



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Ψηφιακή Βιβλιοθήκη διπλωματικών και ερευνητικών
εργασιών»



Του φοιτητή
Χρήστου Κατσάνη
Αρ. Μητρώου: 164814

Επιβλέπων
Αντώνης Σιδηρόπουλος
Αναπληρωτής Καθηγητής

Ιούνιος 2024

Τίτλος Π.Ε.: Ψηφιακή Βιβλιοθήκη διπλωματικών και ερευνητικών εργασιών

Κωδικός Π.Ε.: 24166

Όνοματεπώνυμο φοιτητή: Χρήστος Κατσάνης

Όνοματεπώνυμο εισηγητή: Αντώνης Σιδηρόπουλος

Ημερομηνία ανάληψης Π.Ε.: 26/03/2024

Ημερομηνία περάτωσης Π.Ε.: 26/05/2024

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 και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

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

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

«Αφιέρωση»

Θα ήθελα να αφιερώσω την συγγραφή αυτής της Π.Ε. σε όσους στάθηκαν κοντά μου

Πρόλογος

Επέλεξα την Π.Ε. διότι πιστεύω, ότι τα ανοικτά δεδομένα είναι ένα σημαντικό κομμάτι για τη σημερινή εποχή. Η πρόσβαση πρέπει να είναι ανοικτή και δωρεάν για όλους.

Με την ευκαιρία αυτήν έμαθα αρκετά από την εγκατάσταση της πλατφόρμας του Dspace και την παραμετροποίηση του καθώς στηρίζεται στις τεχνολογίες angular, spring boot καθώς και γνώσεις linux. Επίσης, έμαθα για την σύνδεση και παραμετροποίηση του από εξωτερικές πηγές δεδομένων.

Περίληψη

Στην σημερινή εποχή όλα τα ακαδημαϊκά ιδρύματα έχουν ψηφιακά αποθετήρια εκτός του πανεπιστημίου μας και το πρόβλημα δημιουργείται στην εύρεση ερευνητικών, πτυχιακών και άλλων δεδομένων. Είναι αναγκαίο να εισαχθεί για την ακαδημαϊκή κοινότητα έτσι ώστε να διευκολύνουν το ευρύτερο κοινό στην εύρεση τους. Τα ψηφιακά αποθετήρια είναι διαδικτυακές πλατφόρμες, οι οποίες χρησιμοποιούνται για την αποθήκευση, τη διαχείριση και την κοινοποίηση ψηφιακών περιεχομένων, όπως κείμενα, εικόνες και βίντεο. Τα ακαδημαϊκά αποθετήρια αναφέρονται σε διαδικτυακές πλατφόρμες, που χρησιμοποιούνται από τους ακαδημαϊκούς και τους ερευνητές για τη διαχείριση, την αποθήκευση και την κοινοποίηση επιστημονικών περιεχομένων, όπως εργασίες, άρθρα, διατριβές, δοκίμια και δεδομένα έρευνας. Συχνά, παρέχουν πρόσβαση σε ανοικτά δεδομένα και στην έρευνα, προωθώντας τη διαφάνεια και την ελεύθερη διάδοση της γνώσης. Επιπλέον, πολλά αποθετήρια προσφέρουν υπηρεσίες, όπως διαχείριση και διάχυση δεδομένων καθώς και εργαλεία για την ανάλυση και την οπτικοποίηση των επιστημονικών πληροφοριών. Μέσω των ακαδημαϊκών αποθετηρίων, οι ερευνητές έχουν την δυνατότητα να αυξήσουν την επιρροή τους και να διευκολύνουν τη συνεργασία και την ανταλλαγή ιδεών με άλλους στην ακαδημαϊκή κοινότητα.

Στο πρώτο κεφάλαιο παρουσιάζονται πλατφόρμες που σχετίζονται με την αποθήκευση δεδομένων ακαδημαϊκού ενδιαφέροντος, την σημασία τους καθώς και τα χαρακτηριστικά τους.

Στο δεύτερο κεφάλαιο προσκομίζονται οι τεχνολογίες, οι οποίες χρησιμοποιήθηκαν για την υλοποίηση της πτυχιακής.

Στο τρίτο κεφάλαιο γνωστοποιούνται πληροφορίες για το Dspace και τα βήματα εγκατάστασης της έκδοσης 7.6.

Στο τέταρτο κεφάλαιο παρουσιάζεται η εξειδικευμένη παραμετροποίηση για την υλοποίηση των χαρακτηριστικών της πτυχιακής όπως η αυθεντικοποίηση των χρηστών και η εισαγωγή των δημοσιεύσεων και των πτυχιακών.

Στο πέμπτο κεφάλαιο εκθέτονται τα αποτελέσματα της εφαρμογής.

«Digital Library of theses and research papers»

«Christos Katsanis»

Abstract

Nowadays all academic institutions have digital repositories outside our university and the problem arises in finding research, thesis and other data. It is necessary to introduce them to the academic community to facilitate the wider public in finding them. Digital repositories are online platforms used to store, manage and share digital content such as text, images and videos. Academic repositories refer to online platforms used by academics and researchers to manage, store and share scholarly content such as papers, articles, theses, dissertations, essays and research data. They often provide access to open data and research, promoting transparency and the free dissemination of knowledge. In addition, many repositories offer services such as data management and dissemination as well as tools for analyzing and visualizing scientific information. Through academic repositories, researchers can increase their influence and facilitate collaboration and exchange of ideas with others in the academic community.

The first chapter presents platforms related to the storage of data of academic interest, their importance as well as their characteristics.

The second chapter presents the technologies used to implement the thesis.

In the third chapter information about Dspace and the installation steps of version 7.6 are disclosed.

The fourth chapter presents the specialized configuration to implement the features of the thesis such as user authentication and the import of publications and theses.

Chapter five reports the results of the implementation.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για την υποστήριξη, την οποία μου έδωσαν σε όλη την πορεία, που είχα στην σχολή καθώς και τον καθηγητή Αντώνη Σιδηρόπουλο για την άμεση βοήθεια του στην πτυχιακή.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Σχημάτων	xi
Συντομογραφίες.....	xii
Κεφάλαιο 1ο: Ψηφιακά Αποθετήρια.....	1
1.1 Εισαγωγή.....	1
1.2 Ιστορική αναδρομή.....	1
1.3 Γνωστά ακαδημαϊκά αποθετήρια	2
Κεφάλαιο 2ο: Τεχνολογίες.....	5
2.1 Η γλώσσα Java	5
2.2 Η γλώσσα Java	5
2.3 Η γλώσσα Python	7
2.4 Η γλώσσα PHP	8
2.5 Το πρωτόκολλο LDAP	9
Κεφάλαιο 3ο: Το Dspace	10
3.1 Εισαγωγή.....	10
3.2 Τι είναι το Dspace	10
3.3 Αρχιτεκτονική του Dspace	10
3.4 Χρήση της πλατφόρμας του Dspace	11
3.5 Παραδείγματα χρήσης.....	11
3.6 Οδηγός εγκατάστασης έκδοσης 7.6	11
3.6.1 Εγκατάσταση του Debian 12	12
3.6.2 Εγκατάσταση του Dspace Backend.....	12
3.6.3 Εγκατάσταση του Dspace Frontend	17
Κεφάλαιο 4ο: Εξειδικευμένη Παραμετροποίηση	21
4.1 Εισαγωγή πτυχιακών – διπλωματικών από thesis.....	21
4.2 Εισαγωγή δημοσιεύσεων από ORCID	23
4.3 Εγκατάσταση του LDAP.....	27
Κεφάλαιο 5ο: Αποτελέσματα Εφαρμογής	28
5.1 Εισαγωγή.....	28

5.2	Εγγραφή Είσοδος στην πλατφόρμα.....	28
5.3	Χρήση της Πλατφόρμας.....	28
5.4	Χρήση ως Διαχειριστής.....	31
Κεφάλαιο 6ο:	Συμπεράσματα και προτάσεις βελτίωσης.....	32
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		34
ΠΑΡΑΡΤΗΜΑ Α: Κώδικες Υλοποίησης.....		36

Κατάλογος Σχημάτων

Εικόνα 1.1: Βιβλιοθήκη της αρχαίας Αλεξάνδρειας	1
Εικόνα 1.2: Στιγμιότυπο οθόνης από την κεντρική σελίδα του zenodo.org	3
Εικόνα 1.3: Στιγμιότυπο οθόνης από την κεντρική σελίδα του arxiv.org	3
Εικόνα 1.4: Στιγμιότυπο οθόνης από την κεντρική σελίδα του dspace.lyrasis.org.....	4
Εικόνα 2.1: James Gosling από το site facesofopensource.com	5
Εικόνα 2.2: Σχέδιο JVM από το site Wikipedia.org.....	6
Εικόνα 2.3: Guido Van Rossum από το site peteradamsphoto.com.....	7
Εικόνα 3.1: Αρχιτεκτονική του Dspace	10
Εικόνα 4.1: Διάγραμμα ροής αλγορίθμου update_thesis.php	22
Εικόνα 4.2: Διάγραμμα ροής Αλγορίθμου thesis_api_update.py.....	23
Εικόνα 4.3: Screenshot από το site του wiki.lyrasis.org	24
Εικόνα 4.4: Δημιουργία Community από την σελίδα dspace.iee.ihu.gr	25
Εικόνα 4.5: Δημιουργία collection στην σελίδα dspace.iee.ihu.gr.....	25
Εικόνα 4.6: Διάγραμμα αλγορίθμου update_api_publication.py	26
Εικόνα 5.1: Είσοδος – Εγγραφή στην σελίδα dspace.iee.ihu.gr.....	28
Εικόνα 5.2: Κεντρική σελίδα του dspace.iee.ihu.gr	29
Εικόνα 5.3: Σελίδα δημοσιεύσεων του dspace.iee.ihu.gr.....	29
Εικόνα 5.4: Σελίδα πτυχιακής – διπλωματικής από το dspace.iee.ihu.gr.....	30
Εικόνα 5.5: Σελίδα δημοσίευσης από το dspace.iee.ihu.gr.....	30
Εικόνα 5.6: Σελίδα πρόφιλ από το dspace.iee.ihu.gr.....	31

Συντομογραφίες

ΔΠΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
API	Application Programming Interface
PHP	Hypertext Preprocessor
LDAP	Lightweight Directory Access Protocol
ORCID	Open Researcher and Contributor ID
JSON	JavaScript Object Notation
SSL	Secure Sockets Layer
TLS	Transport Layer Security
OAUTH	Open Authorization
DOI	Digital Object Identifier
SQL	Structured Query Language
UI	User Interface
URL	Uniform Resource Locator
AJP	Apache JServ Protocol
JVM	Java Virtual Machine
OIDC	OpenID Connector Protocol
CSRF	Cross-Site Request Forgery

Κεφάλαιο 1ο: Ψηφιακά Αποθετήρια

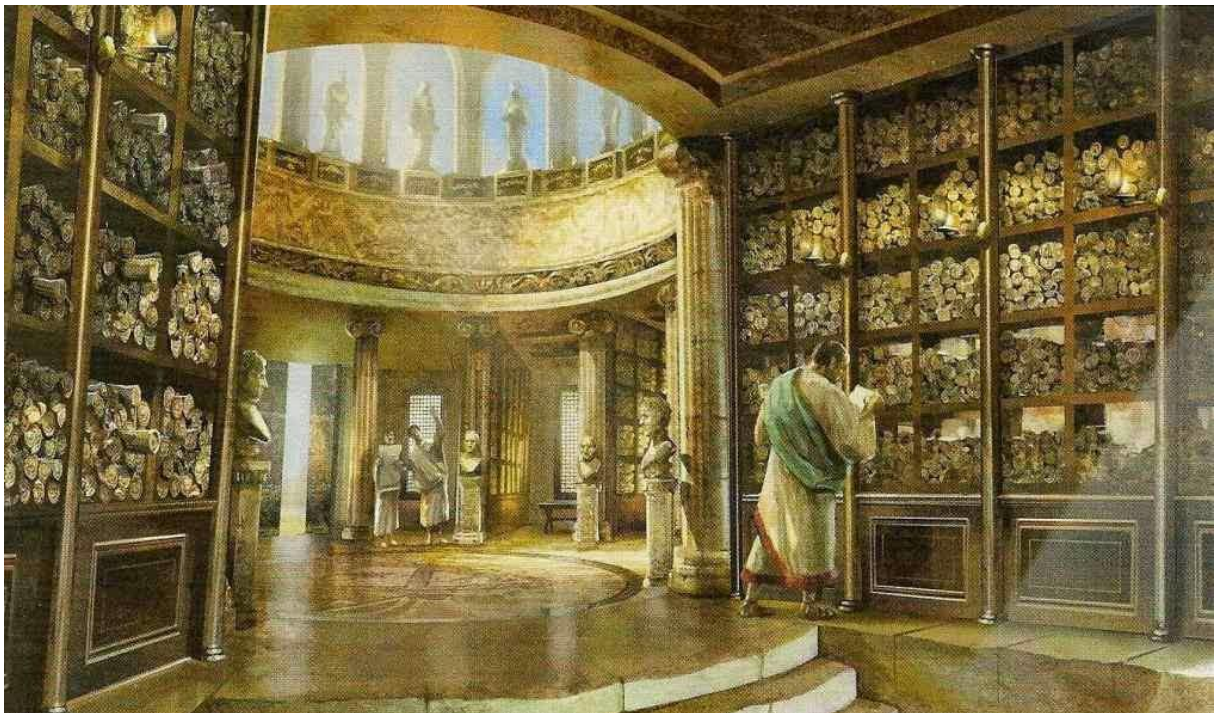
1.1 Εισαγωγή

Τα ψηφιακά αποθετήρια αποτελούν κεντρικό σημείο της ψηφιακής εποχής, παρέχοντας ένα διαδικτυακό χώρο αποθήκευσης, διαχείρισης και κοινοποίησης ψηφιακών περιεχομένων από επιστημονικά άρθρα, διατριβές έως και δεδομένα έρευνας. Επίσης, φιλοξενούν μια πληθώρα πληροφοριών και προσφέρουν δυνατότητες οργάνωσης και πρόσβασης. Τέλος, αναδεικνύουν την σημασία της ελεύθερης πρόσβασης για γνώση στην ακαδημαϊκή κοινότητα επιτρέποντας την ανταλλαγή ιδεών και τη συνεργασία.

1.2 Ιστορική αναδρομή

Πριν την εποχή των ψηφιακών αποθετηρίων, η αποθήκευση και η διάθεση επιστημονικής έρευνας και άλλων διάφορων εγγράφων γινόταν σε βιβλιοθήκες, επιστημονικά περιοδικά και διάφορα άλλα έγγραφα.

Στα αρχαιότερα χρόνια η αποθήκευση γινόταν σε βιβλιοθήκες, όπως η αυτή της Αλεξάνδρειας στην Αίγυπτο.



Εικόνα 1.1: Βιβλιοθήκη της αρχαίας Αλεξάνδρειας

Η βιβλιοθήκη της Αλεξάνδρειας ήταν μια από τις πιο φημισμένες βιβλιοθήκες στην αρχαία εποχή καθώς αποτελούσε σημαντικό μέρος στην ιστορία του πολιτισμού της Αιγύπτου. Ιδρύθηκε περίπου τον 3^ο αιώνα π.Χ. από τον Πτολεμαίο I, τον Σωτήρα βασιλιά της Αιγύπτου.

Επίσης, αποτελούσε ένα κεντρικό σημείο πολιτισμού και γνώσης, συγκεντρώνοντας μια τεράστια συλλογή από γραπτά έργα, αντίτυπα, χειρόγραφα και περγαμηνές.

Περιείχε έργα από διάφορους πολιτισμούς, καθιστώντας την ένα κέντρο, το οποίο προωθούσε ανταλλαγές ιδεών και συζητήσεων μεταξύ πολιτισμών.

Τα ψηφιακά αποθετήρια δημιουργήθηκαν στις αρχές του 1990, με σκοπό την προσπάθεια αποθήκευσης και διαχείρισης εγγράφων και την ηλεκτρονική πρόσβαση του κοινού μέσου του διαδικτύου.

Η ανάπτυξη των ψηφιακών αποθετηρίων οφείλετε στην αύξηση του διαδικτύου και την εξέλιξη των τεχνολογιών. Αυτό οδήγησε την ακαδημαϊκή κοινότητα στην δημιουργία των ακαδημαϊκών αποθετηρίων για την φιλοξενία ερευνητικών διατριβών, επιστημονικών άρθρων, βιβλίων και άλλων ψηφιακών περιεχομένων.

Πλέον, τα ψηφιακά αποθετήρια εξυπηρετούν πολλούς τομείς, παρέχοντας έναν πολύτιμο πόρο για την ανταλλαγή γνώσεων έως την προώθηση της μάθησης και της έρευνας. Ακόμα, αποτελούν ισχυρά εργαλεία, λόγω των χαρακτηριστικών τους. Είναι κατάλληλα για την διαχείριση, τη διάθεση και την ανάπτυξη των ψηφιακών περιεχομένων.

Ανοικτή Πρόσβαση (Open Access): Τα ψηφιακά αποθετήρια, συχνά, υιοθετούν ανοικτές πολιτικές πρόσβασης επιτρέποντας την ελεύθερη πρόσβαση στο περιεχόμενο, για οποιονδήποτε χρήστη, με πρόσβαση στο Διαδίκτυο.

Διατήρηση (Preservation): Τα ψηφιακά αποθετήρια, συχνά, παρέχουν μέσα και μεθόδους για τη μακροπρόθεσμη διατήρηση των ψηφιακών περιεχομένων, εξασφαλίζοντας, ότι θα παραμείνουν προσβάσιμα και χρήσιμα στο μέλλον.

Αναζήτηση και Ανακάλυψη (Search and Discovery): Τα αποθετήρια, συνήθως, παρέχουν προηγμένες δυνατότητες αναζήτησης και φιλτραρίσματος, επιτρέποντας στους χρήστες να εντοπίζουν εύκολα το επιθυμητό περιεχόμενο.

Διαμοιρασμός (Sharing): Τα ψηφιακά αποθετήρια διευκολύνουν τον διαμοιρασμό ψηφιακών περιεχομένων με άλλους χρήστες, επιτρέποντας την ανταλλαγή ιδεών και πληροφοριών.

Διασφάλιση Δικαιωμάτων (Rights Management): Ορισμένα αποθετήρια παρέχουν μηχανισμούς για τη διαχείριση δικαιωμάτων πνευματικής ιδιοκτησίας και άλλων δικαιωμάτων, εξασφαλίζοντας ότι τα περιεχόμενα χρησιμοποιούνται σύμφωνα με τους νόμους και τις συμφωνίες χρήσης.

1.3 Γνωστά ακαδημαϊκά αποθετήρια

Με την εξέλιξη των ψηφιακών αποθετηρίων να ανέρχεται, υπάρχει μια μεγάλη πληθώρα επιλογών στους χρήστες της ακαδημαϊκής κοινότητας. Τα ψηφιακά αποθετήρια, τα οποία παρέχουν φιλικό προς τον χρήστη περιβάλλον και οι εκτεταμένες συλλογές τα καθιστούν ευέλικτη λύση για τους ερευνητές.

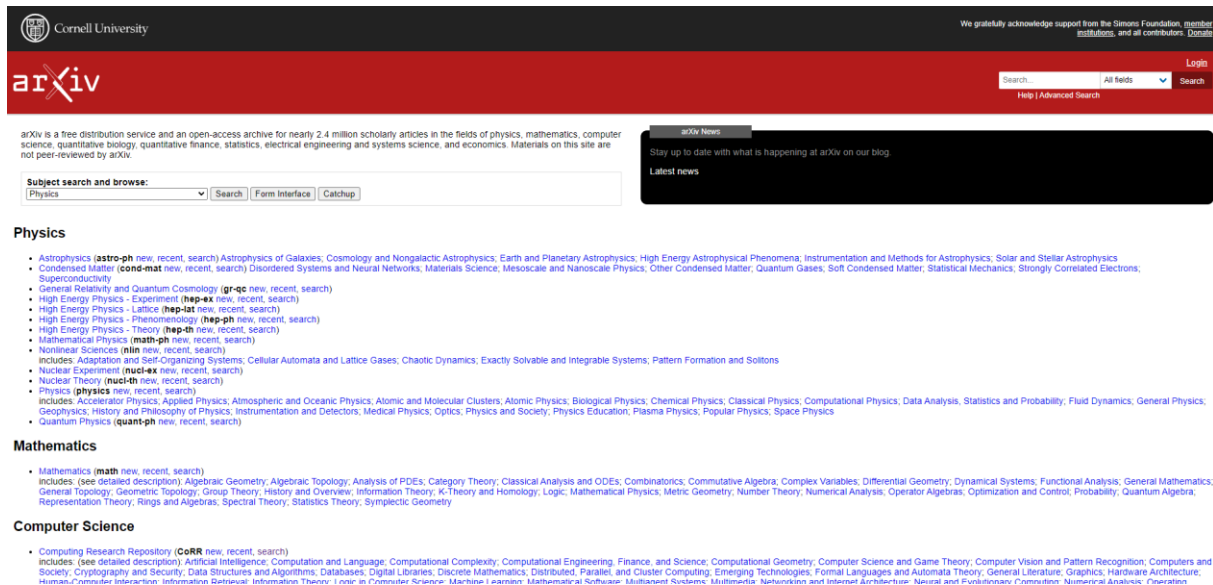
Παρακάτω θα δούμε μια σύντομη αναφορά σε μερικά από αυτά.

