

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Υλοποίηση Διαδικτυακής Πλατφόρμας και εφαρμογής για
έξυπνα κινητά για διαχείριση δεδομένων αθλητικού συλλόγου**



Του φοιτητή

Μαργαρίτη Παναγιώτη

Αρ. Μητρώου: 144255

Επιβλέπων

Αντώνης Σιδηρόπουλος

Βαθμίδα

Θεσσαλονίκη 2020

Τίτλος Π.Ε. Υλοποίηση Διαδικτυακής Πλατφόρμας και
εφαρμογής για έξυπνα κινητά για διαχείριση δεδομένων αθλητικού συλλόγου

Κωδικός Π.Ε. 19062

Όνοματεπώνυμο φοιτητή Μαργαρίτης Παναγιώτης

Όνοματεπώνυμο εισηγητή Σιδηρόπουλος Αντώνης

Ημερομηνία ανάληψης Π.Ε. 06-12-2019

Ημερομηνία περάτωσης Δ.Ε. ...

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

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

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

ΠΡΟΛΟΓΟΣ

Το WordPress είναι το πιο δημοφιλές σύστημα διαχείρισης περιεχομένου (CMS content management system) που μπορεί να χρησιμοποιηθεί στην κατασκευή οποιουδήποτε τύπου website. Είναι ένα λογισμικό ανοιχτού κώδικα (open source), χτισμένο σε μεγάλο βαθμό πάνω στη βάση δεδομένων MySQL και τη γλώσσα προγραμματισμού PHP, ικανό να επεκταθεί και να προσαρμοστεί για μια μεγάλη ποικιλία εφαρμογών. Κάθε χρήστης, που είναι εγγεγραμμένος στη βάση δεδομένων, έχει τη δυνατότητα να συνδεθεί στο αντίστοιχο website με τα κατάλληλα username και password. Τι γίνεται όμως στην περίπτωση όπου, το πλήθος των διαδικτυακών εφαρμογών είναι μεγάλο και απαιτείται η εγγραφή του κάθε χρήστη ξεχωριστά στις βάσεις δεδομένων; Προκύπτει λοιπόν, η ανάγκη για την δημιουργία μίας υπηρεσίας που θα χρησιμοποιεί, αποθηκεύει, επεξεργάζεται και διαμοιράζει τις πληροφορίες των χρηστών για πολλαπλές διαδικτυακές εφαρμογές. Οι υπηρεσίες καταλόγου μας δίνουν την δυνατότητα να συλλέγουμε πληροφορίες που αφορούν χρήστες. Οι πληροφορίες αυτές αποθηκεύονται σε έναν κατάλογο, ο οποίος μπορεί να προσπελαστεί από οποιονδήποτε χρήστη ή διαδικτυακή εφαρμογή, χρησιμοποιώντας τα κατάλληλα διαπιστευτήρια. Ορισμένα είδη καταλόγου παρέχουν απλώς μια στενή και συγκεκριμένη υπηρεσία. Ένας κατάλογος κάποιου server μίας συγκεκριμένης χρήσης, όπως ένα ηλεκτρονικό βιβλίο διευθύνσεων, μπορεί να αποθηκεύει μόνο έναν πολύ συγκεκριμένο τύπο δεδομένων, όπως αριθμούς τηλεφώνου, διευθύνσεις και πληροφορίες ηλεκτρονικού ταχυδρομείου για ένα σύνολο ατόμων. Τέτοιοι κατάλογοι δεν είναι επεκτάσιμοι. Αντ' αυτού, είναι μονής χρήσης. Αλλά το LDAP, σχεδιάστηκε για να είναι κατάλογος γενικής χρήσης, Δεν έχει σχεδιαστεί με σκοπό τη σύλληψη ενός συγκεκριμένου τύπου δεδομένων. Σχεδιάστηκε για να δώσει στους υπεύθυνους της εφαρμογής τη δυνατότητα να καθορίσουν, σαφώς προσεκτικά, ποια δεδομένα πρέπει να αποθηκεύει ο κατάλογος. Ένας τέτοιος κατάλογος, γενικής χρήσης, θα πρέπει να μπορεί να αποθηκεύει πολλά διαφορετικά είδη πληροφοριών σχετικά με διαφορετικά είδη οντοτήτων. Φυσικά, ένας διακομιστής καταλόγου πρέπει επίσης να διαθέτει μέσα, για την προσθήκη, τροποποίηση και διαγραφή πληροφοριών. Όμως, οι πληροφορίες ενός directory server (κατάλογος διακομιστή), θεωρείται ότι διαβάζονται συχνότερα από ότι γράφονται. Παρατηρείται το φαινόμενο «υψηλή ανάγνωση, χαμηλή εγγραφή». Συνεπώς πολλές εφαρμογές της τεχνολογίας LDAP έχουν ως στόχο την ανάγνωση και την αναζήτηση πληροφοριών, ενώ πολλοί κατάλογοι διακομιστών έχουν βελτιστοποιηθεί για γρήγορη ανάγνωση σε βάρος της γρήγορης τροποποίησης. Αυτό δεν ισχύει με το OpenLDAP. Το OpenLDAP είναι αποδοτικό και στις δύο μετρήσεις και μπορεί να χρησιμοποιηθεί για εφαρμογές που απαιτούν συχνή εγγραφή δεδομένων.

ΠΕΡΙΛΗΨΗ

Η παρακάτω Διπλωματική Εργασία επικεντρώνεται στη δημιουργία υπηρεσιών καταλόγου με το OpenLDAP, ξεκινώντας με τη διαδικασία εγκατάστασης του, αφού πρώτα έχει προηγηθεί η εγκατάσταση του λογισμικού WordPress. Στη συνέχεια, γίνεται αναφορά στις ενέργειες που πραγματοποιήθηκαν για την προσθήκη, αναζήτηση και τροποποίηση νέων εγγραφών στο κατάλογο, εισάγοντας πολλές από τις βασικές έννοιες που εμπλέκονται στην εργασία με δεδομένα τύπου LDAP. Έπειτα, αναλύεται η σύνδεση του LDAP με την πλατφόρμα του WordPress, χρησιμοποιώντας το LDAP ως πηγή ελέγχου ταυτότητας και εξουσιοδότησης, καθώς το OpenLDAP μπορεί να αναπαράγει τα περιεχόμενα του καταλόγου του από έναν κεντρικό διακομιστή σε οποιονδήποτε αριθμό δευτερευόντων διακομιστών. Τέλος, παρέχονται πληροφορίες για την δημιουργία μιας εφαρμογής τύπου android που αλληλοεπιδρά με το WordPress Rest API.

Λέξεις Κλειδιά: WordPress Rest API, OpenLDAP, Custom Endpoints, Custom Fields, Push Notifications, authLDAP, Active Directory Integration, LDAP Integration

ABSTRACT

The following essay, focuses on creating a catalog services with OpenLDAP, starting with the process of installing it on a Linux server, after the installation of the WordPress software. Then, a reference is made, about the actions that taken to add, search and modify new entries in the directory, introducing many of the key concepts when a user or a system works with LDAP data. Also, information is provided, about the LDAP's connection to the WordPress platform, using LDAP as a source of authentication and authorization, as OpenLDAP can reproduce the contents of its directory, from a central server to any number of secondary servers. Finally, information is provided, in order to create an android application that interacts with the WordPress Rest API.

Keywords: WordPress Rest API, OpenLDAP, Custom Endpoints, Custom Fields, Push Notifications, authLDAP, Active Directory Integration, LDAP Integration

ΕΥΧΑΡΙΣΤΙΕΣ

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις θερμότερες μου ευχαριστίες σε όλους εκείνους που με βοήθησαν να ολοκληρώσω τη Διπλωματική μου Εργασία.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα της εργασίας μου, καθηγητή κ. Αντώνη Σιδηρόπουλο για την εμπιστοσύνη που μου έδειξε.

Επίσης, θα ήθελα να ευχαριστήσω, τον καλό μου φίλο και συνεργάτη Γιώργο Νικολούση, για την αμέριστη κατανόηση και τεχνική βοήθεια πάνω στα θέματα της Εργασίας.

Τέλος, ένα μεγάλο και εγκάρδιο ευχαριστώ στην οικογένεια μου, η οποία στήριξε τις σπουδές μου με διάφορους τρόπους, φροντίζοντας για την καλύτερη δυνατή μόρφωση μου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	3
ΠΕΡΙΛΗΨΗ	4
ΠΕΡΙΛΗΨΗ	4
ABSTRACT	4
ΕΥΧΑΡΙΣΤΙΕΣ	5
ΠΕΡΙΕΧΟΜΕΝΑ	6
ΠΕΡΙΕΧΟΜΕΝΑ	6
ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ ΚΑΙ ΠΙΝΑΚΩΝ.....	10
ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ ΚΑΙ ΠΙΝΑΚΩΝ.....	11
ΕΙΣΑΓΩΓΗ	12
ΕΙΣΑΓΩΓΗ	12
ΚΕΦΑΛΑΙΟ 1	13
ΚΕΦΑΛΑΙΟ 1	13
1. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΙΣΤΟΣΕΛΙΔΑΣ WORDPRESS.....	13
1.1 ΤΙ ΕΙΝΑΙ ΤΟ WORDPRESS	13
1.2 ΑΓΟΡΑ DOMAIN ΚΑΙ HOSTING	13
1.3 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	14
1.4 ΜΕΤΑΦΟΡΤΩΣΗ ΑΡΧΕΙΩΝ WORDPRESS.....	14
1.5 ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ WORDPRESS	14
1.6 ΣΥΝΔΕΣΗ ΣΤΟΝ ΠΙΝΑΚΑ ΕΛΕΓΧΟΥ	16
1.6.1 ΕΓΚΑΤΑΣΤΑΣΗ ΘΕΜΑΤΟΣ.....	18
1.6.2 ΤΡΟΠΟΠΟΙΗΣΗ ΘΕΜΑΤΟΣ.....	18
1.6.3 ΤΡΟΠΟΠΟΙΗΣΗ ΘΕΜΑΤΟΣ - ACF	19
1.6.4 ADVANCED CUSTOM FIELDS (ACF).....	19
1.7 ΟΛΟΚΛΗΡΩΣΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ.....	21
1.7.1 ΕΙΣΟΔΟΣ ΧΡΗΣΤΗ – ΛΕΙΤΟΥΡΓΙΕΣ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ	22
1.8 ΣΥΜΠΕΡΑΣΜΑ	24

ΕΥΧΑΡΙΣΤΙΕΣ

ΚΕΦΑΛΑΙΟ 2	25
ΚΕΦΑΛΑΙΟ 2	25
2. ΕΠΙΣΚΟΠΗΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ LDAP	25
2.1 ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΟΥ LDAP.....	25
2.2 ΤΙ ΕΙΝΑΙ ΕΝΑ DIRECTORY (ΚΑΤΑΛΟΓΟΣ);	25
2.3 Η ΔΟΜΗ ΜΙΑΣ ΕΓΓΡΑΦΗΣ ΚΑΤΑΛΟΓΟΥ (DIRECTORY ENTRY).....	27
2.4 ΕΝΑ ΜΟΝΑΔΙΚΟ ΟΝΟΜΑ – ΤΟ DN.....	28
2.5 ΠΑΡΑΔΕΙΓΜΑ ΕΓΓΡΑΦΗΣ – ΚΑΡΑΧΩΡΙΣΗΣ LDAP	29
2.6 ΤΟ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ CLASS OBJECT	31
2.7 ΚΑΤΗΓΟΡΙΕΣ ΑΝΤΙΚΕΙΜΕΝΩΝ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	31
2.8 ΛΕΙΤΟΥΡΓΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ.....	33
2.9 ΤΟ ΔΕΝΤΡΟ ΠΛΗΡΟΦΟΡΙΩΝ ΚΑΤΑΛΟΓΟΥ	33
2.10 ΤΙ ΝΑ ΚΑΝΕΤΕ ΜΕ ΕΝΑΝ LDAP.....	35
2.11 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ OPENLDAP	36
2.12 ΤΕΧΝΙΚΗ ΕΠΙΣΚΟΠΙΣΗ ΤΟΥ OPENLDAP	37
2.12.1 Ο ΔΙΑΚΟΜΙΣΤΗΣ (SERVER).....	37
2.12.2 ΟΙ ΠΕΛΑΤΕΣ (CLIENTS)	37
2.12.3 ΒΟΗΘΗΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ (UTILITIES)	38
2.12.4 ΟΙ ΒΙΒΛΙΟΘΗΚΕΣ (LIBRARIES)	38
2.13 ΣΥΜΠΕΡΑΣΜΑ	38
ΚΕΦΑΛΑΙΟ 3	39
ΚΕΦΑΛΑΙΟ 3	39
3 ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ LDAP.....	39
3.1 ΛΗΨΗ ΚΑΙ ΕΓΚΑΤΑΣΤΑΣΗ.....	39
3.2 ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ LDAP SERVER.....	39
3.1.1 ΕΚΚΙΝΗΣΗ SERVER	40
3.1.2 ΔΙΑΚΟΠΗ SERVER	40
3.1.3 ΕΠΕΞΕΡΓΑΣΙΑ ΤΗΣ ΔΙΑΜΟΡΦΩΣΗΣ ΤΟΥ SERVER	40
3.1.4 ΔΙΑΓΡΑΦΗ SERVER.....	40
3.2 ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΔΕΣΗΣ ΜΕ ΤΟ LDAP	41
3.2.1 Η ΔΙΑΔΙΚΑΣΙΑ ΔΕΣΜΕΥΣΗΣ (BINDING PROCEDURE)	42

3.2.2 Η ΔΙΑΔΙΚΑΣΙΑ ΑΝΑΖΗΤΗΣΗΣ (SEARCH PROCEDURE)	44
3.2.3 ΠΕΡΙΣΣΟΤΕΡΕΣ ΔΙΑΔΙΚΑΣΙΕΣ (ΠΡΟΣΘΗΚΗ, ΤΡΟΠΟΠΟΙΗΣΗ, ΔΙΑΓΡΑΦΗ)	46
3.2.4 Η ΔΙΑΔΙΚΑΣΙΑ ΠΡΟΣΘΗΚΗΣ (ADDITION PROCEDURE)	47
3.2.5 Η ΔΙΑΔΙΚΑΣΙΑ ΤΡΟΠΟΠΟΙΗΣΗΣ (MODIFICATION PROCEDURE).....	47
3.2.6 Η ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΓΡΑΦΗΣ (DELETE PROCEDURE)	49
3.3 ΣΥΝΔΕΣΗ ΤΟΥ LDAP ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ WORDPRESS.....	50
3.4 ΣΥΜΠΕΡΑΣΜΑ	53
ΚΕΦΑΛΑΙΟ 4	54
ΚΕΦΑΛΑΙΟ 4	54
4 REST API ΤΟΥ WORDPRESS.....	54
4.1 REPRESENTIVE STATE TRANSFER (REST)	54
4.2 REST APIS	54
4.3 WORDPRESS REST API	55
4.4 ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΟ WORDPRESS REST AP	55
4.5 ΠΡΟΣΘΗΚΗ CUSTOM ENDPOINTS.....	55
4.6 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ANDROID ΜΕ ΤΟ REST API ΤΟΥ WORDPRESS.....	57
4.7 ΣΥΜΠΕΡΑΣΜΑ	62
ΚΕΦΑΛΑΙΟ 5	63
ΚΕΦΑΛΑΙΟ 5	63
5 ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ ANDROID	63
5.1 ANDROID STUDIO	63
5.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ ANDROID STUDIO	63
5.3 ΑΝΑΛΥΣΗ ΕΝΟΣ PROJECT ΣΕ ANDROID	65
5.4 ΔΕΔΟΜΕΝΑ JSON ΑΠΟ URL ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ANDROID	68
5.5 ΟΛΟΚΛΗΡΩΣΗ MOBILE ΕΦΑΡΜΟΓΗΣ	69
5.5.1 ΕΛΕΓΧΟΣ ΕΙΣΟΔΟΥ ΧΡΗΣΤΗ.....	69
5.5.2 ΕΙΣΟΔΟΣ ΠΡΟΠΟΝΗΤΗ.....	70
5.5.3 ΕΠΕΞΕΡΓΑΣΙΑ - ΠΡΟΒΟΛΗ ΣΤΟΙΧΕΙΩΝ ΠΡΟΠΟΝΗΤΗ	71
5.5.4 ΠΡΟΒΟΛΗ ΑΝΑΚΟΙΝΩΣΕΩΝ	72
5.5.5 ΔΗΜΙΟΥΡΓΙΑ ΝΕΑΣ ΑΝΑΚΟΙΝΩΣΗΣ	73
5.5.6 ΔΗΜΙΟΥΡΓΙΑ ΠΑΡΟΥΣΙΟΛΟΓΙΟΥ	75

ΕΥΧΑΡΙΣΤΙΕΣ

ΣΥΜΠΕΡΑΣΜΑΤΑ	76
ΣΥΜΠΕΡΑΣΜΑΤΑ	76
ΒΙΒΛΙΟΓΡΑΦΙΑ	77
ΒΙΒΛΙΟΓΡΑΦΙΑ	77

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ ΚΑΙ ΠΙΝΑΚΩΝ

Εικόνα 1 - Στοιχεία Βάσης Δεδομένων.....	15
Εικόνα 2 - Στοιχεία Διαχειριστή Ιστότοπου.....	15
Εικόνα 3 - Είσοδος στο WordPress.....	16
Εικόνα 4 - Πίνακας Ελέγχου WordPress.....	17
Εικόνα 5 - Τροποποίηση functions.php.....	19
Εικόνα 6 - Παράδειγμα εισαγωγής πεδίου ACF.....	20
Εικόνα 7 - Χρήση ACF στο αρχείο functions.php	20
Εικόνα 8 - Login Page	21
Εικόνα 9 - Προβολή και Επεξεργασία στοιχείων μέσω της Δ.Ε.	22
Εικόνα 10 - Προβολή Ανακοινώσεων.....	23
Εικόνα 11 - Προβολή προγράμματος προπονήσεων.....	23
Εικόνα 12 - Πεδία εγγραφής σε ένα directory	26
Εικόνα 13 - Κλάση αντικειμένου Person	32
Εικόνα 14 - Παράδειγμα DIT	34
Εικόνα 15 - Παράδειγμα DIT με DN	35
Εικόνα 16 - Τα 4 στοιχεία του OpenLDAP	37
Εικόνα 17 - Δημιουργία LDAP	39
Εικόνα 18 - Σύνδεση LDAP σε δίκτυο (Μέρος Α)	41
Εικόνα 19 - Σύνδεση LDAP σε δίκτυο (Μέρος Β).....	41
Εικόνα 20 - Διαδικασία δέσμευσης.....	42
Εικόνα 21 - Φίλτρο Αναζήτησης LDAP	45
Εικόνα 22 - Σύνδεση LDAP - WordPress με PHP	50
Εικόνα 23 - Σύνδεση LDAP - WordPress με Πρόσθετο (Μέρος Α)	51
Εικόνα 24 - Σύνδεση LDAP - WordPress με Πρόσθετο (Μέρος Β)	52
Εικόνα 25 - Σύνδεση LDAP -WordPress με Πρόσθετο (Μέρος Γ).....	53
Εικόνα 26 - Αίτημα - Απάντηση - Rest API	54
Εικόνα 27 - Παράδειγμα μεθόδου για το Rest API	56
Εικόνα 28 - "Εγγραφή" μεθόδου στο WordPress REST AP"	57
Εικόνα 29 - Custom Routes του WordPress Rest API.....	57
Εικόνα 30 - Android Development και WordPress REST API (Μέρος Α).....	58
Εικόνα 31 - Android Development και WordPress REST API (Μέρος Β).....	59
Εικόνα 32 - Android Development και WordPress REST API (Μέρος Γ)	60
Εικόνα 33 - Αποτέλεσμα μεθόδου σε JSON	61
Εικόνα 34 - Εγκατάσταση Android Studio (Μέρος Α)	63
Εικόνα 35 - Εγκατάσταση Android Studio (Μέρος Β).....	64
Εικόνα 36 - Εγκατάσταση Android Studio (Μέρος Γ)	64
Εικόνα 37 - Παράδειγμα Μεθόδου onCreate	65

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ ΚΑΙ ΠΙΝΑΚΩΝ

Εικόνα 38 - Παράδειγμα Activity.....	66
Εικόνα 39 - Παράδειγμα Layout.....	67
Εικόνα 40 - Αξιοποίηση Δεδομένων JSON από URL με χρήση του Android Studio.....	68
Εικόνα 41 - Έλεγχος Εισόδου Χρήστη.....	70
Εικόνα 42 - Βασικές Λειτουργίες Προπονητή	71
Εικόνα 43 - Προβολή και Επεξεργασία Στοιχείων.....	72
Εικόνα 44 - Προβολή Ανακοινώσεων.....	73
Εικόνα 45 - Δημιουργία Νέας Ανακοίνωσης.....	74
Εικόνα 46 - Δημιουργία Νέου Παρουσιολογίου	75

ΕΙΣΑΓΩΓΗ

Η παρούσα Διπλωματική Εργασία έχει τον τίτλο «Υλοποίηση Διαδικτυακής Πλατφόρμας και εφαρμογής για έξυπνα κινητά για διαχείριση δεδομένων αθλητικού συλλόγου». Σκοπός της εργασίας είναι η χρήση του WordPress ως πλατφόρμα αποθήκευσης πληροφορίας και αλληλεπίδραση της ιστοσελίδας με το android application χρησιμοποιώντας το Rest Api του WordPress. Ταυτόχρονα παρουσιάζεται ο τρόπος δημιουργίας και η χρήση ενός εξωτερικού LDAP Server για την οργάνωση των χρηστών, την ομαδοποίησή τους καθώς και τον έλεγχο ταυτότητάς τους. Η σύνδεση των χρηστών στην ιστοσελίδα γίνεται με τα απαραίτητα διαπιστευτήρια που έχουν οριστεί στο LDAP.

Η Διπλωματική Εργασία χωρίζεται σε πέντε (5) κεφάλαια. Κάθε κεφάλαιο χωρίζεται σε επιμέρους ενότητες.

Το πρώτο κεφάλαιο («ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΙΣΤΟΣΕΛΙΔΑΣ WORDPRESS») παρουσιάζει αναλυτικά τα βήματα που πραγματοποιήθηκαν για την υλοποίηση της διαδικτυακής εφαρμογής καθώς και τις τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν.

Στο δεύτερο κεφάλαιο («ΕΠΙΣΚΟΠΗΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ LDAP») αναλύονται τα βασικά στοιχεία των υπηρεσιών καταλόγου τύπου LDAP, κάνοντας ταυτόχρονα μία μικρή εισαγωγή για το Active Directory ως μιας υπηρεσίας καταλόγου και ταυτόχρονα ως μιας υπηρεσίας ελέγχου ταυτότητας , έτσι ώστε να μπορούσαμε να διαχειριστούμε ή να προσπελάσουμε το περιεχόμενό τους. Στην ενότητα («2.10 ») δίνεται μεγαλύτερη βαρύτητα στο πρωτόκολλο επικοινωνίας OpenLDAP, μιας και είναι το πρωτόκολλο επικοινωνίας πάνω στο οποίο θα εργαστούμε.

Στο τρίτο κεφάλαιο («ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ LDAP») αναλύονται οι βασικές λειτουργίες – διαδικασίες που παρέχει ένας διακομιστής LDAP στον πελάτη. Οι διαδικασίες που αναλύονται είναι: δέσμευση, αναζήτηση, προσθήκη, τροποποίηση και διαγραφή. Επιπλέον, δίνεται ιδιαίτερη βαρύτητα στην αλληλεπίδραση ενός καταλόγου LDAP και μίας διαδικτυακής εφαρμογής, χρησιμοποιώντας το Apache Directory Studio και το WordPress.

Στο τέταρτο κεφάλαιο («REST API ΤΟΥ WORDPRESS») γίνεται μία πλήρης περιγραφή και παρουσίαση όλων των ενεργειών που απαιτήθηκαν για τη δημιουργία custom routes χρησιμοποιώντας το Rest Api του WordPress.

