

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ ΚΑΙ ΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ

(Εργασία 2η)

Συντάκτες:	ΣΚΑΡΠΕΤΗ Αθηνά Π08133
	ΣΥΓΓΟΥΡΟΓΛΟΥ Γεώργιος Π08138
	ΔΩΔΕΚΑΤΟΣ Παναγιώτης Π07037

Περιεχόμενα

1.Εκφώνηση.....	3
2.Επίλυση εφαρμογής TCP Client - Server.....	3
2.1.Περιγραφή της λύσης.....	3
2.2.Client.....	4
1.Τρόπος λειτουργίας.....	4
2.Τρόπος δόμησης πακέτων.....	4
2.3.Server.....	5
1.Τρόπος λειτουργίας.....	5
2.Τρόπος δόμησης πακέτων.....	6
3.Σχέδιο βάσης δεδομένων.....	8
4.Αναφορές.....	9

Το παρών έγγραφο δημιουργήθηκε με χρήση του Libre Office 4.3 και επισυνάπτεται επίσης ως pdf. Η εργασία κατασκευάστηκε με χρήση του Netbeans IDE 8.0.2 με χρήση JDK 1.8u25. Ο java builder που χρησιμοποιούμε είναι το maven. Το πρόγραμμα σχεδίασης της βάσης δεδομένων που χρησιμοποιήσαμε ήταν το Visual Paradigm For Uml 11.2. Το λειτουργικό σύστημα στο οποίο πραγματοποιήθηκε και δοκιμάστηκε η εργασία ήταν το Debian 8 Linux.

1. Εκφώνηση

Στόχοι άσκησης: TCP sockets, δημιουργία TCP socket, δημιουργία πρωτοκόλλου, απομακρυσμένη ανάγνωση/εγγραφή αρχείων, χειρισμός ασφαμάτων σύνδεσης.

Σε αυτή την άσκηση θα επεκτείνετε την 1 Άσκηση ώστε να δημιουργήσετε μία client-server εφαρμογή, η οποία θα υλοποιεί μέσα από tcp socket, κάποιες από τις μεθόδους που ορίσατε στην 1 Άσκηση.

Επιπλέον, ένα δεύτερο, ανεξάρτητο βήμα θα είναι ο σχεδιασμός του σχήματος της πραγματικής βάσης δεδομένων που θα χρησιμοποιήσετε στο 3 και τελευταίο μέρος της άσκησης.

Αναλυτικά Βήματα:

1.1. Δημιουργία TCP Server.

1. Υλοποίηση μεθόδου νέων ταινιών

2. Υλοποίηση μεθόδου αναζήτησης προβολών μίας ταινίας

1.2. Δημιουργία σχήματος Βάσης Δεδομένων.

2. Επίλυση εφαρμογής TCP Client - Server

2.1. Περιγραφή της λύσης

Για την υλοποίηση της εργασίας κατασκευάσαμε δύο ανεξάρτητα project. Ο java builder που χρησιμοποιήσαμε ήταν το maven. Και τα δύο project έχουν ως εξαρτήσεις δύο βιβλιοθήκες. Οι βιβλιοθήκες αυτές είναι η log4j και η gson. Η πρώτη βοηθά στη διαχείριση της καταγραφής σε αρχεία log και η δεύτερη στο μετασχηματισμό αντικειμένων σε json και ανάποδα. Ακόμη, τα δύο project έχουν αποθηκευμένες κάποιες ρυθμίσεις τους σε εξωτερικά αρχεία properties.

Τα δύο αυτά προγράμματα ανταλλάσσουν μεταξύ τους δεδομένα. Το πρωτόκολλο επικοινωνίας μεταξύ των δύο προγραμμάτων είναι η γλώσσα json. Έτσι, τα προγράμματα ανταλλάσσουν μηνύματα σε μορφή json. Μηνύματα τα οποία μετατρέπονται εντός της κάθε εφαρμογής σε αντικείμενα.