1. Η Zenodo[1] είναι μια ψηφιακή βιβλιοθήκη, που παρέχει την πρόσβαση σε επιστημονικά άρθρα, δεδομένα και άλλα περιεχόμενα.



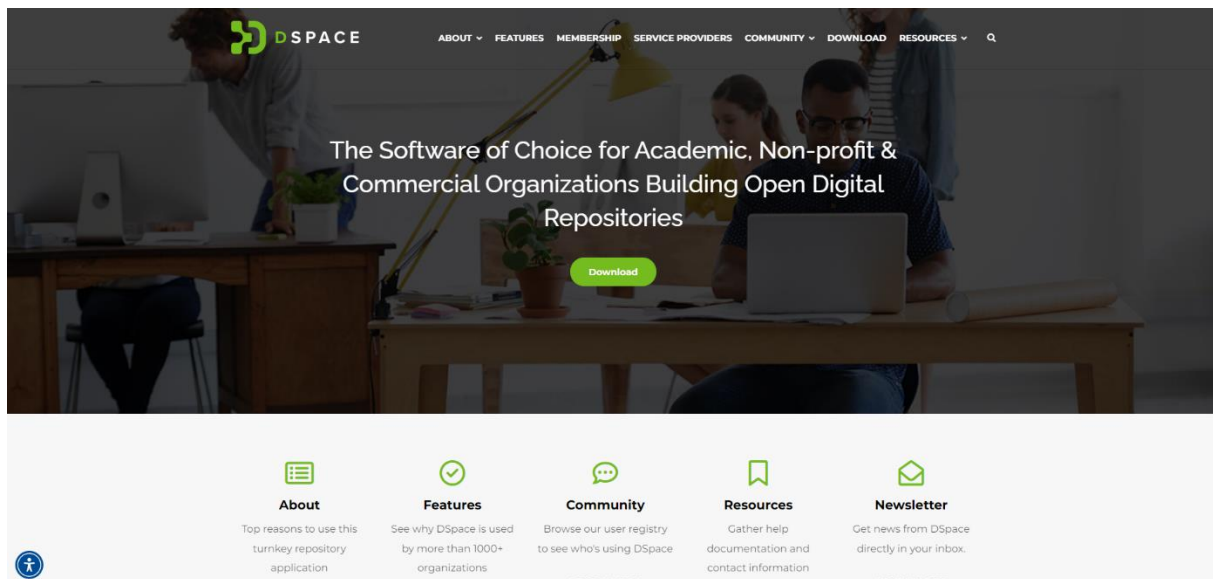
Εικόνα 1.2: Στιγμιότυπο οθόνης από την κεντρική σελίδα του zenodo.org

2. Το `arXiv[2]` είναι ένα ψηφιακό αποθετήριο, το οποίο παρέχει την πρόσβαση σε ακαδημαϊκά άρθρα και έγγραφα των γενικών επιστημών. Επίσης, επιτρέπει σε ερευνητές να μοιράζονται την δουλειά τους με τους χρήστες της ακαδημαϊκής κοινότητας.



Εικόνα 1.3: Στιγμιότυπο οθόνης από την κεντρική σελίδα του arXiv.org

3. Το `Dspace[3]` είναι ένα open source ψηφιακό αποθετήριο, που παρέχει τη δυνατότητα εγκατάστασης από ερευνητικά και ακαδημαϊκά ιδρύματα. Έχει περίπου 3000 εγκαταστάσεις από οργανισμούς και είναι σχεδιασμένο, με σκοπό να υποστηρίζει την διατήρηση και την πρόσβαση σε μια ευρεία γκάμα από ψηφιακές πηγές, επιστημονικά άρθρα και άλλα περιεχόμενα.



Εικόνα 1.4: Στιγμιότυπο οθόνης από την κεντρική σελίδα του dspace.lyrasis.org

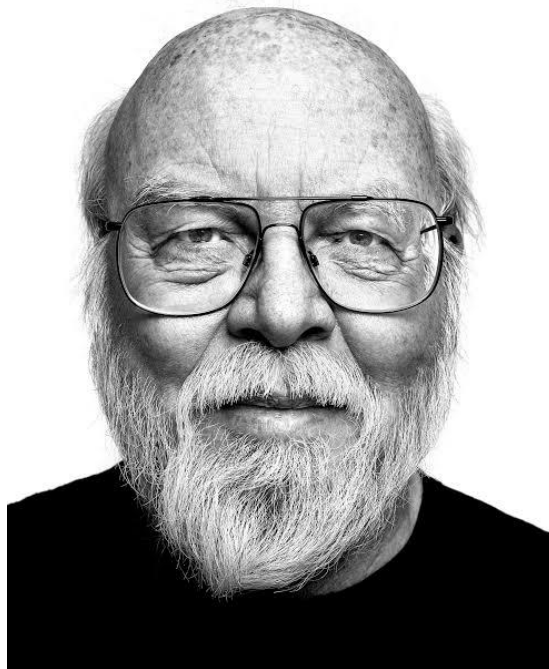
Κεφάλαιο 2ο: Τεχνολογίες

2.1 Η γλώσσα Java

Σε αυτό το κεφάλαιο θα παρουσιαστούν οι τεχνολογίες, οι οποίες έχουν χρησιμοποιηθεί για την ανάπτυξη της εργασίας καθώς και πληροφορίες για αυτές.

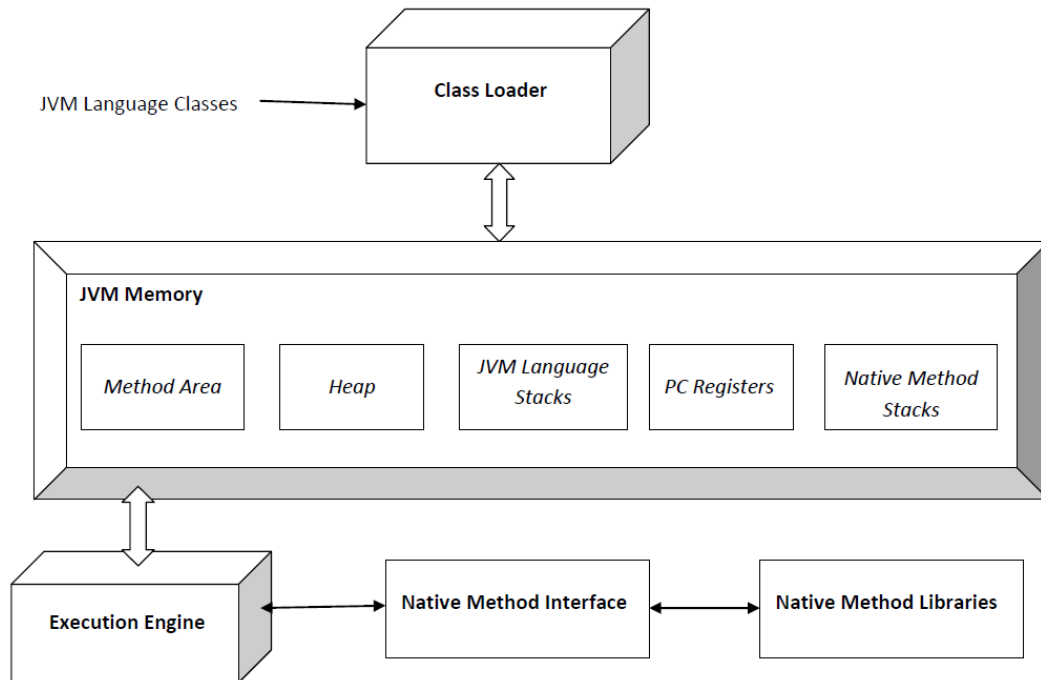
2.2 Η γλώσσα Java

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού, που αναπτύχθηκε από την Sun Microsystems, η οποία εξαγοράστηκε, έπειτα, από την Oracle Corporation. Δημιουργήθηκε το 1995 και από τότε έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού από τον James Gosling για την εταιρεία Sun.



Εικόνα 2.1: James Gosling από το site facesofopensource.com

Ένα από τα βασικότερα χαρακτηριστικά της, έναντι των άλλων γλωσσών, είναι η ανεξαρτησία του λειτουργικού συστήματος καθώς μπορεί να τρέξει σε όλα τα λειτουργικά που έχουν εγκατεστημένη την εικονική μηχανή της.



Εικόνα 2.2: Σχέδιο JVM από το site Wikipedia.org

Ένα από τα πιο γνωστά χαρακτηριστικά της Java είναι το "Write Once, Run Anywhere" (WORA). Ο κώδικας Java μεταγλωττίζεται σε bytecode, ο οποίος μπορεί να εκτελεστεί σε οποιαδήποτε πλατφόρμα έχει εγκατεστημένο τον Java Virtual Machine (JVM).

Επιπρόσθετα, διαθέτει ενσωματωμένο Garbage Collector, όπου δίνει την δυνατότητα στην εικονική μηχανή, να τρέχει ανεξάρτητα την μέθοδο αυτή, χωρίς να χρειαστεί να παρέμβει κάποιος στον κώδικα, εν αντιθέσει με την C. Όταν κριθεί απαραίτητο, η μηχανή εκτελεί τον Garbage Collector και αυτός με την σειρά του αδειάζει την μνήμη από αντικείμενα, τα οποία δεν χρειάζεται πλέον.

Είναι ευέλικτη λόγω της αντικειμενοστρέφειας, που διαθέτει και βασίζεται σε κλάσεις και αντικείμενα. Μία κλάση είναι ουσιαστικά μια φόρμα δημιουργίας αντικειμένων.

Η Java χρησιμοποιείται για την ανάπτυξη διαφόρων τύπων εφαρμογών, όπως εφαρμογές ιστού, κινητές εφαρμογές, εφαρμογές επιχείρησης και ενσωματωμένες συσκευές, καθώς παραμένει μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού λόγω της σταθερότητας και της φορητότητάς της. Ωστόσο, όπως όλες οι άλλες γλώσσες έτσι και η java έχει μειονεκτήματα:

1. Λεπτομερείς συντακτικό: Η σύνταξη για μια απλή εντολή είναι αρκετά λεπτομερείς σε σύγκριση με άλλες γλώσσες, με αποτέλεσμα την δημιουργία μεγάλων project.
2. Ταχύτητα: Η εκτέλεση του προγράμματος έχει αρκετές καθυστερήσεις λόγω ασφαλείας και σταθερότητας.
3. Δέσμευση μνήμης: Η Java σαν γλώσσα δεσμεύει αρκετό χώρο μνήμης για τις μεταβλητές της, σε αντίθεση με άλλες εφαρμογές.

Το backend του dspace έχει υλοποιηθεί σε γλώσσα java καθώς επιτρέπει:

- Να εκτελεστεί ο κώδικας σε οποιοδήποτε λειτουργικό, καθώς τρέχει σε Java Virtual Machine.
- Απόδοση, αφού μπορεί να διαχειριστεί καλύτερα μεγάλα φορτία.

Η επιλογή ήταν στρατηγική λαμβάνοντας υπόψη τα τεχνικά χαρακτηριστικά, την ανεξαρτησία από την πλατφόρμα και την ασφάλεια, την οποία προσφέρει. Γι' αυτόν τον λόγο την καθιστούν κατάλληλη για την ανάπτυξη ενός ψηφιακού αποθετηρίου.

Παρακάτω θα δούμε ένα παράδειγμα κώδικα για την εμφάνιση του Hello World

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

2.3 Η γλώσσα Python

Η Python είναι μια υψηλού επιπέδου, ερμηνευόμενη γλώσσα προγραμματισμού γενικής χρήσης, που δημιουργήθηκε από τον Guido van Rossum και κυκλοφόρησε για πρώτη φορά το 1991.

Είναι γνωστή για την απλότητά της και την ευκολία μάθησης και χρήσης.



Εικόνα 2.3: Guido Van Rossum από το site peteradamsphoto.com

Η Python έχει σχεδιαστεί, για να είναι εύκολη στην ανάγνωση και την γραφή. Η σύνταξη της είναι καθαρή και κατανοητή, καθιστώντας την ιδανική για αρχάριους προγραμματιστές. Αποτελεί μία ερμηνευμένη γλώσσα, όπου ο κώδικας εκτελείται γραμμή-γραμμή από τον διερμηνέα της Python. Αυτό έχει ως αποτέλεσμα να διευκολύνει τον έλεγχο και την αποσφαλμάτωση του κώδικα.

Επίσης, η Python χρησιμοποιεί δυναμική δακτυλογράφηση, που σημαίνει ότι δεν χρειάζεται να δηλώνονται ρητά οι τύποι δεδομένων των μεταβλητών. Αυτό κάνει την ανάπτυξη πιο ευέλικτη και γρήγορη.

Υποστηρίζει τον αντικειμενοστραφή προγραμματισμό, αλλά και άλλες παραδειγματικές προσεγγίσεις, όπως ο διαδικαστικός και ο λειτουργικός προγραμματισμός.

Διαθέτει μια εκτεταμένη βιβλιοθήκη που περιλαμβάνει προκαθορισμένες κλάσεις και μεθόδους για διάφορες λειτουργίες.

Υπάρχουν πολλά δημοφιλή πλαίσια και βιβλιοθήκες, που έχουν αναπτυχθεί για την Python, όπως το Django και το Flask για ανάπτυξη ιστού, το NumPy και το Pandas για επιστημονικούς υπολογισμούς, ανάλυση δεδομένων, το TensorFlow και το PyTorch για μηχανική μάθηση.

Η Python μπορεί να ενσωματωθεί με άλλες γλώσσες προγραμματισμού όπως η C, η C++ και Java καθιστώντας την ευέλικτη για χρήση σε διαφορετικά περιβάλλοντα.

Μπορεί να χρησιμοποιηθεί για εκπαιδευτικούς και επαγγελματικούς λόγους καθώς έχει αρκετές δυνατότητες όπως το game development, ανάλυση δεδομένων και web development.

Η κοινότητα του dspace έχει δείξει αρκετό ενδιαφέρον στην python, λόγω τις ευκολίας, που δίνει στην δημιουργία scripts για την σύνδεση με το api του backend. Οι χρήστες την χρησιμοποιούν για την αλληλεπίδραση με το api ,καθώς το dspace και η python προσφέρουν αιτήματα http, τα οποία η python έχει υλοποιήσει με την κατάλληλη βιβλιοθήκη.

Επίσης, η Python προσφέρει πολλές δυνατότητες για την ενσωμάτωση διάφορων εφαρμογών στην επικοινωνία με το dspace, λόγω της ευκολίας, που προσφέρει με τις βιβλιοθήκες της, οι οποίες την καθιστούν εξαιρετική επιλογή, για την επέκταση λειτουργίας του dspace.

Παρακάτω θα δούμε ένα παράδειγμα κώδικα για την εμφάνιση του Hello World.

```
print("Hello, World!")
```

2.4 Η γλώσσα PHP

Η php (Hypertext Preprocessor) αποτελεί μια ευρέως διαδεδομένη γλώσσα προγραμματισμού server side, που ειδικεύεται στην δημιουργία διαδραστικών εφαρμογών. Αναπτύχθηκε το 1994 από τον Rasmus Lerdorf και από τότε έχει εξελιχθεί αρκετά.

Εκτελείται από την μεριά του διακομιστή, πριν λάβει ο φυλλομετρητής το αποτέλεσμα. Αυτό επιτρέπει την δημιουργία του περιεχομένου, χωρίς καθυστέρηση, καθώς όλες οι υπηρεσίες, που χρησιμοποιούνται εκτελούνται από τον server, πχ SQL. Με αυτόν τον τρόπο υποστηρίζει την σύνδεση με υπηρεσίες, όπως η SQL και άλλες βάσεις δεδομένων.

Η php είναι γλώσσα ανοιχτού κώδικα (opensource) και έχει μεγάλη κοινότητα, η οποία έχει αναπτύξει διάφορα πρόσθετα και frameworks που λειτουργούν στην php.

Επιπρόσθετα, χρησιμοποιείται για την ανάπτυξη:

1. Δυναμικών ιστοσελίδων: Με την ενσωμάτωση της html μπορούμε να δημιουργήσουμε διαδραστικές ιστοσελίδες.
2. Συστήματα διαχείρισης περιεχομένων: Διάφορες πλατφόρμες CMS όπως το WordPress, Drupal και το Joomla έχουν αναπτυχθεί με την php.

3. Api servers: Δημιουργία api διακομιστή για την επικοινωνία με την εφαρμογή πελάτη.

Είναι ευέλικτη καθώς υποστηρίζει πολλούς τύπους βάσεων δεδομένων και μπορεί να συνεργαστεί με διάφορα πρωτόκολλα api.

Η κοινότητα έχει αναπτύξει διάφορα frameworks όπως:

1. Laravel: Ένα ισχυρό και δημοφιλές πλαίσιο που παρέχει κομψή σύνταξη και διάφορα εργαλεία ανάπτυξης λογισμικού.
2. Symfony: Ένα πλαίσιο για ανάπτυξη ισχυρών επεκτάσιμων εφαρμογών.

Υποστηρίζει σχεδόν όλα τα διάσημα πρωτόκολλα SMTP, HTTP, HTTPS, FTP.

Παρακάτω θα δούμε ένα παράδειγμα κώδικα για την εμφάνιση του Hello World

```
<?php
echo "Hello, World!";
?>
```

2.5 Το πρωτόκολλο LDAP

Το LDAP (Lightweight Directory Access Protocol) είναι ένα ελαφρύ πρωτόκολλο πρόσβασης σε καταλόγους, το οποίο χρησιμοποιείται για την αποθήκευση και την ανάκτηση δεδομένων καταλόγου σε ένα δικτυακό περιβάλλον. Δημιουργήθηκε από το Πανεπιστήμιο του Μίσιγκαν και είναι ένα από τα πιο διαδεδομένα πρωτόκολλα αυθεντικοποίησης και υπηρεσίας καταλόγου μέχρι και σήμερα .

Τα δεδομένα οργανώνονται και αποθηκεύονται σε μια ιεραρχική δομή, η οποία είναι γνωστή ως δέντρο καταλόγου ή δέντρο Διευθυνσιολογίου (Directory Information Tree - DIT).

Επίσης, αποτελεί ένα ελαφρύ πρωτόκολλο παρέχοντας αποτελεσματική πρόσβαση σε δεδομένα καταλόγου επιτρέποντας την άμεση αυθεντικοποίηση του χρήστη. Χρησιμοποιεί ένα σύνολο λειτουργιών όπως σύνδεση(bind), αναζήτηση(search), προσθήκη(add), σύγκριση(compare) και διαγραφή(delete). Παρέχει δυνατότητα φιλτραρίσματος στην αναζήτηση στον κατάλογο με βάση διάφορα κριτήρια και δυνατότητα κρυπτογράφησης SSL/TLS για την ασφαλή μεταφορά τους.

Η διαχείριση ταυτότητας και πρόσβασης χρησιμοποιείται για την αποθήκευση και διαχείριση πληροφοριών ταυτότητας όπως ονόματα χρηστών, κωδικούς πρόσβασης και δικαιώματα. Ακόμα, η διαχείριση πόρων επιτρέπει την διαχείριση χρηστών και πόρων δικτύων τα οποία είναι χρήσιμα για μεγάλους οργανισμούς.

Ο κατάλογος που διαθέτει παρέχει ένα ενιαίο σημείο αναφοράς για πληροφορίες χρηστών και πόρων.

Το dspace μας προσφέρει την επιλογή για την χρήση της υπηρεσίας του ldap, καθώς το κάνει αρκετά ευέλικτο στην σύνδεση των χρηστών με το dspace και γρηγορότερο.