Στο πέμπτο κεφάλαιο («ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ ANDROID») γίνεται μία πλήρης περιγραφή και παρουσίαση όλων των ενεργειών που απαιτήθηκαν για τη δημιουργία του android application και της αλληλεπίδρασης του με το Rest API του WordPress.

Για την υλοποίηση της εργασίας, χρησιμοποιήθηκε πληθώρα προγραμματιστικών εργαλείων, τα περισσότερα των οποίων προσφέρονται δωρεάν (Android Studio, Oracle VM VirtualBox, WordPress, Brackets, OpenLDAP, Apache Directory Studio).

ΚΕΦΑΛΑΙΟ 1

1. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΙΣΤΟΣΕΛΙΔΑΣ WORDPRESS

1.1 ΤΙ ΕΙΝΑΙ ΤΟ WORDPRESS

Το WordPress είναι ένα σύστημα διαχείρισης περιεχομένου. Ένα σύστημα διαχείρισης περιεχομένου (CMS) είναι μια εφαρμογή λογισμικού που παρέχει τις δυνατότητες δημιουργίας, οργάνωσης, επεξεργασίας και δημοσίευση περιεχομένου στον Ιστό. Αντί να απαιτούνται πολλές εφαρμογές λογισμικού, απαιτείται μόνο μία εφαρμογή. Αποτελεί λογισμικό ανοιχτού κώδικα και διατίθεται δωρεάν ως προς την χρήση και την τροποποίηση του. Επιπλέον, ως λογισμικό ανοιχτού κώδικα, οι χρήστες μπορούν να κάνουν ελεύθερα αλλαγές στον πηγαίο κώδικα ανά πάσα στιγμή και να τις μοιράζονται με οποιονδήποτε. Αν λείπει ή πρέπει να προστεθεί μια δυνατότητα, οι χρήστες μπορούν να την προσθέσουν, τροποποιώντας τον πηγαίο κώδικα.

Μερικές από τις εταιρίες που χρησιμοποιούν το WordPress για την κατασκευή και συντήρηση των ιστοτόπων και των ιστολογίων τους είναι οι εξής:

- TechCrunch
- Microsoft News
- TED Blog
- BBC America
- PlayStation
- Skype
- The Walt Disney Company
- MTV News
- cPanel Blog
- Usain Bold

1.2 ΑΓΟΡΑ DOMAIN ΚΑΙ HOSTING

Τα πρώτα βήματα για την εγκατάσταση και την δημιουργία ενός ιστοτόπου με την χρήση WordPress είναι η κατοχύρωση ενός domain name και η φιλοξενία της ιστοσελίδας σε ένα διακομιστή.

- Το domain name είναι μια πιο φιλική για τον άνθρωπο γραφή (ψευδώνυμο) μιας IP διεύθυνσης, ενός αριθμού δηλαδή της μορφής "123.23.234.45" που οδηγεί σε μία συγκεκριμένη θέση / ιστοσελίδα στο διαδίκτυο.
- Η φιλοξενία ιστοσελίδων (αγγλικά: Web Hosting ή απλά Hosting) είναι ένα μια διαδικτυακή υπηρεσία που επιτρέπει σε ιδιώτες και εταιρείες να διαθέτουν μία ιστοσελίδα συνεχώς στο Διαδίκτυο, χωρίς να χρειάζεται να διαθέτουν τον ανάλογο εξοπλισμό και να επιβαρύνονται με το σχετικό κόστος (π.χ. Εξυπηρετητές, συνδέσεις δικτύου). Αυτό το αναλαμβάνουν οι εταιρίες φιλοξενίας ιστοσελίδων (Web hosts) που προσφέρουν χώρο στον διακομιστή τους καθώς και μέρος της σύνδεσής τους προς το Internet.

1.3 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Το λογισμικό WordPress είναι ένα προσωπικό σύστημα δημοσίευσης που χρησιμοποιεί PHP και MySQL. Όλο το περιεχόμενό της ιστοσελίδας (επιλογές, δημοσιεύσεις, σχόλια, και άλλα σχετικά δεδομένα) αποθηκεύονται στη βάση δεδομένων MySQL στον λογαριασμό φιλοξενίας. Για αυτό το λόγο πρέπει να ρυθμιστεί η βάση δεδομένων MySQL κατάλληλα ώστε να «συνδεθεί» με τον ιστότοπο. Προκειμένου να ολοκληρωθεί η «σύνδεση», θα πρέπει να πραγματοποιήσετε τα εξής βήματα:

- ✓ Δημιουργία βάσης δεδομένων, «πάνω» στην οποία θα στηθεί το WordPress.
- ✓ Δημιουργία χρήστη, με όλα τα δικαιώματα, για την διαχείριση της βάσης.
- ✓ Ορισμός χρήστη (διαχειριστή) στη βάση.

Η βάση δεδομένων MySQL για τον ιστότοπο είναι πλήρης και απομένει το τελικό βήμα για την εγκατάσταση του λογισμικού.

1.4 ΜΕΤΑΦΟΡΤΩΣΗ ΑΡΧΕΙΩΝ WORDPRESS

Το τελευταίο βήμα για την εκκίνηση της εγκατάστασης του WordPress, είναι η λήψη της τελευταίας έκδοσης του λογισμικού από την επίσημη ιστοσελίδα (wordpress.org) . Όταν ολοκληρωθεί η λήψη, σειρά έχει το «ανέβασμα» (upload) των αρχείων, στο πακέτο φιλοξενίας.

1.5 ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ WORDPRESS

Το τελευταίο βήμα στη διαδικασία εγκατάστασης του WordPress, είναι η σύνδεση του λογισμικού (Βήμα 1.4) με τη βάση MySQL (Βήμα 1.3) .

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΙΣΤΟΣΕΛΙΔΑΣ WORDPRESS

- Πληκτρολογήστε τη διεύθυνση URL του ιστότοπου σας στη γραμμή διευθύνσεων του προγράμματος περιήγησης ιστού (<http://example.com/wp-admin/install.php>).

Κατόπιν θα πρέπει να εισαχθούν τα διαπιστευτήρια για την σύνδεση με τη βάση που δημιουργήθηκε.



The screenshot shows the WordPress installation database configuration screen. At the top is the WordPress logo. Below it, a heading reads: "Below you should enter your database connection details. If you're not sure about these, contact your host." The form contains five input fields with labels and explanatory text:

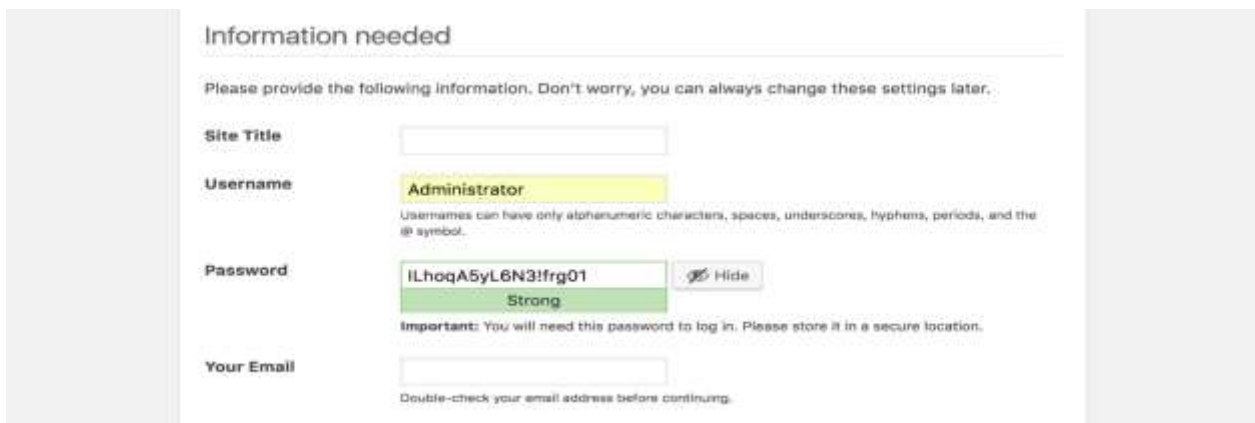
- Database Name:** Input field contains "wordpress". Text: "The name of the database you want to use with WordPress."
- Username:** Input field contains "username". Text: "Your database username."
- Password:** Input field contains "password". Text: "Your database password."
- Database Host:** Input field contains "localhost". Text: "You should be able to get this info from your web host, if localhost doesn't work."
- Table Prefix:** Input field contains "wp_". Text: "If you want to run multiple WordPress installations in a single database, change this."

A "Submit" button is located at the bottom left of the form.

Εικόνα 1 - Στοιχεία Βάσης Δεδομένων

- Database Name: Το όνομα της βάσης δεδομένων MySQL που δημιουργήθηκε πριν από αυτήν την εγκατάσταση.
- Username: Το όνομα χρήστη που ορίστηκε διαχειριστής στην αντίστοιχη βάση δεδομένων.
- Password: Ο κωδικός πρόσβασης του διαχειριστή.
- Database Host: Στην πλειοψηφία των περιπτώσεων η επιλογή localhost είναι η σωστή, διαφορετικά απαιτείται επικοινωνία με τον πάροχο φιλοξενίας.

Εάν τα στοιχεία που δοθούν είναι σωστά, η εγκατάσταση ολοκληρώνεται με το παρακάτω βήμα.



The screenshot shows the "Information needed" step of the WordPress installation. The heading is "Information needed". Below it, a message says: "Please provide the following information. Don't worry, you can always change these settings later." The form contains four input fields:

- Site Title:** An empty input field.
- Username:** Input field contains "Administrator". Text below: "Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol."
- Password:** Input field contains "iLhoqA5yL6N3!frg01". A "Hide" button is next to it. Text below: "Important: You will need this password to log in. Please store it in a secure location."
- Your Email:** An empty input field. Text below: "Double-check your email address before continuing."

Εικόνα 2 - Στοιχεία Διαχειριστή Ιστότοπου

Κατόπιν θα πρέπει να εισαχθούν τα διαπιστευτήρια για την σύνδεση στον ιστότοπο WordPress.

ΚΕΦΑΛΑΙΟ 1

- Site Title: Ο τίτλος της ιστοσελίδας.
- Username: Όνομα διαχειριστή της ιστοσελίδας.
- Password: Κωδικός πρόσβασης διαχειριστή για την ιστοσελίδα.
- Your Email: Εισαγωγή διεύθυνσης email για εισερχόμενες ειδοποιήσεις για την ιστοσελίδα.

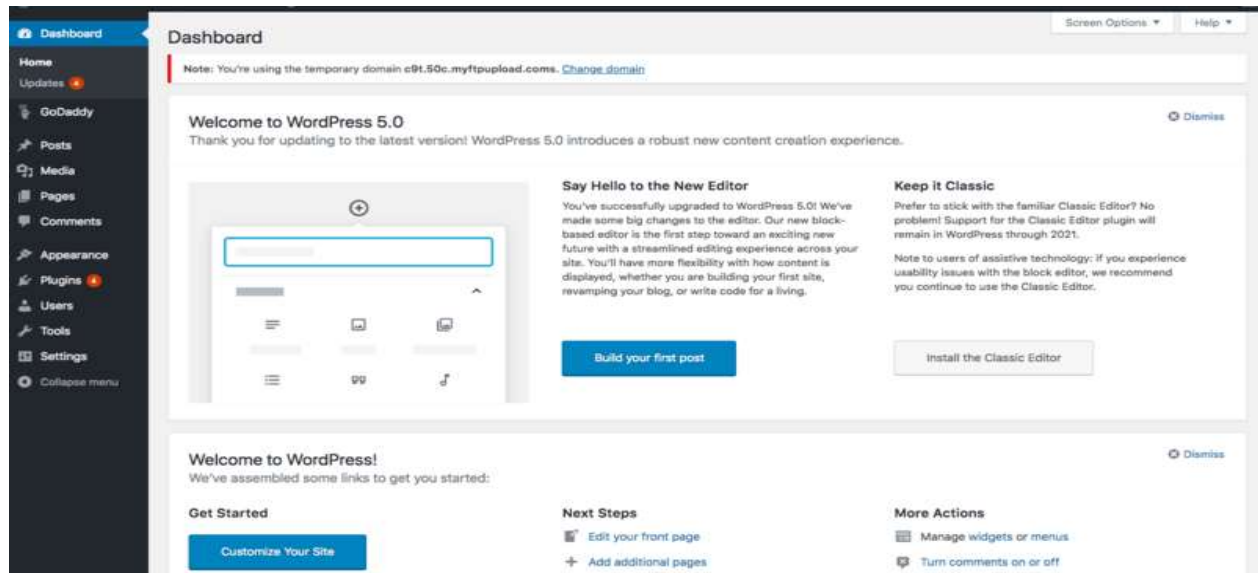
Το μόνο που απομένει είναι η πληκτρολόγηση των παραπάνω username και password.



Εικόνα 3 - Είσοδος στο WordPress

1.6 ΣΥΝΔΕΣΗ ΣΤΟΝ ΠΙΝΑΚΑ ΕΛΕΓΧΟΥ

Τώρα που το WordPress έχει εγκατασταθεί με επιτυχία, το επόμενο βήμα, είναι η επιλογή του σωστού θέματος με βάση το οποίο θα κατασκευαστεί η ιστοσελίδα. Δεδομένου ότι ο ιστότοπος θα είναι για έναν αθλητικό σύλλογο, το θέμα που θα επιλεγεί είναι το Trophy - Soccer and Football Club Theme. Όλες οι ενέργειες που απαιτούνται για την κατασκευή της ιστοσελίδας πραγματοποιούνται μέσω του πίνακα ελέγχου του WordPress.



Εικόνα 4 - Πίνακας Ελέγχου WordPress

Ο πίνακας ελέγχου περιέχει διάφορες ρυθμίσεις που βοηθούν τον χρήστη να επεξεργαστεί αποτελεσματικά τον ιστότοπο του, χωρίς να χρειάζεται να κάνει FTP στον ιστότοπο ή να τροποποιήσει τη βάση δεδομένων του WordPress.

Ο πίνακας ελέγχου χωρίζεται σε διάφορες περιοχές που παρέχουν πρόσβαση σε εργαλεία και λειτουργίες. Μερικά από αυτά είναι:

1. Posts (Άρθρα) Τα άρθρα αποτελούν το περιεχόμενο του ιστολογίου της ιστοσελίδας, που παρατίθενται με αντίστροφη χρονολογική σειρά (νεότερο περιεχόμενο στην κορυφή).
2. Pages (Σελίδες) Οι σελίδες αντιπροσωπεύουν το στατικό περιεχόμενο του ιστότοπου. Είναι δυνατό να χρησιμοποιηθούν, για να δημιουργηθούν αμετάβλητα δεδομένα ή σπάνια τροποποιημένα δεδομένα στον ιστότοπο.
3. Widgets (Μικροεφαρμογές) Οι μικροεφαρμογές παρέχουν μια γρήγορη επισκόπηση του τι συμβαίνει με τον ιστότοπο, από ορισμένα στατιστικά στοιχεία. Επιπλέον, προσφέρουν μερικούς επωφελείς συνδέσμους για τη διαχείριση, προσθήκη, πλοήγηση και προσαρμογή υλικού στον ιστότοπο.
4. Appearance (Εμφάνιση) Η εμφάνιση είναι μία, αν όχι η πιο, σημαντική ρύθμιση στον πίνακα ελέγχου του WordPress. Μέσω αυτής, μπορεί να αλλάξει ολόκληρη η εμφάνιση, η διάταξη, ο σχεδιασμός και η πλοήγηση του ιστότοπου. Στον τομέα αυτό, πραγματοποιείται η εγκατάσταση του θέματος, ενώ δίνεται η δυνατότητα τροποποίησης του θέματος ή προσθήκη νέων λειτουργιών στο θέμα, μέσω του αρχείου functions.php.

5. Plugins (Πρόσθετα) Μαζί με την Εμφάνιση, αποτελεί μία από τις πιο σημαντικές λειτουργίες στον πίνακα ελέγχου του WordPress. Ένα πρόσθετο είναι ένα μικρό πρόγραμμα που όταν προστίθεται στο WordPress, αλληλεπιδρά με το λογισμικό για να παρέχει κάποια δυνατότητα επέκτασης στο λογισμικό. Τα πρόσθετα συνήθως, δεν λειτουργούν ως αυτόνομο λογισμικό. Απαιτούν το λογισμικό του WordPress για λειτουργία.

1.6.1 ΕΓΚΑΤΑΣΤΑΣΗ ΘΕΜΑΤΟΣ

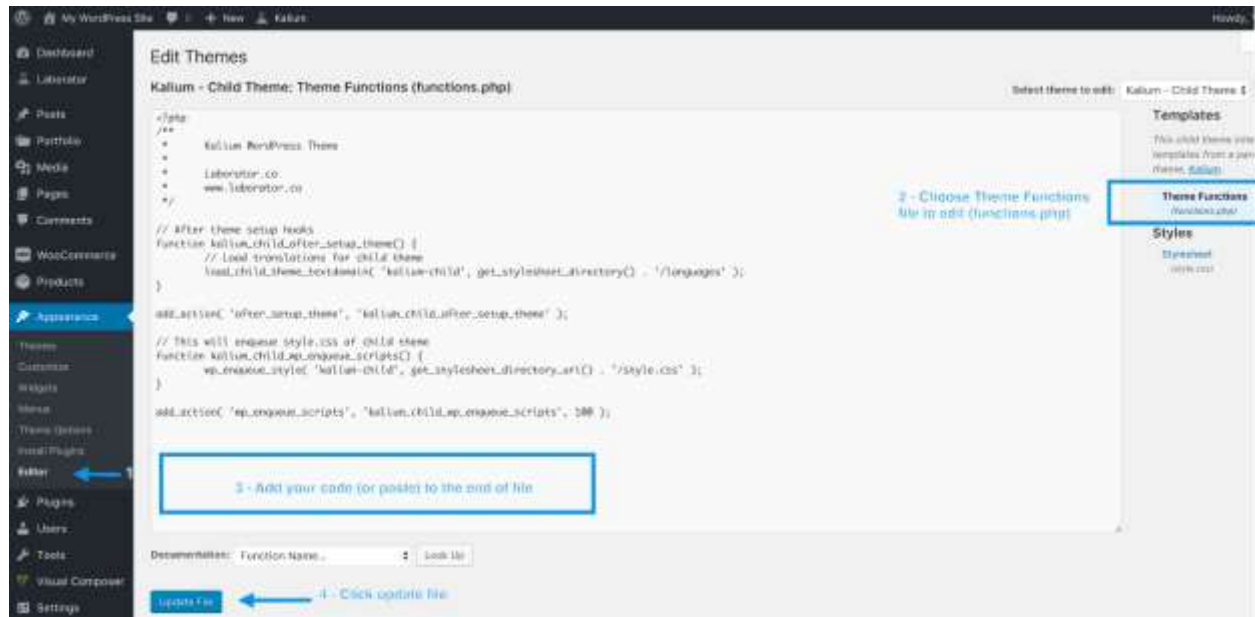
Σε αυτήν την ενότητα, όπως προαναφέρθηκε, επιλέγεται το θέμα για το πως θα φαίνεται ο ιστότοπος και ολοκληρώνεται η εγκατάσταση του.

Για την εγκατάσταση ενός θέματος μέσω του Πίνακα Ελέγχου του WordPress, πρέπει να πραγματοποιηθούν τα εξής βήματα:

1. Να «κατέβει» (download), το αρχείο του θέματος από τον κατάλογο θεμάτων στην επιφάνεια εργασίας. Συνήθως, τα αρχεία θεμάτων παρέχονται σε συμπιεσμένη μορφή (αρχείο .zip).
2. Εφόσον, ο χρήστης έχει συνδεθεί στον πίνακα ελέγχου, τότε επιλέγοντας τη λειτουργία Εμφάνιση και κατόπιν Θέματα.
3. Ακολουθεί η μεταφόρτωση του αρχείου που είχε «κατέβει» στο Βήμα 1.
4. Στο τέλος, απομένει η ενεργοποίηση του αρχείου

1.6.2 ΤΡΟΠΟΠΟΙΗΣΗ ΘΕΜΑΤΟΣ

Στο WordPress, το αρχείο `functions.php` είναι ένα πρότυπο που περιλαμβάνεται σε κάθε θέμα. Λειτουργεί σαν μία προσθήκη για τον ιστότοπο, που ενεργοποιείται αυτόματα με το τρέχον θέμα. Το αρχείο `functions.php`, χρησιμοποιεί κώδικα PHP για να προσθέσει δυνατότητες ή να αλλάξει προεπιλεγμένες δυνατότητες σε μια τοποθεσία WordPress. Για παράδειγμα, εάν ο χρήστης θέλει να προσθέσει μια νέα Μικροεφαρμογή (Widget) στο υποσέλιδο ή να προσθέσει ένα προσαρμοσμένο μήνυμα καλωσορίσματος στον πίνακα ελέγχου του WordPress, τότε αρκεί να προσθέσει λίγο κώδικα στο αρχείο `functions.php`. Η επεξεργασία του αρχείου `functions.php` επιτρέπει στον χρήστη να προσθέτει τύπους δημοσιεύσεων, συντομεύσεις και άλλες πολλές λειτουργίες για να βελτιωθεί ο ιστότοπος.



Εικόνα 5 - Τροποποίηση functions.php

1.6.3 ΤΡΟΠΟΠΟΙΗΣΗ ΘΕΜΑΤΟΣ - ACF

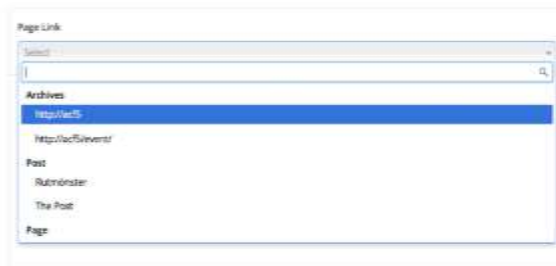
Κατά την δημιουργία ενός ιστότοπου με το WordPress, ο χρήστης έχει τη δυνατότητα να προσαρμόσει τόσο την εμφάνιση όσο και τη λειτουργικότητα του ιστότοπου. Το WordPress, παρέχει αρκετή ευελιξία καθώς αποτελεί λογισμικό ανοιχτού κώδικα. Όπως προαναφέρθηκε, μέσω της προσθήκης προσαρμοσμένου κώδικα, υπάρχει η δυνατότητα υλοποίησης πολλαπλών λειτουργιών. Ωστόσο, πέραν από σημαντικές γνώσεις, πολύ σημαντικό ρόλο παίζουν τα εργαλεία που θα χρησιμοποιηθούν για την υλοποίηση των λειτουργιών.

1.6.4 ADVANCED CUSTOM FIELDS (ACF)

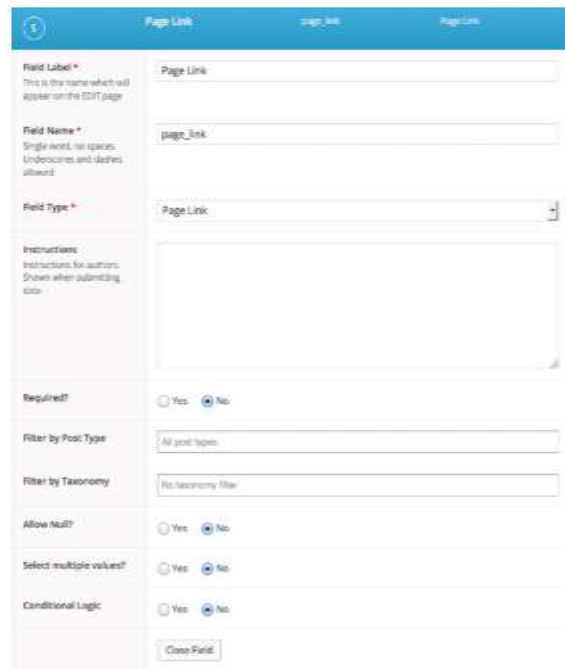
Το Advanced Custom Fields είναι ένα πρόσθετο που βοηθά τον χρήστη να επεξεργάζεται πολύπλοκο περιεχόμενο με απλό και λογικό τρόπο. Το ACF επιτρέπει στον χρήστη, να προσθέτει επιπλέον πεδία, εκτός από αυτά που παρέχει ο επεξεργαστής του WordPress, μέσα από ένα ευρύ φάσμα τύπων πεδίων όπως κειμένου, πολυμέσων, επιλογών κ.α. Κάθε πεδίο μπορεί να προστεθεί και να επεξεργαστεί σύμφωνα με τις προτιμήσεις του χρήστη.

