμε αξία στην ύδρα, καθώς την αξιοποιούμε με τέτοιο τρόπο, ώστε ο χρήστης να ενημερώνεται αυτόματα μέσω των ενημερώσεων που δημοσιεύονται στο Facebook. Η υπηρεσία μας παρέχεται δωρεάν και ο καθένας μπορεί να την επεκτείνει, έτσι ώστε να παρέχει ακόμη περισσότερες υπηρεσίες, όπως τη δημοσίευση περαιτέρω δημόσιων ανακοινώσεων (για παράδειγμα ανακοινώσεις γραμματείας, εκδηλώσεις που γίνονται στο τμήμα κ.ά.) και βέβαια, την δημοσίευση τους σε περισσότερα κοινωνικά δίκτυα, ώστε να καλύπτεται μεγαλύτερο φάσμα χρηστών (καθώς δεν χρησιμοποιούν όλοι μόνο το Facebook).

Στο επόμενο κεφάλαιο, θα μελετήσουμε πώς μπορεί κανείς να επεκτείνει την υπηρεσία μας στο μέλλον.

8 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Η εφαρμογή μας προσφέρει μια καινοτόμα λύση για την ενημέρωση των συμφοιτητών μας, ωστόσο στο μέλλον θα μπορούσε να επεκταθεί τόσο ως προς την πηγή που θα λαμβάνει τα δεδομένα, όσο και προς την επεξεργασία τους και την πλατφόρμα που χρησιμοποιεί για να τα δημοσιεύσει.

Στην παρούσα της μορφή η εφαρμογή μας λαμβάνει δεδομένα από την ιστοσελίδα <http://hydra.it.teithe.gr/s/index.php?m=itdep-bbpublic&bid=news>, η οποία όμως έχει συγκεκριμένη δομή. Σε περίπτωση που συμβούν αλλαγές στη σελίδα οι οποίες θα μεταβάλλουν και τον πηγαίο της κώδικα, η εφαρμογή θα είναι ανίκανη να λάβει τα δεδομένα, καθώς λειτουργεί βασιζόμενη στην παρούσα δομή του πηγαίου κώδικα της σελίδας. Θα χρειαστούν κάποιες μικρές αλλαγές για να λειτουργήσει σωστά. Επειδή όμως η εφαρμογή μετατρέπει την πληροφορία που λαμβάνει σε XML έγγραφο, και στη συνέχεια το επεξεργάζεται για να δημοσιεύσει την ζητούμενη πληροφορία, μία εύκολη επέκταση της θα ήταν να λειτουργεί με είσοδο απ' ευθείας XML αρχείων, εφόσον αυτά γινόντουσαν διαθέσιμα από την σελίδα.

Επίσης, ανάλογα βέβαια και με τον τρόπο που τα δεδομένα προστίθενται αρχικά στην σελίδα, θα μπορούσε η εφαρμογή να λαμβάνει τα δεδομένα αυτά από SQL πίνακες που θα είναι αποθηκευμένοι στον διακομιστή όπου φιλοξενείται η ιστοσελίδα. Η λειτουργία αυτή θα πραγματοποιείται με την εκτέλεση SQL ερωτημάτων στους πίνακες για την λήψη της ζητούμενης πληροφορίας.

Η επεξεργασία των δεδομένων, θα μπορούσε να γίνει και με την χρήση της τεχνολογίας JSON (JavaScript Object Notation), η οποία είναι ένα ανοιχτό πρότυπο ανάλυσης και παραγωγής δεδομένων, ανεξάρτητο από γλώσσα προγραμματισμού που θεωρείται εναλλακτική λύση της XML. Ένα από τα πλεονεκτήματά της έναντι στην XML είναι ότι παράγει αρχεία μικρότερα σε μέγεθος σε σύγκριση με τα αρχεία XML. Επίσης, σε συνδυασμό με την γλώσσα Javascript απαιτεί λιγότερη συγγραφή κώδικα στη μεριά του client (client-side).

Όσον αφορά το πού δημοσιεύονται τελικά τα δεδομένα, θα μπορούσε εύκολα η εφαρμογή να επεκταθεί, έτσι ώστε να δημοσιεύει τις ανακοινώσεις και σε παραπάνω από ένα κοινωνικά δίκτυα ταυτόχρονα, όπως για παράδειγμα στο Twitter, το Google+ ή το LinkedIn. Εφόσον η πληροφορία είναι ήδη κατάλληλα διαμορφωμένη προς δημοσίευση, το μόνο που απαιτείται στη συγκεκριμένη περίπτωση είναι η μελέτη των API των κοινωνικών αυτών δικτύων, έτσι ώστε να γίνουν κατανοητές οι μέθοδοι που χρησιμοποιεί το καθένα για τη δημοσίευση περιεχομένου.