Κεφάλαιο 3ο: Το Dspace

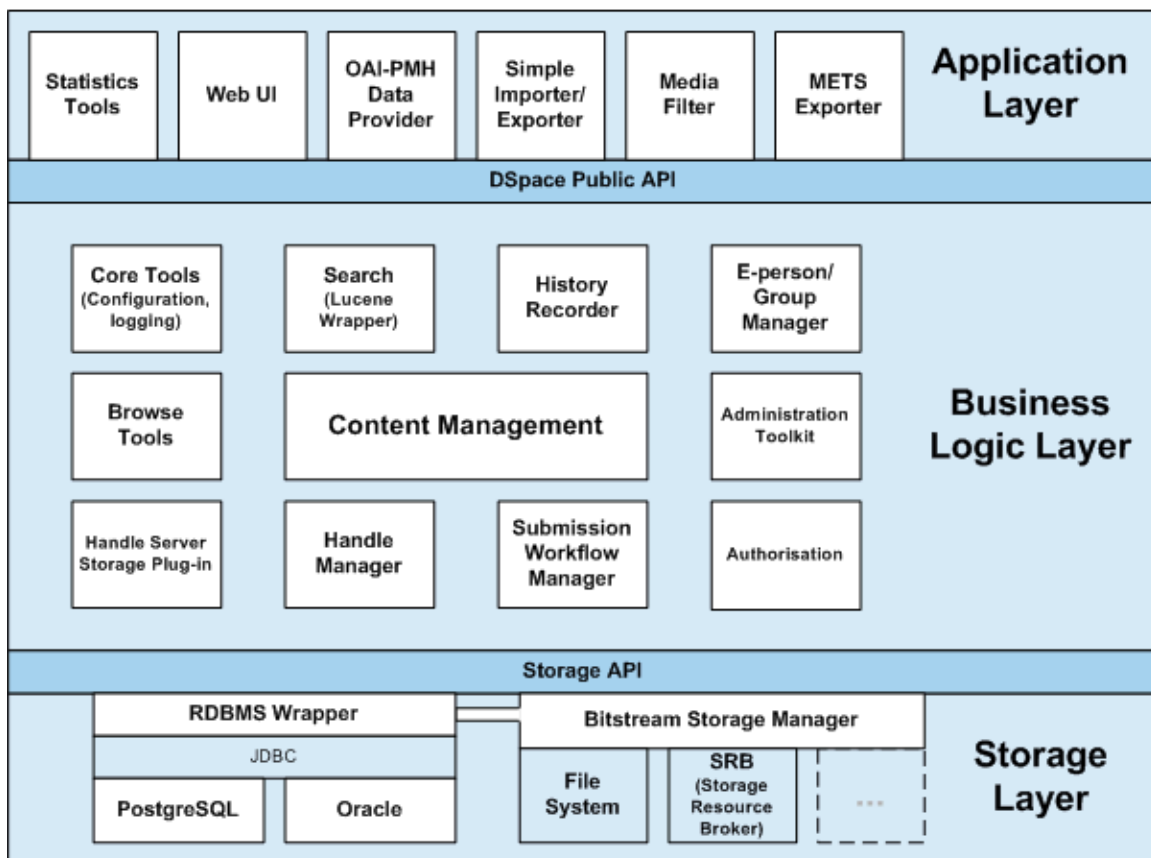
3.1 Εισαγωγή

Σε αυτήν την ενότητα θα παρουσιαστεί τι ακριβώς είναι το Dspace και για ποιόν λόγο υλοποιείται στην σημερινή εποχή. Ο λόγος αυτός είναι η ενσωμάτωση του στην ακαδημαϊκή κοινότητα του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙΠΑΕ, όπου οι χρήστες θα έχουν την δυνατότητα να αναζητήσουν ερευνητικά άρθρα, διατριβές και διπλωματικές, που έχουν εκδοθεί από τους χρήστες του τμήματος.

3.2 Τι είναι το Dspace

Το Dspace είναι μια πλατφόρμα ψηφιακού αποθετηρίου, όπου επιτρέπει στους χρήστες της να εγκαταστήσουν την πλατφόρμα και να προσθέσουν ακαδημαϊκό ή ερευνητικό υλικό. Είναι αρκετά διαδεδομένο στην ακαδημαϊκή και ερευνητική κοινότητα επιτρέποντας στους χρήστες να ανεβάσουν ερευνητικό περιεχόμενο, όπως επιστημονικά περιοδικά, διατριβές, έρευνες και άλλα ψηφιακά μέσα. Επίσης, διαθέτει επιλογές διασύνδεσης με άλλα μέσα αυθεντικοποίησης και ενσωμάτωσης ερευνητικών άρθρων από άλλες πηγές, όπως το ORCID[4]

3.3 Αρχιτεκτονική του Dspace



Εικόνα 3.1: Αρχιτεκτονική του Dspace

Η αρχιτεκτονική του Dspace αποτελείται από τρία επίπεδα:

- Storage Layer – Αυτό το επίπεδο είναι υπεύθυνο για την αποθήκευση των δεδομένων της πλατφόρμας και την διασύνδεση της σε μέσα αποθήκευσης, όπως η PostgreSQL[5]
- Business Logic Layer – Αυτό το επίπεδο είναι υπεύθυνο για την διαχείριση της εφαρμογής, του περιεχομένου και της αναζήτησης μέσω του API, που παρέχει η εφαρμογή.
- Application Layer – Αυτό το επίπεδο εκτελεί την διεπαφή χρήστη Web UI, όπου παρέχεται ένα φιλικό προς τον χρήστη περιβάλλον.

3.4 Χρήση της πλατφόρμας του Dspace

Το DSpace είναι μια πλατφόρμα λογισμικού ανοιχτού κώδικα, ο οποίος χρησιμοποιείται για τη δημιουργία, διαχείριση και διάδοση ψηφιακών αποθετηρίων. Κύρια χρήση του DSpace είναι η δημιουργία ψηφιακών αποθετηρίων, για την αποθήκευση και προβολή ψηφιακού περιεχομένου, όπως επιστημονικά άρθρα, θέσεις εργασίας, διατριβές, παραδοσιακές δημοσιεύσεις, πολυμέσα, και άλλα. Το DSpace παρέχει εργαλεία για τη διαχείριση, την αναζήτηση και την προβολή αυτών των ψηφιακών περιεχομένων. Συνήθως, χρησιμοποιείται από ακαδημαϊκά ιδρύματα, βιβλιοθήκες και άλλους οργανισμούς που επιθυμούν να διατηρούν και να προβάλλουν ψηφιακό περιεχόμενο.

3.5 Παραδείγματα χρήσης

Η πλατφόρμα έγινε διάσημη λόγω της απλότητας χρήσης, που παρέχει και της ευελιξίας στις λειτουργίες του. Επιλέγεται, συχνά, επειδή είναι ανοικτού κώδικα και δεν χρειάζεται συνδρομή για την εγκατάσταση της. Αυτός είναι και ο λόγος, που πολλά ακαδημαϊκά και ερευνητικά ιδρύματα επιλέγουν την χρήση της.

Παρακάτω θα δούμε παραδείγματα χρήσης από άλλα ιδρύματα που παρέχουν ανοικτή πρόσβαση:

1. Ψηφιακό αποθετήριο Εθνικού Μετσόβιου Πολυτεχνείου[6]: Η ψηφιακή βιβλιοθήκη διαθέτει δημοσιεύσεις, διατριβές, διπλωματικές και ιστορικά βιβλία.
2. Ψηφιακό καταθετήριο Πανεπιστημίου Δυτικής Μακεδονίας[7]: Το ψηφιακό καταθετήριο διαθέτει δημοσιεύσεις, διατριβές, ερευνητικό υλικό και πτυχιακές.
3. Ψηφιακό αποθετήριο Γεωπονικού Πανεπιστημίου Αθηνών[8]: Το ψηφιακό αποθετήριο διαθέτει επιστημονικές δημοσιεύσεις ανοιχτής πρόσβασης και διατριβές από το 1929 έως και σήμερα.
4. Ψηφιακό αποθετήριο Πανεπιστημίου Μακεδονίας[9]: Το ψηφιακό αποθετήριο Ψηφίδα διαθέτει ψηφιακή βιβλιοθήκη και υλικό για Εντυποανάπηρους φοιτητές καθώς και ερευνητικές εργασίες και δημοσιεύσεις.

3.6 Οδηγός εγκατάστασης έκδοσης 7.6

Για την ανάπτυξη της πτυχιακής επιλέχθηκε η έκδοση 7.6 του Dspace. Εδώ θα αναλύσουμε τα βήματα που χρειάστηκαν για την εγκατάσταση και την παραμετροποίηση του.

Σκοπός του project ήταν:

- Η σύνδεση του χρήστη μέσω του ιδρυματικού λογαριασμού από την σελίδα του τμήματος <https://login.iee.ihu.gr> με χρήση πρωτοκόλλου LDAP.
- Η άντληση όλων των ολοκληρωμένων πτυχιακών από το σύστημα Thesis <https://thesis.iee.ihu.gr>

- Η άντληση όλων των επιστημονικών δημοσιεύσεων των μελών του τμήματος που υπάρχουν στο ORCID.

Η παραπάνω ενέργειες θα εξηγηθούν στην επόμενη ενότητα, για την παραμετροποίηση των API για την εισαγωγή μέσου του Dspace API με χρήση της python.

Η εγκατάσταση βασίζεται στο online documentation “Ubuntu22 DSPace 7.4 Instuctions”[10] και την επίσημη ιστοσελίδα της Lyrasis <https://wiki.lyrasis.org/display/DSDOC7x> η οποία πραγματοποιήθηκε σε Debian 12[11]

3.6.1 Εγκατάσταση του Debian 12

- 1 Κατεβάσουμε το ISO του Debian 12
- 2 Δημιουργούμε ένα εικονικό μηχάνημα και το κάνουμε εγκατάσταση.
- 3 Ακολουθούμε τις οδηγίες εγκατάστασης

3.6.2 Εγκατάσταση του Dspace Backend

Για την εγκατάσταση του backend θα πρέπει να εγκατασταθούν κάποια πακέτα, τα οποία χρειάζονται και θα παρουσιαστούν παρακάτω.

1. Διενεργείται ενημέρωση του λειτουργικού συστήματος, για να υπάρχουν όλα τα ενημερωμένα πακέτα:

```
sudo apt update && sudo apt upgrade -y
```

2. Πραγματοποιείται εγκατάσταση της έκδοσης 11 ή 17 της java[12]:

```
sudo apt install default-jdk
```

Η εγκατεστημένη έκδοση εμφανίζεται με την εντολή:

```
java -version
```

3. Ελέγχεται, αν η έκδοση, που έγινε εγκατάσταση έχει τοποθετηθεί σωστά στην διαδρομή:

```
cd /usr/lib/jvm
```

```
ls -la
```

```
find / -name "jvm"
```

Θα πρέπει να εμφανιστεί ένας φάκελος με όνομα “java-11-openjdk-amd64” ή “java-17-openjdk-amd64”

4. Η έκδοση αυτή τοποθετείται στο αρχείο environment του συστήματος:

```
sudo nano /etc/environment
```

Προσθέτεται το JAVA_HOME="/usr/lib/jvm/[Το όνομα του φακέλου]"

Έπειτα ενεργοποιείται στο terminal session την java ή ανανεώνεται το session:

```
source /etc/environment
```

```
echo &JAVA_HOME
```

Θα πρέπει να εμφανιστεί η διαδρομή του φακέλου που εγκαταστάθηκε.

5. Εγκατάσταση Apache Maven[13] 3.6.X και εμφάνιση έκδοσης για την εγκατάσταση των πακέτων του dspace

```
sudo apt install maven -y
```

```
mvn -v
```

6. Εγκατάσταση Apache Ant[14] 1.10.X και εμφάνιση έκδοσης για την εγκατάσταση του web app του dspace

```
sudo apt install ant -y
```

```
ant -version
```

7. Εγκατάσταση και δημιουργία χρήστη PostgreSQL για την βάση δεδομένων

```
sudo useradd postgres
```

```
sudo apt install postgresql postgresql-client postgresql-contrib -y
```

Ενεργοποίηση του Cluster της PostgreSQL:

```
pg_ctlcluster 15 main start
```

Ελέγχεται αν η υπηρεσία είναι ενεργή και τρέχει:

```
sudo systemctl status postgres
```

Αν δεν είναι, η εκκίνησή της πρέπει να γίνει ως εξής:

```
sudo systemctl start postgres
```

Δημιουργείται ο κωδικός για τον χρήστη postgres:

```
sudo passwd postgres
```

Ακολουθεί η είσοδος στον χρήστη:

```
su postgres
```

Εκτελείται η εντολή για να είναι σίγουρο ότι υπάρχει το UTF8:

```
psql -c "SHOW SERVER_ENCODING"
```

Για την έξοδο του χρήστη εκτελείται η εντολή:

```
exit
```

Γίνεται επεξεργασία στο αρχείο postgresql.conf με την εντολή:

```
sudo nano /etc/postgres/15/main/postgresql.conf
```

Αναζητείται η γραμμή #listen_addresses = 'localhost' και αφαιρείται το # για να μπορεί η σύνδεση να γίνει μόνο τοπικά

Έπειτα γίνεται επεξεργασία στο αρχείο pg_hba.conf

```
sudo nano /etc/postgres/15/main/pg_hba.conf
```

Έπειτα γίνεται προσθήκη της παρακάτω εγγραφής πριν την γραμμή # Database administrative login by Unix domain socket

```
host dspace dspace 127.0.0.1 255.255.255.255 md5
```

Εκτελείται επανεκκίνηση της υπηρεσίας postgres

```
sudo systemctl restart postgres
```

8. Εγκατάσταση Solr 8.11.3 [15] για την υπηρεσία ευρετηριοποίησης

Στον φάκελο optional (opt)

```
cd /opt
```

Με την χρήση της wget γίνεται εγκατάσταση της υπηρεσίας Solr

```
sudo wget https://downloads.apache.org/lucene/solr/8.11.3/solr-8.11.3.zip
```

Έπειτα γίνεται αποσυμπίεση του φακέλου

```
sudo unzip solr-8.11.3.zip
```

Δίνονται δικαιώματα στον χρήστη που θα το τρέξει την υπηρεσία στον φάκελο solr-8.11.3

```
sudo chown -R <user> solr-8.11.3
```

Έπειτα η υπηρεσία προσθέτεται στο αρχείο .profile για την εκτέλεση από τον χρήστη

```
cd ~/
```

```
nano .profile
```

Στο τέλος της γραμμής προσθέτεται η γραμμή `/opt/solr-8.11.3/bin/solr start`

Έπειτα εκτελείται επανεκκίνηση του μηχανήματος

```
sudo reboot now
```

Το αποτέλεσμα είναι διαθέσιμο στο <http://localhost:8983/solr/>

9. Εγκατάσταση Apache Tomcat 9[16] για την λειτουργία του backend server

Δημιουργείται ένα group και έναν χρήστη tomcat

```
sudo groupadd tomcat
```

```
sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

Έπειτα η εγκατάσταση του tomcat γίνεται από τον σύνδεσμο παρακάτω

```
curl -O http://www-eu.apache.org/dist/tomcat/tomcat-9/v9.0.11/bin/apache-tomcat-9.0.11.tar.gz
```

Σε έναν νέο φάκελο εκτελείται η αποσυμπίεση του αρχείου

```
sudo mkdir /opt/tomcat
```

```
sudo tar xzvf apache-tomcat-9*.tar.gz -C /opt/tomcat --strip-components=1
```

```
cd /opt/tomcat
```

Δίνονται τα απαραίτητα δικαιώματα για να μπορεί ο tomcat να έχει πρόσβαση σε αυτά

```
sudo chown -R tomcat /opt/tomcat
```

```
sudo chmod -R g+r conf
```

```
sudo chmod g+r conf
```

```
sudo chown -R tomcat webapps/ work/ temp/ logs/
```

Έπειτα ενημερώνονται τα alternatives

```
sudo update-java-alternatives -l
```

Θα εμφανιστεί ένα αποτέλεσμα με μια διαδρομή από την Java όπου θα χρησιμοποιηθεί στο αρχείο που θα δημιουργηθεί

```
« Output java-1.11.0-openjdk-amd64 1111 /usr/lib/jvm/java-1.11.0-openjdk-amd-64 »
```

Στην συνέχεια, δημιουργείται το αρχείο συστήματος για να αναγνωρίζεται ως service tomcat.service στην διαδρομή /etc/systemd/system/tomcat.service

```
sudo /etc/system/system/tomcat.service
```

Γίνεται επικόλληση του κώδικα στο Κώδικας 6.2 Αρχείο tomcat.service

Ανανεώνουμε το system service και εκκινούμε την υπηρεσία και την κάνουμε active στο boot

```
sudo systemctl daemon-reload
sudo systemctl start tomcat
sudo systemctl enable tomcat
```

Σε περίπτωση που υπάρχει το Undefined Firewall ufw[17] προστίθεται η πόρτα για να μην το μπλοκάρει το τοίχος προστασίας

```
sudo ufw allow 8080
sudo ufw reload
```

Ακολουθεί η δημιουργία ενός φακέλου στον οποίο θα γίνει η εγκατάσταση του Dspace μέσα στον optional φάκελο

```
cd /opt
sudo mkdir dspace
sudo chown -R tomcat /opt/dspace/
```

Πρέπει να δηλωθεί στο JAVA_OPTS η απαραίτητη μνήμη για να τρέξει το Dspace backend

```
sudo nano /etc/environment
```

Τοποθετείται η παρακάτω γραμμή

```
JAVA_OPTS="-Xmx512M -Xms64M -Dfile.encoding=UTF-8"
```

Έπειτα γίνεται η πρόσθεση του Terminal Session την αλλαγή ή ανανεώνουμε το session

```
source /etc/environment
```

Έπειτα πρέπει να αλλάξει μια ρύθμιση στην διαδρομή /etc/tomcat/server.xml το οποίο φαίνεται στο Κώδικας 6.3 Αρχείο server.xml

```
Sudo nano /etc/tomcat/server.xml
```

Γίνεται ανανέωση του system service και επανεκκίνηση του tomcat για να διαβάσει τις αλλαγές

```
sudo systemctl daemon-reload
sudo systemctl restart tomcat
```

Ο tomcat λειτουργεί στον σύνδεσμο: <http://localhost:8080/>

10. Εγκατάσταση Git για να έχουμε πρόσβαση στα repositories του github

```
sudo apt install git -y
```

11. Εγκατάσταση Dspace Backend

Πάμε στον home directory ή στο optional φάκελο και κατεβάζουμε από το repository του dspace

```
cd ~/
wget https://github.com/DSPACE/DSPACE/archive/refs/tags/dspace-7.6.1.zip
unzip dspace-7.6.1.zip
mv ~/dspace-7.6.1 ~/dspace-backend
```

12. Δημιουργία βάσης και χρήστη στην postgres

Μπαίνουμε στον χρήστη postgres

```
su postgres
```

```
cd ~/
```

Δημιουργούμε τον χρήστη για το dspace για την βάση δεδομένων

```
createuser --username=postgres --no-superuser --pwprompt dspace
```

Δημιουργούμε την βάση δεδομένων για το dspace

```
createdb --username=postgres --owner=dspace --encoding=UNICODE dspace
```

Ενεργοποιούμε το πρόσθετο κρυπτογράφησης στην βάση δεδομένων dspace

```
psql dspace -c "CREATE EXTENSION pgcrypto;"
```

```
exit
```

13. Παραμετροποίηση του Dspace config

Ανοίγουμε το αρχείο ~/dspace-backend/dspace/config/local.cfg

```
cd ~/dspace-backend/dspace/config/local.cfg
```

```
cp local.cfg.EXAMPLE local.cfg
```

```
nano local.cfg
```

Αυτό το αρχείο είναι το configuration file του backend. Από εδώ ορίζουμε το path για το front & back end, επίσης μπορούμε να ενεργοποιήσουμε τις επιπρόσθετες υπηρεσίες ldap, oauth(oidc), orcid, shib

Για το production θα βάλουμε το Κώδικας 6.4 αρχείο local.cfg

Ολοκληρώνοντας τις απαραίτητες ρυθμίσεις αποθηκεύουμε

Έπειτα μπαίνουμε στον φάκελο ~/dspace-backend

```
cd ~/dspace-backend
```

Ξεκινάμε το build κάνοντας εγκατάσταση τις απαραίτητες βιβλιοθήκες