ΚΕΦΑΛΑΙΟ 1

Παρακάτω προστίθεται, ένα πεδίο «Page Link», το οποίο δημιουργεί μία αναπτυσσόμενη λίστα επιλογής μεταξύ αναρτήσεων, σελίδων ή αρχείων αρχειοθέτησης.



The Page Link field interface



The Page Link field settings

Εικόνα 6 - Παράδειγμα εισαγωγής πεδίου ACF

```
<?php
$url = get_field('urls');
if( $url ): ?>
<h3>Further reading</h3>
<ul>
  <?php foreach( $url as $url ): ?>
  <li>
    <a href="<?php echo esc_url( $url ); ?>"><?php echo esc_html( $url ); ?></a>
  </li>
  <?php endforeach; ?>
</ul>
<?php endif; ?>
```

Εικόνα 7 - Χρήση ACF στο αρχείο functions.php

1.7 ΟΛΟΚΛΗΡΩΣΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ

Όπως προαναφέρθηκε, οι κατηγορίες των χρηστών είναι 3 (τρεις).

1. Αθλητής – Αθλήτρια που ανήκουν στο σύλλογο.
2. Προπονητής – Προπονήτρια που εργάζονται στο σύλλογο.
3. Χρήστης που δεν ανήκει στο σύλλογο.

Για τις πρώτες δύο κατηγορίες, ο κάθε χρήστης θα πρέπει να χρησιμοποιήσει τα διαπιστευτήρια LDAP (βλέπε Κεφάλαιο 2, Κεφάλαιο 3) προκειμένου να συνδεθεί στη εφαρμογή και να χρησιμοποιήσει τις λειτουργίες της.



Εικόνα 8 - Login Page

Όσον αφορά, την τρίτη κατηγορία χρηστών, επειδή δεν ανήκουν στο σύλλογο, δεν έχουν πρόσβαση στις επιπλέον λειτουργίες της διαδικτυακής εφαρμογής παρά μόνο στις γενικές πληροφορίες που παρέχει η ιστοσελίδα.

1.7.1 ΕΙΣΟΔΟΣ ΧΡΗΣΤΗ – ΛΕΙΤΟΥΡΓΙΕΣ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ

Εάν τα διαπιστευτήρια (username και password) που δώσει ο υποψήφιος χρήστης είναι σωστά τότε έχει πρόσβαση σε:

- Προβολή και επεξεργασία των στοιχείων του
- Ανακοινώσεις της ομάδας
- Πρόγραμμα προπονήσεων

Όνομα *

Admin

Επώνυμο *

Admin

Τμήμα *

ΤΖΟΥΝΘΡ

Έτος γέννησης *

1996

Ύψος (cm)

189

Βάρος (kg)

90

ΑΝΑΝΕΩΣΗ ΣΤΟΙΧΕΙΩΝ

ΜΕΝΟΥ ΕΠΙΛΟΓΩΝ

ΣΤΟΙΧΕΙΑ

ΑΝΑΚΟΙΝΩΣΕΙΣ

ΠΡΟΓΡΑΜΜΑ

Εικόνα 9 - Προβολή και Επεξεργασία στοιχείων μέσω της Δ.Ε.

Όπως φαίνεται και από την εικόνα 9, ο αθλητής ή η αθλήτρια που ανήκουν στο σύλλογο έχουν τη δυνατότητα να δουν τα στοιχεία τους και να τα τροποποιήσουν, καθώς για παράδειγμα, πιθανώς το ύψος ή το βάρος του δεν θα παραμένουν σταθερά. Στα δεξιά, υπάρχει ένα μενού επιλογών το οποίο περιλαμβάνει τις υπόλοιπες λειτουργίες της διαδικτυακής εφαρμογής.

Στατιστικά στοιχεία παιδικού τμήματος
23/04/2020
Παιδικό
Διαβάστε

Πρόγραμμα γυμναστικής-διατάσεων
23/04/2020
Κορίτσια, Προπαιδικό
Διαβάστε

Φιλικά με το Πλαγιάρι
17/02/2020
Προπαιδικό, Μίνι, Τζούνιορ
Διαβάστε

Κοπή Πρωτοχρονιάτικης πίτας
16/02/2020
Νέα
Διαβάστε

Ενημέρωση για τους λογαριασμούς
02/01/2020
Νέα
Διαβάστε

MENΟΥ ΕΠΙΛΟΓΩΝ

- ΣΤΟΙΧΕΙΑ
- ΑΝΑΚΟΙΝΩΣΕΙΣ
- ΠΡΟΓΡΑΜΜΑ

Εικόνα 10 - Προβολή Ανακοινώσεων

Στη δεύτερη επιλογή ο χρήστης που ανήκει στο σύλλογο έχει τη δυνατότητα να δει τις ανακοινώσεις των τμημάτων του συλλόγου.

Δευτέρα	Τετάρτη	Σάββατο
		<p>Φυσική Κατάσταση Ταχύτητα - Συντονισμός - Ευλυγισία - Ισορροπία. 13:00 - 14:00</p>
<p>Ατομική Τεχνική Διδασκαλία των βασικών ειδών ντρίμπλ και πάσας. 18:00 - 19:00</p>		
	<p>Ομαδική Τακτική Προπαρασκευαστικά ομαδικά παιχνίδια - βασικά του αθλήματος. 19:00 - 20:00</p>	

MENΟΥ ΕΠΙΛΟΓΩΝ

- ΣΤΟΙΧΕΙΑ
- ΑΝΑΚΟΙΝΩΣΕΙΣ
- ΠΡΟΓΡΑΜΜΑ

Εικόνα 11 - Προβολή προγράμματος προπονήσεων

Στην τρίτη και τελευταία επιλογή, ο χρήστης μπορεί να δει το πρόγραμμα των προπονήσεων του τμήματος στο οποίο ανήκει.

1.8 ΣΥΜΠΕΡΑΣΜΑ

Εν κατακλείδι, στο κεφάλαιο αυτό αναφέρθηκαν βασικές πληροφορίες του WordPress και τις δυνατότητες του, οι οποίες εξελίσσονται ραγδαία με την πάροδο των ετών. Το WordPress είναι λογισμικό ανοιχτού κώδικα και έχει μία ολόκληρη κοινότητα που απαρτίζεται από άτομα που επενδύουν στη συνεχή βελτίωση του και συμβάλλουν στην επέκταση νέου κώδικα, ως προς τη λειτουργικότητα του. Οι χρήστες, οι προγραμματιστές και οι σχεδιαστές του WordPress σκέφτονται πάντα νέους τρόπους χρήσης του WordPress και νέων προσθηκών, που διατίθενται στο ευρύ κοινό. Στην συνέχεια αναλύθηκαν τα βήματα που οφείλει να ακολουθήσει οποιοσδήποτε χρήστης προκειμένου να εγκαταστήσει το λογισμικό του WordPress και με βάση αυτό να κατασκευάσει τον ιστότοπο του. Τέλος, αναφέρθηκαν σημαντικές μέθοδοι τροποποίησης του εκάστοτε θέματος για βελτίωση της λειτουργικότητας της ιστοσελίδας και περιήγησης του υποψήφιου χρήστη.

ΚΕΦΑΛΑΙΟ 2

2. ΕΠΙΣΚΟΠΗΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ LDAP

2.1 ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΟΥ LDAP

Ο όρος LDAP σημαίνει Lightweight Directory Access Protocol. Όπως υποδηλώνει το όνομα, το LDAP είχε αρχικά σχεδιαστεί για να είναι ένα πρωτόκολλο δικτύου που παρέχει μια εναλλακτική μορφή πρόσβασης σε υπάρχοντες διακομιστές καταλόγου, αλλά καθώς η ιδέα του LDAP – και των τεχνολογιών που το περιείχαν – ωριμάζει, ο όρος LDAP έγινε συνώνυμος με έναν συγκεκριμένο τύπο αρχιτεκτονικής καταλόγου. Χρησιμοποιούμε τον όρο LDAP όταν γίνεται αναφορά σε υπηρεσίες καταλόγου που συμμορφώνονται με αυτήν την αρχιτεκτονική, όπως ορίζεται στις προδιαγραφές LDAP. Η δομή των προτύπων LDAP, συμπεριλαμβανομένων των πρωτοκόλλων δικτύου, της δομής καταλόγου και των υπηρεσιών που παρέχονται από έναν διακομιστή LDAP, είναι όλα διαθέσιμα με τη μορφή RFC (Request For Comments). Η τρέχουσα έκδοση του LDAP είναι LDAP v.3 (έκδοση 3), ένα πρότυπο που αναπτύχθηκε το 1997 ως RFC 2251, και εφαρμόζεται ευρέως σε ολόκληρο τον κλάδο.

2.2 ΤΙ ΕΙΝΑΙ ΕΝΑ DIRECTORY (ΚΑΤΑΛΟΓΟΣ);

Όταν σκεφτόμαστε έναν κατάλογο, δημιουργούμε εικόνες τηλεφωνικών καταλόγων ή βιβλίων διευθύνσεων. Χρησιμοποιούμε αυτούς τους καταλόγους για να βρούμε πληροφορίες για άτομα ή οργανισμούς. Ένας directory server (κατάλογος διακομιστή) χρησιμοποιείται επίσης με αυτόν τον τρόπο. Διατηρεί πληροφορίες σχετικά με ορισμένες ομάδες οντοτήτων (οντότητες όπως άτομα ή οργανισμούς) και παρέχει υπηρεσίες πρόσβασης σε αυτές τις πληροφορίες. Φυσικά, ένας κατάλογος πρέπει επίσης να διαθέτει τρόπους και μέσα για την προσθήκη, τροποποίηση και διαγραφή πληροφοριών. Όμως, ακόμη όπως και όταν ένας τηλεφωνικός θεωρείται πρωταρχικά πόρος για ανάγνωση, οι πληροφορίες ενός καταλόγου ενός διακομιστή θεωρείται ότι διαβάζονται συχνότερα από ότι γράφονται. Αυτή η υπόθεση σχετικά με τη χρήση ενός καταλόγου ενός διακομιστή συνοψίζεται στη φράση «υψηλή ανάγνωση, χαμηλή εγγραφή». Συνεπώς, πολλές εφαρμογές της τεχνολογίας LDAP έχουν ως στόχο την ανάγνωση και την αναζήτηση πληροφοριών. Ενώ πολλοί κατάλογοι διακομιστών έχουν βελτιστοποιηθεί για γρήγορη ανάγνωση σε βάρος της γρήγορης τροποποίησης, αυτό δεν ισχύει απαραίτητα με το OpenLDAP. Το OpenLDAP είναι αποδοτικό και στις δύο μετρήσεις και μπορεί να χρησιμοποιηθεί για εφαρμογές που

απαιτούν συχνή εγγραφή δεδομένων. Ορισμένα είδη καταλόγων διακομιστών παρέχουν απλώς μια συγκεκριμένη υπηρεσία. Ένας κατάλογος διακομιστή «μίας χρήσης», όπως ένα ηλεκτρονικό βιβλίο διευθύνσεων, μπορεί να αποθηκεύει μόνο έναν πολύ συγκεκριμένο τύπο δεδομένων, όπως αριθμούς τηλεφώνου, διευθύνσεις και πληροφορίες ηλεκτρονικού ταχυδρομείου για ένα σύνολο ατόμων. Τέτοιοι είδους κατάλογοι («μίας χρήσης») δεν είναι επεκτάσιμοι. Αλλά το LDAP σχεδιάστηκε για να είναι κατάλογος διακομιστή «γενικής χρήσης». Δεν έχει σχεδιαστεί με σκοπό τη σύλληψη ενός συγκεκριμένου τύπου δεδομένων όπως αριθμοί τηλεφώνου ή διευθύνσεις email. Σχεδιάστηκε για να παρέχει στους υπεύθυνους της εφαρμογής τη δυνατότητα να καθορίσουν, προσεκτικά, ποια δεδομένα πρέπει να αποθηκεύει ο κατάλογος. Ένας κατάλογος διακομιστή «γενικής χρήσης», θα πρέπει να αποθηκεύει πολλά διαφορετικά είδη πληροφοριών σχετικά με διαφορετικά είδη οντοτήτων. Ένα άτομο μπορεί να έχει Όνομα, Επώνυμο, Διεύθυνση, Email κ.λπ. όπως φαίνεται στο παρακάτω σχήμα.

Attribute	New Value	Skip
cn=Test User,ou=people,dc=my-domain,dc=com		
First name	Test	<input type="checkbox"/>
Last name	User	<input type="checkbox"/>
Common Name	Test User	<input type="checkbox"/>
Street	123 Any Way	<input type="checkbox"/>
Organisation	My Company	<input type="checkbox"/>
City	Anytown	<input type="checkbox"/>
State	State	<input type="checkbox"/>
Postal code	01234	<input type="checkbox"/>
Work phone	1-555-1212	<input type="checkbox"/>
Email	test_user@eapps-example.com	<input type="checkbox"/>
objectClass	inetOrgPerson	<input type="checkbox"/>

Εικόνα 12 - Πεδία εγγραφής σε ένα directory

2.3 Η ΔΟΜΗ ΜΙΑΣ ΕΓΓΡΑΦΗΣ ΚΑΤΑΛΟΓΟΥ (DIRECTORY ENTRY)

Όπως προαναφέρθηκε, ένας κατάλογος ενός διακομιστή μπορεί να συγκριθεί με έναν τηλεφωνικό κατάλογο. Ένας τηλεφωνικός κατάλογος περιέχει έναν πολύ συγκεκριμένο τύπο πληροφοριών, οργανωμένο με πολύ συγκεκριμένο τρόπο και σχεδιασμένο για να εκπληρώνει έναν πολύ συγκεκριμένο σκοπό. Ένα παράδειγμα εγγραφής σε ένα τηλεφωνικό κατάλογο θα μπορούσε να είναι:

Internet Services Company

Κωνσταντίνου Καραμανλή 78 Καλαμαριά

551 36

231 045 4250 231 045 4256

Παρατηρείται λοιπόν, ότι αυτό το είδος καταλόγου έχει συγκεκριμένες πληροφορίες, οργανωμένες με πολύ συγκεκριμένο τρόπο. Είναι πληροφορίες που έχουν σχεδιαστεί σχετικά με τον τρόπο επικοινωνίας με ένα συγκεκριμένο οργανισμό (Internet Service Company), που είναι οργανωμένος σε ένα οικείο μοτίβο (διεύθυνση και αριθμός τηλεφώνου). Έχει σχεδιαστεί με τέτοιο τρόπο, ώστε ένα άτομο, εάν θέλει να αναζητήσει ένα συγκεκριμένο όνομα, να μπορεί να αναζητήσει γρήγορα τον κατάλογο, ο οποίος πιθανότατα έχει ταξινομηθεί αλφαβητικά με βάση το όνομα του οργανισμού. Συνεπάγεται, ότι τα δεδομένα του καταλόγου τακτοποιούνται για αναζήτηση με βάση μία μόνο τιμή. Στη συγκεκριμένη περίπτωση είναι το όνομα του οργανισμού. Τι γίνεται όμως στην περίπτωση που ο χρήστης διαθέτει τον αριθμό τηλεφώνου ή την διεύθυνση και πρέπει η αναζήτηση να γίνει με βάση αυτά; Θα πρέπει οι πληροφορίες να οργανώνονται με τρόπο όπως τον παρακάτω:

Όνομα Επιχείρησης: Internet Services Company

Οδός: Κωνσταντίνου Καραμανλή 78

Περιοχή: Άγιος Ιωάννης

Πόλη: Θεσσαλονίκη

Πολιτεία: Καλαμαριά

Χώρα: Ελλάδα

Ταχυδρομικός Κώδικας: 55132

Αριθμός Τηλεφώνου: 231 045 4250

Αριθμός Τηλεφώνου: 231 045 4256

Στο παραπάνω παράδειγμα παρατηρείται ότι η έννοια των πληροφοριών γίνεται πιο σαφής. Πριν από κάθε τιμή υπάρχει ένα όνομα που προσδιορίζει τον τύπο των πληροφοριών που δίνονται. Το Internet Service Company αναγνωρίζεται πλέον ως το όνομα ενός οργανισμού. Επίσης, οι πληροφορίες χωρίζονται σε μικρότερα κομμάτια, όπως για παράδειγμα η πόλη και η περιοχή βρίσκονται σε ξεχωριστές γραμμές, ενώ ορισμένες πληροφορίες που ήταν έμμεσες στην αρχική εγγραφή, όπως η χώρα, πλέον έχουν καταστεί σαφείς. Επιπροσθέτως, όπου δύο κομμάτια πληροφοριών, όπως είναι οι δύο αριθμοί τηλεφώνου, αρχικά είχαν συμπειστεί σε μία γραμμή, τώρα έχουν διαχωριστεί, καθιστώντας τις πληροφορίες πιο σαφείς. Αυτή η μορφή καταχώρισης είναι πιο κοντά στον τρόπο εμφάνισης μιας εγγραφής σε έναν κατάλογο LDAP. Ωστόσο, υπάρχει ακόμη ένα ζήτημα που πρέπει να αντιμετωπιστεί. Πως είναι δυνατόν, να διαχωριστούν – διακριθούν δύο αρκετά όμοιες εγγραφές. Παραδειγματος χάρη, έστω ότι ένας χρήστης διαθέτει τον τηλεφωνικό κατάλογο της Καλαμαριάς. Στην Καλαμαριά, υπάρχουν δύο εταιρείες που διαθέτουν το ίδιο όνομα (Internet Services Company) και οι εγκαταστάσεις των δύο εταιρειών βρίσκονται αρκετά κοντά μεταξύ του (η μία εταιρεία έχει βάση τον Άγιο Ιωάννη και η άλλη εταιρεία την Νέα Κρήνη). Στην περίπτωση αυτή, η απλή γνώση της επωνυμίας του οργανισμού, δεν αποτελεί αρκετή πληροφορία για την απομόνωση μόνο μίας καταχώρισης στον τηλεφωνικό κατάλογο. Προκειμένου να επιτευχθεί η απομόνωση της πληροφορίας, απαιτείται ένα είδος μοναδικού ονόματος που υπάρχει μόνο μία φορά σε ολόκληρο τον κατάλογο, το οποίο μπορεί να χρησιμοποιηθεί για αναφορά σε μία συγκεκριμένη καταχώριση.

2.4 ΕΝΑ ΜΟΝΑΔΙΚΟ ΟΝΟΜΑ – ΤΟ DN

Ένας τρόπος διάκρισης μεταξύ δύο παρόμοιων εγγραφών είναι η δημιουργία ενός μοναδικού ονόματος για κάθε εγγραφή στον κατάλογο. Αυτή είναι η στρατηγική που υιοθέτησε το LDAP. Δηλαδή κάθε εγγραφή στον κατάλογο να έχει ένα διακεκριμένο όνομα. Το διακεκριμένο όνομα είναι ένας σημαντικός όρος για το LDAP και συμβολίζεται με την ονομασία DN (Distinguished Name). Σε έναν κατάλογο LDAP, ο υπεύθυνος σχεδιασμού του LDAP, είναι αυτός που αποφασίζει ποια στοιχεία θα συνθέσουν ένα DN, αλλά συνήθως το DN αντικατοπτρίζει που βρίσκεται η εγγραφή στον κατάλογο, καθώς και ορισμένες πληροφορίες που διακρίνουν αυτήν την εγγραφή από άλλες κοντινές εγγραφές. Ένα DN, αποτελείται από έναν συνδυασμό πληροφοριών καταλόγου και μοιάζει κάπως έτσι:

DN: o=Internet Service Company, l=Άγιος Ιωάννης, st=Καλαμαριά, c=ΕΛ

Ενώ το DN της άλλης εταιρείας μοιάζει κάπως έτσι:

DN: o=Internet Service Company, l=Νέα Κρήνη, st=Καλαμαριά, c=ΕΛ

Είναι φανερό λοιπόν ότι αυτό το μοναδικό χαρακτηριστικό αρκεί για να ξεχωρίσουν δύο ή περισσότερες πανομοιότυπες εγγραφές. Στο παραπάνω παράδειγμα, αποδεικνύεται ότι όταν ορίζονται ποια πεδία θα συνθέσουν ένα DN, αποτελεί προϋπόθεση, τα πεδία αυτά να είναι αρκετά λεπτομερής για να διακρίνουν δύο παρόμοιες καταχωρίσεις.

2.5 ΠΑΡΑΔΕΙΓΜΑ ΕΓΓΡΑΦΗΣ – ΚΑΤΑΧΩΡΙΣΗΣ LDAP

Μία καταχώριση ή εγγραφή LDAP, είναι μία μονάδα καταλόγου που αποθηκεύει πληροφορίες σχετικά με ένα μεμονωμένο στοιχείο στον κατάλογο. Όπως μία καταχώριση σε ένα τηλεφωνικό κατάλογο περιγράφει μια συγκεκριμένη ενότητα πληροφοριών σε αυτόν τον κατάλογο, ομοίως, μία εγγραφή σε ένα κατάλογο LDAP περιέχει πληροφορίες σχετικά με μία συγκεκριμένη μονάδα. Μπορεί αυτή η εγγραφή να είναι ένα άτομο, μία εταιρεία ή και ακόμη κάποια εικονική οντότητα όπως ένα αντικείμενο JAVA. Μια εγγραφή – καταχώριση LDAP αποτελείται από ένα DN και ένα ή περισσότερα χαρακτηριστικά. Το DN χρησιμεύει ως ένα μοναδικό αναγνωριστικό μέσα σε ένα δέντρο πληροφοριών καταλόγου LDAP. Τα χαρακτηριστικά παρέχουν πληροφορίες σχετικά με αυτήν την εγγραφή. Εάν μετατραπεί το προηγούμενο παράδειγμα σε μία εγγραφή LDAP, θα μοιάζει έτσι:

dn: o=Internet Service Company, l=Άγιος Ιωάννης, st=Καλαμαριά, c=ΕΛ

o=Internet Service Company

postalCode: 55132

l:Άγιος Ιωάννης

st: Καλαμαριά

tw: Θεσσαλονίκη

c: Ελλάδα

telephoneNumber: 231 045 4250

telephoneNumber: 231 045 4256

objectclass: organization

ΚΕΦΑΛΑΙΟ 2

Η πρώτη γραμμή αντιπροσωπεύει το DN (Distinguished Name). Όλες οι άλλες γραμμές σε αυτήν την εγγραφή αντιπροσωπεύουν χαρακτηριστικά. Η κύρια διαφορά μεταξύ του τελευταίου και των υπόλοιπων παραδειγμάτων είναι το ονόματα κάθε πεδίου σε κάθε εγγραφή – καταχώριση. Αυτά συμπιέζονται σε μία μορφή που ο κατάλογος LDAP μπορεί εύκολα να ερμηνεύσει. Αυτά τα ονόματα χαρακτηριστικών, όπως το postalCode, αναφέρονται σε καλά καθορισμένους ορισμούς χαρακτηριστικών που περιέχονται σε ένα σχήμα LDAP. Η δημιουργία νέων χαρακτηριστικών απαιτεί τη σύνταξη ενός σχήματος. Ένα χαρακτηριστικό περιγράφει έναν συγκεκριμένο τύπο πληροφοριών. Στο παράδειγμα μας υπάρχουν 8 (οχτώ) χαρακτηριστικά που αντιπροσωπεύουν τα ακόλουθα:

- o (Organization Name) = Ονομασία οργανισμού
- postalCode = Ταχυδρομικός κώδικας
- l (Locality) = Τοποθεσία, που μπορεί να είναι το όνομα της περιοχής, χωριού κ.λπ.
- st (State) = Πολιτεία
- tw (Town) = Πόλη
- c (Country) = Χώρα
- telephoneNumber = Αριθμός τηλεφώνου
- objectclass = Αντικείμενο κλάσης, η οποία καθορίζει τον τύπο ή τους τύπους της εγγραφής αυτής της καταχώρισης

Σε οποιαδήποτε εγγραφή, ένα χαρακτηριστικό ,μπορεί να έχει μία ή περισσότερες τιμές (αφού πρώτα έχει ρυθμιστεί μέσω της σχεδίασης του LDAP, ότι το συγκεκριμένο πεδίο επιτρέπει περισσότερες από μία τιμές) . Η παραπάνω εγγραφή έχει μόνο ένα χαρακτηριστικό που περιέχει περισσότερες από μία τιμές. Το χαρακτηριστικό telephoneNumber έχει δύο τιμές, καθεμία από αυτές αντιπροσωπεύει έναν διαφορετικό αριθμό τηλεφώνου.