Στην προσπάθεια μας να μοντελοποιήσουμε την επικοινωνία των προγραμμάτων ώστε να ταιριάζει με αυτή του http, ορίσαμε κάποιες αρχές στον τρόπο επικοινωνίας. Μία αρχή είναι πως κάθε request θα εκτελεί μια συγκεκριμένη ενέργεια στο server η οποία θα ακολουθείτε με ένα response. Μετά το response η επικοινωνία μεταξύ των δύο προγραμμάτων θα τερματίζεται. Στο request θα ορίζεται η μέθοδος που ο client επιθυμεί να εκτελέσει. Οι μέθοδοι που ορίζουμε είναι οι SEARCH, CREATE, READ, UPDATE, DELETE (SCRUD) καθώς επίσης και οι LOGIN, LOGOUT. Ακόμη, κάθε request θα περιέχει κάποια δεδομένα. Το ίδιο

ισχύει και για τα responses. Τέλος, για λόγους ασφάλειας προσθέσαμε την έννοια του sessionId. Πρόκειται για έναν αριθμό που παίρνει κάποιος χρήστης μετά από επιτυχής σύνδεση (login). Ο αριθμός αυτός ακολουθεί κάθε επόμενο request του και τον ταυτοποιεί στο server. Έτσι, για κάθε request ο server γνωρίζει ποιος χρήστης το έκανε και του δίνει τα ανάλογα δικαιώματα του ρόλου στον οποίο ανήκει.

Παρακάτω ακολουθεί η επεξήγηση του τρόπου δόμησης των πακέτων και της λειτουργίας του client και του server.

2.2. Client

1. Τρόπος λειτουργίας

Ο client περιέχει το αρχείο settings.properties στο οποίο υπάρχουν αποθηκευμένα τα στοιχεία σύνδεσης προς το server. Τα στοιχεία αυτά είναι η διεύθυνση και η πόρτα στην οποία ακούει ο server.

Θεωρούμε πως κάθε ολοκληρωμένη (request-response) σύνδεση στο server είναι ένα σενάριο. Ο τρόπος λειτουργίας του client είναι η αποστολή κάποιον σεναρίων στο server. Οπότε έχουμε δημιουργήσει μια κλάση ανά σενάριο. Ανάλογα το σενάριο που θέλουμε να εκτελέσουμε, δημιουργούμε αντικείμενα σεναρίων, τα τοποθετούμε σε μια λίστα και τα δίνουμε ως όρισμα στην κλάση που διαχειρίζεται τη σύνδεση με το server. Η κλάση αυτή, εκτελεί τα σενάρια με τη σειρά που έχουν τοποθετηθεί στη λίστα. Τα δύο ερωτήματα της εργασίας είναι, από τη μεριά του client, δύο λίστες σεναρίων.

Η κλάση που υλοποιεί τη σύνδεση με το server είναι κλάση τύπου Thread. Κάθε αντικείμενό της τρέχει ως ξεχωριστό νήμα στην εφαρμογή. Το response κάθε request εκτυπώνεται στην κονσόλα και σε αρχείο καταγραφής (log).

2. Τρόπος δόμησης πακέτων

Το βασικό πακέτο της εφαρμογής ακολουθεί τη μορφή του βασικού πακέτου της προηγούμενης εργασίας και είναι το "gr.unipi.mainpackage.client". Εκεί βρίσκονται τέσσερα πακέτα το lib, το model, το scenario και το service.

Το πακέτο lib περιέχει κλάσεις γενικής χρήσης. Η κλάση Encryption περιέχει μεθόδους για encryption. Ο μοναδικός τρόπος που υποστηρίζεται είναι το md5 hash. Η κλάση PropertiesUtilities είναι κλάση τύπου singleton. Αντιπροσωπεύει το αρχείο ρυθμίσεων.

Το πακέτο model περιέχει όλα τα model τύπου Java Beans της εφαρμογής. Αυτά είναι δύο κατηγοριών, τα μοντέλα δεδομένων και τα μοντέλα επικοινωνίας με το server. Έτσι, το υπο-πακέτο data περιέχει όλα τα μοντέλα δεδομένων. Το υπο-πακέτο client περιέχει τις κλάσεις RequestMethod, RequestModel, ResponseErrorModel και ResponseModel. Το enum RequestMethod περιέχει όλες τις μεθόδους επικοινωνίας που προαναφέρθηκαν. Οι κλάσεις

RequestModel, ReponseModel και ResponseErrorModel είναι μοντέλα που δέχεται και στέλνει ο client με το server κατά τη διάρκεια της επικοινωνίας τους.