```
mvn package
```

Όταν ολοκληρωθεί η εγκατάσταση (μπορεί να πάρει 10 λεπτά) πάμε στον φάκελο ~/dspace-backend/dspace/target/dspace-installer

```
cd ~/dspace-backend/dspace/target/dspace-installer
```

Τώρα ήρθε η ώρα της εγκατάστασης με την χρήση της ant

```
sudo ant fresh_install
```

Το επόμενο βήμα είναι να κάνουμε migrate την βάση για να είμαστε σίγουροι ότι είναι όλα εντάξει μετά από αναβάθμιση

```
sudo /opt/dspace/bin/dspace database migrate
```

Τώρα πρέπει να αντιγράψουμε τα αρχεία της εγκατάστασης του dspace στον φάκελο του tomcat και τα αρχεία του solr για να έχουμε πρόσβαση από το διαδίκτυο

```
sudo cp -R /opt/dspace/webapps/* /opt/tomcat/webapps*
```

```
sudo cp -R /opt/dspace/solr/* /opt/solr-8.11.3/server/solr/configsets
```

Κάνουμε επανεκκίνηση την υπηρεσία Solr

```
/opt/solr-8.11.3/bin/solr restart
```

Τώρα μας έμεινε να δημιουργήσουμε λογαριασμό διαχειριστή

```
sudo /opt/dspace/bin/dspace create-administrator
```

Ακολουθούμε τις οδηγίες για την δημιουργία λογαριασμού

Δίνουμε ως νέο ιδιοκτήτη τον tomcat στον φάκελο του dspace

```
sudo chown -R tomcat:tomcat /opt/dspace
```

Κάνουμε επανεκκίνηση το service του tomcat

```
sudo systemctl restart tomcat
```

Εδώ η εγκατάσταση του backend έχει τελειώσει. Απαραίτητη προϋπόθεση να έχουμε συμπληρώσει σωστά το local.cfg για να μπορέσει ο ant να κάνει install. Το dspace.url & server.url θα δούμε παρακάτω πως θα είναι ως τελικό url σε πόρτες 80, 443 μέσω proxy από τον apache[18].

3.6.3 Εγκατάσταση του Dspace Frontend

1. Εγκατάσταση του Node.js[19] v18.20.2 για την εφαρμογή του frontend

Κατεβάζουμε το αρχείο install.sh

```
wget -qOhttps://raw.githubusercontent.com/nvm-sh/nvm/v0.39.2/install.sh
```

Βλέπουμε τις εκδόσεις που είναι διαθέσιμες και επιλέγουμε την 18.20.2

```
nvm ls-remote
```

Κατεβάζουμε την έκδοση 18.20.2

```
nvm install 18.20.2
```

Βλέπουμε την έκδοση που έχουμε εγκαταστήσει

```
node -v
```

2. Εγκατάσταση Yarn και PM2

Εγκατάσταση του yarn για το build και την δοκιμή του rest api

```
npm install --global yarn
```

Εγκατάσταση του PM2 για να τρέξει η εφαρμογή του frontend

```
npm install --global pm2
```

3. Εγκατάσταση Apache και CertBot

Εγκαθιστούμε τον Apache ως proxy για να ανακατευθύνει τα αιτήματα και να εγκαταστήσουμε ssl

```
sudo apt install apache2
```

Ενεργοποιούμε τα πρόσθετα για την διαχείριση header, ανακατεύθυνση και ssl

```
sudo a2enmod headers;
```

```
sudo a2enmod proxy
```

```
sudo a2enmod proxy_ajp
```

```
sudo a2enmod ssl
```

Έπειτα πρέπει να φτιάξουμε ένα αρχείο στην διαδρομή /etc/apache2/sites-available/dspace90.conf και να βάλουμε από το Κώδικας 6.5 Αρχείο apache τις ρυθμίσεις για την πόρτα 80

```
sudo nano /etc/apache2/sites-available/dspace90.conf
```

Έπειτα ενεργοποιούμε το config dspace90.conf στον apache

```
sudo a2ensite dspace90.conf
```

Έπειτα πρέπει να εγκαταστήσουμε το ssl

```
sudo apt -y install python3-certbot-apache
```

Εγκαθιστούμε το πιστοποιητικό από το certbot

```
sudo certbot --apache -d yourdomain.com -d www.yourdomain.com
```

Το πιστοποιητικό εγκαθιστά αυτόματα στο /etc/apache2/sites-enabled/default-ssl.conf τις απαραίτητες γραμμές

Το ανοίγουμε και βάζουμε από το Κώδικας 6.5 Αρχείο apache τις ρυθμίσεις για την πόρτα 443

Θα πρέπει να αντικαταστήσουμε το domain name και την διαδρομή με τα αρχεία του πιστοποιητικού που εκδόθηκε

```
sudo nano /etc/apache2/sites-enabled/default-ssl.conf
```

Ανοίγουμε το αρχείο του /opt/tomcat/server.xml για να ενεργοποιήσουμε την δυνατότητα επικοινωνίας με τον AJP

```
sudo nano /opt/tomcat/server.xml
```

Βγάζουμε το # από το

```
<Connector protocol="AJP/1.3" port="8009" redirectPort="8443"
URIEncoding="UTF-8" />
```

Κάνουμε επανεκκίνηση τον apache και tomcat

```
sudo systemctl restart apache2
```

```
sudo systemctl restart tomcat
```

4. Εγκατάσταση Dspace Frontend

Πάμε στον φάκελο ~/ ή στον optional φάκελο

```
cd ~/
```

Διαγράφουμε το αρχείο του backend

```
rm dspace-7.6.1.zip
```

Κατεβάζουμε την έκδοση 7.6.1 για το frontend από το github

```
wget https://github.com/DSpace/dspace-angular/archive/refs/tags/dspace-7.6.1.zip
```

Αποσυμπιέζουμε το αρχείο dspace-7.6.1.zip

```
unzip dspace-7.6.1.zip
```

Μετονομάζουμε τον φάκελο dspace-angular-dspace-7.6.1

```
mv ~/dspace-angular-dspace-7.6.1/ ~/dspace-frontend
```

Μπαίνουμε στον φάκελο

```
cd ~/dspace-frontend
```

Τώρα εγκαθιστούμε το frontend με την εντολή yarn

```
yarn install
```

Κάνουμε build prod για production dev για development

```
yarn build:prod
```

Έπειτα πάμε στον φάκελο του config

```
cd ~/dspace-frontend/config
```

Αντιγράφουμε το αρχείο config.example.yml στο config.prod.yml

```
cp config.example.yml config.prod.yml
```

```
nano ~/dspace-frontend/config/config.prod.yml
```

Θα δούμε όπως στο Κώδικας 6.6 Αρχείο config.prod.yml

Πρέπει στο host να είναι το όνομα του domain και το port του rest 443

Έπειτα θα πάμε στον φάκελο ~/dspace-frontend

```
cd ~/dspace-frontend
```

Εδώ θα δοκιμάσουμε την σύνδεση μεταξύ του frontend και του rest(backend)

```
yarn test:rest
```

Θα πρέπει να πάρουμε response 200 αλλιώς έχουμε κάνει κάποιο λάθος στην παραμετροποίηση του config.prod.yml και local.cfg

Πάμε στον φάκελο ~/dspace-frontend

```
cd ~/dspace-frontend
```

Δημιουργούμε ένα αρχείο angular.json και βάζουμε τα παρακάτω στον φάκελο ~/dspace-frontend

```
nano angular.json
```

Βάζουμε τον κώδικα από το Κώδικας 6.7 Αρχείο angular.json αποθηκεύουμε και τρέχουμε

```
pm2 startup
```

και ακολουθούμε τις οδηγίες

Έπειτα αποθηκεύουμε το pm

```
pm2 save
```

Θα πρέπει να υπάρχει ένα αρχείο με το όνομα που έχει δοθεί «dspace-angular»

```
sudo nano /etc/systemd/system/dspace-angular.service
```

Με το περιεχόμενο από το Κώδικας 6.8 Αρχείο dspace-angular.service

Όπου katsanis το όνομα χρήστη σας

Πλέον μπορούμε να το εκκινήσουμε – σταματήσουμε τη υπηρεσία του frontend

Κεφάλαιο 3

```
sudo systemctl start / stop / status dspace-angular
```

Κεφάλαιο 4ο: Εξειδικευμένη Παραμετροποίηση

Σε αυτό το κεφάλαιο θα προσκομιστεί, πως έγινε η παραμετροποίηση, για να μπορεί ο χρήστης να συνδεθεί μέσω του login.iee.ihu.gr μέσω της υπηρεσίας ldap καθώς και το πώς θα εισαχθούν οι ολοκληρωμένες διπλωματικές – πτυχιακές και δημοσιεύσεις στο dspace μέσω του api. Θα ασχοληθούμε με την rython[20] και το repository, που είναι διαθέσιμο στο dspace-client-python[21] καθώς και την rhp [22], όπου θα αντλήσουμε τα δεδομένα από το thesis.

4.1 Εισαγωγή πτυχιακών – διπλωματικών από thesis

Θα δούμε αναλυτικά, πως κάνουμε την εισαγωγή των δεδομένων από το <https://orcid.org/> API μέσω του Dspace API.

Αρχικά κάνουμε εγκατάσταση της rython

Εγκατάσταση Python & Pip & rhp για την εκτέλεση των αρχείων που θα χρησιμοποιήσουμε για την άντληση των δεδομένων μας

```
sudo apt -y install python3
sudo apt -y install python3-pip
sudo apt -y install php8
```

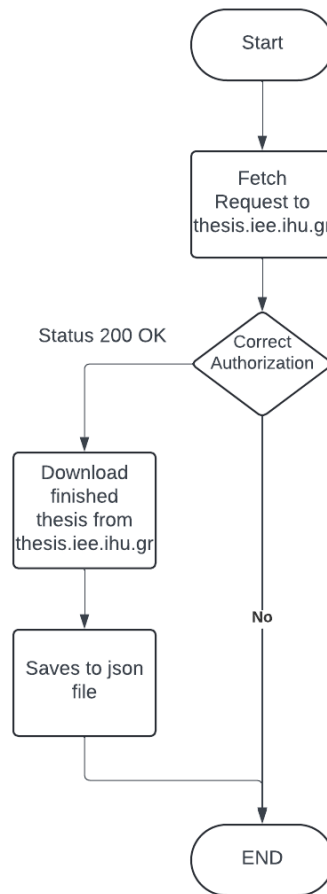
Δημιουργούμε τον φάκελο thesis και βάζουμε το rhp αρχείο για την άντληση των πτυχιακών.

```
cd ~
```

```
mkdir thesis
```

```
nano update_thesis.php
```

και βάζουμε το Κώδικας 6.1.Αρχείο update_thesis.php



Εικόνα 4.1: Διάγραμμα ροής αλγορίθμου update_thesis.php

Με το αρχείο αυτό κάνουμε κλήση στο `api.thesis.iee.ihu.gr` με το κατάλληλο `access token` και παίρνουμε αποθηκεύουμε τα δεδομένα στο αρχείο `thesis_finished.json` όπου μετά θα το περάσουμε στο `dspace` μέσω της `python`.

Έπειτα στον φάκελο `thesis` κατεβάζουμε το `rest client`

```
cd ~/thesis
```

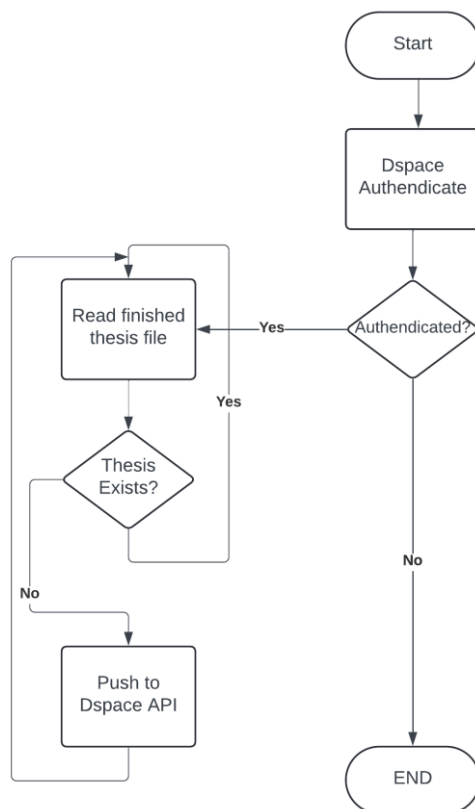
```
git clone https://github.com/the-library-code/dspace-rest-python.git
```

```
pip3 install -r requirements.txt
```

Έπειτα δημιουργούμε ένα αρχείο `thesis_api_update.py`

```
nano thesis_api_update.py
```

θα βάλουμε τον κώδικα από το Κώδικας 6.11 Αρχείο `update_thesis.py`



Εικόνα 4.2: Διάγραμμα ροής Αλγορίθμου thesis_api_update.py

Σε αυτό το αρχείο γίνεται η σύνδεση με το dspace, όπου παίρνουμε τα CSRF Tokens, ανοίγουμε το αποθηκευμένο αρχείο, που έχουμε πάρει παραπάνω το thesis_finished.json και μέσω της python γίνεται εισαγωγή στο API του Dspace όπου αποθηκεύουμε τα id για την αποφυγή διπλότυπων εγγραφών.

Σε όλα αυτά τα βήματα αποθηκεύονται τα αρχεία:

- ids.txt: Αποθήκευση id και παρεμπόδιση δημιουργίας διπλότυπων εγγραφών.
- pdf: Αποθήκευση των εισαχθέντων pdf και άμεση διαγραφή με την ολοκλήρωση της εισαγωγής.

4.2 Εισαγωγή δημοσιεύσεων από ORCID

Για την εισαγωγή των χρηστών και δημοσιεύσεων πρέπει να ρυθμίσουμε το dspace κατάλληλα

Αρχικά ενεργοποιούμε το SolrAuthorAuthority αφαιρώντας το «#»

```
sudo nano /opt/dspace/config/modules/authority.cfg
```

```
plugin.named.org.dspace.content.authority.ChoiceAuthority = \
    org.dspace.content.authority.SolrAuthority = SolrAuthorAuthority
```

Έπειτα στο local.cfg προσθέτουμε στο τέλος τον κώδικα στο Κώδικας 6.9 Αρχείο local.cfg Για ενεργοποίηση ρυθμίσεων orcid

Μετά στο config/mofules/solrauthority.cfg

```
# This setting should already be specified in your solrauthority.cfg
solr.authority.server=${solr.server}/${solr.multicorePrefix}authority
```

Κάνουμε indexing το authority για να είναι up to date

Κεφάλαιο 4

```
sudo /opt/dspace/bin/dspace index-authority
```

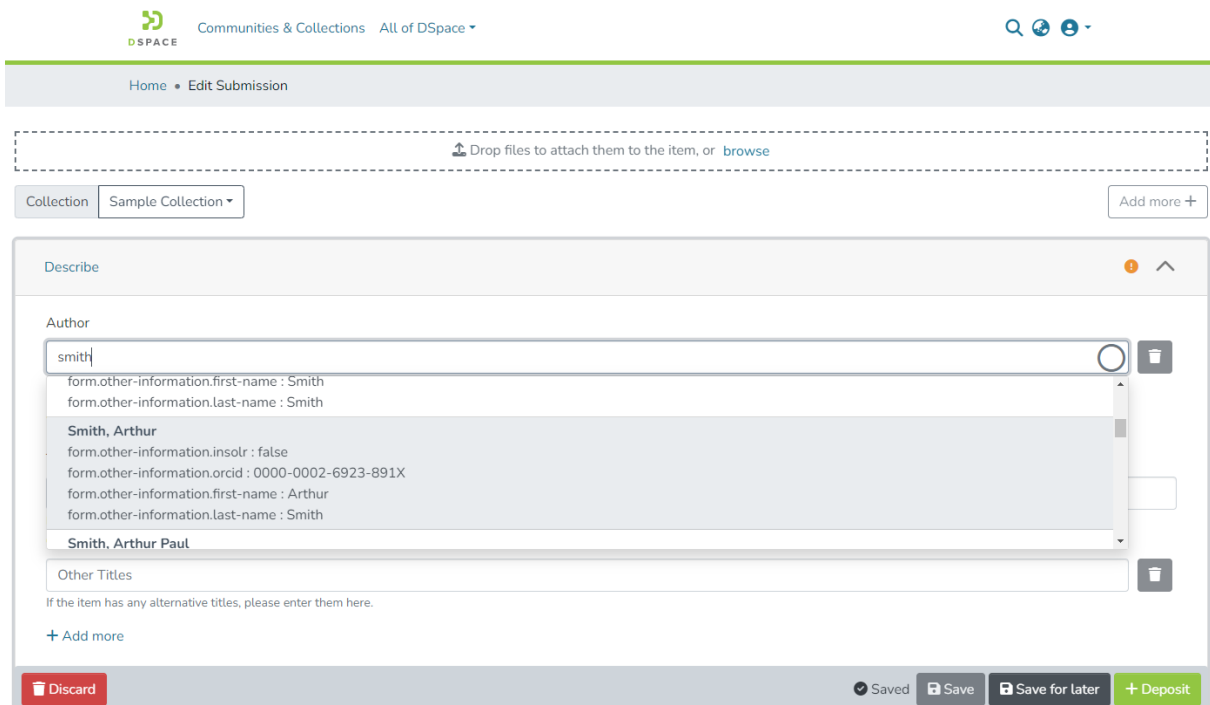
Στο config/modules/authority.cfg πρέπει να βγάλουμε τα «#»