Τα χαρακτηριστικά ορίζονται στους ορισμούς χαρακτηριστικών. Αυτοί οι ορισμοί παρέχουν πληροφορίες σχετικά με τη σύνταξη και το μήκος των πληροφοριών που αποθηκεύονται σε τιμές. Οι εγγραφές που είναι αποθηκευμένες σε καταλόγους LDAP πρέπει να συμμορφώνονται με τους ορισμούς των χαρακτηριστικών.

Για παράδειγμα, ο ορισμός χαρακτηριστικών για ένα όνομα χώρας δίνει τις ακόλουθες πληροφορίες:

- ✓ Τα προεπιλεγμένο όνομα για την ονομασία μίας χώρας είναι το c.
- ✓ Ένα όνομα χώρας αποθηκεύεται ως συμβολοσειρά.
- ✓ Το όνομα χώρας δεν μπορεί να υπερβαίνει τους 30 χαρακτήρες.
- ✓ Επιτρέπεται μόνο ένα όνομα χώρας ανά εγγραφή.

Όλες οι παραπάνω είναι πληροφορίες ενσωματωμένες σε έναν συμπαγή ορισμό σχήματος που «διαβάζει» ο κατάλογος διακομιστή όταν ξεκινά. Όσον αφορά τις τιμές των χαρακτηριστικών, η «ευαισθησία» των κεφαλαίων εξαρτάται από τον ορισμό των χαρακτηριστικών.

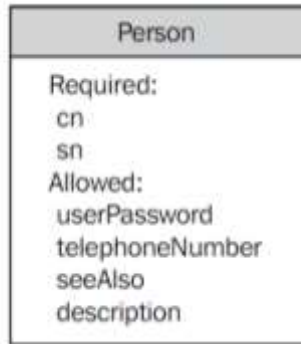
2.6 ΤΟ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ CLASS OBJECT

Το τελευταίο χαρακτηριστικό στο παράδειγμα μας είναι το χαρακτηριστικό `objectclass`. Αυτό είναι ένα ειδικό χαρακτηριστικό που παρέχει πληροφορίες σχετικά με τον τύπο εγγραφής ή καταχώρισης. Μια κλάση αντικειμένων καθορίζει ποια χαρακτηριστικά μπορούν να δοθούν σε μια εγγραφή. Στην περίπτωση μας, η κλάση αντικειμένων μας δείχνει ότι η εγγραφή αυτή περιγράφει έναν οργανισμό. Σύμφωνα με τον ορισμό αυτής της κλάσης αντικειμένων, μια εγγραφή οργανισμού μπορεί να περιέχει μια τοποθεσία (`Locality, l`), έναν ταχυδρομικό κώδικα (`postalCode`) καθώς και όλα τα άλλα χαρακτηριστικά που υπάρχουν στην εγγραφή. Δεδομένου του χαρακτηριστικού κλάσης αντικειμένου, το οποίο απαιτείται για οποιαδήποτε καταχώριση, ο κατάλογος μπορεί να καθορίσει ποια χαρακτηριστικά, πρέπει, μπορεί και δεν μπορεί να υπάρχουν στην καταχώριση. Όπως και με άλλα χαρακτηριστικά, το χαρακτηριστικό `objectclass` μπορεί να έχει πολλαπλές τιμές. Αν και ποιες τιμές μπορούν να δοθούν υπόκεινται στον ορισμό της κλάσης αντικειμένων και στον ορισμό σχήματος. Ένα σχήμα LDAP αποτελείται από κανόνες που καθορίζουν τους τύπους εγγραφών σε έναν κατάλογο και πως αυτές οι εγγραφές ενδέχεται να σχετίζονται μεταξύ τους. Μια εγγραφή μπορεί να έχει πολλές κλάσεις αντικειμένων, ωστόσο, μία από αυτές τις κλάσεις αντικειμένων πρέπει να είναι η κλάση δομικού αντικειμένου για την εγγραφή. Μία κλάση δομικού αντικειμένου καθορίζει τι είδους αντικείμενο είναι η εγγραφή.

2.7 ΚΑΤΗΓΟΡΙΕΣ ΑΝΤΙΚΕΙΜΕΝΩΝ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Υπάρχουν δύο διαφορετικοί τύπου ορισμού σχήματος που χρειαζόμαστε για να επεκτείνουμε τους τύπους πληροφοριών που θα αποθηκεύσει ο διακομιστής καταλόγου.

- ✓ Ορισμός τύπου χαρακτηριστικού: Ένας ορισμός τύπου χαρακτηριστικού ορίζει ένα χαρακτηριστικό, συμπεριλαμβανομένων των ονομάτων χαρακτηριστικών που μπορεί να έχει (όπως για παράδειγμα, `cn` και `commonName`), τι είδους τιμές μπορεί να περιέχει ένα χαρακτηριστικό (αριθμούς, συμβολοσειρές, DN και ούτω καθεξής). Κάθε δεδομένο χαρακτηριστικό μπορεί να απαιτεί οι τιμές του να αποτελούνται από συγκεκριμένους χαρακτήρες ή τύπους δεδομένων. Για παράδειγμα, ένα χαρακτηριστικό μπορεί να επιτρέπει μεγάλες σειρές χαρακτήρων, γεγονός που καθιστά δυνατή τη συμπερίληψη μιας πρότασης σε ένα πεδίο περιγραφής
- ✓ Ορισμός κλάσης αντικειμένων: Ο ορισμός κλάσης αντικειμένων καθορίζει το όνομα της κλάσης αντικειμένων, ποια χαρακτηριστικά πρέπει να έχει, ποια χαρακτηριστικά μπορεί να έχει και τι είδους αντικείμενο είναι. Μια γραφική αναπαράσταση της κατηγορίας αντικειμένου "Person" μπορεί αν είναι:



Εικόνα 13 - Κλάση αντικειμένου Person

Η κλάση αντικειμένου “Person” έχει δύο απαιτούμενα χαρακτηριστικά (cn και sn) και τέσσερα ακόμη χαρακτηριστικά που επιτρέπονται, αλλά δεν είναι υποχρεωτικά. Αυτά είναι: userPassword, telephoneNumber, seeAlso και description. Μια νέα εγγραφή που ανήκει στην κλάση αντικειμένου “Person” (και δεν έχει άλλες κατηγορίες αντικειμένων) μπορεί να μοιάζει με:

dn: cn: Thomas Reid, dc:example, dc:com

objectclass: person

cn: Thomas Reid

sn: Reid

userPassword: 3246t89jfbv

telephoneNumber: 555-555-5555

seeAlso: uid=david,ou=users,dc=example,dc=com

description: A basic user

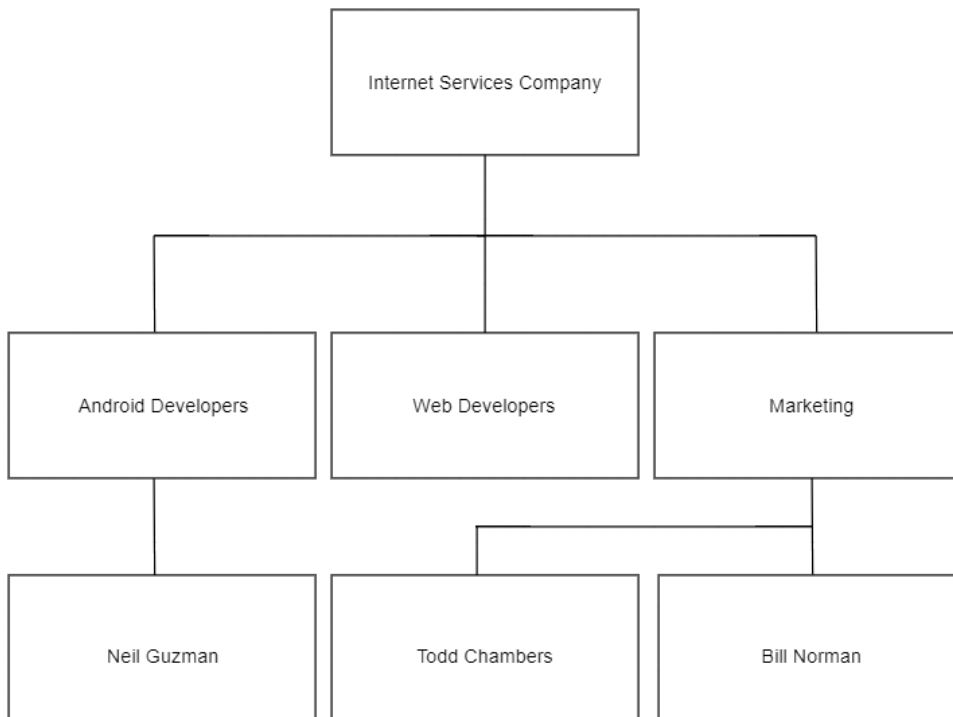
Αυτή η εγγραφή περιέχει όλα τα χαρακτηριστικά της κλάσης “Person”. Η απόπειρα προσθήκης διαφορετικού τύπου χαρακτηριστικού που δεν αναφέρεται στο σχήμα θα οδηγούσε σε σφάλμα. Ομοίως, η προσπάθεια κατάργησης όλων των τιμών για τα χαρακτηριστικά cn ή sn θα οδηγούσε επίσης σε σφάλμα καθώς τα χαρακτηριστικά είναι απαιτούμενα. Αλλά πως γνωρίζει ο LDAP ποια χαρακτηριστικά απαιτούνται και ποια επιτρέπονται; Αυτές οι πληροφορίες αποθηκεύονται στον ορισμό σχήματος για την κλάση “Person”

2.8 ΛΕΙΤΟΥΡΓΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Εκτός από τα κανονικά χαρακτηριστικά, ο κατάλογος του διακομιστή μπορεί επίσης να επισυνάψει ειδικά λειτουργικά χαρακτηριστικά σε μία καταχώριση. Τα λειτουργικά χαρακτηριστικά χρησιμοποιούνται από τον ίδιο τον ίδιο κατάλογο του διακομιστή για την αποθήκευση πληροφοριών σχετικά με τις καταχωρίσεις. Τέτοιου είδους χαρακτηριστικά, δεν έχουν σχεδιαστεί για χρήση από τους τελικούς χρήστες και συνήθως δεν επιστρέφονται κατά τη διάρκεια αναζητήσεων LDAP.

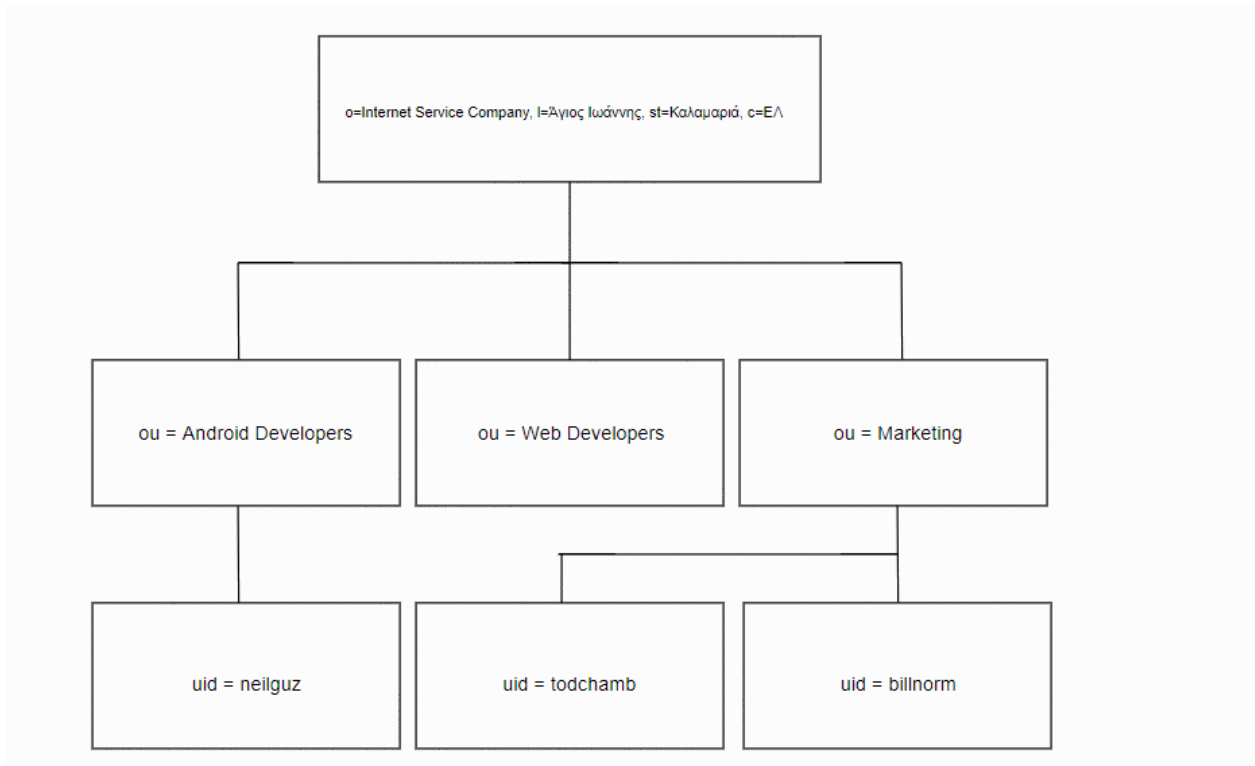
2.9 ΤΟ ΔΕΝΤΡΟ ΠΛΗΡΟΦΟΡΙΩΝ ΚΑΤΑΛΟΓΟΥ

Η μέχρι πρότινος σύγκριση ήταν μεταξύ ενός καταλόγου LDAP και ενός τηλεφωνικού καταλόγου. Ωστόσο υπάρχουν πρωταρχικές διαφορές μεταξύ της δομής δεδομένων σε έναν κατάλογο LDAP και εκείνης των άλλων μορφών καταλόγων. Οι πληροφορίες σε έναν τηλεφωνικό κατάλογο, συνήθως αποθηκεύονται σε μία μακρά αλφαβητική λίστα. Αλλά σε έναν κατάλογο LDAP η οργανωτική δομή είναι πιο εξελιγμένη. Οι πληροφορίες σε έναν κατάλογο LDAP οργανώνονται σε μία ή περισσότερες ιεραρχίες όπου, στην κορυφή της ιεραρχίας, υπάρχει μία καταχώριση βάσης, ενώ οι άλλες καταχωρίσεις οργανώνονται σε δομές που μοιάζουν με δέντρο. Οι καταχωρίσεις αυτές βρίσκονται κάτω από την καταχώριση βάσης. Κάθε κόμβος στην ιεραρχία είναι μία καταχώριση με DN και περισσότερα από ένα χαρακτηριστικά. Αυτή η ιεραρχικά οργανωμένη συλλογή καταχωρίσεων ονομάζεται δέντρο πληροφοριών καταλόγου, ενώ μερικές φορές αναφέρεται απλώς ως δέντρο καταλόγου η πιο συχνά DIT (Directory Information Tree). Προκειμένου να κατανοηθεί αυτή η μέθοδος οργάνωσης πληροφοριών, αρκεί να ληφθεί υπόψη το οργανόγραμμα μίας εταιρείας. Η κορυφή της ιεραρχίας είναι η ίδια η εταιρεία. Κάτω από την εταιρεία, υπάρχουν ορισμένα τμήματα της όπως οργανωτικές μονάδες, ενώ κάτω από αυτές βρίσκονται οι εργαζόμενοι και άλλα άτομα που έχουν επίσημη σχέση με την εταιρεία. Μπορούμε να σχεδιάσουμε το δέντρο πληροφοριών καταλόγου ως ιεραρχία:



Εικόνα 14 - Παράδειγμα DIT

Οι κατάλογοι LDAP αποθηκεύουν δεδομένα σε ιεραρχικές σχέσεις. Στην κορυφή του δέντρου πληροφοριών καταλόγου βρίσκεται η καταχώριση ρίζας. Κάτω από αυτήν είναι μια δευτερεύουσα καταχώριση, η οποία με τη σειρά της, μπορεί να έχει τις δικές της δευτερεύουσες καταχωρήσεις. Κάθε μία από αυτές τις εγγραφές έχει το δικό της DN και τα δικά της χαρακτηριστικά. Προσαρμόζοντας αυτό, στο παράδειγμα που προαναφέρθηκε θα μπορούσε εύκολα να δημιουργηθεί ένα δέντρο πληροφοριών καταλόγου LDAP που αντιπροσωπεύει το διάγραμμα:



Εικόνα 15 - Παράδειγμα DIT με DN

Αξίζει να σημειωθεί ότι το DN κάθε καταχώρηση περιέχει πληροφορίες σχετικά με την ανώτερη καταχώρηση του. Στην πραγματικότητα, ένα DN αποτελείται από δύο μέρη: το πρώτο μέρος είναι το σχετικό DN (Relative Distinguished Name) και περιέχει ένα ή περισσότερα χαρακτηριστικά από την καταχώρηση. Το δεύτερο μέρος είναι το πλήρες DN (Full Distinguished Name) της ανώτερης συμμετοχής. Οι εγγραφές που αποτελούνται από DN και ορισμένα χαρακτηριστικά, οργανώνονται σε μία ιεραρχία. Στην κορυφή της ιεραρχίας βρίσκεται η βασική καταχώρηση και κάτω από αυτές, οι εγγραφές οργανώνονται σε κλάδους.

2.10 ΤΙ ΝΑ ΚΑΝΕΤΕ ΜΕ ΕΝΑΝ LDAP

Ποια είναι η λειτουργία ενός διακομιστή LDAP; Τι πρόβλημα σκοπεύει να λύσει; Η πρώτη προφανής απάντηση είναι ότι το LDAP έχει σχεδιαστεί για να παρέχει έναν ψηφιακό κατάλογο. Η απάντηση θα μπορούσε να επεκταθεί, επισημαίνοντας ότι το LDAP παρέχει ένα στιβαρό επίπεδο υπηρεσιών, όπως είναι η αναζήτηση με σύνθετα φίλτρα, που αντιπροσωπεύουν σύνθετες οντότητες με χαρακτηριστικά, επιτρέποντας λεπτομερή πρόσβαση σε δεδομένα που παρέχουν εξελιγμένες υπηρεσίες καταλόγου. Μια πιο κλασική εξήγηση, που βασίζεται στην ιστορική ανάπτυξη του LDAP από τον κατάλογο X.500, είναι ότι το LDAP έχει σχεδιαστεί για να αντιπροσωπεύει οργανισμούς. Στην πραγματικότητα, αυτός είναι ένας

από τους πιο συνηθισμένους τρόπους χρήσης καταλόγων LDAP. Ένας κατάλογος LDAP αποθηκεύει πληροφορίες σε κάθε λογαριασμό χρήστη για κάποιο δίκτυο (πληροφορίες όπως το όνομα χρήστη, ο κωδικός πρόσβασης ή μία διεύθυνση email). Άλλες υπηρεσίες, στο ίδιο δίκτυο, χρησιμοποιούν το LDAP ως έγκυρη πηγή πληροφοριών χρήστη. Οι υπηρεσίες αυτές (ή εφαρμογές) μπορούν να κάνουν έλεγχο ταυτότητας των χρηστών στον κατάλογο. Επίσης, μπορεί να γίνει κοινή χρήση ενός λογαριασμού χρήστη σε πολλές (ίσως και σε όλες) τις εταιρικές εφαρμογές που ανήκουν στο ίδιο δίκτυο. Τέλος, υπάρχει μια πιο γενική άποψη της λειτουργίας των υπηρεσιών LDAP. Το LDAP δεν είναι τίποτα άλλο από ένα ειδικό είδος βάσης δεδομένων που οργανώνει δεδομένα σε δομές δέντρων, όπως μια ιεραρχία συστήματος αρχείων. Αυτή η άποψη, επαληθεύεται, συγκρίνοντας έναν κατάλογο LDAP με ένα σύστημα σχεσιακής βάσης δεδομένων (RDB). Οι σχεσιακές βάσεις δεδομένων αποθηκεύουν πληροφορίες σε πίνακες και οι πίνακες αποτελούνται από εγγραφές. Οι σχεσιακές βάσεις δεδομένων υποστηρίζουν εργασίες ανάγνωσης και γραφής σε δεδομένα, που συνήθως υλοποιούνται μέσω κάποιου SQL (Standard Query Language) και συνήθως επικοινωνούν με συνδέσεις δικτύου, καθιστώντας τα δεδομένα διαθέσιμα σε άλλες εφαρμογές στο δίκτυο. Σε σύγκριση λοιπόν με μία σχεσιακή βάση δεδομένων, το LDAP μπορεί επίσης να θεωρηθεί ως σύστημα αποθήκευσης. Το LDAP αποθηκεύει καταχωρήσεις όπως ένα σύστημα αρχείων. Οι βασικές σχέσεις σε ένα LDAP συνίστανται:

1. Superior to Subordinate - Στη σχέση ανώτερης προς υποδεέστερης (μία έως πολλές)
2. Subordinate to Superior - Της δευτερεύουσας σχέσης προς την ανώτερη (ένα προς ένα)

2.11 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ OPENLDAP

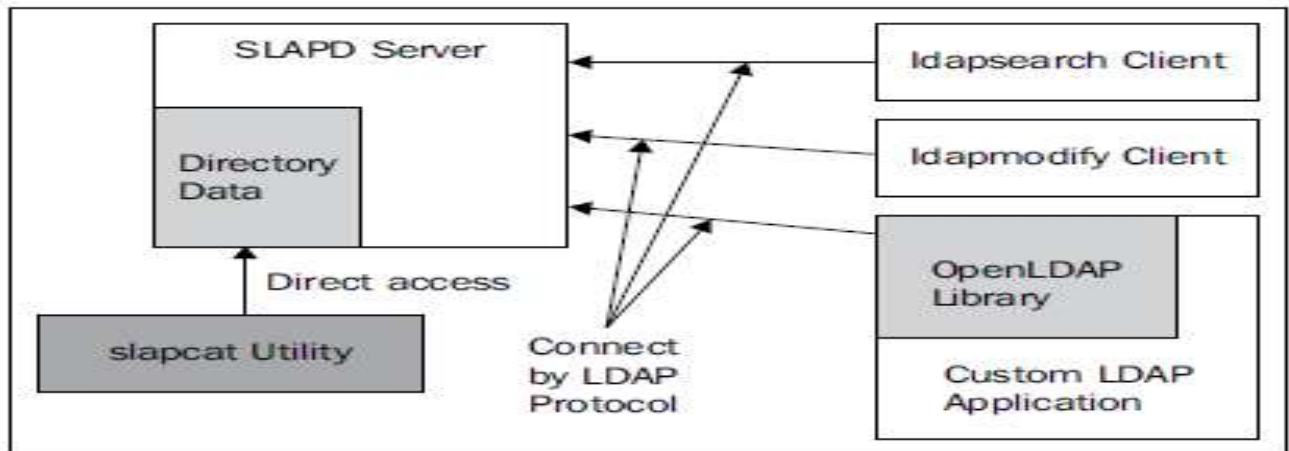
Το 1998 ξεκίνησε το έργο OpenLDAP από τον Kurt Zeilenga. Λίγο αργότερα, ο Χάουαρντ Τσου (ο σημερινός αρχιτέκτονας του έργου) εντάχθηκε και αυτός στο έργο. Μαζί, «έσωσαν» τη βάση του κώδικα του Πανεπιστημίου του Μίσιγκαν, ξεκινώντας εκ νέου την ανάπτυξη του. Το αποτέλεσμα αυτής της προσπάθειας, ήταν το OpenLDAP 2.0. Το OpenLDAP 2.0 ήταν πολύ επιτυχημένο και εισήλθε σχεδόν σε κάθε σημαντική «διανομή» Linux. Από τα τέλη της δεκαετίας του '90, το OpenLDAP, συνέχιζε να ωριμάζει, εποπτευόμενο από το OpenLDAP Foundation και υποστηρίζεται από συνεισφορές από χορηγούς του κλάδου. Η έκδοση 2.49 είναι η σταθερή έκδοση. Το OpenLDAP εξακολουθεί να τηρεί τα πρότυπα LDAP. Στην πραγματικότητα, ο Kurt Zeilenga είναι υπεύθυνος για πολλές από τις ενημερώσεις που έγιναν στα πρότυπα LDAP τον Ιούνιο του 2006. Ωστόσο, εκτός από τον υψηλό βαθμό συμμόρφωσης προς τα πρότυπα, το OpenLDAP είναι επίσης ένας από τους ταχύτερους καταλόγους διακομιστών, ξεπερνώντας κατά πολύ τις ταχύτητες από άλλες υλοποιήσεις καταλόγων διακομιστών ανοιχτού κώδικα.