Τέλος, η εφαρμογή αυτή θα μπορούσε να επεκταθεί και ως προς το περιεχόμενο της, εφόσον αυτό είναι θεμιτό, δηλαδή θα μπορούσε να δημοσιεύει και μη δημόσιες ανακοινώσεις του τμήματος, με διερεύνηση φυσικά και του πλαισίου μέσω του οποίου δημοσιεύονται, για παράδειγμα δημοσίευση τους σε κλειστές ομάδες του Facebook, έτσι ώστε να είναι ελεγχόμενη η πρόσβαση σε αυτές.

9 Βιβλιογραφία

- [1] Όγκος δεδομένων του Facebook - <http://techcrunch.com/2012/08/22/how-big-is-facebooks-data-2-5-billion-pieces-of-content-and-500-terabytes-ingested-every-day/>
- [2] The Influence of Social Media Sites on the College Search Process - <http://www.artsci.com/studentpoll/0913/index.aspx>
- [3] Wikipedia – Facebook features - http://en.wikipedia.org/wiki/Facebook_features
- [4] Wikipedia – Push technology - http://en.wikipedia.org/wiki/Push_technology
- [5] Wikipedia – Pull technology - http://en.wikipedia.org/wiki/Pull_technology
- [6] What is pull technology - <http://www.wifinotes.com/mobile-communication-technologies/what-is-pull-technology.html>
- [7] Push technology - http://en.wikipedia.org/wiki/Push_technology
- [8] http://cgi.di.uoa.gr/~btsetsos/Thesis_LBS_Push.pdf
- [9] What is push technology - <http://www.library.ca.gov/CRB/97/notes/v4n6.pdf>
- [10] Real time web applications - <http://bstavroulakis.com/blog/web/real-time-web-applications/>
- [11] Wikipedia – Server side events - http://en.wikipedia.org/wiki/Server-sent_events
- [12] Server Sent Events - http://www.tutorialspoint.com/html5/html5_server_sent_events.htm
- [13] W3schools - Server Sent Events - http://www.w3schools.com/html/html5_serversentevents.asp

[14] Websockets - http://www.tutorialspoint.com/html5/html5_websocket.htm

[15] Wikipedia – WebSocket - <http://en.wikipedia.org/wiki/WebSocket>

[16] Διαφορές μεταξύ WebSocket και Server Sent Events -
<http://stackoverflow.com/questions/5195452/websockets-vs-server-sent-events-eventsources>

[17] Διαφορές μεταξύ websocket και long polling -
<http://stackoverflow.com/questions/10028770/html5-websocket-vs-long-polling-vs-ajax-vs-webrtc>

[18] Διαφορές μεταξύ websocket, server sent events και long polling -
<http://dsheiko.com/weblog/websockets-vs-sse-vs-long-polling/>

[19] Διαφορές μεταξύ push και pull email - https://motorola-global-eng.custhelp.com/app/answers/detail/a_id/15859/~what-is-the-difference-between-push-and-pull-email%3F

[20] Πλεονεκτήματα και μειονεκτήματα newsletter -
<http://www.matthewfarmer.com.au/the-5-pros-and-cons-of-newsletters/>

[21] Τεχνολογία RSS - <http://rss.softwaregarden.com/aboutrss.html>

[22] Βασικά στοιχεία RSS -
<http://computer.howstuffworks.com/internet/basics/rss.htm>

[23] RSS feeds - <http://www.2createawebsite.com/traffic/create-rss-feeds.html>

[24] Wikipedia – Atom - [http://en.wikipedia.org/wiki/Atom_\(standard\)](http://en.wikipedia.org/wiki/Atom_(standard))

[25] Atom XML - <http://whatis.techtarget.com/definition/Atom-XML>

[26] Facebook tools - <https://blog.kissmetrics.com/6-expert-facebook-tools/>

[27] Πλεονεκτήματα εφαρμογής Buffer -
<http://www.thedigitalfa.com/toddgreider/marketing-automation-made-easy-4-reasons-you-should-be-using-buffer-app/>