```
# Configuration settings required for Researcher Profiles
# These settings ensure "dspace.object.owner" field are indexed by Authority
Control
choices.plugin.dspace.object.owner = EPersonAuthority
choices.presentation.dspace.object.owner = suggest
authority.controlled.dspace.object.owner = true
```

Για την ενεργοποίηση των Researcher Profiles εισάγουμε τα παρακάτω στο config/modules/researcher-profile.cfg

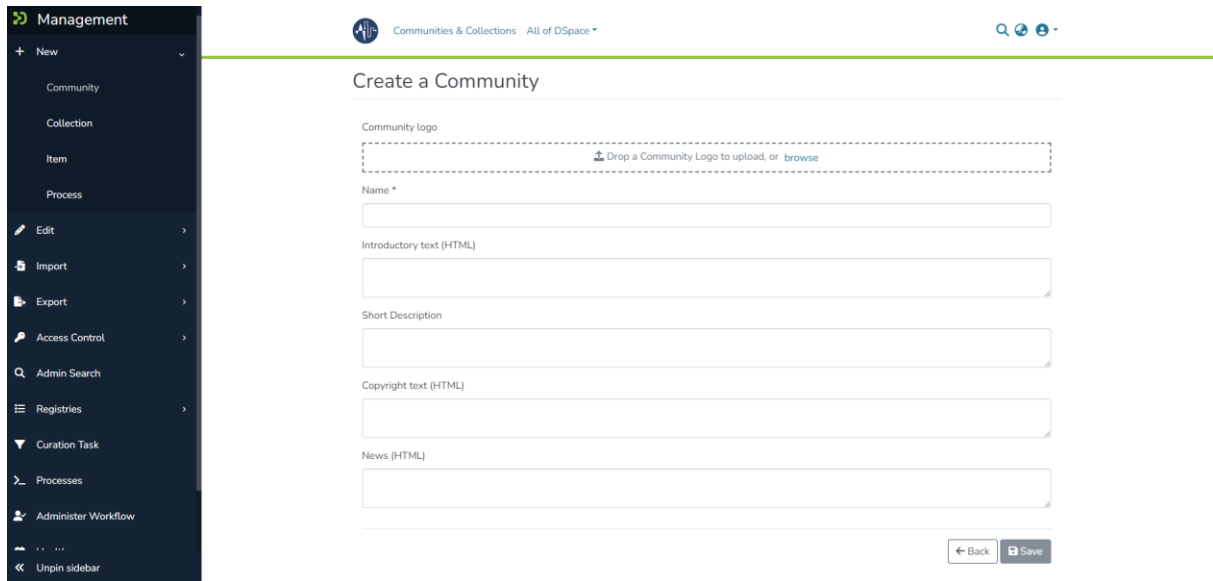
```
researcher-profile.entity-type = Person
researcher-profile.hard-delete.enabled = false
researcher-profile.set-new-profile-visible = false
```

Πλέον μπορούμε να εισάγουμε publications κάνοντας αναζήτηση από το my dspace -> import publications

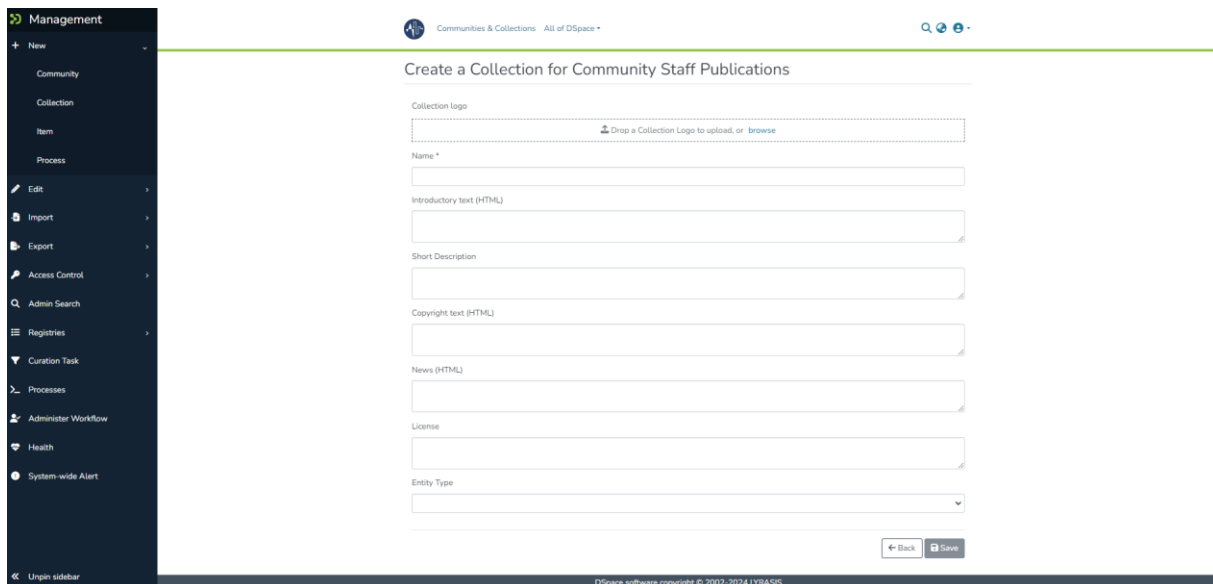


Εικόνα 4.3: Screenshot από το site του wiki.lyrasis.org

Για να μπορέσουν να λειτουργήσουν όλα πρέπει να δημιουργήσουμε ένα Community και δύο collections



Εικόνα 4.4: Δημιουργία Community από την σελίδα dspace.iee.ihu.gr

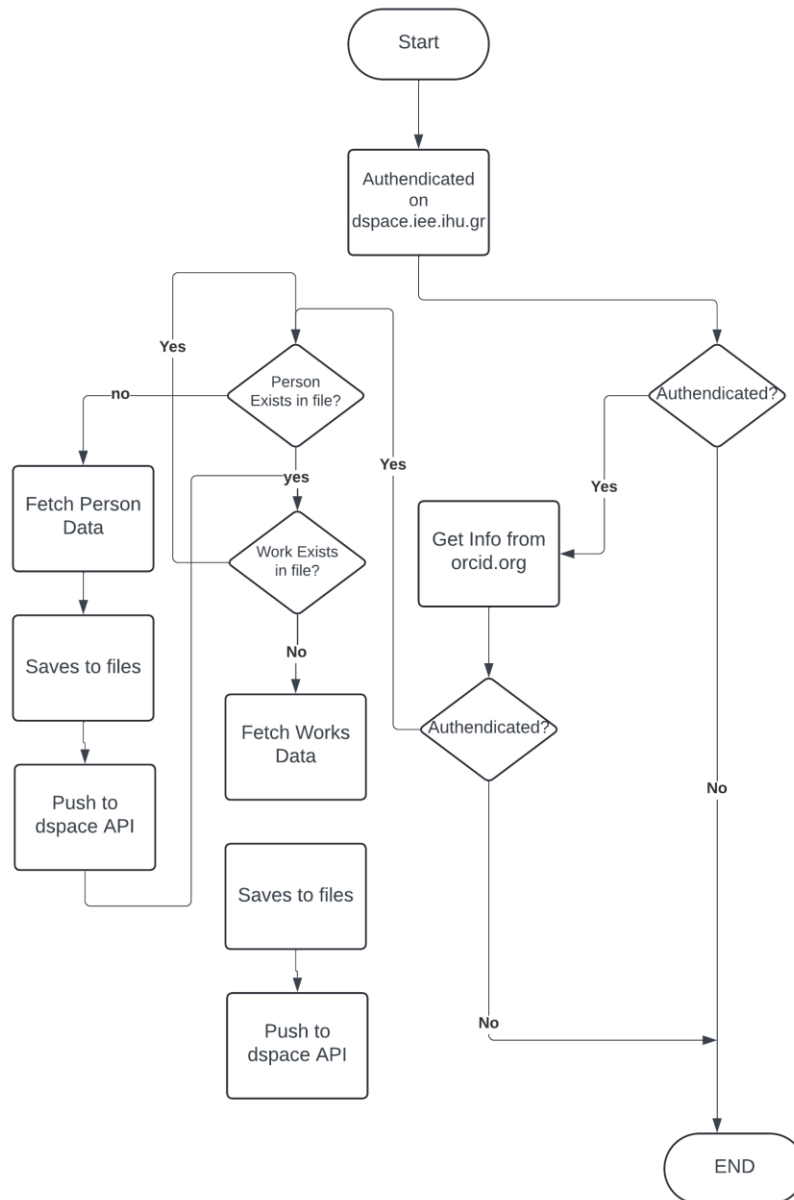


Εικόνα 4.5: Δημιουργία collection στην σελίδα dspace.iee.ihu.gr

Στα collections θα πρέπει να επιλέξουμε το αντίστοιχο Entity Type: Publication ή Person.

Έπειτα θα κρατήσουμε τα ids τους όπως <https://dspace.iee.ihu.gr/collections/8b1dc0fe-d79d-41ae-8579-4ea4dcbab2fa> (8b1dc0fe-d79d-41ae-8579-4ea4dcbab2fa) για να τα βάλουμε στα αντίστοιχα item στην python.

Δημιουργούμε το αρχείο publications_api_update.py και βάζουμε τον κώδικα από το Κώδικας 6.10
Αρχείο update_publications.py



Εικόνα 4.6: Διάγραμμα αλγορίθμου update_api_publication.py

Σε αυτό το αρχείο γίνεται η σύνδεση με το dspace, όπου παίρνουμε τα CSRF Tokens, κάνουμε κλήση στο API του ORCID για συγκεκριμένα ids και πριν την εισαγωγή των δημοσιεύσεων, πρώτα ελέγχουμε αν υπάρχει ο χρήστης ως Person μέσω του αρχείου savedpreson.txt, αν δεν υπάρχει γίνεται εισαγωγή, έπειτα ελέγχουμε αν υπάρχει η δημοσίευση μέσω των αρχείων title, savedpubs και αντλούμε όλα τα works που είναι διαθέσιμα, ταυτόχρονα γίνεται κλήση στο API του Crossref[23], όπου αντλούμε τα δεδομένα authors.

Σε όλα αυτά τα βήματα αποθηκεύονται τα αρχεία:

- Titles.txt: Αποθήκευση τίτλων και παρεμπόδιση δημιουργίας διπλότυπων εγγραφών.
- Savedpubs.txt: Αποθήκευση των εισαχθέντων δημοσιεύσεων για την παρεμπόδιση διπλοτυπίας.
- Savedperson.txt: Αποθήκευση των Person που έχουν εισαχθεί.
- Profiles.txt: Αρχείο εισαγωγής ORCID για την άντληση δεδομένων.

- Relations.json: Αποθηκεύει τα στοιχεία των Person μαζί.
- {oidcid}-person.json: Αποθηκεύει τα αναλυτικά στοιχεία του ατόμου
- {oidcid}-relations.json: Αποθηκεύει τα στοιχεία του Person
- {oidcid}.json: Αποθηκεύει όλες τις δημοσιεύσεις του ατόμου

Για την λειτουργία των παραπάνω Script πρέπει να βάλετε το username και password από τον λογαριασμό σας στο Dspace με την προϋπόθεση ότι έχετε δικαιώματα διαχειριστή, επίσης θα χρειαστείτε τα client id και secret από το orcid.org για την άντληση κάποιων στοιχείων.

Στο βήμα για την δημιουργία των collections κρατήσαμε τα id όπου θα πρέπει να βάλουμε τα αντίστοιχα στο parent

```
new_item = d.create_item(parent="8b1dc0fe-d79d-41ae-8579-4ea4dcbab2fa",
item=item)
```

4.3 Εγκατάσταση του LDAP

Θα δούμε αναλυτικά πως έγινε η σύνδεση με τον LDAP[24] Server του users.iee.ihu.gr και πως αυτό μπορεί να πάρει τα στοιχεία του χρήστη καθώς και να τον τοποθετήσει απευθείας σε ομάδα.

Αρχικά πρέπει να ενεργοποιήσουμε το plugin του ldap μέσα από το local.cfg.

Ανοίγουμε το αρχείο local.cfg

```
sudo nano /opt/dspace/config/local.cfg
```

Βάζουμε το παρακάτω και αποθηκεύουμε για την ενεργοποίηση των μεθόδων εισόδου

```
plugin.sequence.org.dspace.authenticate.AuthenticationMethod =
org.dspace.authenticate.PasswordAuthentication,
org.dspace.authenticate.LDAPAuthentication
```

Έπειτα πρέπει να κάνουμε τις κατάλληλες ρυθμίσεις, ώστε να μπορέσει να συνδεθεί με τον ldap server.

Στο αρχείο local.cfg μπορούμε να κάνουμε override απευθείας, χωρίς να πειράζουμε τα αρχεία μέσα στον φάκελο /opt/dspace/config/modules

Θα πρέπει να βάλουμε την παραμετροποίηση που φαίνεται στον Κώδικας A.12 Αρχείο για την ενεργοποίηση του ldap στο local.cfg

Μετά την αποθήκευση του local.cfg χρειάζεται επανεκκίνηση του tomcat