2.12 ΤΕΧΝΙΚΗ ΕΠΙΣΚΟΠΗΣΗ ΤΟΥ OPENLDAP

Το OpenLDAP μπορεί να χωριστεί σε τέσσερα στοιχεία:

1. Servers (Διακομιστές) : Παρέχονται υπηρεσίες LDAP
2. Clients (Πελάτες) : Χειρισμός δεδομένων LDAP
3. Utilities (Βοηθητικά Προγράμματα) : Υποστήριξη διακομιστών LDAP
4. Libraries (Βιβλιοθήκες) : Παροχή διεπαφών προγραμματισμού στο LDAP

Το παρακάτω διάγραμμα εξηγεί πως αυτά τα τέσσερα στοιχεία σχετίζονται μεταξύ τους.



Εικόνα 16 - Τα 4 στοιχεία του OpenLDAP

2.12.1 Ο ΔΙΑΚΟΜΙΣΤΗΣ (SERVER)

Ο κύριος διακομιστής στον LDAP είναι το SLAPD (Stand – Alone LDAP Daemon). Αυτός ο διακομιστής παρέχει πρόσβαση σε ένα ή περισσότερα δέντρα πληροφοριών καταλόγου. Οι πελάτες συνδέονται με το διακομιστή μέσω του πρωτοκόλλου LDAP, συνήθως χρησιμοποιώντας μια σύνδεση μέσω δικτύου (αν και το SLAPD παρέχει και έναν ακροατή υποδοχής UNIX). Ένας διακομιστής μπορεί να αποθηκεύσει δεδομένα καταλόγου τοπικά, ή απλά πρόσβαση σε εξωτερικές πηγές. Συνήθως, παρέχει υπηρεσίες ελέγχου ταυτότητας και αναζήτησης και μπορεί επίσης να υποστηρίζει την προσθήκη, την αφαίρεση και την τροποποίηση δεδομένων καταλόγου, ενώ παρέχει λεπτομερή έλεγχο πρόσβασης στον κατάλογο.

2.12.2 ΟΙ ΠΕΛΑΤΕΣ (CLIENTS)

Οι πελάτες έχουν πρόσβαση σε διακομιστές LDAP μέσω του πρωτοκόλλου δικτύου του LDAP. Λειτουργούν ζητώντας από τον διακομιστή να εκτελεί λειτουργίες για λογαριασμό τους. Συνήθως, ένας πελάτης θα συνδεθεί πρώτα με τον κατάλογο του διακομιστή, (αφού έχει προηγηθεί ο έλεγχος ταυτότητας του πελάτη) και στη συνέχεια θα εκτελέσει μηδενικές ή περισσότερες λειτουργίες (αναζητήσεις, τροποποιήσεις, προσθήκες, διαγραφές κ.λπ.) πριν τελικά καταργήσει τη σύνδεση και αποσυνδεθεί.

2.12.3 ΒΟΗΘΗΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ (UTILITIES)

Σε αντίθεση με τους πελάτες, τα βοηθητικά προγράμματα δεν εκτελούν λειτουργίες χρησιμοποιώντας το πρωτόκολλο LDAP. Αντ' αυτού, χειρίζονται δεδομένα σε χαμηλότερο επίπεδο και χωρίς διαμεσολάβηση από τον διακομιστή. Χρησιμοποιούνται κυρίως για τη συντήρηση του διακομιστή.

2.12.4 ΟΙ ΒΙΒΛΙΟΘΗΚΕΣ (LIBRARIES)

Υπάρχουν πολλές βιβλιοθήκες OpenLDAP που είναι κοινόχρηστες μεταξύ εφαρμογών LDAP. Οι βιβλιοθήκες παρέχουν λειτουργίες LDAP σε αυτές της εφαρμογές. Οι πελάτες, τα βοηθητικά προγράμματα και οι διακομιστές μοιράζονται πρόσβαση σε ορισμένες από αυτές τις βιβλιοθήκες. Οι διεπαφές προγραμματισμού εφαρμογών (API) παρέχονται για να επιτρέπουν στους προγραμματιστές λογισμικού να γράφουν τις δικές τους εφαρμογές που «γνωρίζουν» το LDAP χωρίς να χρειάζεται να ξαναγράψουν βασικό κώδικα LDAP. Ενώ τα API που παρέχονται με το OpenLDAP είναι γραμμένα σε C, το OpenLDAP παρέχει επίσης δύο API γραμμένα σε JAVA.

2.13 ΣΥΜΠΕΡΑΣΜΑ

Σε αυτό το κεφάλαιο αναλύθηκαν όλες οι βασικές ορολογίες των καταλόγων LDAP γενικά και του διακομιστή OpenLDAP ειδικότερα. Παράλληλα, πραγματοποιήθηκαν αναφορές για την ιστορία του LDAP, τη σημαντική ορολογία και ορισμένες από τις υψηλού επιπέδου τεχνικές πτυχές του OpenLDAP.

ΚΕΦΑΛΑΙΟ 3

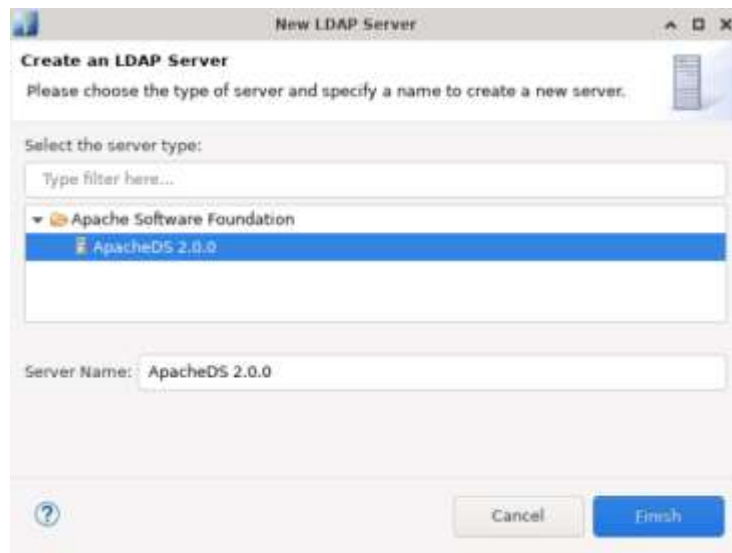
3 ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ LDAP

3.1 ΛΗΨΗ ΚΑΙ ΕΓΚΑΤΑΣΤΑΣΗ

Το πρόγραμμα που χρησιμοποιήθηκε για την εγκατάσταση και διαμόρφωση του LDAP είναι το Apache Directory Studio. Το Apache Directory Studio αποτελεί λογισμικό ανοιχτού κώδικα που αρχικά γράφτηκε από τον Alex Karasulu. Είναι ένας ενσωματωμένος διακομιστής καταλόγων που είναι γραμμένος εξ ολοκλήρου σε JAVA. Είναι συμβατός με το LDAP v3 και υποστηρίζει και άλλα πρωτόκολλα και έναν διακομιστή Kerberos. Η τελευταία έκδοση του Apache Directory Studio μπορεί να «κατέβει» από τον κατάλογο λήψεων, σε αυτή την διεύθυνση (<https://directory.apache.org/studio/downloads.html>).

3.2 ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ LDAP SERVER

Για να δημιουργηθεί ένας νέος διακομιστής LDAP, στη γραμμή εργαλείων, κάντε κλικ στην επιλογή Νέος Διακομιστής. Εμφανίζεται ο ακόλουθος οδηγός:



Εικόνα 17 - Δημιουργία LDAP

Όπως φαίνεται από τα πεδία, πρέπει να δοθεί ένα όνομα στο διακομιστή.

Επιλογή	Περιγραφή	Προεπ. Τιμή
Τύπος Διακομιστή	Ο τύπος του διακομιστή LDAP. Αυτή τη στιγμή ο μόνος διαθέσιμος είναι ο Apache Directory Server (ApacheDS).	Null
Όνομα Διακομιστή	Το όνομα του διακομιστή LDAP. Το όνομα του διακομιστή πρέπει να είναι μοναδικό.	null

3.1.1 ΕΚΚΙΝΗΣΗ SERVER

Για την εκκίνηση του διακομιστή, στην προβολή διακομιστών, πρέπει να επιλεγθεί ο διακομιστής τον οποίο θέλετε να ξεκινήσετε και κάντε κλικ στο κουμπί Εκτέλεση στη γραμμή εργαλείων.

3.1.2 ΔΙΑΚΟΠΗ SERVER

Για τη διακοπή ενός διακομιστή, στην προβολή διακομιστών, πρέπει να επιλεγθεί ο διακομιστής τον οποίο θέλετε να διακόψετε και κάντε κλικ στο κουμπί Διακοπή στη γραμμή εργαλείων.

3.1.3 ΕΠΕΞΕΡΓΑΣΙΑ ΤΗΣ ΔΙΑΜΟΡΦΩΣΗΣ ΤΟΥ SERVER

Για την επεξεργασία της διαμόρφωσης ενός διακομιστή, στην προβολή διακομιστών, πρέπει να επιλεγθεί ο διακομιστής τον οποίο θέλετε να επεξεργαστείτε, κάνοντας διπλό κλικ σε αυτόν και ανοίγοντας την ενέργεια Διαμόρφωση στο μενού περιβάλλοντος.

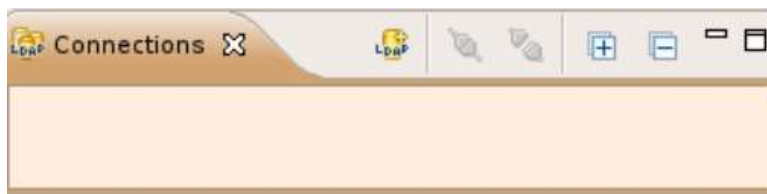
3.1.4 ΔΙΑΓΡΑΦΗ SERVER

Για τη διαγραφή ενός διακομιστή, στην προβολή διακομιστών, πρέπει να επιλεγθεί ο διακομιστής τον οποίο θέλετε να διαγράψετε και κάντε κλικ στην ενέργεια Διαγραφή στο μενού περιβάλλοντος.

3.2 ΔΗΜΙΟΥΡΓΙΑ ΣΥΝΔΕΣΗΣ ΜΕ ΤΟ LDAP

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, ένας κατάλογος LDAP αποθηκεύει πληροφορίες σε κάθε λογαριασμό χρήστη για κάποιο δίκτυο (πληροφορίες όπως το όνομα χρήστη, ο κωδικός πρόσβασης ή μία διεύθυνση email). Άλλες υπηρεσίες, στο ίδιο δίκτυο, χρησιμοποιούν το LDAP ως έγκυρη πηγή πληροφοριών χρήστη. Έτσι λοιπόν, θα πρέπει να δημιουργηθεί κάποια σύνδεση μεταξύ του LDAP και κάποιου δικτύου.

Στην κάτω αριστερή γωνία εμφανίζονται όλες οι συνδέσεις LDAP. Όπως διακρίνεται στην εικόνα, δεν υπάρχει προεπιλεγμένη τιμή, άρα ο χρήστης πρέπει να καθορίσει μία σύνδεση. Για να δημιουργηθεί μία νέα σύνδεση κάντε κλικ στο κουμπί Νέα Σύνδεση.

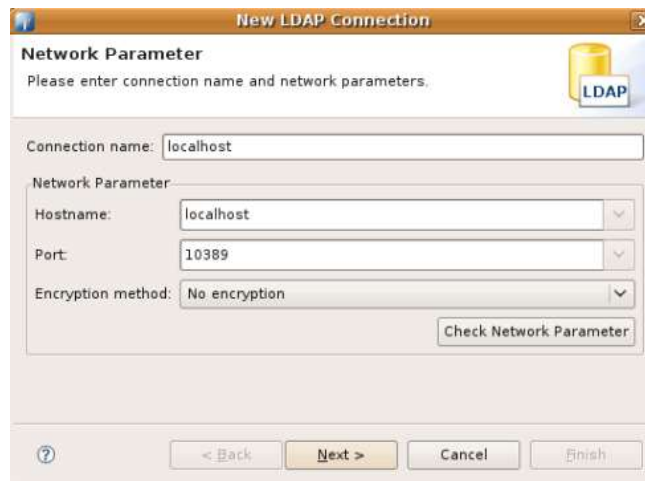


Εικόνα 18 - Σύνδεση LDAP σε δίκτυο (Μέρος Α)

Στην πρώτη σελίδα του οδηγού θα πρέπει να εισαχθούν:

1. Ένα όνομα σύνδεσης
2. Όνομα κεντρικού υπολογιστή
3. Θύρα του διακομιστή LDAP

Για την ορθότητα των παραμέτρων που εισήγατε, μπορείτε να κάνετε κλικ στην επιλογή Έλεγχος παραμέτρου δικτύου.



Εικόνα 19 - Σύνδεση LDAP σε δίκτυο (Μέρος Β)

Παρατηρείται από την εικόνα, ότι η ονομασία της σύνδεσης μπορεί να είναι οποιαδήποτε αρκεί να είναι μοναδική. Η ονομασία της σύνδεσης θα πρέπει να διαφέρει από τις υπόλοιπες. Όσον αφορά, το όνομα του κεντρικού υπολογιστή (Hostname), ανάλογα σε ποιο δίκτυο επιθυμεί ο χρήστης να «βλέπει» το LDAP. Εάν για παράδειγμα ο LDAP, επιθυμούμε να βλέπει το τοπικό δίκτυο του υπολογιστή μας, χρησιμοποιούμε την επιλογή localhost.

3.2.1 Η ΔΙΑΔΙΚΑΣΙΑ ΔΕΣΜΕΥΣΗΣ (BINDING PROCEDURE)

Στη δεύτερη σελίδα του οδηγού εγκατάστασης, ο χρήστης πρέπει να επιλέξει τη μέθοδο , για το πως θα γίνεται ο έλεγχος ταυτότητας.



Εικόνα 20 - Διαδικασία δέσμευσης

Η διαδικασία αυτή πιστοποιεί τον πελάτη στον εξυπηρετητή. Η πιστοποίηση των χρηστών γίνεται μέσω ενός bind request στον διακομιστή LDAP με το login/password που παρέχει ο χρήστης κατά την σύνδεση. Κατά αυτόν τον τρόπο υποστηρίζονται όλοι οι τρόποι κρυπτογράφησης του password του χρήστη τους οποίους υποστηρίζει ο διακομιστής LDAP. Επιπλέον παρέχεται η δυνατότητα η αποστολή του login και του password στον ldap server να γίνει μέσω ασφαλούς σύνδεσης SSL.

Το πρώτο πράγμα που πρέπει να συμβεί είναι ότι ο πελάτης πρέπει να πραγματοποιήσει έλεγχο ταυτότητας στον διακομιστή. Πρέπει να ληφθεί υπόψη ότι για να αλληλεπιδράσει με ένα διακομιστή LDAP, ο πελάτης πρέπει να παρέχει δύο πληροφορίες:

1. Ένα DN
2. Και έναν κωδικό πρόσβασης.

Συνήθως, υπάρχουν δύο διαφορετικοί τρόποι με τους οποίους ένας πελάτης μπορεί να πραγματοποιήσει έλεγχο ταυτότητας σε ένα διακομιστή. Ο πρώτος είναι μέσω ενός Simple Bind και ο δεύτερος μέσω ενός SASL Bind. Εκτός από αυτές τις δύο μεθόδους, είναι δυνατόν να γραφτούν προσαρμοσμένες μέθοδοι δέσμευσης ωστόσο αυτό είναι ένα δύσκολο και σημαντικό εγχείρημα.

Ο παραπάνω τρόπος που βλέπουμε στην Εικόνα 15, είναι ο τρόπος σύνδεσης των πελατών στο LDAP χρησιμοποιώντας τη μέθοδο Simple Bind. Συνήθως, για τον έλεγχο της ταυτότητας ενός χρήστη, το SLAPD αναζητά το DN (και το χαρακτηριστικό userPassword του DN) στον κατάλογο και επαληθεύει τα ακόλουθα:

- ✓ Το παρεχόμενο DN, αν υπάρχει στον κατάλογο.
- ✓ Ο κωδικός πρόσβασης που παρέχεται, αν ταιριάζει με την τιμή του χαρακτηριστικού userPassword του DN.

Στο παράδειγμα μας, ο χρήστης admin θέλει να δεσμευτεί στον κατάλογο. Για να δεσμευτεί ο admin, σύμφωνα με τα περιγραφόμενα βήματα, ο πελάτης θα πρέπει να παρέχει το πλήρες DN του admin, το οποίο (όπως το παράδειγμα) μπορεί να είναι κάτι σαν:

```
dn: uid=admin, ou=system
```

Όμως υπάρχει μεγάλη πιθανότητα, όλοι οι πελάτες να μην γνωρίζουν το πλήρες DN του χρήστη. Οι περισσότερες εφαρμογές απαιτούν μόνο ένα όνομα χρήστη και έναν κωδικό πρόσβασης, όχι ένα πλήρες DN. Για την επίλυση αυτού του προβλήματος, οι διακομιστές LDAP υποστηρίζουν την ιδέα του Ανώνυμου χρήστη. Όταν ο διακομιστής LDAP λαμβάνει ένα αίτημα δέσμευσης με ένα κενό DN και ένα κενό πεδίο κωδικού πρόσβασης, ο διακομιστής αντιμετωπίζει το χρήστη ως Ανώνυμος. Ο Ανώνυμος χρήστης, είτε μπορεί να του επιτραπεί είτε να του απαγορευτεί η πρόσβαση στις πληροφορίες στον κατάλογο βάσει των ACL που καθορίζονται για το SLAPD. Οι ACL παρέχουν υπηρεσίες εξουσιοδότησης, οι οποίες καθορίζουν σε ποιες πληροφορίες έχει πρόσβαση ένα δεδομένο DN. Γενικά το έργο του Ανώνυμου χρήστη είναι να βγάλει το DN του (π.χ.) Mike από τον κατάλογο και να ζητήσει τον έλεγχο ταυτότητας του Mike. Ο τρόπος με τον οποίο επιτυγχάνεται αυτό είναι ότι ο πελάτης συνδέεται πρώτα με τον διακομιστή ως Ανώνυμος και έπειτα αναζητά στον κατάλογο για την καταχώρηση Mike με ένα είδους φίλτρο όπως: «καταχωρήσεις των οποίων το CN είναι Mike και (π.χ.) το αντικείμενο είναι organizationalPerson». Το πραγματικό φίλτρο για αυτό το αίτημα θα μοιάζει με:

```
(& (cn = Mike) (objectClass = organizationalPerson))
```

Υποθέτοντας ότι το φίλτρο είναι αρκετά συγκεκριμένο και ο κατάλογος έχει πράγματι μία καταχώριση για τον Mike, τότε η απάντηση του διακομιστή στη συνέχεια προς τον πελάτη θα ήταν ένα DN: cn = Mike, dc = όνομα οργανισμού. Στην συνέχεια, ο πελάτης θα πραγματοποιούσε ξανά bind request (αίτημα δέσμευσης), αυτή τη φορά ως cn = Mike, dc = όνομα οργανισμού και φυσικά με τον κωδικό πρόσβασης του Mike, αντί για Ανώνυμος Χρήστης. Για να λειτουργήσει φυσικά, όλη αυτή η μέθοδος του Ανώνυμου Χρήστη στον έλεγχο ταυτότητας, θα πρέπει οι ACL να επιτρέψουν στον Ανώνυμο Χρήστη να κάνει αίτηση για δέσμευση και να επιχειρήσει να πραγματοποιήσει έλεγχο ταυτότητας. Ωστόσο, στην δική μας

περίπτωση, χρησιμοποιείται το Simple Binding. Το Simple Binding στέλνει το uid και τον κωδικό πρόσβασης από τον πελάτη στον διακομιστή.

3.2.2 Η ΔΙΑΔΙΚΑΣΙΑ ΑΝΑΖΗΤΗΣΗΣ (SEARCH PROCEDURE)

Στο παράδειγμα μας, ο admin αφού ελεγχθεί από το διακομιστή, ξεκινάει η διαδικασία της αναζήτησης. Για να πραγματοποιηθεί η διαδικασία της αναζήτησης στον κατάλογο πρέπει να είναι γνωστά τα ακόλουθα πράγματα:

- Βασικό DN: Από που ξεκινάει ο κατάλογος
- Πεδίο Εφαρμογής: Πόσο «βαθιά» θα αναζητήσει στο δέντρο
- Χαρακτηριστικά: Τι πληροφορίες θα ανακτηθούν
- Φίλτρο: Τι ακριβώς θα αναζητήσει

Εφόσον το παράδειγμα μας αφορά δεδομένα αθλητικού συλλόγου, ας δούμε τι θέλει ο admin να αναζητήσει στον κατάλογο. Ο admin θέλει να πάρει μια λίστα με όλα τα άτομα στον οργανισμό του, για το marathonbc.gr., που έχουν διευθύνσεις ταχυδρομείου που ξεκινούν με το γράμμα t. Από αυτές τις πληροφορίες, μπορούμε να δημιουργήσουμε μία αναζήτηση. Πρώτον, ο admin θέλει να μάθει για όλους στον οργανισμό marathonbc.gr. Στον κατάλογο, αυτά είναι πάντα κάτω από μία καταχώριση:

dc=marathonbc, dc=gr

Επίσης γνωρίζουμε, ότι ο admin, θέλει όλες τις διευθύνσεις ηλεκτρονικού ταχυδρομείου που ξεκινούν με το γράμμα t και όχι μόνο ένα επίπεδο κάτω. Έτσι λοιπόν ο admin, θέλει να ψάξει σε ολόκληρο το υπόστρωμα στο *dc=marathonbc, dc=gr*. Έχουμε λοιπόν:

- Base DN: *dc=marathonbc, dc=gr*
- Scope: *Entire Tree*

Στη συνέχεια, θέλουμε να μάθουμε ποια χαρακτηριστικά θέλει ο admin να επιστρέψει ο διακομιστής. Το DN θα επιστραφεί αυτόματα. Εκτός όμως από το DN, ο admin ασχολείται μόνο με το χαρακτηριστικό που αποθηκεύει τη διεύθυνση email. Οι διευθύνσεις email αποθηκεύονται στα χαρακτηριστικά του χρήστη. Θα μπορούσε ο admin να ασχολείται επίσης με οποιοδήποτε χαρακτηριστικό, ή περισσότερα από ένα χαρακτηριστικά, όπως το όνομα χρήστη (cn) και τον αριθμό τηλεφώνου. Επομένως πρέπει να δημιουργηθεί ένα φίλτρο από τα κριτήρια του admin. Ο admin θέλει όλες τις καταχωρήσεις, που είναι χρήστες και η διεύθυνση email τους ξεκινά με το γράμμα t.