- [28] Πλεονεκτήματα και μειονεκτήματα των Hootsuite και Argyle Social - <http://www.slideshare.net/bordeauxmktg/5-social-media-dashboards-their-pros-and-cons-social-media-marketing>
- [29] Facebook Developers Docs – Graph API - <https://developers.facebook.com/docs/graph-api/quickstart/v2.0>
- [30] Sprout Social - <http://www.convinceandconvert.com/social-media-tools/favorite-social-media-management-tools-for-small-business/>
- [31] Εφαρμογές Facebook - <http://computer.howstuffworks.com/internet/social-networking/networks/facebook3.htm>
- [32] Δημιουργία Facebook εφαρμογών - <http://www.qwerty.gr/howto/facebook-applications>
- [33] Ανάπτυξη Facebook εφαρμογών <http://www.emanueleferonato.com/2009/09/20/developing-a-facebook-application-for-absolute-beginners/>
- [34] Μονολιθικές εφαρμογές - http://www.cs.toronto.edu/~matz/instruct/csc407/lectures/10c_monolithic-4.pdf
- [35] Διαφορά Open Graph και Graph API <http://stackoverflow.com/questions/16639799/is-there-a-difference-between-facebooks-open-graph-api-and-its-graph-api>
- [36] Wikipedia – Open Graph Protocol - http://en.wikipedia.org/wiki/Open_Graph_Protocol
- [37] Facebook Developers Docs – Open Graph - <https://developers.facebook.com/docs/opengraph/overview/>
- [38] Wikipedia – GraphAPI - https://fbdevwiki.com/wiki/Graph_API
- [39] Facebook Developers Docs – Χρήση API - <https://developers.facebook.com/docs/graph-api/using-graph-api/v2.0>
- [40] Facebook Developers Docs – Άλλα APIs - <https://developers.facebook.com/docs/graph-api/other-apis>
- [41] PHP - <https://php.net/manual/en/intro-what-is.php>
- [42] Facebook Developers Docs – PHP SDK - <https://developers.facebook.com/docs/reference/php/4.0.0>
- [43] Apache - <http://httpd.apache.org/>
- [44] Wikipedia – XAMPP - <http://el.wikipedia.org/wiki/XAMPP>
- [45] XML - <http://www.w3.org/TR/REC-xml>

[46] Xpath - <http://www.w3.org/TR/xpath>

[47] W3schools – HTTP μέθοδοι -
http://www.w3schools.com/tags/ref_httpmethods.asp

[48] Αυθεντικοποίηση OAuth 2.0 - <http://tools.ietf.org/pdf/draft-ietf-oauth-v2-12.pdf>

[49] Facebook Developers Docs – Αυθεντικοποίηση -
<http://developers.facebook.com/docs/authentication>

10 ΠΑΡΑΡΤΗΜΑ ΜΕ ΤΟΝ ΚΩΔΙΚΑ

Παρατίθεται ολόκληρος ο κώδικας.

10.1 Index.php

```
<?php
require_once 'php-sdk/facebook.php';

ini_set('default_charset', 'UTF-8');

$hydraBaseLink = "http://hydra.it.teithe.gr";
$s = 0;
$f = 0;

session_start();

if (isset($_SESSION['lastPost'])) {
    $lastPost = $_SESSION['lastPost'];
} else {
    $lastPost = "";
}

$hydraContents = new DOMDocument;

$hydraContents->loadHTML(file_get_contents($hydraBaseLink . "/s/index.php?m=itdep-bbpublic&bid=news"));
file_put_contents('data.xml', $hydraContents->saveXml());
$xml = simplexml_load_file('data.xml', 'SimpleXMLElement', LIBXML_NOCDATA);

$path = "//div[@class='new']";
$news = array_reverse($xml->xpath($path);
```

```
$foundLast = false;
$lastSuccess = $lastPost;
foreach ($news as $currentNew) {

    $countChildren = count($currentNew);

    $post = $currentNew->div[0];
    $post .= $currentNew->div[1];

    if ($countChildren == 3) {
        $post .= $currentNew->div[2];
    } else {
        $post .= $hydraBaseLink . $currentNew->div[2]->a["href"];
        $post .= $currentNew->div[3];
    }

    if($_SESSION["lastPost"] != null && $post != $lastPost && $foundLast == false) {
        continue;
    } elseif ($post == $lastPost) {
        $foundLast = true;
        continue;
    } else {
```

```
$pageid = '542273795822262';

// initialize Facebook class using our own Facebook App credentials
$config = array();
$config['appId'] = '1377208439213066';
$config['secret'] = 'xxxxxxxx';

$fb = new Facebook($config);

// define your POST parameters
$params = array(
    "access_token" => "CAATkkFPNDacBAFDZCx44V7p7hQQPotFDvCeOEwXP08q4sekBBAg1Wd",
    "message" => $post,
);

try {
    $ret = $fb->api('/'. $pageid .'/feed', 'POST', $params);
    echo 'Successfully posted to Facebook';
    $lastSuccess = $post;
    $s++;
} catch(Exception $e) {
    echo $e->getMessage();
    $f++;
}
}
```

```
$_SESSION["lastPost"] = $lastSuccess;
if ($s+$f==0){
    echo "There are no new announcements!";
} else {
    echo "Succeeded posting: ".$s."<br>Failed Posting: ".$f;
}
```

10.2 Βιβλιοθήκες Facebook

Λόγω του μεγέθους των βιβλιοθηκών, το οποίο υπερβαίνει τις 100 σελίδες, οι βιβλιοθήκες διατίθενται σε ηλεκτρονική μορφή.