```
sudo systemctl restart tomcat
```

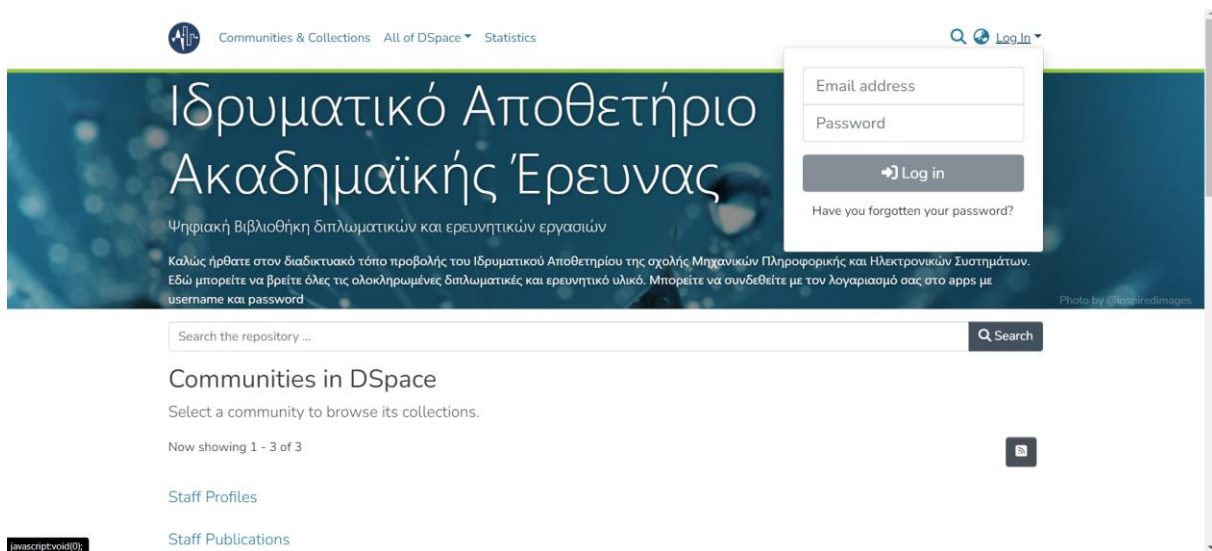
Κεφάλαιο 5ο: Αποτελέσματα Εφαρμογής

5.1 Εισαγωγή

Στην ενότητα αυτήν θα δούμε τα αποτελέσματα της εφαρμογής του Dspace. Θα δούμε πως μπορούμε να συνδεθούμε με τα στοιχεία μας από το login.iee.ihu.gr καθώς και πως αντλήσαμε από τα API του thesis και του orcid τα δεδομένα για τα οποία κληθήκαμε να παρέμβουμε.

5.2 Εγγραφή Είσοδος στην πλατφόρμα

Για να μπορέσουμε να συνδεθούμε στην πλατφόρμα ως χρήστης μπαίνουμε αρχικά στον <https://dspace.iee.ihu.gr> και επιλέγουμε log in.



Εικόνα 5.1: Είσοδος – Εγγραφή στην σελίδα dspace.iee.ihu.gr

Με αυτόν τον τρόπο μπορούμε να συνδεθούμε στην πλατφόρμα με τα στοιχεία που έχουμε στο apps, βάζοντας στο Email address το όνομα χρήστη, που έχουμε στο apps (it164814) και τον κωδικό μας. Θα δούμε, ότι με αυτόν τον τρόπο θα λάβουμε έναν ρόλο Student, αν είμαστε φοιτητής ή Staff, αν ανήκουμε στο προσωπικό του τμήματος.

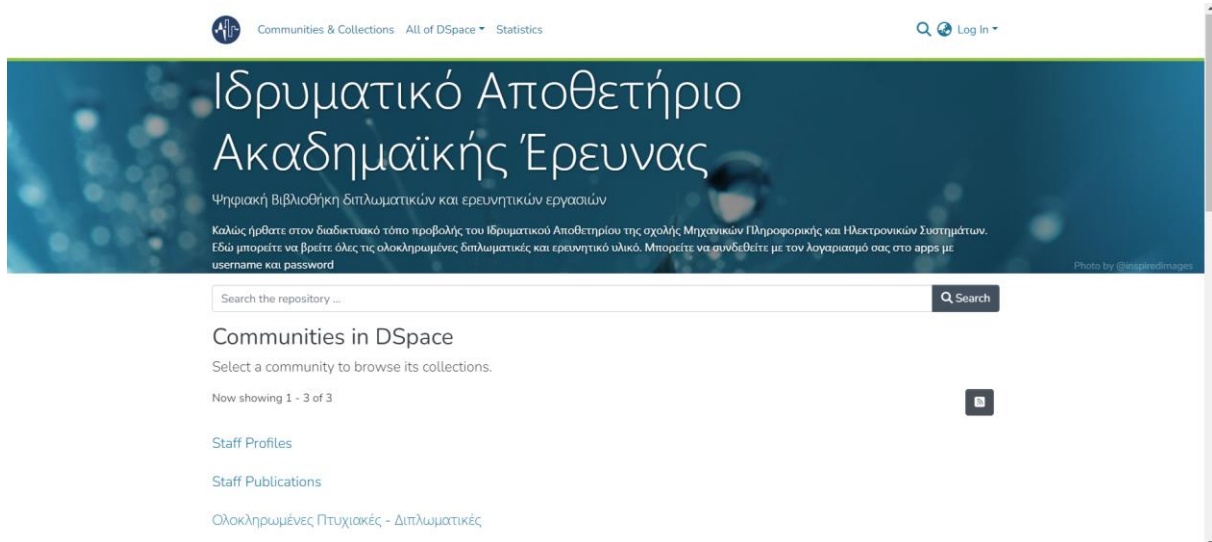
1. Ο ρόλος του φοιτητή μπορεί απλά να συνδεθεί και να δει τα ήδη ανοικτά δεδομένα που υπάρχουν.
2. Ο ρόλος Staff επιτρέπει στο προσωπικό την διαχείριση του περιεχομένου των Publication και των ολοκληρωμένων πτυχιακών – διπλωματικών, όπως την προσθήκη, την επεξεργασία τους καθώς και την δημιουργία του public Person Profile για την συστοιχία των δημοσιεύσεων.
3. Ο ρόλος διαχειριστή παρέχει πλήρη πρόσβαση στον χρήστη σε δικαιώματα διαχειριστή.

Η πλατφόρμα δεν επιτρέπει την εγγραφή άλλων μελών μέσω email, αλλά μόνο την πιστοποίηση μέσω του login.iee.ihu.gr.

5.3 Χρήση της Πλατφόρμας

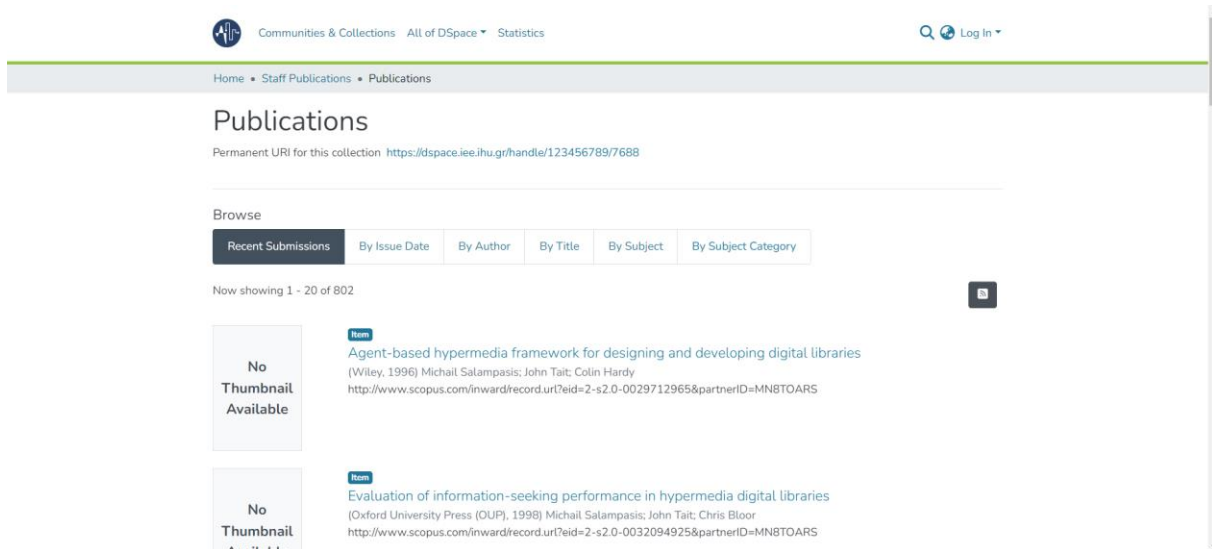
Στην κεντρική σελίδα μπορούμε να δούμε τα διαθέσιμα community που υπάρχουν στην πλατφόρμα όπως:

1. **Staff Profiles – Profiles:** Σε αυτό το collection εμφανίζονται όλα τα προφίλ του ερευνητικού προσωπικού, τα οποία δημιουργούνται αυτόματα κατά την άντληση των δημοσιεύσεων.
2. **Staff Publications – Publications:** Σε αυτό το collection εμφανίζονται όλες οι δημοσιεύσεις, που έχουν αντληθεί από το API του orcid του ερευνητικού προσωπικού.
3. **Ολοκληρωμένες Πτυχιακές – Διπλωματικές:** Σε αυτό το collection εμφανίζονται όλες οι ολοκληρωμένες πτυχιακές – διπλωματικές, που έχουν αντληθεί από το API του thesis.



Εικόνα 5.2: Κεντρική σελίδα του dspace.iee.ihu.gr

Παρακάτω δίνεται ένα παράδειγμα της διαδρομής Staff Publications – Publications, όπου εμφανίζονται οι δημοσιεύσεις του ερευνητικού προσωπικού.

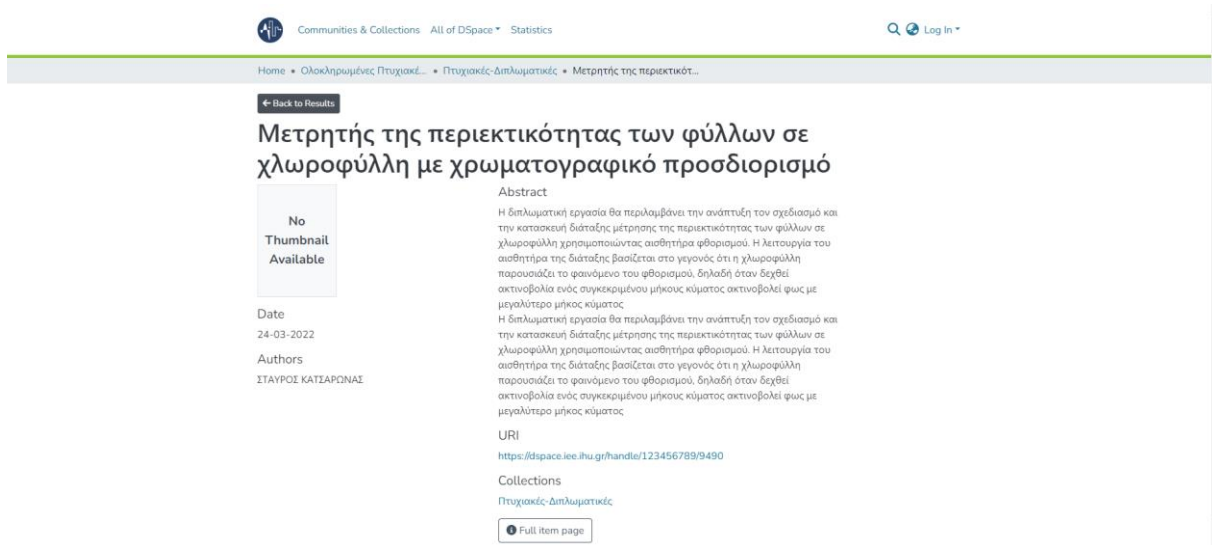


Εικόνα 5.3: Σελίδα δημοσιεύσεων του dspace.iee.ihu.gr

Μέσα από κάθε collection μπορούμε να δούμε τις αντίστοιχες δημοσιεύσεις για πτυχιακές – διπλωματικές, προφίλ, διατριβές.

Κεφάλαιο 5

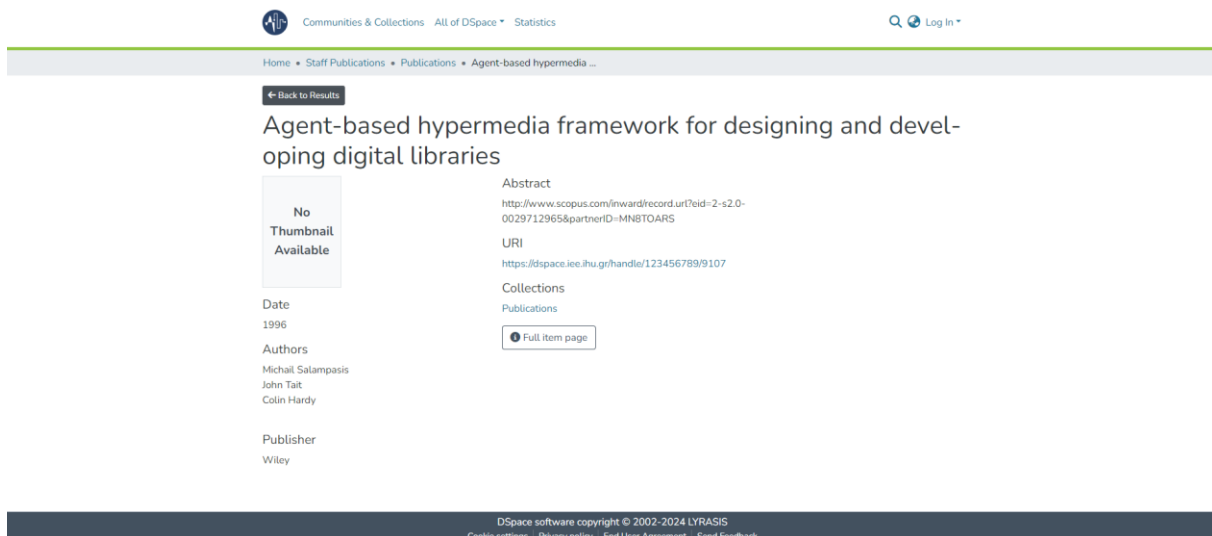
Μια ολοκληρωμένη πτυχιακή – διπλωματική εμφανίζεται στα Ελληνικά ή στα Αγγλικά με τον φοιτητή ως author και τον υπεύθυνο καθηγητή ως advisor, το αρχείο της πτυχιακής καθώς και τον τίτλο και την περιγραφή.



The screenshot shows a DSpace repository page for a thesis. The title is "Μετρητής της περιεκτικότητας των φύλλων σε χλωροφύλλη με χρωματογραφικό προσδιορισμό". The author is listed as ΣΤΑΥΡΟΣ ΚΑΤΑΡΩΝΑΙ. The date is 24-03-2022. The abstract discusses the development and use of a chlorophyll content meter. The page includes a "Full item page" button and a "No Thumbnail Available" message.

Εικόνα 5.4: Σελίδα πτυχιακής – διπλωματικής από το dspace.iee.ihu.gr

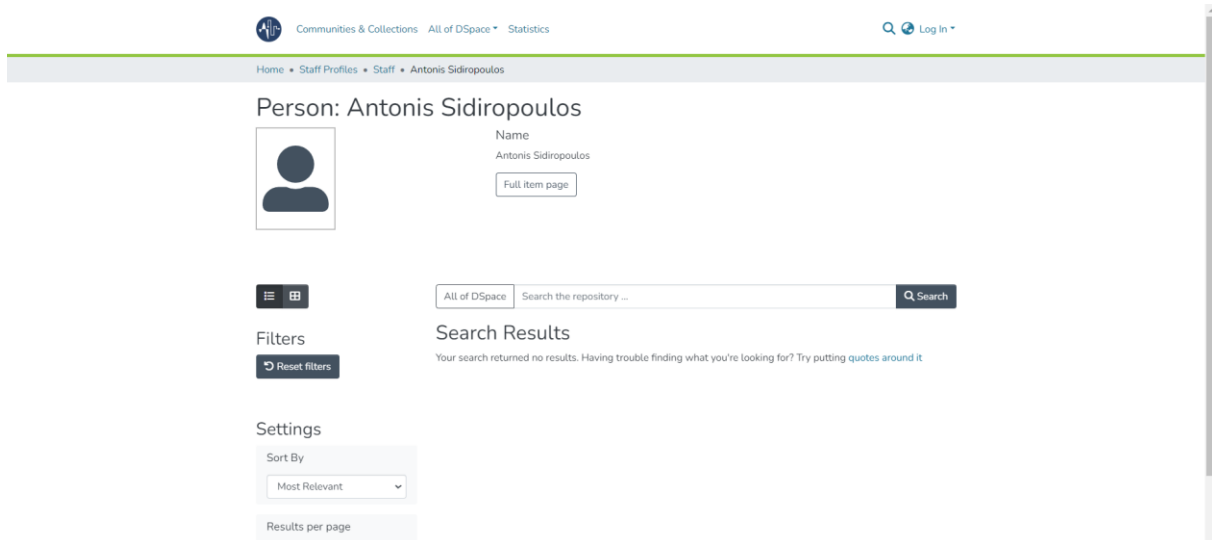
Μια δημοσίευση Publication εμφανίζεται στα Αγγλικά με τους συγγραφείς ως authors, τον publisher, τον τίτλο με την περιγραφή και τον σύνδεσμο της δημοσίευσης.



The screenshot shows a DSpace repository page for a publication. The title is "Agent-based hypermedia framework for designing and developing digital libraries". The authors are Michael Salampasis, John Tait, and Colin Hardy. The publisher is Wiley. The date is 1996. The abstract discusses the development of an agent-based hypermedia framework. The page includes a "Full item page" button and a "No Thumbnail Available" message.

Εικόνα 5.5: Σελίδα δημοσίευσης από το dspace.iee.ihu.gr

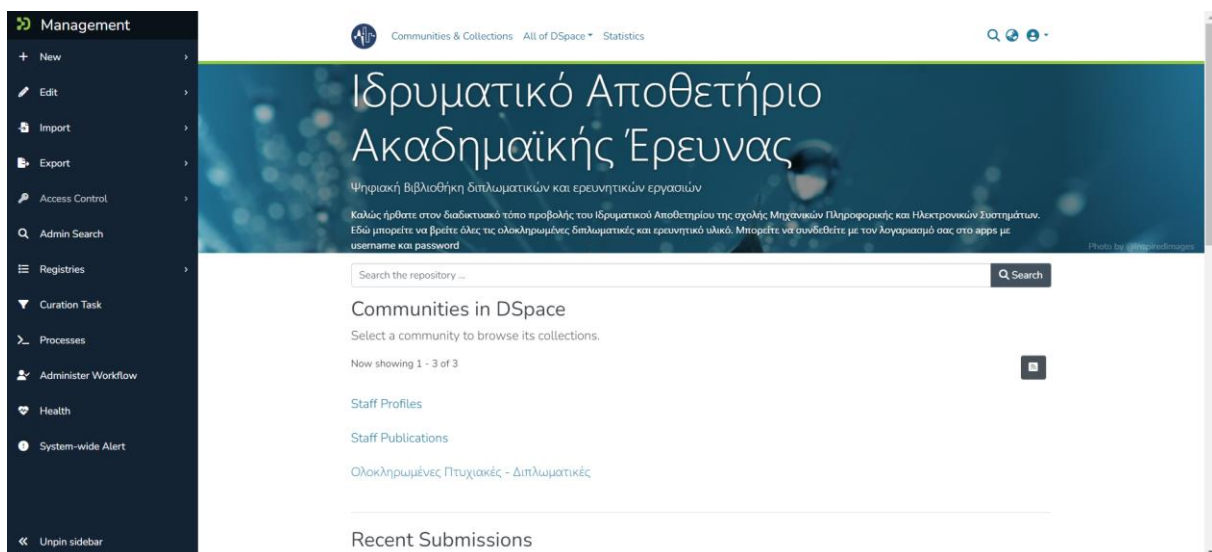
Ένα προφίλ εμφανίζεται στα Αγγλικά με ονοματεπώνυμο και ORCID.



Εικόνα 5.6: Σελίδα προφίλ από το dspace.iee.ihu.gr

5.4 Χρήση ως Διαχειριστής

Ο διαχειριστής του συστήματος έχει περισσότερα δικαιώματα, τα οποία του επιτρέπουν την εύκολη διαχείριση των περιεχομένων (Publications, Person, Πτυχιακές – Διπλωματικές), των χρηστών, των ομάδων και των ρυθμίσεων του συστήματος.



Εικόνα 15: Διαχειριστικό περιβάλλον του dspace.iee.ihu.gr

Το σύστημα του διαχειριστή επιτρέπει την μαζική εισαγωγή και εξαγωγή των μεταδιδόμενων, την εισαγωγή και διαχείριση όλων των περιεχομένων, που υπάρχουν καθώς και έξτρα λειτουργιών που διαθέτει το dspace.

Κεφάλαιο 6ο: Συμπεράσματα και προτάσεις βελτίωσης

Κατά την εκτέλεση της πτυχιακής εργασίας υλοποιήθηκαν όλα τα ζητούμενα με επιτυχία. Η εφαρμογή παρέχει την δυνατότητα διασύνδεσης μέσω του Idap server της σχολής για την αυθεντικοποίηση των χρηστών καθώς και την αυτόματη άντληση των διπλωματικών – πτυχιακών και των διατριβών του ερευνητικού προσωπικού μέσω της μετατροπής των JSON δεδομένων με την βοήθεια της python και την εισαγωγή τους με την βοήθεια της βιβλιοθήκης, που διαθέτει το site pygi.org για το Dspace API.

Στην υλοποίηση αυτής της πτυχιακής έμαθα αρκετά για την γλώσσα python και την σύνδεση με το Dspace Api από την βιβλιοθήκη, η οποία υπάρχει καθώς και την εισαγωγή και εξαγωγή δεδομένων json μορφής. Επίσης, ήταν η πρώτη φορά που ασχολήθηκα με την υπηρεσία Idap και με βοήθησε αρκετά, καθώς με αυτήν αντιμετώπισα το πρόβλημα που είχα με την σύνδεση στον διακομιστή του users και κατάφερα την σύνδεση των χρηστών μέσω του καταλόγου, που προσφέρει. Γενικά, η εγκατάσταση ήταν εύκολη καθώς είχα κάποιες γνώσεις από παλαιότερες εγκαταστάσεις σε εικονικές μηχανές για επαγγελματικούς και μαθησιακούς σκοπούς.

Το κομμάτι, που δυσκολεύτηκα ήταν το κομμάτι της σύνδεσης με το oauth του users και το api του dspace γιατί:

1. Στο oauth υπήρχε πρόβλημα με κάποιες παραμέτρους που έπρεπε να στείλω από το dspace και η υλοποίηση έγινε με την υπηρεσία Idap
2. Στην σύνδεση με το api του dspace καθώς προσπαθούσα να το υλοποιήσω με χρήση php αλλά δεν μπορούσα να πάρω το επιθυμητό CSRF Token που χρειαζόταν οπότε έγινε χρήση της έτοιμης βιβλιοθήκης από την python.

Θα μπορούσαν να βελτιωθούν ή να προστεθούν κάποια πράγματα σε μεταγενέστερη διπλωματική όπως αυτά αναλύονται παρακάτω:

1. Υλοποίηση της συσχέτισης του Person με τα Publications που έχουν αντληθεί. Έχει υλοποιηθεί σε κάθε δημοσίευση, που εισάγεται μέσα στο dspace από το api να ελέγχει, αν οι authors είναι αποθηκευμένοι στο αρχείο relations.json, να παίρνει το uid και να το βάζει στο entity relation.isAuthorOfPublication αλλά μέχρι στιγμής η συγκεκριμένη κίνηση για την συσχέτιση πρέπει να γίνει χειροκίνητα.
2. Υλοποίηση συσχέτισης των ολοκληρωμένων πτυχιακών – διπλωματικών με τους αντίστοιχους advisors καθώς και η δημιουργία ενός νέου collection “Advisors” αν χρειαστεί.
3. Έλεγχος προβλημάτων που υπάρχουν στο api του orcid καθώς κάποια άρθρα – διατριβές έχουν περαστεί στο orcid διπλά ή και τριπλά, σύνδεσμοι που δεν υπάρχουν και doi που δεν υπάρχουν σε κάποια works.
4. Έλεγχος προβλημάτων, τα οποία υπάρχουν στο api του crossref για την εύρεση του doi του work καθώς κάποια μπορεί να μην έχουν authors.
5. Δημιουργία αυτόματης άντλησης από τα αντίστοιχα api κάθε X μέρες.
6. Δημιουργία documentation για την χρήση των endpoints του api για χρήση σε εφαρμογές εντός του τμήματος. (Έχει υλοποιηθεί με python αλλά δεν υπάρχει doc)

Η εφαρμογή αναπτύχθηκε για την έκδοση 7.6 η οποία είναι η released έκδοση που υπάρχει τώρα καθώς έρχεται η έκδοση 8.

Αυτή η πτυχιακή μου έμαθε πόσο σημαντικό είναι το να έχεις σε ένα σημείο τις ατελείωτες ώρες δουλειάς αμέτρητων ανθρώπων καθηγητών και φοιτητών, την οποία με τόσο κόπο και αφοσίωση δίνουν ελεύθερα στο εκπαιδευτικό κοινό, ώστε να μπορούν νέοι επιστήμονες και μη να κάνουν τα πρώτα τους βήματα στην μελέτη. Το dspace και όλες οι πλατφόρμες της ίδιας χρήσης είναι χρήσιμα εργαλεία, που μπορούν να φέρουν το κομμάτι της εκπαίδευσης ένα βήμα παραπέρα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] “Zenodo.” Accessed: May 25, 2024. [Online]. Available: <https://zenodo.org/>
- [2] “arXiv.org e-Print archive.” Accessed: May 25, 2024. [Online]. Available: <https://arxiv.org/>
- [3] “DSpace Home - DSpace.” Accessed: May 25, 2024. [Online]. Available: <https://dspace.lyrasis.org/>
- [4] “ORCID.” Accessed: May 25, 2024. [Online]. Available: <https://orcid.org/>
- [5] “PostgreSQL: The world’s most advanced open source database.” Accessed: May 25, 2024. [Online]. Available: <https://www.postgresql.org/>
- [6] “Welcome to the Digital Library of National Technical University of Athens.” Accessed: May 25, 2024. [Online]. Available: <https://dspace.lib.ntua.gr/xmlui/>
- [7] “Ψηφιακή Βιβλιοθήκη του Πανεπιστημίου Δυτικής Μακεδονίας.” Accessed: May 25, 2024. [Online]. Available: <https://dspace.uowm.gr/xmlui/>
- [8] “Ψηφιακό Αποθετήριο Γεωπονικού Πανεπιστημίου Αθηνών.” Accessed: May 25, 2024. [Online]. Available: <http://dspace.aua.gr/xmlui/>
- [9] “ΨΗΦΙΑΔΑ: Αρχική Σελίδα.” Accessed: May 25, 2024. [Online]. Available: <https://dspace.lib.uom.gr/?locale=el>
- [10] “EOU Library.” Accessed: May 25, 2024. [Online]. Available: <https://library.eou.edu/>
- [11] “Debian -- Debian ‘stretch’ Release Information.” Accessed: May 25, 2024. [Online]. Available: <https://www.debian.org/releases/stretch/>
- [12] “Java | Oracle.” Accessed: May 25, 2024. [Online]. Available: <https://www.java.com/en/>
- [13] “Maven – Welcome to Apache Maven.” Accessed: May 25, 2024. [Online]. Available: <https://maven.apache.org/>
- [14] “Apache Ant - Welcome.” Accessed: May 25, 2024. [Online]. Available: <https://ant.apache.org/>
- [15] “Welcome to Apache Solr - Apache Solr.” Accessed: May 25, 2024. [Online]. Available: <https://solr.apache.org/>
- [16] “Apache Tomcat® - Welcome!” Accessed: May 25, 2024. [Online]. Available: <https://tomcat.apache.org/>
- [17] “UncomplicatedFirewall - Ubuntu Wiki.” Accessed: May 25, 2024. [Online]. Available: <https://wiki.ubuntu.com/UncomplicatedFirewall>
- [18] “Welcome to The Apache Software Foundation!” Accessed: May 25, 2024. [Online]. Available: <https://apache.org/>
- [19] “Node.js — Run JavaScript Everywhere.” Accessed: May 25, 2024. [Online]. Available: <https://nodejs.org/en>
- [20] “Welcome to Python.org.” Accessed: May 25, 2024. [Online]. Available: <https://www.python.org/>

- [21] “dspace-rest-client · PyPI.” Accessed: May 25, 2024. [Online]. Available: <https://pypi.org/project/dspace-rest-client/>
- [22] “PHP: Hypertext Preprocessor.” Accessed: May 25, 2024. [Online]. Available: <https://www.php.net/>
- [23] “You are Crossref - Crossref.” Accessed: May 25, 2024. [Online]. Available: <https://www.crossref.org/>
- [24] “LDAP.com – Lightweight Directory Access Protocol.” Accessed: May 25, 2024. [Online]. Available: <https://ldap.com/>

ΠΑΡΑΡΤΗΜΑ Α: Κώδικες Υλοποίησης

Στο παράρτημα αυτό βρίσκονται οι κώδικες που έχουν υλοποιηθεί βοηθητική ή μη για την ολοκλήρωση της πτυχιακής.

Κώδικας 6.1.Αρχείο update_thesis.php

```
<?php
$authorization=ACCESS_TOKEN;

$additional_headers = array(
    'Accept: application/json',
    'authority: api.thesis.iee.ihu.gr',
    "authorization: $authorization"
);

$teacher_names = [
45=>'Elvira-Maria Arvanitou',
68=>'Christina Volioti'
];

foreach(['finished'] as $s) {
    $fin=[];
    if($s=='finished') {$fin = read_local($s);}
    get_thesis($fin,$s);
    save_local($s,$fin);
}

function get_thesis(&$fin,$status) {
    //$csrf_token = loginAndGetCSRFToken();

    //var_dump($csrf_token);
    //echo '\n';

    global $additional_headers;
    global $teacher_names;

    $stat=6;
    switch ($status) {
        case 'finished': $stat=6; break;
        case 'available': $stat=1; break;
        case 'working': $stat=2; break;
    }