Το φίλτρο αναζήτησης:

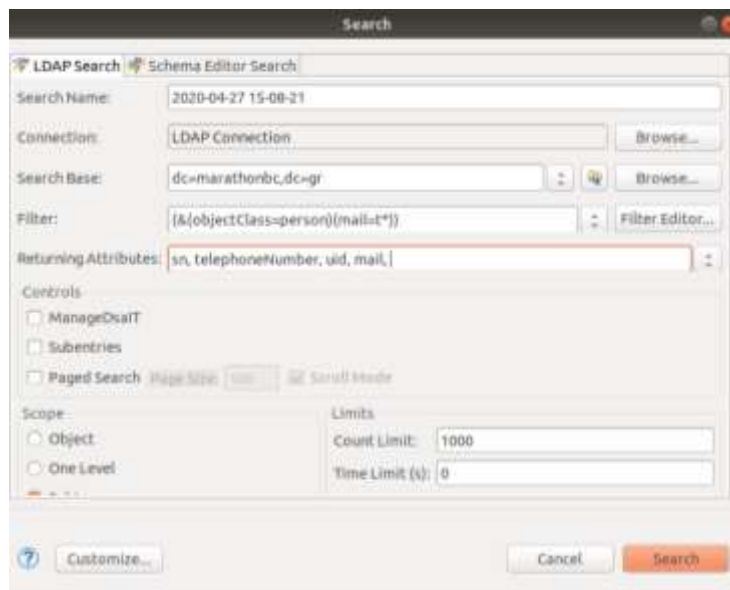
`(& (objectClass=person) (mail=t*))`

Αυτό το απλό φίλτρο αποτελείται από τέσσερα μέρη:

1. Πρώτον, το φίλτρο περικλείεται από παρενθέσεις. Οι παρενθέσεις χρησιμοποιούνται για την ομαδοποίηση στοιχείων εντός του φίλτρου. Για οποιοδήποτε φίλτρο, ολόκληρο το φίλτρο πρέπει πάντα να περικλείεται σε παρένθεση.
2. Δεύτερον, το φίλτρο ξεκινά είτε με κάποιον τελεστή (υπάρχουν τρεις λογικοί τελεστές που μπορούν να χρησιμοποιηθούν σε φίλτρα: ΚΑΙ (&), Ή (|) και ΟΧΙ !), είτε με μία περιγραφή χαρακτηριστικού (objectClass).
3. Τρίτον, υπάρχει οι κανόνες της αντιστοίχισης. Υπάρχουν τέσσερις κανόνες αντιστοίχισης:
 1. Ισότητα (=)
 2. Κατά προσέγγιση ισότητα (~=)
 3. Μεγαλύτερη από ή ίση με (>=)
 4. Μικρότερη από ή ίση με (<=)

Ο τρόπος με τον οποίο χρησιμοποιούνται (και αν μπορούν να χρησιμοποιηθούν) καθορίζεται σε μεγάλο βαθμό από το σχήμα καταλόγου.

4. Τέλος, έχουμε την τιμή επιβεβαίωσης, τη συμβολοσειρά ή το μοτίβο που θέλουμε να ταιριάζουν τα αποτελέσματα. Στη δική μας περίπτωση αποτελείται από τον χαρακτήρα t, σε συνδυασμό με τον χαρακτήρα μπαλαντέρ, ενώ υπάρχει και συμβολοσειρά person. Ο χαρακτήρας t μαζί με το μπαλαντέρ (*) υποδηλώνει ότι η συμβολοσειρά πρέπει να ξεκινάει με t και στη συνέχεια μπορεί να έχει μηδέν ή περισσότερους χαρακτήρες μετά τον χαρακτήρα t.



Εικόνα 21 - Φίλτρο Αναζήτησης LDAP

Μόλις ο admin εκτελέσει την αναζήτηση, με το βασικό DN, το εύρος, τα χαρακτηριστικά και το φίλτρο, θα λάβει μία απάντηση από τον διακομιστή που θα περιέχει μία λίστα με εγγραφές ή μία εγγραφή που μοιάζουν με αυτό.

```
dn: cn=Turner Parker, dc=marathonbc, dc=gr
```

```
sn: Parker
```

```
telephoneNumber: 6944553246
```

```
uid: turpar
```

```
mail: turnerparker@gmail.com
```

Η αναζήτηση επιστρέφει όλα όσα εμφανίζονται στο υποδέντρο κάτω από το DN: dc = marathonbc, dc = gr που ταιριάζουν στο φίλτρο μας. Οι επιστρεφόμενη εγγραφής (ή εγγραφές) έχουν μόνο το DN και τα χαρακτηριστικά που καθορίσαμε: sn, mail, telephoneNumber, uid και το mail. Ωστόσο, κάτι τελευταίο που αξίζει να αναφερθεί, όσον αφορά την αναζήτηση, είναι ότι κατά την διάρκεια μίας αναζήτησης, ολόκληρο το αίτημα ελέγχεται έναντι της λίστας των στοιχείων ελέγχου πρόσβασης. Εάν ένας ACL καθορίσει ότι ο admin (ή οποιοσδήποτε χρήστης το επιχειρήσει) , δεν έχει πρόσβαση στο χαρακτηριστικό (π.χ. telephoneNumber), τότε η αναζήτηση θα επιστρέψει τα ίδια DN αλλά χωρίς το χαρακτηριστικό telephoneNumber. Ομοίως, εάν ένας ACL αρνηθεί την πρόσβαση του admin στα αρχεία ορισμένων ατόμων στον κατάλογο, τότε ο διακομιστής θα στείλει πίσω μόνο τα αποτελέσματα εκείνα που ο admin έχει άδεια να δει. Ο διακομιστής, επίσης, δεν θα δώσει στον admin καμία ένδειξη ότι έχουν παρακρατηθεί πληροφορίες λόγω ACL.

3.2.3 ΠΕΡΙΣΣΟΤΕΡΕΣ ΔΙΑΔΙΚΑΣΙΕΣ (ΠΡΟΣΘΗΚΗ, ΤΡΟΠΟΠΟΙΗΣΗ, ΔΙΑΓΡΑΦΗ)

Στο παράδειγμα μας, ο admin πραγματοποιεί αναζήτηση για διευθύνσεις ηλεκτρονικού ταχυδρομείου, καλύπτοντας ως διαδικασίες μόνο τη δέσμευση (binding) και την αναζήτηση (searching). Φυσικά, το LDAP υποστηρίζει επίσης την προσθήκη, την τροποποίηση και τη διαγραφή. Και τα τρία από αυτά απαιτούν, ο χρήστης να έχει δεσμευτεί πρώτα, ενώ και τα τρία υπόκεινται σε περιορισμούς ACL.

3.2.4 Η ΔΙΑΔΙΚΑΣΙΑ ΠΡΟΣΘΗΚΗΣ (ADDITION PROCEDURE)

Στη διαδικασία προσθήκης, μία νέα εγγραφή προστίθεται στον διακομιστή. Σε αυτή την περίπτωση, ο πελάτης θα πρέπει να παράσχει ένα νέο και μοναδικό DN και ένα σύνολο χαρακτηριστικών με τις αντίστοιχες τιμές τους. Τα ζεύγη χαρακτηριστικών/τιμών πρέπει να περιλαμβάνουν μία λίστα κλάσεων αντικειμένων στις οποίες ανήκει η καταχώριση. Για παράδειγμα, εάν η καταχώριση πρόκειται να είναι ένας νέος χρήστης με αναγνωριστικό χρήστη (uid) και λογαριασμό, τότε η εγγραφή για τον χρήστη που θα προστεθεί μπορεί να μοιάζει με αυτό:

```
dn: uid=turpar, dc=marathonbc, dc=gr
```

```
cn=Turner Parker
```

```
uid: turpar
```

```
mail: turnerparker@gmail.com
```

```
objectClass: inetOrgPerson
```

```
objectClass: organizationalPerson
```

```
objectClass: person
```

3.2.5 Η ΔΙΑΔΙΚΑΣΙΑ ΤΡΟΠΟΠΟΙΗΣΗΣ (MODIFICATION PROCEDURE)

Η τροποποίηση ενεργεί σε μία συγκεκριμένη εγγραφή, που καθορίζεται από το DN. Οποιοσδήποτε αριθμός αλλαγών μπορεί να γίνει σε μία εγγραφή αρκεί να δημιουργηθεί ένα αίτημα τροποποίησης. Για μία συγκεκριμένη εγγραφή, μία λειτουργία τροποποίησης μπορεί να προσθέσει, να αντικαταστήσει ή να αφαιρέσει χαρακτηριστικά. Η τροποποίηση μπορεί να συνδυάσει λειτουργίες στο ίδιο αίτημα. Δηλαδή μπορεί να αφαιρέσει ένα χαρακτηριστικό και να αντικαταστήσει ένα άλλο χαρακτηριστικό σε ένα αίτημα. Ας δούμε αυτά τα χαρακτηριστικά. Ένα αίτημα προσθήκης παίρνει ένα όνομα χαρακτηριστικού και μία ή περισσότερες τιμές. Θα προστεθούν αυτές οι τιμές στο υπάρχων σύνολο τιμών για αυτό το χαρακτηριστικό. Έτσι λοιπόν για το προηγούμενο παράδειγμα θα έχουμε:

(Διαδικασία τροποποίησης που χρησιμοποιεί την διαδικασία της προσθήκης)

```
dn: cn=Turner Parker, dc=marathonbc, dc=gr
```

```
sn: Parker
```

ΚΕΦΑΛΑΙΟ 3

telephoneNumber: 2310458656

telephoneNumber: 6944553246

uid: turpar

mail: turnerparker@gmail.com

objectClass: person

(Διαδικασία τροποποίησης που χρησιμοποιεί την διαδικασία της τροποποίησης)

dn: cn=Turner Parker, dc=marathonbc, dc=gr

sn: Parker

telephoneNumber: 6944553247

uid: turpar

mail: turnerparker@gmail.com

Οι διαδικασίες τροποποίησης υποβάλλονται σε επεξεργασία με «όλα ή τίποτα» τρόπο. Όταν αποστέλλονται πολλές τροποποιήσεις σε ένα αίτημα, είτε όλες επιτυγχάνουν είτε όλες αποτυγχάνουν.

(Διαδικασία τροποποίησης που χρησιμοποιεί την διαδικασία της αντικατάστασης)

dn: cn=Turner Parker, dc=marathonbc, dc=gr

sn: Parker

telephoneNumber: 6978496581

uid: turpar

Ένα αίτημα αντικατάστασης λαμβάνει επίσης ένα χαρακτηριστικό και μία ή περισσότερες τιμές. Αλλά η λίστα τιμών αντικαθιστά τις υπάρχουσες τιμές. Για παράδειγμα, εάν ο Turner άλλαξε αριθμός τηλεφώνου, τότε η αντικατάσταση με το νέο χαρακτηριστικό τηλεφώνου (*telephoneNumber*) θα είναι η συμβολοσειρά 6978496581. Ο νέος αριθμός προστίθεται και οι παλιοί αριθμοί καταργούνται.

(Διαδικασία τροποποίησης που χρησιμοποιεί την διαδικασία της διαγραφής)

dn: cn=Turner Parker, dc=marathonbc, dc=gr

sn: Parker

telephoneNumber: 6944553246

uid: turpar

mail: turnerparker@gmail.com

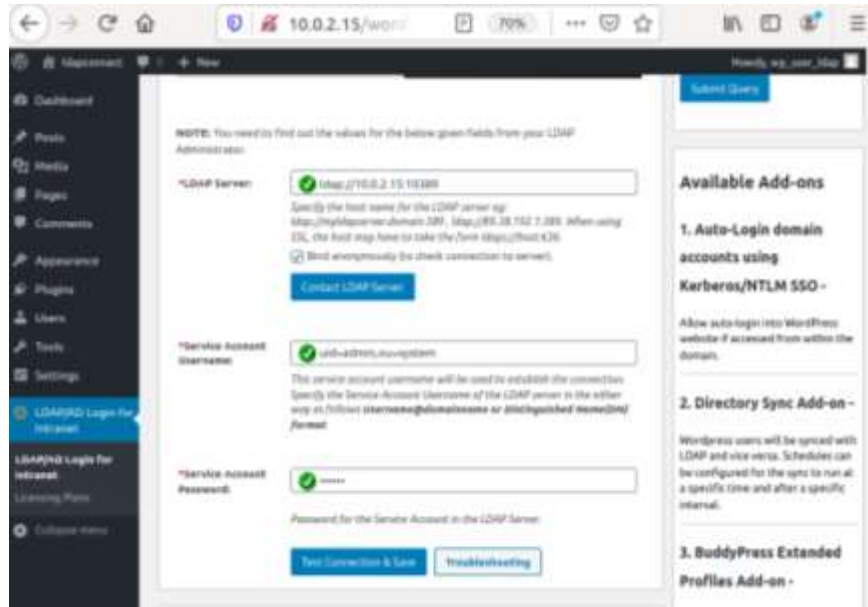
Ένα αίτημα διαγραφής λαμβάνει επίσης ένα χαρακτηριστικό και μία λίστα τιμών, Διαγράφει μόνο τις τιμές για ένα χαρακτηριστικό που καθορίζονται στη λίστα τιμών. Όπως βλέπουμε στο παραπάνω παράδειγμα, καταργήθηκε μόνο το αντίστοιχο sn. Εάν ωστόσο, μόνο ένα αίτημα διαγραφής καθορίζει το χαρακτηριστικό, τότε όλες οι εμφανίσεις αυτού το χαρακτηριστικού θα καταργηθούν.

3.2.6 Η ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΓΡΑΦΗΣ (DELETE PROCEDURE)

Τέλος, μία ολόκληρη εγγραφή LDAP μπορεί να διαγραφεί. Όπως οι τροποποιήσεις, η διαγραφή λειτουργεί σε μία συγκεκριμένη εγγραφή, στο DN της εγγραφής. Κατά τη διάρκεια μιας λειτουργίας διαγραφής, ολόκληρη η εγγραφή αφαιρείται από τον κατάλογο (το DN και όλα τα χαρακτηριστικά. Μόνο εγγραφές που δεν έχουν «παιδιά» μπορούν να διαγραφούν από τον κατάλογο. Εάν μία καταχώριση έχει «παιδιά» πρέπει να αφαιρεθούν από τον κατάλογο (ή να μετεγκατασταθεί σε άλλο τμήμα του δέντρου) προτού καταργηθεί η γονική καταχώριση.

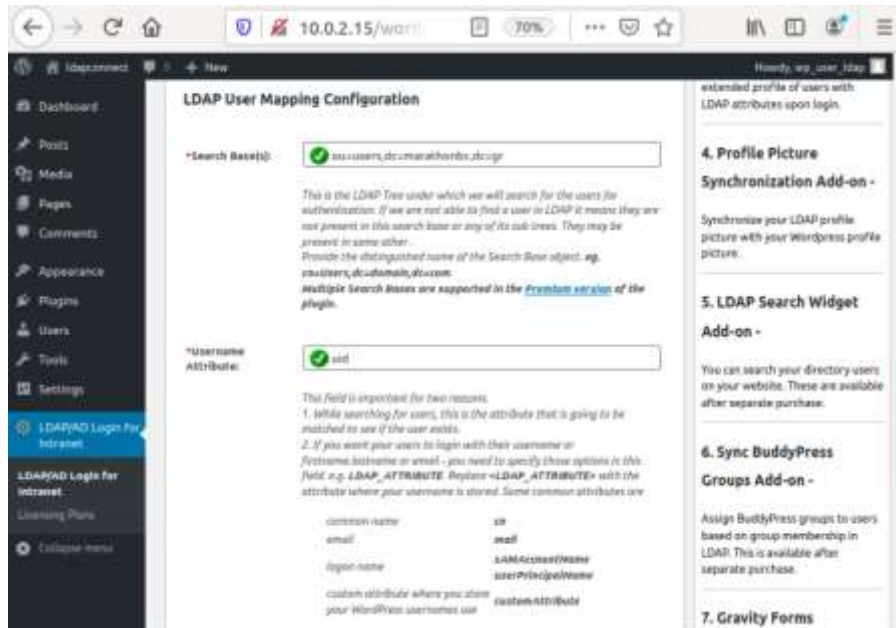
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ LDAP

και ούτω καθεξής μέσω πρόσθετων λειτουργιών. Οι πληροφορίες χρήστη είναι συγχρονισμένες με τις πληροφορίες στο LDAP.



Εικόνα 23 - Σύνδεση LDAP - WordPress με Πρόσθετο (Μέρος Α)

- LDAP Server: Σε αυτό το πεδίο καθορίζεται το όνομα του κεντρικού υπολογιστή για το διακομιστή LDAP. Π.χ. ldaps://myldapserver.domain:389 ή ldaps://89.38.192.1:389. Αξίζει να τονιστεί, ότι όταν χρησιμοποιείται το πρωτόκολλο Secure Sockets Layer, θα πρέπει να είναι της μορφής ldaps://89.38.192.1:389.
- Service Account Username: Αυτό το όνομα χρήστη θα χρησιμοποιηθεί για τη δημιουργία της σύνδεσης (binding). Το όνομα χρήστη του διακομιστή LDAP μπορεί να καθοριστεί είτε ως Username@domainname είτε ως DN (Distinguished Name).
- Service Account Password: Ο κωδικός του χρήστη που χρησιμοποιήθηκε για τη σύνδεση (binding) με το διακομιστή LDAP.



Εικόνα 24 - Σύνδεση LDAP - WordPress με Πρόσθετο (Μέρος Β)

- Search Bases(s): Αυτό είναι το δέντρο LDAP κάτω από το οποίο θα αναζητηθούν οι χρήστες για έλεγχο ταυτότητας. Εάν δεν βρεθεί κάποιος χρήστης σε αυτό το δέντρο LDAP, αυτό σημαίνει ότι ο συγκεκριμένος χρήστης δεν είναι παρών σε αυτή τη βάση αναζήτησης. Μπορεί όμως να βρίσκεται σε κάποιο άλλο δευτερεύον δέντρο LDAP. Έτσι λοιπόν η τιμή του συγκεκριμένου πεδίου είναι το διακεκριμένο όνομα (Distinguished Name) της βάσης αναζήτησης.
- Username Attribute: Το συγκεκριμένο πεδίο είναι πολύ σημαντικό για δύο λόγους:
 - a. Κατά τη διάρκεια της αναζήτησης των χρηστών, αποτελεί το χαρακτηριστικό με το οποίο θα γίνει η ταυτοποίηση του χρήστη.
 - b. Εάν ο υποψήφιος προγραμματιστής, επιθυμεί οι πελάτες να πραγματοποιούν σύνδεση με το κατάλογο LDAP, χρησιμοποιώντας το username τους, ή το firstname, ή το lastname, ή το email τους, θα πρέπει να «δηλώσει» στο συγκεκριμένο πεδίο, το όνομα του χαρακτηριστικού με το οποίο θα γίνει ο έλεγχος ταυτότητας

Test Authentication

WordPress username is mapped to the LDAP attribute defined in the Search Filter attribute in LDAP. Ensure that you have an administrator user in LDAP with the same attribute value.

*Username:

*Password:

[Test Authentication](#) [Troubleshooting](#)

Thank you for creating with [WordPress](#)

Version 5.3.2

9. Memberpress Integration -

Login to Memberpress protected content with LDAP Credentials. This is available after separate purchase.

10. Ultimate Member Integration -

Login to Ultimate Member with LDAP Credentials. This is available after separate purchase.

Εικόνα 25 - Σύνδεση LDAP -WordPress με Πρόσθετο (Μέρος Γ)

Εφόσον, τα δύο παραπάνω βήματα έχουν πραγματοποιηθεί με επιτυχία, το μόνο που απομένει, είναι η δοκιμή για τον έλεγχο ταυτότητας ενός χρήστη, εφόσον αυτός ανήκει στον κατάλογο LDAP και τα διαπιστευτήρια που έχει δώσει είναι σωστά (username και password). Διαφορετικά δεν μπορεί να πραγματοποιηθεί σύνδεση.

3.4 ΣΥΜΠΕΡΑΣΜΑ

Σε αυτό το κεφάλαιο, εξετάστηκαν ορισμένες από τις λειτουργίες που διαθέτει ένας κατάλογος LDAP στους χρήστες – πελάτες τους. Οι πιο κοινές λειτουργίες είναι η δέσμευση, η αναζήτηση, η τροποποίηση, η προσθήκη και η διαγραφή. Επίσης αναλύθηκαν οι δύο βασικοί τρόποι αλληλεπίδρασης μεταξύ ενός LDAP καταλόγου και μίας ιστοσελίδας που έχει κατασκευαστεί με τη χρήση του λογισμικού WordPress.

ΚΕΦΑΛΑΙΟ 4

4 REST API ΤΟΥ WORDPRESS

4.1 REPRESENTATIVE STATE TRANSFER (REST)

Αφού εγκαταστάθηκε το Android Studio, ήρθε η ώρα να αναλυθεί η τεχνολογία με την οποία θα αλληλεπιδρά το mobile application με την ιστοσελίδα. Το 2000, μετά την αποτροπή της κρίσης επεκτασιμότητας του Ιστού, ο Fielding ονόμασε και περιέγραψε το αρχιτεκτονικό στυλ του Ιστού στη διδακτορική του διατριβή. Το «Representative State Transfer» (REST) είναι το όνομα που έδωσε ο Fielding στην περιγραφή του για το αρχιτεκτονικό στυλ του Web.

4.2 REST APIs

Η υπηρεσίες Διαδικτύου είναι διαδικτυακοί διακομιστές κατασκευασμένοι έτσι ώστε να υποστηρίζουν τις ανάγκες μίας ιστοσελίδας ή οποιαδήποτε άλλη διαδικτυακή εφαρμογή. Τα προγράμματα ή οι εφαρμογές των προγραμματιστών χρησιμοποιούν εφαρμογές διασύνδεσης (API) για επικοινωνία με υπηρεσίες Ιστού. Γενικά, ένα API «εκθέτει» ένα σύνολο δεδομένων και λειτουργιών για να διευκολύνει τις αλληλεπιδράσεις μεταξύ προγραμμάτων ή εφαρμογών και τους επιτρέπει να ανταλλάσσουν πληροφορίες. Η παρακάτω εικόνα συμβάλει στην κατανόηση ενός API, όσον αφορά την άμεση ακρόαση και απάντηση σε αιτήματα πελατών.

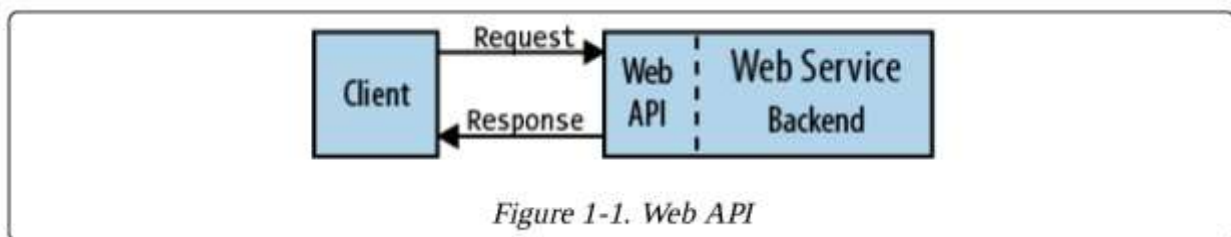


Figure 1-1. Web API

Εικόνα 26 - Αίτημα - Απάντηση - Rest API

Το αρχιτεκτονικό στυλ των REST εφαρμόζεται συνήθως στο σχεδιασμό των API για σύγχρονες διαδικτυακές υπηρεσίες. Ένα Web API “συμμορφώνεται» με το αρχιτεκτονικό στυλ των REST και σχηματίζουν ένα REST API. Η κατοχή ενός REST API δημιουργεί μία διαδικτυακή υπηρεσία που ονομάζεται RESTful. Ένα REST API αποτελείται από ένα συγκρότημα διασυνδεδεμένων πόρων. Αυτό το σύνολο πόρων είναι γνωστό ως οι πόροι του μοντέλου REST API. Φυσικά, για να υλοποιηθούν όλα τα παραπάνω, απαιτείται προσεκτικός σχεδιασμός του REST API.

4.3 WORDPRESS REST API

Όπως προαναφέρθηκε, το API είναι μία διεπαφή προγραμματισμού εφαρμογών. Το REST που αντιπροσωπεύει το «Representative State Transfer», η οποία είναι ένα σύνολο εννοιών για μοντελοποίηση και πρόσβαση στα δεδομένα της εφαρμογής ως αλληλένδετα αντικείμενα. Το WordPress Rest API παρέχει endpoints (URLs) που μπορεί να αντιπροσωπεύουν τις αναρτήσεις μια ιστοσελίδας, το περιεχόμενο των στατικών σελίδων κ.α. Η αίτηση που στέλνει ο πιθανός προγραμματιστής, θα επιστρέψει ως απάντηση δεδομένα JSON. Το JSON είναι μια τυπική μορφή δεδομένων που είναι εύκολα αναγνώσιμη από τον άνθρωπο και μοιάζει με τα αντικείμενα σε JavaScript. Όταν ένας χρήστης ζητήσει περιεχόμενο ή αποστείλει περιεχόμενο στο Rest Api του WordPress, η απάντηση θα επιστρέφεται σε JSON. Αυτό συμβαίνει διότι το JSON υποστηρίζεται ευρέως σε πολλές γλώσσες προγραμματισμού. Το γεγονός αυτό βοηθάει τους προγραμματιστές να μπορούν να δημιουργήσουν εφαρμογές WordPress σε JavaScript.

4.4 ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΟ WORDPRESS REST AP

Το WordPress από μόνο του παρέχει ήδη ένα πλούσιο σύνολο εργαλείων και διεπαφών για την κατασκευή ιστότοπων. Εάν ωστόσο, κάποιος προγραμματιστής, επιθυμεί να πραγματοποιήσει μία εξωτερική εφαρμογή, είτε αυτή είναι μία εφαρμογή σε JavaScript, είτε ένα αυτόνομο πρόγραμμα σε μία διαφορετική γλώσσα από την PHP, τότε η εφαρμογή θα χρειαστεί ένα δομημένο τρόπο για να αποκτήσει πρόσβαση στο περιεχόμενο της ιστοσελίδας του WordPress. Οποιαδήποτε γλώσσα προγραμματισμού μπορεί να στείλει αιτήματα HTTP και να λάβει απαντήσεις με την μορφή JSON.

4.5 ΠΡΟΣΘΗΚΗ CUSTOM ENDPOINTS

Το WordPress REST API είναι κάτι παραπάνω από ένα σύνολο προεπιλεγμένων διαδρομών. Αποτελεί εργαλείο για τη δημιουργία προσαρμοσμένων διαδρομών και τελικών σημείων. Ας δούμε ένα σύντομο παράδειγμα με την δημιουργία μίας μεθόδου. Θα δημιουργήσουμε μία μέθοδο η οποία θα επιστρέφει το τελευταίο post (ανάρτηση) , με βάση το id του author (συγγραφέα).

```

1  function my_awesome_func( $data ) {
2      $posts = get_posts( array(
3          'author' => $data['id'],
4      ) );
5
6      if ( empty( $posts ) ) {
7          return null;
8      }
9
10     return $posts[0]->post_title;
11 }
12

```

Εικόνα 27 - Παράδειγμα μεθόδου για το Rest API

- Ονομασία μεθόδου: my_awesome_func
- Παράμετρος μεθόδου: η μεταβλητή \$data
- Ανάλυση της μεθόδου: Αρχικά, δημιουργείτε η μεταβλητή \$posts η οποία θα λάβει όλα τα posts , τις αναρτήσεις, με βάση το id του συγγραφέα που δόθηκε ως παράμετρος στο προηγούμενο βήμα. Εάν δεν υπάρχει κάποια ανάρτηση τότε επιστρέφεται null, ενώ στην αντίθετη περίπτωση επιστρέφεται το τελευταίο post (η τελευταία ανάρτηση).

Έχοντας πλέον δημιουργήσει τη μέθοδο που θέλουμε, αυτή η μέθοδος πρέπει να «γραφτεί» κάπου εάν ο προγραμματιστής θέλει να τη χρησιμοποιήσει σε κάποια εξωτερική εφαρμογή. Οπότε για να γίνει διαθέσιμη η μέθοδος μέσω του WordPress REST API, πρέπει να καταχωρηθεί μία διαδρομή. Η διαδρομή αυτή «αναφέρει» στο API να ανταποκριθεί σε ένα δεδομένο αίτημα σχετικά με μια λειτουργία. Αυτή η λειτουργία ονομάζεται register_rest_route και ουσιαστικά και καλείται σε μία επιστροφή κλήσης στο rest_api_init. Οπότε, για να δημιουργηθεί η διαδρομή απαιτούνται τρία δεδομένα

- Ονομασία
- Διαδρομή
- Παράμετρος

Με βάση λοιπόν αυτά, η δημιουργία διαδρομής για αξιοποίηση της μεθόδου my_awesome_func μοιάζει κάπως έτσι:

```

add_action('rest_api_init', function(){
    register_rest_route('yourname/v1', '/author/(?P<id>\d+)', array(
        'methos' => 'GET',
        'callback' => 'my_awesome_func',
    ));
});

```

Εικόνα 28 - "Εγγραφή" μεθόδου στο WordPress REST AP"

Όπως βλέπουμε, η διαδρομή που καταχωρήθηκε η καινούργια μέθοδος που δημιουργήθηκε στο προηγούμενο βήμα είναι: [http://domain.com/wp-json/yourname/v1/author/\(?P\d+\)](http://domain.com/wp-json/yourname/v1/author/(?P\d+)). Αξίζει να τονιστεί, ότι προκειμένου η συγκεκριμένη διαδρομή να επιστρέψει τα αποτελέσματα που επιθυμεί ο χρήστης, θα πρέπει να δοθεί σαν όρισμα το id του συγγραφέα.

4.6 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ANDROID ME TO REST API TOY WORDPRESS

Εφόσον έχουν ολοκληρωθεί και κατανοηθεί πλήρως τα προηγούμενα βήματα, σειρά έχει η δημιουργία μίας μεθόδου που θα επιστρέφει κάποια JSON δεδομένα από το REST API του WordPress , στο mobile application που δημιουργήθηκε.

Αρχικά με παρόμοιο τρόπο, όπως στην ενότητα 4.7, θα δημιουργηθούν 2 (δύο) μέθοδοι για την αποθήκευση κάθε post (ανάρτησης) μεμονωμένα αλλά και όλες τις αναρτήσεις μαζί.

Resource Methods	URL	Description
GET, POST, PUT	https://marathonbc.gr/wp-json/wp/v2/posts	Επιστρέφει τα στοιχεία της κάθε δημοσίευσης
GET, POST, PUT	https://marathonbc.gr/wp-json/wp/v2/posts?fields=id,title,content	Επιστρέφει μόνο το id, τον τίτλο και το περιεχόμενο της κάθε δημοσίευσης
GET, POST, PUT	https://marathonbc.gr/wp-json/wp/v2/pages?slug=paidiko	Επιστρέφει τα στοιχεία της σελίδας με slug value = <u>paidiko</u>

Εικόνα 29 - Custom Routes του WordPress Rest API

Παρακάτω, περιγράφεται αναλυτικά ο τρόπος με τον οποίο δημιουργείται η μέθοδος:

```
function wl_post( $slug ) {
    $args = [
        'name' => $slug['slug'],
        'post_type' => 'post'
    ];

    $post = get_posts($args);

    $data['id'] = $post[0]->ID;
    $data['title'] = $post[0]->post_title;
    $data['content'] = $post[0]->post_content;
    $data['slug'] = $post[0]->post_name;
    $data['featured_image']['thumbnail'] = get_the_post_thumbnail_url($post[0]
        ->ID, 'thumbnail');
    $data['featured_image']['medium'] = get_the_post_thumbnail_url($post[0]->ID,
        'medium');
    $data['featured_image']['large'] = get_the_post_thumbnail_url($post[0]->ID,
        'large');
    $data['featured_image']['full'] = get_the_post_thumbnail_url($post[0]->ID,
        'full');

    return $data;
}
```

Εικόνα 30 - Android Development και WordPress REST API (Μέρος Α)

Το όνομα της μεθόδου είναι `wl_post`. Ως παράμετρο, δέχεται τη μεταβλητή `$slug`. Το `slug` αποτελεί το μέρος μίας διεύθυνσης URL που προσδιορίζει μία συγκεκριμένη σελίδα (στην περίπτωση μας, μία συγκεκριμένη ανάρτηση) σε έναν ιστότοπο σε μια ευανάγνωστη μορφή. Στη συνέχεια δημιουργείται η μεταβλητή `$args`. Η μεταβλητή `$args` είναι ένα array στο οποίο αποθηκεύονται οι τιμή του `$slug` καθώς και το είδος της ανάρτησης. Όπως αναφέρθηκε, η ανάρτηση μπορεί να είναι κάποιο άρθρο ή μία νέα σελίδα. Στην δική μας περίπτωση είναι άρθρο. Κατόπιν, δημιουργείται επίσης η μεταβλητή `$post` που ισούται με τη μέθοδο `get_posts`. Η μέθοδος `get_posts`, γενικά, δέχεται σαν όρισμα, ένα array με χαρακτηριστικά που αφορούν μία δημοσίευση. Ωστόσο, αν δεν καθοριστεί σε κάποια μεταβλητή, η συγκεκριμένη μέθοδος θα επιστρέψει όλα τα χαρακτηριστικά μίας δημοσίευσης. Ο υποψήφιος προγραμματιστής θα πρέπει να καθορίσει μέσω μίας επιπλέον μεταβλητής τα χαρακτηριστικά που επιθυμεί να επιστραφούν. Το ρόλο αυτό στην προκειμένη περίπτωση παίζει η μεταβλητή `$data`, στην οποία αποθηκεύονται μερικά από τα χαρακτηριστικά ενός άρθρου, όπως για παράδειγμα το ID, ο τίτλος, το περιεχόμενο κ.λπ.

Έχοντας ολοκληρώσει την παραπάνω μέθοδο, στόχος είναι οι συγκεκριμένες δημοσιεύσεις, να μην προβάλλονται μόνο μεμονωμένα, αλλά σε μία μορφή λίστας. Παρακάτω παρουσιάζεται ο κώδικας στον οποίο αποθηκεύονται οι παραπάνω δημοσιεύσεις.

```
function wl_posts() {
    $args = [
        'numberposts' => 10,
        'post_type' => 'post',
    ];

    $posts = get_posts($args);

    $data = [];
    $i = 0;

    foreach($posts as $post) {
        $data[$i]['id'] = $post->ID;
        $data[$i]['title'] = $post->post_title;
        $data[$i]['content'] = $post->post_content;
        $data[$i]['slug'] = $post->post_name;
        $data[$i]['full'] = get_the_post_thumbnail_url($post->ID, 'large');
        $data[$i]['date'] = get_the_date('d/m/Y', $post->ID);
        $data[$i]['image'] = $post -> post_image;
        $data[$i]['post_content'] = $post -> post_content;
        $data[$i]['trim-post_content'] = $post -> post_excerpt;

        $i++;
    }

    return $data;
}
```

Εικόνα 31 - Android Development και WordPress REST API (Μέρος Β)

Η φιλοσοφία της παραπάνω μεθόδου, όπως φαίνεται και από την εικόνα, είναι παρόμοια με την μέθοδο που αναπτύχθηκε στο πρώτο μέρος. Ωστόσο υπάρχουν διαφορές που αξίζει να αναφερθούν.

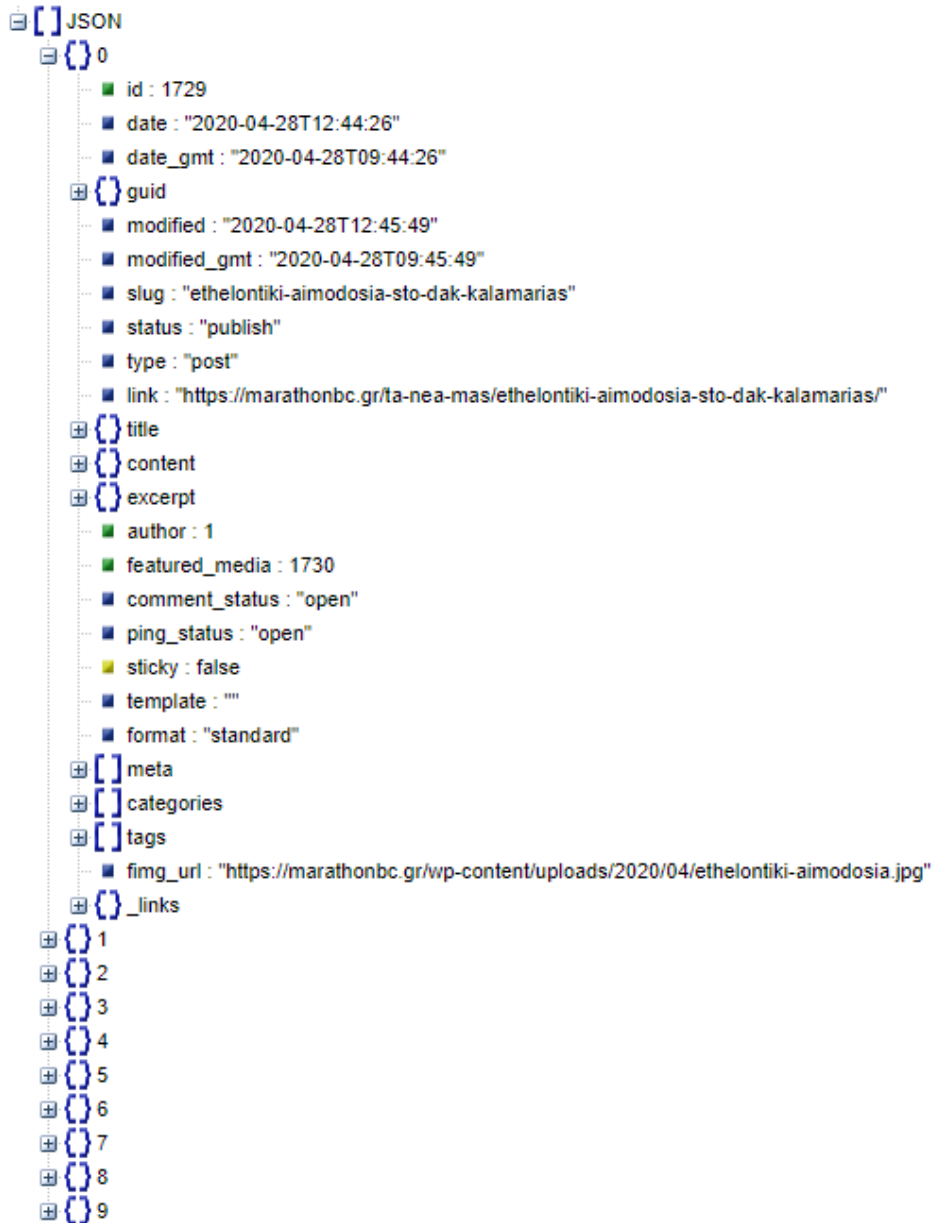
Παρατηρείται ότι πάλι αρχικοποιείται η μεταβλητή `$args`. Ενώ πάλι, η μεταβλητή αποτελεί ένα `array`, σε αυτή την περίπτωση τα χαρακτηριστικά στοιχεία διαφέρουν. Και αυτό διότι, όπως τονίστηκε, το επιστρεφόμενο αποτέλεσμα, δεν θα είναι ένα μόνο `post` (δημοσίευση) αλλά σύμφωνα με τη μεταβλητή `'numberposts' => 10`, θα είναι 10 (δέκα) συνολικά. Έχουμε πάλι την εμφάνιση της μεθόδου `get_posts` και ακολουθεί ένας βρόχος που αναλαμβάνει να επιστρέψει στη μεταβλητή `$data` τα χαρακτηριστικά της δημοσίευσης που επιθυμεί να προβάλλει ο χρήστης. Ωστόσο, όπως και πριν, η υλοποίηση των παραπάνω μεθόδων δεν αρκεί, ώστε οι μέθοδοι να χρησιμοποιηθούν σε κάποια εξωτερική εφαρμογή μέσω του WordPress REST API. Θα πρέπει ο υποψήφιος προγραμματιστής, να καταχωρίσει μία διαδρομή, ώστε οι μέθοδοι που θα χρησιμοποιηθούν να είναι διαθέσιμοι μέσω του WordPress REST API. Στην παρακάτω εικόνα, παρουσιάζεται ο τρόπος με τον οποίο πραγματοποιείται η σύνδεση.

```
add_action('rest_api_init', function() {
    register_rest_route('wl/v1', 'posts', [
        'methods' => 'POST',
        'callback' => 'wl_posts',
    ]);

    register_rest_route( 'wl/v1', 'posts/(?P<slug>[a-zA-Z0-9-]+)', array(
        'methods' => 'POST',
        'callback' => 'wl_post',
    ) );
});
```

Εικόνα 32 - Android Development και WordPress REST API (Μέρος Γ)

Σε σύγκριση με το αρχικό παράδειγμα, παρατηρείται ότι η διαδρομή που προκύπτει από τον παραπάνω κώδικα είναι η: <https://marathonbc.gr/wp-json/wl/v1/posts>. Όμως η διαδρομή αυτή επιστρέφει ως αποτέλεσμα τη λίστα με τις 10 (δέκα) τελευταίες δημοσιεύσεις. Εάν, ο υποψήφιος προγραμματιστής επιθυμεί να, χρησιμοποιήσει κάποια συγκεκριμένα χαρακτηριστικά των δημοσιεύσεων, όπως για παράδειγμα το `id`, ο τίτλος και το περιεχόμενο της δημοσίευσης, τότε η διαδρομή που πρέπει να πληκτρολογήσει είναι η: <https://marathonbc.gr/wp-json/wp/v2/posts?fields=id,title,content>. Όπως και οι υπόλοιπες διαδρομές στο WordPress REST API, έτσι κι αυτή, θα επιστρέψει το αποτέλεσμα σε μορφή JSON.



Εικόνα 33 - Αποτέλεσμα μεθόδου σε JSON

Βλέπουμε λοιπόν, ότι στο επιστρεφόμενο αποτέλεσμα περιλαμβάνονται τα χαρακτηριστικά των τελευταίων 10 (δέκα) άρθρων. Είναι στο χέρι του προγραμματιστή ποια χαρακτηριστικά των άρθρων θέλει να συμπεριλάβει στην εφαρμογή του. Εάν επιθυμεί να συμπεριλάβει άλλα χαρακτηριστικά και όχι τα προηγούμενα που συμπεριλάμβανε η προηγούμενη αναζήτηση, τότε θα πρέπει να αλλάξει τα ορίσματα στη διαδρομή στο πεδίο `posts?fields`. Ακολουθώντας, τα παραπάνω βήματα ολοκληρώνεται η διαδικασία υλοποίησης μεθόδων και καταγραφής τους στο WordPress REST API. Το μόνο που απομένει είναι η αξιοποίηση των δεδομένων JSON.

4.7 ΣΥΜΠΕΡΑΣΜΑ

Σε αυτό το κεφάλαιο, αναφέρθηκαν αναλυτικά τα βήματα εγκατάστασης του Android Studio σε κάποιον υπολογιστή. Έπειτα, αναλύθηκαν οι έννοιες των APIs και ο ρόλος που παίζουν στη μεταξύ τους διάδραση με κάποια εξωτερική εφαρμογή. Κατόπιν, μελετήθηκαν σε βάθος οι όροι που αφορούν το WordPress REST API όπως είναι τα Custom Endpoints και τα Customs Routes. Τέλος, παρέχονται αναλυτικά βήματα για το πως ένας προγραμματιστής μπορεί να δημιουργήσει τις δικές του μεθόδους και να τα εκχωρήσει στο WordPress REST API, αξιοποιώντας τα δεδομένα σε μία android εφαρμογή.

ΚΕΦΑΛΑΙΟ 5

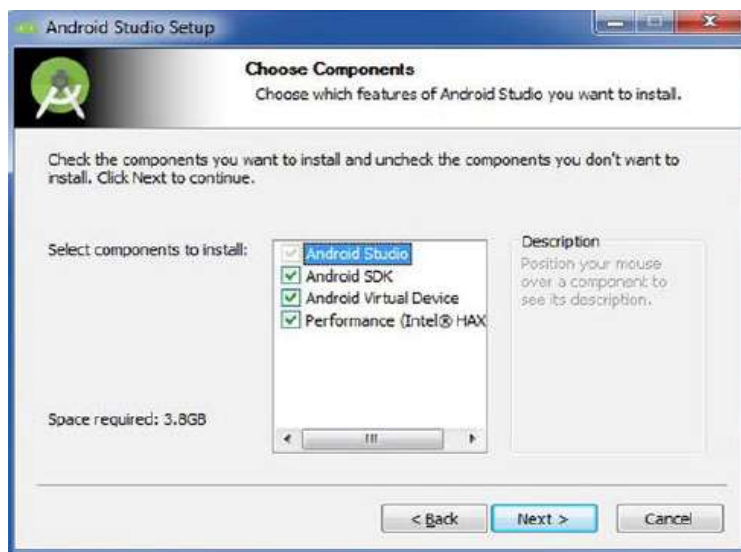
5 ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ ANDROID

5.1 ANDROID STUDIO

Το πρόγραμμα που χρησιμοποιήθηκε για την δημιουργία του mobile application είναι το Android Studio. Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για ανάπτυξη εφαρμογών Android. Βασίζεται στο IntelliJ IDEA, ένα ολοκληρωμένο περιβάλλον ανάπτυξης Java για λογισμικό και ενσωματώνει τα εργαλεία επεξεργασίας κώδικα και προγραμματιστών.

5.2 ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ ANDROID STUDIO

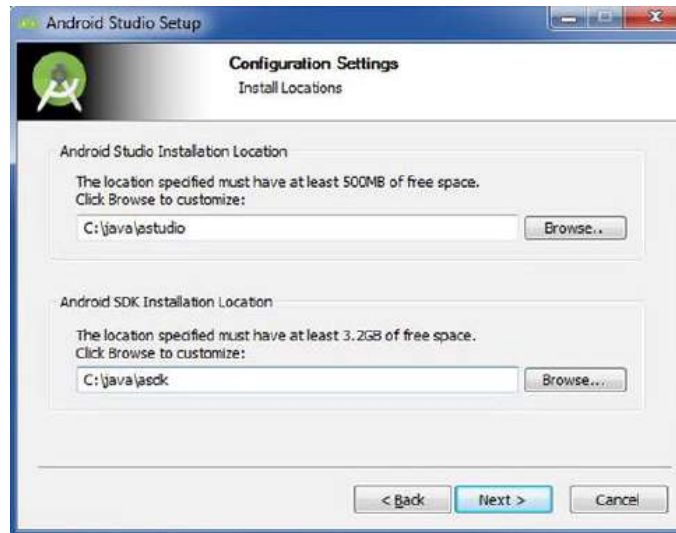
Πριν ξεκινήσετε τη λήψη του Android Studio, απαιτείται η δημιουργία ενός βασικού φακέλου στον οποίο θα αποθηκεύονται τα πρότζεκτ. Ένα παράδειγμα, είναι C:\AndroidApps. Η λήψη του Android Studio είναι απλή. Μπορεί οποιοσδήποτε να κατεβάσει δωρεάν το Android Studio από το σύνδεσμο developer.android.com/sdk/installing/studio.html. Αφού ξεκινήσει ο Οδηγός εγκατάστασης, μετακινηθείτε στις οθόνες του κάνοντας κλικ στα κουμπιά Επόμενο μέχρι να φτάσετε στην οθόνη Επιλογή στοιχείων. Εκεί, θα πρέπει να επιλέξετε όλα τα τετραγωνίδια ελέγχου που εμφανίζονται στην παρακάτω εικόνα.