    $ok=0;
    $page=1;
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_HTTPHEADER, $additional_headers);
    $ch2 = curl_init();
    curl_setopt($ch2, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch2, CURLOPT_HTTPHEADER, $additional_headers);

    while(! $ok) {

        $url="https://api.thesis.iee.ihu.gr/theses/?page=$page&tag=all&status
=$stat&search=&order=code&dir=DESC";
        curl_setopt($ch, CURLOPT_URL, $url);
        $output = curl_exec($ch);
        $a = json_decode($output,true);
        if($a['total_pages']<=$page) {$ok=1;}
        else {$page++;}

        foreach($a['results'] as $p){
            if(!isset($fin[$p['code']])) {
```

```

        $url2="https://api.thesis.iee.ihu.gr/thesis/$p[id]";
        print "New theses $p[code] $p[id] ($url2)\n";
        curl_setopt($ch2, CURLOPT_URL, $url2);
        $t = curl_exec($ch2);
        $t2= json_decode($t,true);
        // print_r($t2);
        if(is_array($t2)) {
            $fin[$p['code']]=$t2[0];

            $fin[$p['code']]['teacher1_nameEn']=$teacher_names[$p['teacher1_id']]
;
                $fin[$p['code']]['totalgrade']=null;
                /*$metadata = array(
                    "dc.title" => $fin[$p['code']]['titleEl'],
                    "dc.contributor.author" =>
array($fin[$p['code']]['student1_name'],$fin[$p['code']]['student2_name'],
$fin[$p['code']]['teacher1_name'],$fin[$p['code']]['teacher2_name']),
                    "dc.date.issued" =>
$fin[$p['code']]['graded_date'],
                    "dc.publisher" =>
$fin[$p['code']]['teacher1_name'],
                    "dc.description.abstract" =>
$fin[$p['code']]['abstractEl'],
                );*/
                }
                // $result = uploadPublication($metadata, $csrf_token);
                // echo $result;
            }
        }
        curl_close($ch2);
        curl_close($ch);
    }

function read_local($status) {
    $a=[];
    $file_prefix = "thesis_";
    $file="$file_prefix$status.json";
    if (file_exists($file)) {
        $a = json_decode(file_get_contents($file),true);
    }
    return($a);
}

function save_local($status,$p){
    $file_prefix = "thesis_";
    $file="$file_prefix$status.json";

    copy($file, "$file.bak");
    $fp = fopen($file,'w');
    fwrite($fp, json_encode($p,JSON_UNESCAPED_UNICODE));
    fclose($fp);
}

    curl_setopt($ch2, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch2, CURLOPT_HTTPHEADER, $additional_headers);

    while(! $ok) {

        $url="https://api.thesis.iee.ihu.gr/theses/?page=$page&tag=all&status
=$stat&search=&order=code&dir=DESC";
        curl_setopt($ch, CURLOPT_URL, $url);
        $output = curl_exec($ch);
        $a = json_decode($output,true);
        if($a['total_pages']<=$page) {$ok=1;}
        else {$page++;}
    }
}

```

Κώδικας 6.2 Αρχείο tomcat.service

```
[Unit]
Description=Apache Tomcat 9 Web Application Server
Documentation=https://tomcat.apache.org/tomcat-9.0-doc/index.html
After=network.target

[Service]
Type=forking
User=tomcat
Group=tomcat
PrivateTmp=yes
AmbientCapabilities=CAP_NET_BIND_SERVICE
NoNewPrivileges=true
CacheDirectory=tomcat9
CacheDirectoryMode=750
ProtectSystem=strict
ReadWritePaths=/opt/tomcat/work/Catalina/
ReadWritePaths=/opt/tomcat/webapps/
ReadWritePaths=/opt/tomcat/logs/
ReadWritePaths=/opt/dspace/

Environment=JAVA_HOME=/usr/lib/jvm/java-1.17.0-openjdk-amd64 # εδώ βάζουμε το
παραπάνω αποτέλεσμα της διαδρομής
Environment="CATALINA_HOME=/opt/tomcat9"
Environment="CATALINA_BASE=/opt/tomcat9"
Environment="JAVA_OPTS=-Djava.awt.headless=true"

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

Κώδικας 6.3 Αρχείο server.xml

```
Ψάχνουμε με ctrl+x το "connect port="8080"
Θα βρούμε κάτι σαν:
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />

#Το διαγράφουμε και το αντικαθιστούμε με:
<Connector port="8080" protocol="HTTP/1.1"
minSpareThreads="25"
enableLookups="false"
redirectPort="8443"
connectionTimeout="20000"
disableUploadTimeout="true"
URIEncoding="UTF-8"/>
```

Κώδικας 6.4 αρχείο local.cfg

```
# Ο προορισμός εγκατάστασης του dspace backend
dspace.dir=/opt/dspace
# Εδώ πρέπει να βάλουμε το τελικό url του backend
dspace.server.url = https://dspace.iee.ihu.gr/server # Ένα παράδειγμα
# Εδώ πρέπει να βάλουμε το τελικό url του frontend
dspace.ui.url = https://dspace.iee.ihu.gr # Ένα παράδειγμα
# Εδώ εισάγουμε το όνομα του ψηφιακού αποθετηρίου
```

```

dspace.name = Αποθετήριο Άρθρων & Πτυχιακών
# Εδώ πρέπει να βάλουμε το url από την υπηρεσία του Solr
solr.server = http://localhost:8983/solr
# Εδώ πρέπει να βάλουμε το όνομα χρήστη και τον κωδικό που δημιουργήσαμε στο
προηγούμενο βήμα με την postgres
db.username = dspace
db.password = Password
# Αυτό το μέρος πρέπει να ορίσουμε να url που θέλουμε ο χρήστης να μπορεί να
προηγηθεί
rest.cors.allowed-origins = ${dspace.ui.url}

```

Κώδικας 6.5 Αρχείο apache

```

<VirtualHost *:80>
    ServerName dspace.iee.ihu.gr # Όνομα domain
    #ServerAlias www.dspace.iee.ihu.gr # Όνομα domain

    Redirect / https://dspace.iee.ihu.gr
RewriteEngine on
RewriteCond %{SERVER_NAME} =dspace.iee.ihu.gr
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>

# Όπου dspace.iee.ihu.gr το domain name που διαθέτουμε

<VirtualHost _default_:443>
    # Add your domain here. We've added "my.dspace.edu" as an example
    ServerName dspace.iee.ihu.gr
    #.. setup your host how you want, including log settings...

    # Most installs will need these options enabled to ensure DSpace knows its
hostname and scheme (http or https)
    # Also required to ensure correct sitemap URLs appear in /robots.txt for User
Interface.
    ProxyPreserveHost On
    RequestHeader set X-Forwarded-Proto https

    SSLEngine On
    Include /etc/letsencrypt/options-ssl-apache.conf
    SSLCertificateFile /etc/letsencrypt/live/dspace.iee.ihu.gr/cert.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/dspace.iee.ihu.gr/privkey.pem
    SSLCertificateChainFile /etc/letsencrypt/live/dspace.iee.ihu.gr/chain.pem
    # These Proxy settings are for the backend. They are described in the backend
installation (see above)
    # If you already have the backend running HTTPS, just append the new Proxy
settings below.
    # Proxy all HTTPS requests to "/server" from Apache to Tomcat via AJP connector
    # (In this example: https://my.dspace.edu/server/ will display the REST API)
    #ProxyPass "/server" "ajp://localhost:8009/server"
    #ProxyPassReverse "/server" "ajp://localhost:8009/server"
    #ProxyRequests On
    ProxyPass "/server" "http://dspace.iee.ihu.gr:8080/server"
    ProxyPassReverse "/server" "http://dspace.iee.ihu.gr:8080/server"
    #ProxyPass "/error" "http://dspace.iee.ihu.gr:4000/error"
    #ProxyPassReverse "/error" "http://dspace.iee.ihu.gr:4000/error"

    # [NEW FOR UI:] Proxy all HTTPS requests from Apache to PM2, port 4000
    # NOTE that this proxy URL must match the "ui" settings in your config.prod.yml
    # (In this example: https://my.dspace.edu/ will display the User Interface)
    ProxyPass "/" "http://dspace.iee.ihu.gr:4000"
    ProxyPassReverse "/" "http://dspace.iee.ihu.gr:4000"
</VirtualHost>

```

Κώδικας 6.6 Αρχείο config.prod.yml

```
ui:
  ssl: false
  host: dspace.iee.ihu.gr
  port: 4000
  # NOTE: Space is capitalized because 'namespace' is a reserved string in
  TypeScript
  nameSpace: /
  # The rateLimiter settings limit each IP to a 'max' of 500 requests per 'windowMs'
  (1 minute).
  rateLimiter:
    windowMs: 60000 # 1 minute
    max: 500 # limit each IP to 500 requests per windowMs
  # Trust X-FORWARDED-* headers from proxies (default = true)
  useProxies: true

# The REST API server settings
# NOTE: these settings define which (publicly available) REST API to use. They are
usually
# 'synced' with the 'dspace.server.url' setting in your backend's local.cfg.
rest:
  ssl: true
  host: dspace.iee.ihu.gr
  port: 443
  # NOTE: Space is capitalized because 'namespace' is a reserved string in
  TypeScript
  nameSpace: /server
```

Κώδικας 6.7 Αρχείο angular.json

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "dspace-angular": {
      "projectType": "application",
      "schematics": {
        "@schematics/angular:component": {
          "style": "scss"
        }
      },
      "root": "",
      "sourceRoot": "src",
      "prefix": "ds",
      "architect": {
        "build": {
          "builder": "@angular-builders/custom-webpack:browser",
          "options": {
            "preserveSymlinks": true,
            "customWebpackConfig": {
              "path": "./webpack/webpack.browser.ts",
              "mergeStrategies": {
                "loaders": "prepend"
              }
            }
          },
          "allowedCommonJsDependencies": [
            "serialize",
            "core-js",
            "lodash",
            "jwt-decode",
            "uuid",
            "webfontloader",
            "zone.js"
          ]
        },

```

```

"outputPath": "dist/browser",
"index": "src/index.html",
"main": "src/main.browser.ts",
"polyfills": "src/polyfills.ts",
"tsConfig": "tsconfig.app.json",
"aot": true,
"assets": [
  "src/assets",
  "src/robots.txt"
],
"styles": [
  "src/styles/startup.scss",
  {
    "input": "src/styles/base-theme.scss",
    "inject": false,
    "bundleName": "base-theme"
  },
  {
    "input": "src/themes/custom/styles/theme.scss",
    "inject": false,
    "bundleName": "custom-theme"
  },
  {
    "input": "src/themes/dspace/styles/theme.scss",
    "inject": false,
    "bundleName": "dspace-theme"
  }
],
"scripts": [],
"baseHref": "/"
},
"configurations": {
  "development": {
    "buildOptimizer": false,
    "optimization": false,
    "vendorChunk": true,
    "extractLicenses": false,
    "sourceMap": true,
    "namedChunks": true
  },
  "production": {
    "fileReplacements": [
      {
        "replace": "src/environments/environment.ts",
        "with": "src/environments/environment.production.ts"
      },
      {
        "replace": "src/config/store/devtools.ts",
        "with": "src/config/store/devtools.prod.ts"
      }
    ],
    "optimization": true,
    "outputHashing": "all",
    "namedChunks": false,
    "aot": true,
    "extractLicenses": true,
    "vendorChunk": false,
    "buildOptimizer": true,
    "budgets": [
      {
        "type": "initial",
        "maximumWarning": "3mb",
        "maximumError": "5mb"
      },
      {
        "type": "anyComponentStyle",
        "maximumWarning": "200kb",
        "maximumError": "300kb"
      }
    ]
  }
}

```



```

    "e2e": {
      "builder": "@cypress/schematic:cypress",
      "options": {
        "devServerTarget": "dspace-angular:serve",
        "watch": true,
        "headless": false
      },
      "configurations": {
        "production": {
          "devServerTarget": "dspace-angular:serve:production"
        }
      }
    },
    "server": {
      "builder": "@angular-builders/custom-webpack:server",
      "options": {
        "customWebpackConfig": {
          "path": "./webpack/webpack.prod.ts",
          "mergeStrategies": {
            "loaders": "prepend"
          }
        },
        "outputPath": "dist/server",
        "main": "server.ts",
        "tsConfig": "tsconfig.server.json"
      },
      "configurations": {
        "development": {
          "sourceMap": true,
          "optimization": false
        },
        "production": {
          "sourceMap": false,
          "optimization": true,
          "fileReplacements": [
            {
              "replace": "src/environments/environment.ts",
              "with": "src/environments/environment.production.ts"
            },
            {
              "replace": "src/config/store/devtools.ts",
              "with": "src/config/store/devtools.prod.ts"
            }
          ]
        }
      }
    },
    "serve-ssr": {
      "builder": "@nguniversal/builders:ssr-dev-server",
      "options": {
        "browserTarget": "dspace-angular:build",
        "serverTarget": "dspace-angular:server",
        "port": 4000
      },
      "configurations": {
        "production": {
          "browserTarget": "dspace-angular:build:production",
          "serverTarget": "dspace-angular:server:production"
        }
      }
    },
    "prerender": {
      "builder": "@nguniversal/builders:prerender",
      "options": {
        "browserTarget": "dspace-angular:build:production",
        "serverTarget": "dspace-angular:server:production",
        "routes": [
          "/"
        ]
      }
    }
  }
}

```

```

    ]
  },
  "configurations": {
    "production": {}
  }
},
"cypress-run": {
  "builder": "@cypress/schematic:cypress",
  "options": {
    "devServerTarget": "dspace-angular:serve"
  },
  "configurations": {
    "production": {
      "devServerTarget": "dspace-angular:serve:production"
    }
  }
},
"cypress-open": {
  "builder": "@cypress/schematic:cypress",
  "options": {
    "watch": true,
    "headless": false
  }
},
"lint": {
  "builder": "@angular-eslint/builder:lint",
  "options": {
    "lintFilePatterns": [
      "src/**/*.ts",
      "src/**/*.html",
      "src/**/*.json5"
    ]
  }
}
}
}
},
"cli": {
  "analytics": false,
  "schematicCollections": [
    "@angular-eslint/schematics"
  ]
},
"schematics": {
  "@angular-eslint/schematics:application": {
    "setParserOptionsProject": true
  },
  "@angular-eslint/schematics:library": {
    "setParserOptionsProject": true
  }
}
}
}

```

Κώδικας 6.8 Αρχείο dspace-angular.service

```

[Unit]
Description=PM2 process manager
Documentation=https://pm2.keymetrics.io/
After=network.target

[Service]
Type=forking
User=katsanis
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity

```

```

Environment=PATH=/home/katsanis/.nvm/versions/node/v18.20.2/bin:/usr/local/bin:/u
sr/bin:/bin:/usr/local/games:/usr/game>Environment=PM2_HOME=/home/katsanis/.pm2
PIDFile=/home/katsanis/.pm2/pm2.pid
Restart=on-failure

ExecStart=/home/katsanis/.nvm/versions/node/v18.20.2/lib/node_modules/pm2/bin/pm2
resurrect
ExecReload=/home/katsanis/.nvm/versions/node/v18.20.2/lib/node_modules/pm2/bin/pm
2 reload all
ExecStop=/home/katsanis/.nvm/versions/node/v18.20.2/lib/node_modules/pm2/bin/pm2
kill

[Install]
WantedBy=multi-user.target

```

Κώδικας 6.9 Αρχείο local.cfg Για ενεργοποίηση ρυθμίσεων orcid

```

# These settings can be found in your authority.cfg (or could be added to local.cfg)
choices.plugin.dc.contributor.author = SolrAuthorAuthority
choices.presentation.dc.contributor.author = authorLookup
authority.controlled.dc.contributor.author = true
authority.author.indexer.field.1=dc.contributor.author

event.dispatcher.default.consumers = authority, versioning, discovery, eperson

<!-- This bean & alias are commented out by default. Simply uncomment them -->
<alias name="OrcidSource" alias="AuthoritySource"/>
<bean
                                name="OrcidSource"
class="org.dspace.authority.orcid.Orcidv3SolrAuthorityImpl">
  <property name="clientId" value="{orcid.application-client-id}" />
  <property name="clientSecret" value="{orcid.application-client-secret}" />
  <property name="OAUTHUrl" value="{orcid.token-url}" />
  <property name="orcidRestConnector" ref="orcidRestConnector"/>
</bean>

<!-- Also uncomment Orcidv3AuthorityValue in the list of supported types below.
The other settings in this AuthorityTypes bean can be left as-is. -->
<bean name="AuthorityTypes" class="org.dspace.authority.AuthorityTypes">
  <property name="types">
    <list>
      <bean class="org.dspace.authority.orcid.Orcidv3AuthorityValue"/>
      <bean class="org.dspace.authority.PersonAuthorityValue"/>
    </list>
  </property>
  ...
</bean>

```

Κώδικας 6.10 Αρχείο update_upblications.py

```

# This software is licenced under the BSD 3-Clause licence
# available at https://opensource.org/licenses/BSD-3-Clause
# and described in the LICENCE file in the root of this project

"""
Example Python 3 application using the dspace.py API client library to create
some resources in a DSpace 7 repository.
"""

from dspace_rest_client.client import DSpaceClient
from dspace_rest_client.models import Community, Collection, Item, Bundle,
Bitstream
import json

```

```

import requests
from requests_oauthlib import OAuth2Session
from oauthlib.oauth2 import BackendApplicationClient
from bs4 import BeautifulSoup
import re

url = 'https://dspace.iee.ihu.gr/server/api'
username = 'USERNAME'
password = 'PASSWORD'
client_id = 'ORCID-CLIENT-ID'
client_secret = 'ORCID-CLIENT-SECRET'
scope = "/read-public" # Adjust scope as needed
titles = "/home/katsanis/thesis/dspace-rest-python/saves/txt/title.txt" #for
restrict duplicates posts
savedpubs = "/home/katsanis/thesis/dspace-rest-python/saves/txt/savedpubs.txt" #
all saved pubs
savedProfiles = "/home/katsanis/thesis/dspace-rest-
python/saves/txt/savedperson.txt" # saved orcid profiles
orcid_ids_file = "/home/katsanis/thesis/dspace-rest-python/saves/profiles.txt"
#get_orcidid(filename)

def doi_req(doi):
    # Base URL of the CrossRef API
    base_url = "https://api.crossref.org/works/"

    # Construct the full URL with the provided DOI
    url = base_url + doi

    # Make a GET request to the CrossRef API
    response = requests.get(url)

    # Check if the request was successful (status code 200)
    if response.status_code == 200:
        # Parse the JSON response
        data = response.json()

        # Extract publisher and authors from the response
        publisher = data["message"]["publisher"] if "publisher" in data["message"]
    else ""

        authors = [author["given"] + " " + author["family"] for author in
data["message"]["author"] if "given" in author and "family" in author] if "author"
in data["message"] else []

        orcid_authors = [(author.get("ORCID") or "") for author in
data["message"]["author"] if "ORCID" in author] if "author" in data["message"] else
[]

        return publisher, authors, orcid_authors
    else:
        # Print an error message if the request fails
        print(f"Failed to fetch metadata for DOI: {doi}")
        return None, None, None

def get_orcidid(filename):
    last_position = 0 # Default value if last_position.txt is empty or invalid
    try:
        with open("/home/katsanis/thesis/dspace-rest-python/last_position.txt",
'r') as position_file:
            last_position_str = position_file.readline().strip()
            if last_position_str:
                last_position = int(last_position_str)
    except (IOError, ValueError):
        pass

    with open(filename, 'r') as file:
        # Move the file pointer to the last processed position

```

```

        file.seek(last_position)
        # Read the next line
        line = file.readline().strip()
        if line:
            # Save the current file position for next time
            current_position = file.tell()
            with open("last_position.txt", 'w') as position_file:
                position_file.write(str(current_position))

def get_access_token(client_id, client_secret, scope):
    client = BackendApplicationClient(client_id=client_id)
    oauth = OAuth2Session(client=client)
    token_url = 'https://orcid.org/oauth/token'
    token = oauth.fetch_token(token_url=token_url, client_id=client_id,
client_secret=client_secret, scope=scope)
    return token['access_token']

def get_publications(orcid_id, access_token):
    url = f"https://pub.orcid.org/v3.0/{orcid_id}/works"
    headers = {
        'Accept': 'application/json',
        'Authorization': f'Bearer {access_token}'
    }
    response = requests.get(url, headers=headers)
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Error fetching publications: {response.status_code}")
        return None

def get_person(orcid_id, access_token):
    url = f"https://pub.orcid.org/v3.0/{orcid_id}/person"
    headers = {
        'Accept': 'application/json',
        'Authorization': f'Bearer {access_token}'
    }
    response = requests.get(url, headers=headers)
    if response.status_code == 200:
        return response.json()
    else:
        print(f"Error fetching person: {response.status_code}")
        return None

def orcid_ids(id, filename):
    with open(filename, 'a') as file:
        file.write(str(id) + '\n')

# Instantiate DSpace client
# Note the 'fake_user_agent' setting here -- this will set a string like the
following, to get by Cloudfront:
# Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/39.0.2171.95 Safari/537.36
# The default is to *not* fake the user agent, and instead use the default of DSpace
Python REST Client.
# To specify a custom user agent, set the USER_AGENT env variable and leave/set
fake_user_agent as False
d = DSpaceClient(api_endpoint=url, username=username, password=password,
fake_user_agent=True)

# Authenticate against the DSpace client
authenticated = d.authenticate()
if not authenticated:
    print(f'Error logging in! Giving up.')
    exit(1)

def save_ids(id, filenames):
    with open(filenames, 'a') as file:
        file.write(str(id) + '\n')

```

```

def check_id(id, filenames):
    try:
        with open(filenames, 'r') as file:
            for line in file:
                if str(id) == line.strip():
                    return True
            return False
    except FileNotFoundError:
        print("File not found. No IDs saved yet.")
        return False

def save_orcid(orcid, filenames):
    with open(filenames, 'a') as file:
        file.write(str(orcid) + '\n')

def check_orcid(orcid, filenames):
    try:
        with open(filenames, 'r') as file:
            for line in file:
                if str(orcid) == line.strip():
                    return True
            return False
    except FileNotFoundError:
        print("File not found. No IDs saved yet.")
        return False

def save_orcid_title(orcid, filenames):
    with open(filenames, 'a') as file:
        file.write(str(orcid) + '\n')

def check_orcid_title(orcid, filenames):
    try:
        with open(filenames, 'r') as file:
            for line in file:
                if str(orcid) == line.strip():
                    return True
            return False
    except FileNotFoundError:
        print("File not found. No IDs saved yet.")
        return False

def check_id_to_file(id):
    try:
        file = open(f"/home/katsanis/thesis/dspace-rest-
python/saves/json/relations.json")
        savedPerson = json.load(file)
        for person in savedPerson:
            if (person.get("orcid") == id):
                return person.get("id")
        return ""
    except FileNotFoundError:
        print("File not found. No orcid found in files.")
        return ""

def save_profile(orcid):
    p = open(f'/home/katsanis/thesis/dspace-rest-python/saves/json/{orcid}-
person.json')

    webPerson = json.load(p)
    fname = webPerson["name"]["given-names"]["value"]
    lname = webPerson["name"]["family-name"]["value"]
    try:
        emails = webPerson["emails"]["email"]
        email = emails[0]['email']
    except:
        print("Email not visible")

```

```

        email = ""
        name = fname+ " " +lname

        item_data = {
            "name": name,
            "metadata": {
                "dc.title": [
                    {
                        "value": name,
                        "language": "en",
                        "authority": None,
                        "confidence": -1
                    }
                ],
                "person.identifier.orcid": [
                    {
                        "value": orcid,
                        "language": "en",
                        "authority": None,
                        "confidence": -1
                    }
                ],
                "person.email": [
                    {
                        "value": email,
                        "language": "en",
                        "authority": None,
                        "confidence": -1
                    }
                ],
                "dspace.object.owner": [
                    {
                        "value": name,
                        "language": "en",
                        "authority": None,
                        "confidence": -1
                    }
                ],
                "dspace.entity.type": [
                    {
                        "value": "Person"
                    }
                ]
            },
            "inArchive": True,
            "discoverable": True,
            "withdrawn": False,
            "type": "Person"
        }
        # The creation process for an item is a bit different, because they're typically
        # a bit more
        # complex than a container object. However, they're all DSOs at the end of the
        # day so we
        # could make usage fully consistent if we wanted.
        #
        # Instantiate an empty Item, passing the full API resource data to the
        # constructor.
        item = Item(item_data)
        # Create the item using the new collection as the parent container.
        # Note that we're passing the full Item object here, not just the dict data
        # (though it will be serialised to a dict before the final POST)

        new_item = d.create_item(parent="71748b8c-e8ce-4a53-ae2f-9c8ed85c6e19",
            item=item)
        if isinstance(new_item, Item) and new_item.uuid is not None:
            print(f'New Person created! Handle: {new_item.handle}')
            url = "http://orcid.org/"
            data = {

```

```

        "name": name,
        "orcid": url + orcid,
        "itemname": new_item.name,
        "uuid": new_item.uuid,
        "handle": new_item.handle,
        "id": new_item.id
    }

    with open(f"/home/katsanis/thesis/dspace-rest-python/saves/json/{orcid}-
relations.json", "w") as json_file:
        json.dump(data, json_file, indent=4)

    try:
        with open("/home/katsanis/thesis/dspace-rest-
python/saves/json/relations.json", 'r') as file:
            existing_data = json.load(file)
        except FileNotFoundError:
            existing_data = []

        # Append new response to existing data list
        existing_data.append(data)

        # Write combined data back to JSON file
        with open("/home/katsanis/thesis/dspace-rest-
python/saves/json/relations.json", 'w') as file:
            json.dump(existing_data, file, indent=4)
    else:
        print(f'Error! Giving up.')
        exit(1)
    # Add a single metadata field+value to the item (PATCH operation)
    #updated_item = d.add_metadata(dso=new_item, field='dc.description.abstract',
value='Added abstract to an existing item',
    # language='en', authority=None, confidence=-1)
    # Create a new ORIGINAL bundle
    # See https://github.com/DSpace/RestContract/blob/main/bundles.md
    new_bundle = d.create_bundle(parent=new_item, name='ORIGINAL')
    if isinstance(new_bundle, Bundle) and new_bundle.uuid is not None:
        print(f'New bundle created! UUID: {new_bundle.uuid}')
    else:
        print(f'Error! Giving up.')
        exit(1)
    p.close()

def submit(orcid_file):
    f = open(f"/home/katsanis/thesis/dspace-rest-
python/saves/json/{orcid_file}.json')

    data = json.load(f)
    works = data.get("group", [])

    for obj in works:
        print("-----")
        for work_summary in obj.get("work-summary", []):
            code = work_summary.get("put-code", "")
            title = work_summary.get("title", {}).get("title", {}).get("value", "")

            journal_title = work_summary.get("journal-title", {})
            if journal_title:
                titlealt = journal_title.get("value", "")
            else:
                titlealt = "Unknown"

            try:
                doi = next((id_data["external-id-value"] for id_data in
work_summary.get("external-ids", {}).get("external-id", []) if id_data["external-
id-type"] == "doi"), None)
            except:
                print("Doi not found")

```

```

        doi = "Unknown"
    try:
        url = work_summary.get("url", {}).get("value", "")
    except:
        print("Null url")
        url = "Unknown"
    try:
        publication_date = work_summary.get("publication-date",
        {}).get("year", {}).get("value", "")
    except:
        print("Publication doesnt exists")
        publication_date = ""
    work_type = work_summary.get("type", "")

    if doi:
        publisher, authors, orcid = doi_req(doi)

    orcid_ids = []
    if orcid:
        print (orcid)
        for url in orcid:
            orcid_ids.append(check_id_to_file(url))
            print(orcid_ids)
    else:
        orcid_ids.append("")
        print("orcid Not found")

    if check_id(code, savedpubs):
        print(f"Code {code} exists.")
    else:
        if check_orcid_title(title, titles):
            print(f"{title} from {orcid_file} Exists with another id and
doi")
        else:
            save_orcid_title(title, titles)
            print(f"{title} from {orcid_file} saved")
            print(f"Code {code} does not exist.")
            save_ids(code, savedpubs)

    item_data = {
        "name": title,
        "metadata": {
            "dc.contributor.author": authors,
            "dc.publisher": [
                {
                    "value": publisher,
                    "language": "en",
                    "authority": None,
                    "confidence": -1
                }
            ],
            "dc.title": [
                {
                    "value": title,
                    "language": "en",
                    "authority": None,
                    "confidence": -1
                }
            ],
            "dc.title.alternative": [
                {
                    "value": titlealt,
                    "language": "en",
                    "authority": None,
                    "confidence": -1
                }
            ],
            "dc.type": [

```

```

        {
            "value": work_type,
            "language": "en",
            "authority": None,
            "confidence": -1
        }
    ],
    'dc.description.abstract': [
        {
            "value": url,
            "language": 'en',
            "authority": None,
            "confidence": -1
        }
    ],
    'dc.date.issued': [
        {
            "value": publication_date,
            'authority': None,
            'confidence': -1,
            'place': 0
        }
    ],
    "relation.isAuthorOfPublication": orcid_ids
},
"inArchive": True,
"discoverable": True,
"withdrawn": False,
"type": "item"
}
# The creation process for an item is a bit different, because
they're typically a bit more
# complex than a container object. However, they're all DSOs at the
end of the day so we
# could make usage fully consistent if we wanted.
#
# Instantiate an empty Item, passing the full API resource data to
the constructor.
item = Item(item_data)
# Create the item using the new collection as the parent container.
# Note that we're passing the full Item object here, not just the
dict data
# (though it will be serialised to a dict before the final POST)
new_item = d.create_item(parent="46a6b6e7-cb15-4c53-97a5-
af98e92b1e8e", item=item)
new_item = d.create_item(parent="e7eb5f53-9881-4081-838a-
bd180d1bfd7f", item=item)
if isinstance(new_item, Item) and new_item.uuid is not None:
    print(f'New item created! Handle: {new_item.handle}')
else:
    print(f'Error! Giving up.')
    exit(1)
# Add a single metadata field+value to the item (PATCH operation)
#updated_item = d.add_metadata(dso=new_item,
field='dc.description.abstract', value='Added abstract to an existing item',
# language='en', authority=None,
confidence=-1)
# Create a new ORIGINAL bundle
# See https://github.com/Dspace/RestContract/blob/main/bundles.md
new_bundle = d.create_bundle(parent=new_item, name='ORIGINAL')
if isinstance(new_bundle, Bundle) and new_bundle.uuid is not None:
    print(f'New bundle created! UUID: {new_bundle.uuid}')
else:
    print(f'Error! Giving up.')
    exit(1)

f.close()

```

```

def check_prof():
    orcid = open(orcid_ids_file, 'r')
    for line in orcid:
        orcid_id = line.strip()
        if check_orcid(orcid_id, savedProfiles):
            print(f"{orcid_id} Exists")
        else:
            save_profile(orcid_id)
            save_orcid(orcid_id, savedProfiles)
            print(f"{orcid_id} saved")

check_prof()
orcid = open(orcid_ids_file, 'r')
for line in orcid:
    orcid_id = line.strip()
    access_token = get_access_token(client_id, client_secret, scope)
    if access_token:
        person = get_person(orcid_id, access_token)
        publications = get_publications(orcid_id, access_token)
        if person:
            pers = json.dumps(person, indent=4)

            with open(f"/home/katsanis/thesis/dspace-rest-
python/saves/json/{orcid_id}-person.json", "w") as file:
                file.write(pers)

            if publications:
                #print(json.dumps(publications, indent=4)) # Print publications JSON
data
                # Put together some basic Item data.
                # (See: https://github.com/DSpace/RestContract/blob/main/items.md)
                pubs = json.dumps(publications, indent=4)

                with open(f"/home/katsanis/thesis/dspace-rest-
python/saves/json/{orcid_id}.json", "w") as file:
                    file.write(pubs)

                submit(orcid_id)

            else:
                print("No publications found or error occurred.")
        else:
            print("Failed to obtain access token.")

print(f'All finished with example data creation. Visit your test repository to
review created objects')

```

Κώδικας 6.11 Αρχείο update_thesis.py

```

# This software is licenced under the BSD 3-Clause licence
# available at https://opensource.org/licenses/BSD-3-Clause
# and described in the LICENCE file in the root of this project

"""
Example Python 3 application using the dspace.py API client library to create
some resources in a DSpace 7 repository.
"""

from dspace_rest_client.client import DSpaceClient
from dspace_rest_client.models import Community, Collection, Item, Bundle,
Bitstream
import json
import requests
import os

```

```

f = open ('/home/katsanis/thesis/thesis_finished.json')
data = json.load(f)

url = 'https://dspace.iee.ihu.gr/server/api'
username = 'USERNAME'
password = 'PASSWORD'

# Instantiate DSpace client
# Note the 'fake_user_agent' setting here -- this will set a string like the
following, to get by Cloudfront:
# Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/39.0.2171.95 Safari/537.36
# The default is to *not* fake the user agent, and instead use the default of DSpace
Python REST Client.
# To specify a custom user agent, set the USER_AGENT env variable and leave/set
fake_user_agent as False
d = DSpaceClient(api_endpoint=url, username=username, password=password,
fake_user_agent=True)

# Authenticate against the DSpace client
authenticated = d.authenticate()
if not authenticated:
    print(f'Error logging in! Giving up.')
    exit(1)

def get_pdf(id):
    authorization = 'THESIS API AUTHORIZATION TOKEN'
    url = f'https://api.thesis.iee.ihu.gr/downloadthesisfile/{id}'
    headers = {
        'Accept': 'application/json',
        'authority': 'api.thesis.iee.ihu.gr',
        'authorization': f'{authorization}'
    }
    response = requests.get(url, headers=headers)
    if response.status_code == 200:
        with open(f"/home/katsanis/thesis/dspace-rest-
python/saves/pdf/thesis_{id}.pdf", 'wb') as f:
            f.write(response.content)
            print("PDF downloaded successfully")
    else:
        print(f"Error fetching PDF: {response.status_code}")

def save_ids(id, filename):
    with open(filename, 'a') as file:
        file.write(str(id) + '\n')

def check_id(id, filename):
    try:
        with open(filename, 'r') as file:
            for line in file:
                if str(id) == line.strip():
                    return True
            return False
    except FileNotFoundError:
        print("File not found. No IDs saved yet.")
        return False

# Example usage
filenagme = "/home/katsanis/thesis/dspace-rest-python/saves/txt/ids.txt"

# Put together some basic Item data.
# (See: https://github.com/DSpace/RestContract/blob/main/items.md)
for obj in data:
    if check_id(data[obj]['id'], filenagme):
        print(f"ID {data[obj]['id']} exists.")
    else:
        print(f"ID {data[obj]['id']} does not exist.")
        save_ids(data[obj]['id'], filenagme)

```

```

print(data[obj]['id'])
print(data[obj]['graded_date'])
print(data[obj]['file'])
print(data[obj]['student1_name'])
print(data[obj]['student1_am'])
print(data[obj]['student2_name'])
print(data[obj]['student2_am'])
print(data[obj]['teacher1_name'])
print(data[obj]['teacher2_name'])
print(data[obj]['teacher3_name'])
print(data[obj]['student2_am'])
print(data[obj]['titleEn'])
print(data[obj]['abstractEn'])
print(data[obj]['titleEl'])
print(data[obj]['abstractEl'])
print(data[obj]['statustext'])
name = data[obj]['titleEl']
id = data[obj]['id']
item_data = {
    "name": data[obj]['titleEl'],
    "metadata": {
        "dc.contributor.author": [
            {
                "value": data[obj]['student1_name'],
                "language": "el",
                "authority": None,
                "confidence": -1
            }
        ],
        "dc.contributor.advisor": [
            {
                "value": data[obj]['teacher1_name'],
                "language": "el",
                "authority": None,
                "confidence": -1
            }
        ],
        "dc.relation.ispartofseries": [
            {
                "value": data[obj]['id'],
                "authority": None,
                "confidence": -1
            }
        ],
        "dc.title": [
            {
                "value": data[obj]['titleEl'],
                "language": "el",
                "authority": None,
                "confidence": -1
            },
            {
                "value": data[obj]['titleEn'],
                "language": "en",
                "authority": None,
                "confidence": -1
            }
        ],
        "dc.type": [
            {
                "value": "Thesis",
                "language": "en",
                "authority": None,
                "confidence": -1
            }
        ],
        'dc.description.abstract': [
            {

```

```

        "value": data[obj]['abstractEl'],
        "language": 'el',
        "authority": None,
        "confidence": -1
    },
    {
        "value": data[obj]['abstractEn'],
        "language": 'en',
        "authority": None,
        "confidence": -1
    }
],
'dc.date.issued': [
    {
        "value": data[obj]['graded_date'],
        'authority': None,
        'confidence': -1,
        'place': 0
    }
]
},
"inArchive": True,
"discoverable": True,
"withdrawn": False,
"type": "item"
}
# The creation process for an item is a bit different, because they're
typically a bit more
# complex than a container object. However, they're all DSOs at the end of
the day so we
# could make usage fully consistent if we wanted.
#
# Instantiate an empty Item, passing the full API resource data to the
constructor.
item = Item(item_data)
# Create the item using the new collection as the parent container.
# Note that we're passing the full Item object here, not just the dict data
# (though it will be serialised to a dict before the final POST)
new_item = d.create_item(parent="8b1dc0fe-d79d-41ae-8579-4ea4dcbab2fa",
item=item)
if isinstance(new_item, Item) and new_item.uuid is not None:
    print(f'New item created! Handle: {new_item.handle}')
else:
    print(f'Error! Giving up.')
    exit(1)
# Add a single metadata field+value to the item (PATCH operation)
#updated_item = d.add_metadata(dso=new_item,
field='dc.description.abstract', value='Added abstract to an existing item',
# language='en', authority=None, confidence=-
1)

# Create a new ORIGINAL bundle
# See https://github.com/DSpace/RestContract/blob/main/bundles.md
new_bundle = d.create_bundle(parent=new_item, name='ORIGINAL')
if isinstance(new_bundle, Bundle) and new_bundle.uuid is not None:
    print(f'New bundle created! UUID: {new_bundle.uuid}')
else:
    print(f'Error! Giving up.')
    exit(1)

get_pdf(id)
# Create and upload a new bitstream using the LICENSE.txt file in this
project
# Set bitstream metadata
# See https://github.com/DSpace/RestContract/blob/main/bitstreams.md
bitstream_metadata = {
    'dc.description':
        [
            {

```

```

        "value": "Αρχείο Διατριβής",
        'language': 'el',
        'authority': None,
        'confidence': -1,
        'place': 0
    },
    {
        "value": "Thesis File",
        'language': 'en',
        'authority': None,
        'confidence': -1,
        'place': 0
    }
]

# # Set the mime type (using mimetypes.guess_type is recommended for real
uploads if you don't want to set manually)
file_mime = 'mimetypes.guess_type'
# # Set a better file name for our test upload
file_name = f'{name}.pdf'
# # Create the bitstream and upload the file
new_bitstream = d.create_bitstream(bundle=new_bundle, name=file_name,
path=f'/home/katsanis/thesis/dspace-rest-python/saves/pdf/thesis_{id}.pdf',
mime=file_mime, metadata=bitstream_metadata)
if isinstance(new_bitstream, Bitstream) and new_bitstream.uuid is not None:
    print(f'New bitstream created! UUID: {new_bitstream.uuid}')
else:
    print(f'Error! Giving up.')
    exit(1)

print("PDF deleted from local space")
file_path = f"/home/katsanis/thesis/dspace-rest-
python/saves/pdf/thesis_{id}.pdf"
try:
    os.remove(file_path)
    print(f"{file_path} removed")
except FileNotFoundError:
    print(f"{file_path} file not found")

f.close()
print(f'All finished with example data creation. Visit your test repository to
review created objects')

```

Κώδικας A.12 Αρχείο για την ενεργοποίηση του ldap στο local.cfg

Αρχείο παραμετροποίησης LDAP

```

# Ενεργοποίηση του ldap module
authentication-ldap.enable = true

# Ενεργοποίηση αυτόματης εγγραφής χρήστη
authentication-ldap.autoregister = true

# Ldap Server
authentication-ldap.provider_url = ldap://it.teithe.gr:389/

# Συστοιχία του id με του uid
authentication-ldap.id_field = uid

# Διαδρομή αναζήτησης χρηστών
authentication-ldap.search_context = ou=people\,dc=it\,dc=teithe\,dc=gr

# Συστοιχία του email με το mail
authentication-ldap.email_field = mail

```

```
# Συστοιχία του surname με το sn (Επώνυμο)
authentication-ldap.surname_field = sn

# Συστοιχία του givenname (Όνομα)
authentication-ldap.givenname_field = givenName

# Αναζήτηση με scope subtree
authentication-ldap.search_scope = 2

# Ανώνυμη αναζήτηση
authentication-ldap.search.anonymous = true

# Συστοιχία group χρηστών
authentication-ldap.login.groupmap.1 = ou=student:Student
authentication-ldap.login.groupmap.2 = ou=staff:Staff
```