Εικόνα 34 - Εγκατάσταση Android Studio (Μέρος Α)

ΚΕΦΑΛΑΙΟ 5

Στην συνέχεια, αφού αποδεχτείτε για άλλη μια φορά τους όρους και τις προϋποθέσεις, θα πρέπει να καθορίσετε τις τοποθεσίες για το Android Studio και το Android SDK.



Εικόνα 35 - Εγκατάσταση Android Studio (Μέρος Β)

Η εγκατάσταση του Android Studio μπορεί να διαρκέσει λίγα λεπτά, ανάλογα με την ταχύτητα του υπολογιστή σας. Όταν ολοκληρωθεί, θα σας εμφανιστεί η εικόνα ολοκλήρωσης εγκατάστασης. Κάντε κλικ στο κουμπί τέλος και είστε έτοιμοι να ξεκινήσετε το πρότζεκτ σας.



Εικόνα 36 - Εγκατάσταση Android Studio (Μέρος Γ)

5.3 ΑΝΑΛΥΣΗ ΕΝΟΣ PROJECT ΣΕ ANDROID

Με την εγκατάσταση του Android, ο υποψήφιος προγραμματιστής έχει τη δυνατότητα να υλοποιήσει το project του. Για να συμβεί αυτό, θα πρέπει να δημιουργηθούν τις αντίστοιχες Activities. Μία Activity κλάση αποτελεί ίσως το πιο κρίσιμο συστατικό μίας εφαρμογής Android καθώς αποτελούν τη μέθοδο με την οποία θα υλοποιηθούν οι λειτουργίες που επιθυμεί ο εκάστοτε προγραμματιστής για την εφαρμογή του. Σε αντίθεση με άλλα παραδείγματα προγραμματισμού, όσον αφορά το Android, η εφαρμογή δεν ξεκινά με μία κύρια μέθοδο. Το σύστημα Android ξεκινά τον κώδικα με μία συγκεκριμένη Activity που έχει οριστεί από τον προγραμματιστή, η οποία καλεί συγκεκριμένες μεθόδους. Μερικές από αυτές τις μεθόδους είναι οι εξής:

- onCreate ()

Με τη συγκεκριμένη μέθοδο, το σύστημα «πυροδοτεί» τη δημιουργία της συγκεκριμένης δραστηριότητας.

- onStart ()

Με τη συγκεκριμένη μέθοδο, η δραστηριότητα εισέρχεται σε κατάσταση έναρξης και γίνεται «ορατή» από το χρήστη.

- onStop ()

Με τη συγκεκριμένη μέθοδο, η δραστηριότητα δεν είναι πλέον ορατή από το χρήστη. Αυτό μπορεί να συμβεί είτε επειδή η δραστηριότητα «καταστρέφεται» είτε επειδή ξεκινά μία νέα.

- onRestart ()

Με τη συγκεκριμένη μέθοδο, επικαλείται η επανάκληση μίας συγκεκριμένης δραστηριότητας που είχε διακοπεί.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.sections_details);
}
```

Εικόνα 37 - Παράδειγμα Μεθόδου onCreate

Όπως φαίνεται από την παραπάνω εικόνα, με τη δημιουργία της δραστηριότητας καλείται η μέθοδος onCreate () για να υλοποιηθεί η λειτουργία που προαναφέρθηκε. Ωστόσο, αξίζει να σημειωθεί, ότι για να εμφανιστεί η οποιαδήποτε δραστηριότητα – λειτουργία στο χρήστη της εφαρμογής θα πρέπει

να δημιουργηθούν τα αντίστοιχα layouts. Το layout καθορίζει τη δομή για μία διεπαφή χρήστη σε μία εφαρμογή, όπως σε μία δραστηριότητα. Όλα τα στοιχεία ενός layout δημιουργούνται χρησιμοποιώντας μία ιεραρχία των αντικειμένων View και ViewGroup. Στην εικόνα 37, χρησιμοποιείται η μέθοδος setContentView, η οποία «παρουσιάζει» την συγκεκριμένη δραστηριότητα μέσω του XML αρχείου sections_details. Χρησιμοποιώντας XML του Android, ο υποψήφιος προγραμματιστής μπορεί γρήγορα να σχεδιάσει layouts με παρόμοιο τρόπο όπως θα δημιουργούσε ιστοσελίδες σε HTML. Με βάση τα παραπάνω, η λογική προκειμένου να ολοκληρωθεί μία εφαρμογή ή απλούστερα, η λειτουργία μίας εφαρμογής, βασίζεται στην ολοκλήρωση μίας ή περισσότερων δραστηριοτήτων (Activities) και φυσικά η δημιουργία των αντίστοιχων layouts που θα αποτελέσουν την «εικόνα» με την οποία θα αλληλεπιδρά ο χρήστης.

```
public class Splash extends AppCompatActivity {

    private TextView splashText;
    private ImageView splashImage;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        splashText = (TextView) findViewById(R.id.splashTextView);
        splashImage = (ImageView) findViewById(R.id.splashImageView);
        Animation myanim = AnimationUtils.loadAnimation(context: this, R.anim.mytransition);
        splashText.startAnimation(myanim);
        splashImage.startAnimation(myanim);
        final Intent splashIntent = new Intent(packageContext: this, MainActivity.class);
        Thread timer = run() -> {
            try{
                sleep(millis: 3000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            finally {
                startActivity(splashIntent);
                finish();
            }
        };
        timer.start();
    }
}
```

Εικόνα 38 - Παράδειγμα Activity

Η παραπάνω δραστηριότητα «εμφανίζεται» κάθε φορά με την εκκίνηση της εφαρμογής για τρία (3) δευτερόλεπτα και μετά τερματίζεται. Τα στοιχεία που «δηλώνονται» στη δραστηριότητα είναι ένα TextView (κείμενο) και ένα ImageView (εικόνα). Αποτελεί απαραίτητη προϋπόθεση να δηλωθούν τα συγκεκριμένα στοιχεία, εάν ο υποψήφιος προγραμματιστής επιθυμεί να τα «προβάλει» στο αντίστοιχο layout της δραστηριότητας.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/activity_splash"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#fff9b8">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/splashImageView"
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:layout_gravity="center"
            android:src="@drawable/marathon_bc_logo" />

        <TextView
            android:id="@+id/splashTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="10dp"
            android:text="WE OWN THE PRESENT"
            android:textColor="@color/colorPrimaryDark"
            android:textSize="23sp"
            android:textStyle="bold" />

    </LinearLayout>

</RelativeLayout>

```

Εικόνα 39 - Παράδειγμα Layout

Παρατηρείται λοιπόν ότι τα ImageView και TextView χρησιμοποιούνται στο συγκεκριμένο layout με ξεχωριστό id. Η διαδικασία με τις δραστηριότητες και τα layouts επαναλαμβάνεται μέχρι ο προγραμματιστής να υλοποιήσει την εφαρμογή που επιθυμεί.

5.4 ΔΕΔΟΜΕΝΑ JSON ΑΠΟ URL ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ANDROID

Έχοντας πλέον τα δεδομένα που επιθυμεί ο πιθανός προγραμματιστής σε μορφή JSON, μπορεί να τα αξιοποιήσει σε οποιαδήποτε γλώσσα προγραμματισμού. Στη δική μας περίπτωση η διαδικασία θα γίνει μέσω του Android Studio, χρησιμοποιώντας τη γλώσσα προγραμματισμού της JAVA. Παρακάτω παρουσιάζεται ο μέθοδος με την οποία αποστέλλεται ένα HTTP αίτημα προκειμένου να λάβει τα συγκεκριμένα χαρακτηριστικά που επιθυμεί ο προγραμματιστής.

```

public void sendRequestForPosts() {
    JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(Method.POST, request_url_for_posts_list, jsonObject, null, new Response.Listener<JsonObject>() {
        @Override
        public void onResponse(JSONObject response) {
            progressBar.setVisibility(View.GONE);
            recyclerView.setVisibility(View.VISIBLE);

            for (int i = 0; i < response.length(); i++) {
                PostInList postInList = new PostInList();

                try {
                    jsonObject = response.getJSONObject(i);
                    postInList.setName(jsonObject.getString("title"));
                    postInList.setUrlImage(jsonObject.getString("url"));
                    postInList.setDate(jsonObject.getString("date"));
                    postInList.setDescription(jsonObject.getString("description"));
                    postInList.setContent(jsonObject.getString("content"));

                } catch (JSONException e) {
                    e.printStackTrace();
                }

                postInList.add(postInList);
            }

            mAdapter = new CustomRecyclerViewAdapterForPosts(postInList);
            recyclerView.setAdapter(mAdapter);
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.d("Valley Forum", error.toString());
        }
    });

    request_for_posts.add(jsonObjectRequest);
}

```

Εικόνα 40 - Αξιοποίηση Δεδομένων JSON από URL με χρήση του Android Studio

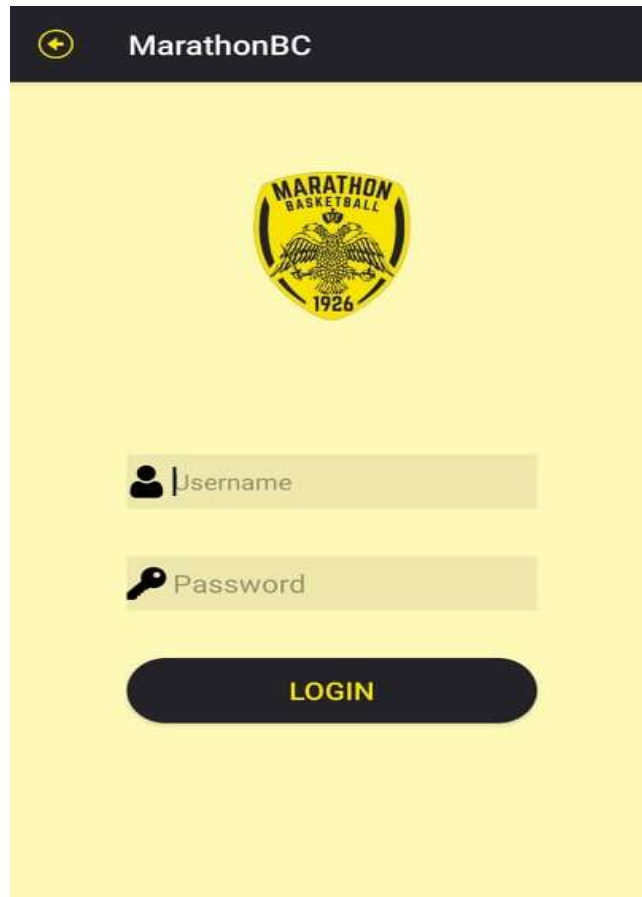
Το επιστρεφόμενο αποτέλεσμα της μεθόδου ήταν ένα array. Άρα σκοπός του πιθανού προγραμματιστή είναι να αποστείλει ένα αίτημα HTTP και να λάβει ως απάντηση ένα JSON array. Κάθε στοιχείο αυτό του array, είναι ένα JSON Object το οποίο περιλαμβάνει τα χαρακτηριστικά που επιθυμεί ο προγραμματιστής για κάθε δημοσίευση. Απαιτείται μεγάλη προσοχή στην ονομασία των πεδίων καθώς τα ονόματα των

χαρακτηριστικών στο JSON θα πρέπει να είναι ίδια με αυτά στο Android Studio ή σε οποιοδήποτε πρόγραμμα χρησιμοποιεί ο χρήστης.

5.5 ΟΛΟΚΛΗΡΩΣΗ MOBILE ΕΦΑΡΜΟΓΗΣ

Με βάση τις ενότητες 4.6 και 5.3, ο πιθανός προγραμματιστής είναι σε θέση να ολοκληρώσει μία εφαρμογή τύπου android που θα αλληλεπιδρά με το Rest API του WordPress. Στην περίπτωση μας, όπως προαναφέρθηκε υπάρχουν τρεις κατηγορίες χρηστών, όπου ο κάθε τύπος έχει πρόσβαση σε διαφορετικές λειτουργίες με βάση το ρόλο του. Παρακάτω αναλύονται στα screenshots οι λειτουργίες που πραγματοποιεί ένας προπονητής με το android application.

5.5.1 ΕΛΕΓΧΟΣ ΕΙΣΟΔΟΥ ΧΡΗΣΤΗ

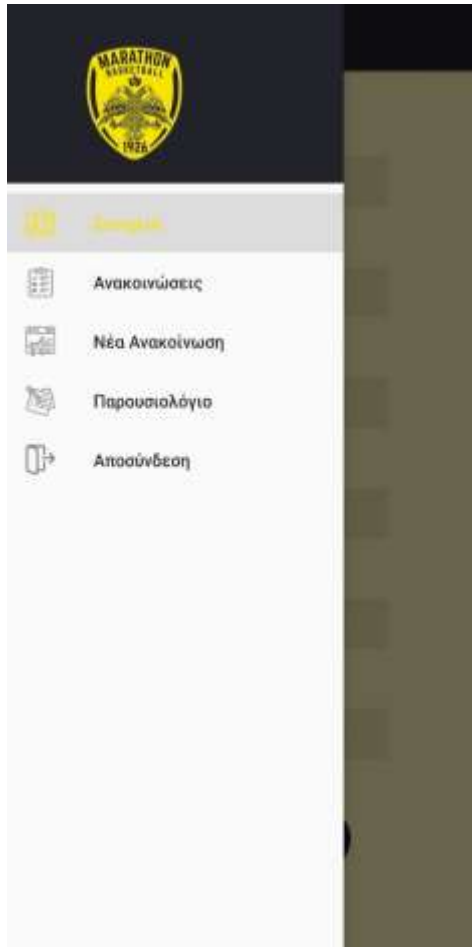


Εικόνα 41 - Έλεγχος Εισόδου Χρήστη

Τα διαπιστευτήρια που χρησιμοποιούνται για την ταυτοποίηση των χρηστών είναι ίδια με αυτά της διαδικτυακής εφαρμογής. Για να επιτευχθεί αυτό δημιουργήθηκε ένα custom endpoint το οποίο δέχεται ως παράμετρο το username και το password του χρήστη και τον ή την ταυτοποιεί με τη μέθοδο wp_signon.

5.5.2 ΕΙΣΟΔΟΣ ΠΡΟΠΟΝΗΤΗ

Εφόσον πιστοποιηθεί ότι ο ρόλος του χρήστη, ότι είναι προπονητής ή προπονήτρια, εισέρχεται στο βασικό μενού λειτουργιών της εφαρμογής.



Εικόνα 42 - Βασικές Λειτουργίες Προπονητή

5.5.3 ΕΠΕΞΕΡΓΑΣΙΑ - ΠΡΟΒΟΛΗ ΣΤΟΙΧΕΙΩΝ ΠΡΟΠΟΝΗΤΗ

Η πρώτη επιλογή «Στοιχεία» δίνει τη δυνατότητα στον προπονητή ή στην προπονήτρια να δει τα στοιχεία του/της και να τα επεξεργαστεί.



Εικόνα 43 - Προβολή και Επεξεργασία Στοιχείων

5.5.4 ΠΡΟΒΟΛΗ ΑΝΑΚΟΙΝΩΣΕΩΝ

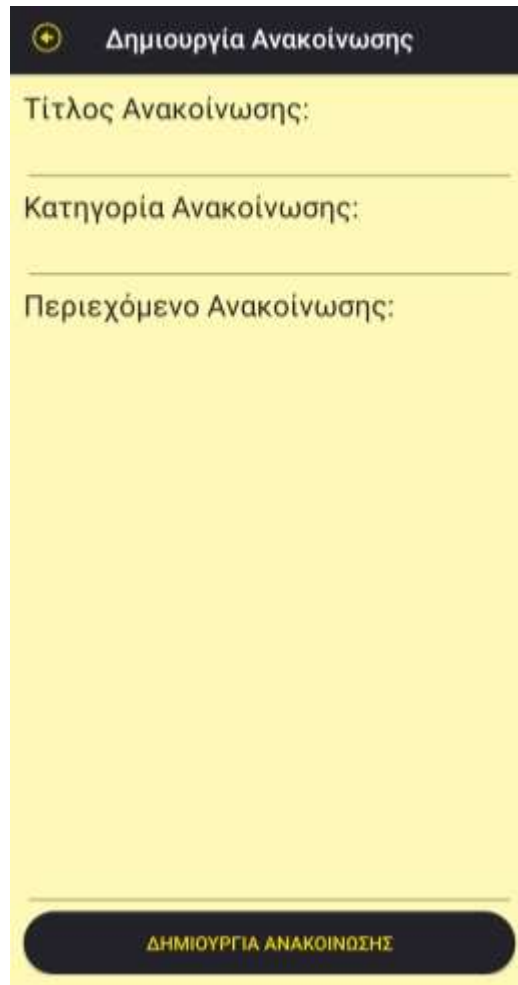
Εφόσον έχει δημιουργήσει κάποια ανακοίνωση, ο προπονητής ή προπονήτρια, έχουν την δυνατότητα να δουν τη συγκεκριμένη ανακοίνωση ή τις ανακοινώσεις από την επιλογή «Ανακοινώσεις».



Εικόνα 44 - Προβολή Ανακοινώσεων

5.5.5 ΔΗΜΙΟΥΡΓΙΑ ΝΕΑΣ ΑΝΑΚΟΙΝΩΣΗΣ

Ο προπονητής ή προπονήτρια, που ανήκουν στο σύλλογο μπορούν να ενημερώνουν τους αθλητές – αθλήτριες του συλλόγου με τη δημιουργία μιας νέας ανακοίνωσης.



Δημιουργία Ανακοίνωσης

Τίτλος Ανακοίνωσης:

Κατηγορία Ανακοίνωσης:

Περιεχόμενο Ανακοίνωσης:

ΔΗΜΙΟΥΡΓΙΑ ΑΝΑΚΟΙΝΩΣΗΣ

Εικόνα 45 - Δημιουργία Νέας Ανακοίνωσης

Όπως παρατηρείται από την εικόνα 42, ο προπονητής – προπονήτρια θα πρέπει να συμπληρώσει 3 (τρία) πεδία προκειμένου να δημιουργήσει μία νέα ανακοίνωση.

1. Τίτλος Ανακοίνωσης
2. Κατηγορία Ανακοίνωσης (το όνομα του τμήματος για το οποίο δημιουργήθηκε η ανακοίνωση)
3. Περιεχόμενο Ανακοίνωσης

5.5.6 ΔΗΜΙΟΥΡΓΙΑ ΠΑΡΟΥΣΙΟΛΟΓΙΟΥ

Τελευταία επιλογή στο μενού λειτουργιών όσον αφορά το ρόλο του προπονητή είναι η δημιουργία παρουσιολογίου. Με την επιλογή αυτή, προβάλλονται τα ονόματα των αθλητών του κάθε τμήματος ξεχωριστά και ο προπονητής, ανάλογα εάν ο αθλητής απουσιάζει ή παρευρίσκεται στην προπόνηση, επιλέγει το κατάλληλο εικονίδιο και προστίθεται αντίστοιχα απουσία ή παρουσία στον εκάστοτε αθλητή – αθλήτρια.

☰ Marathon BC	
ΚΑΡΑΪΤΗΣ ΑΓΓΕΛΟΣ	 
ΑΝΑΓΝΩΣΤΑΚΗΣ ΑΠΟΣΤΟΛΟΣ	 
ΤΖΙΒΟΠΟΥΛΟΣ ΦΙΛΙΠΠΟΣ	 
ΠΑΡΑΣΚΕΥΑΣ ΦΟΙΒΟΣ	 
ΓΚΙΟΚΑΣ ΓΙΑΝΝΗΣ	 
ΑΒΡΑΜΙΔΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ	 
ΤΑΒΑΝΙΔΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ	 
ΤΟΥΜΠΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ	 
ΛΟΥΓΓΟΣ ΝΙΚΟΛΑΟΣ	 
ΚΟΡΕΤΑ ΡΑΦΑΗΛ ΛΟΥΚΑΣ	 

Εικόνα 46 - Δημιουργία Νέου Παρουσιολογίου

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα Πτυχιακή Εργασία προτάθηκε η δημιουργία ενός LDAP καταλόγου και η σύνδεση του με μία δικτυακή εφαρμογή βασισμένη στο λογισμικό WordPress. Η υλοποίηση του καταλόγου έγινε με το Apache Directory Studio, βασισμένη στις βασικές αρχές που διέπουν ένα κατάλογο LDAP, αναλύοντας τα βασικά χαρακτηριστικά του και τις βασικές λειτουργίες του. Ταυτόχρονα, μελετήθηκε η αλληλεπίδραση μίας διαδικτυακής εφαρμογής που έχει κατασκευαστεί με WordPress με μία android εφαρμογή χρησιμοποιώντας το REST API του WordPress. Η πραγματοποίηση των παραπάνω εργασιών οδήγησαν, εν τέλει, στην ανάπτυξη δύο εφαρμογών (διαδικτυακή και android), ενώ δημιουργήθηκε με επιτυχία ο κατάλογος LDAP. Η ολοκλήρωση τους έδωσε τα παρακάτω συμπεράσματα:

1. Το WordPress αποτελεί ένα πανίσχυρο εργαλείο για όποιον επιθυμεί να δημιουργήσει μία διαδικτυακή εφαρμογή. Ωστόσο θα πρέπει να μελετηθούν αναλυτικά οι τεχνολογίες με τις οποίες επιτυγχάνεται η παραμετροποίηση του.
2. Αν και πρόκειται για μία τεχνολογία που δημιουργήθηκε το 1997, το LDAP προσφέρει σημαντικές υπηρεσίες καταλόγου, οι οποίες μπορούν να αξιοποιηθούν κατάλληλα από μεγάλους οργανισμούς που αναζητούν τον έλεγχο ταυτότητας των χρηστών τους από πολλές διαδικτυακές εφαρμογές.
3. Το REST API του WordPress δίνει τη δυνατότητα στους υποψήφιους προγραμματιστές, να δημιουργήσουν εξωτερικές εφαρμογές, οι οποίες θα συνδέονται με εφαρμογές που έχουν δημιουργηθεί με βάση το WordPress.

BIBΛΙΟΓΡΑΦΙΑ

Βιβλία

1. Understanding and Deploying LDAP Directory Services by Gordon Good and Mark Smith
2. LDAP Programming with Java by Rob Weltman and Tony Dahbura
3. LDAP System Administration by Gerald Carter

Internet Sites

1. <https://developer.android.com/>
2. <https://developer.wordpress.org/rest-api/>
3. <https://www.openldap.org/>
4. <https://directory.apache.org/studio/>
5. <https://www.virtualbox.org/>

Thesis

1. Υπηρεσίες Καταλόγου Active Directory και OpenLDAP του Μουμίν Ογλού
2. Εγκατάσταση και παραμετροποίηση συστήματος LDAP των Καπράνου Δ και Πετρουτζάκου Α.

Udemy Courses

1. The Complete Android & Java Developer Course
2. cPanel Crash Course
3. LDAP Directory Services : Concepts, Setup and Configuration