Το πακέτο scenario περιέχει όλα τα scenario που κατασκευάσαμε για λόγους δοκιμών και για λόγους υλοποίησης των ερωτήσεων της εργασίας. Αυτά είναι τα σενάρια δημιουργίας και σύνδεσης(login) ενός διαχειριστή, τα σενάρια δημιουργίας και σύνδεσης ενός διαχειριστή περιεχομένου, τα σενάρια δημιουργίας και σύνδεσης ενός πελάτη, τα σενάρια δημιουργίας και αναζήτησης μίας ταινίας και τα σενάρια δημιουργίας και αναζήτησης προβολών μιας ταινίας.

Το πακέτο service περιέχει μόνο μια κλάση. Αυτή είναι η κλάση-νήμα η οποία διαχειρίζεται τη σύνδεση με τον server.

2.3. Server

1. Τρόπος λειτουργίας

Ο server περιέχει το αρχείο settings.properties στο οποίο υπάρχουν αποθηκευμένα η πόρτα στην οποία θα ακούει ο server, το χρονικό διάστημα μετά του οποίου θα λήγει ένας session κωδικός καθώς επίσης μια ρύθμιση για το αν θα γίνεται έλεγχος για τη σύνδεση χρήστη ή όχι.

Τα δεδομένα της εφαρμογής αποθηκεύονται σε αρχεία τα οποία βρίσκονται στο φάκελο database. Στο φάκελο αυτό βρίσκεται ένα αρχείο ανά οντότητα (μοντέλο δεδομένων). Όλα τα αρχεία περιέχουν δεδομένα σε μορφή json. Τα αρχεία αυτά τα διαχειρίζονται κλάσεις τύπου DbFileManager. Υπάρχει μια κλάση ανά αρχείο ή αλλιώς μια κλάση ανά οντότητα. Οι κλάσεις τύπου DBFileManager υποστηρίζουν δυνατότητες search, create, readAll, read, update και delete. Ακόμη, οι κλάσεις αυτές διασφαλίζουν ότι το format των αρχείων θα παραμείνει json μετά από ταυτόχρονες αλλαγές τους από διαφορετικά νήματα. Το πρόβλημα με το οποίο δεν ασχοληθήκαμε περαιτέρω είναι πως μετά από ταυτόχρονη δημιουργία ή αλλαγή οντοτήτων στο αρχείο, θα υλοποιηθεί μόνο η τελευταία από αυτές.

Οι κλάσεις τύπου Service, είναι αυτές που επικοινωνούν με τις κλάσεις τύπου DbFileManager. Οι μέθοδοι των κλάσεων τύπου service είναι αυτές που εκτελούνται κατόπιν αιτημάτων από τους client. Σε αυτές τις κλάσεις γίνεται ο έλεγχος δικαιωμάτων πρόσβασης. Σε αυτές υπάρχουν οι μέθοδοι search, create, update, delete, read, login και logout. Τα ονόματα των μεθόδων αυτών ακολουθούν τη σύμβαση να συνδυάζονται με την οντότητα που διαχειρίζονται. Δηλαδή, για παράδειγμα, η δημιουργία ενός πελάτη επιτελείται από την κλάση CustomerService, από τη μέθοδο createCustomer. Η σύμβαση του ονόματος των services και των μεθόδων τους είναι πολύ σημαντική για λόγους που θα

περιγράψουμε παρακάτω.

Τα δικαιώματα της εφαρμογής, όπως περιγράφηκε παραπάνω, ελέγχονται μέσα από τις μεθόδους των Service κλάσεων. Πιο συγκεκριμένα, όλα τα δικαιώματα της εφαρμογής βρίσκονται σε ένα enumeration. Κάθε οντότητα, η οποία έχει δικαίωμα σύνδεσης, είναι τύπου AuthorizedUser. Οι κλάσεις τύπου AuthorizedUser, περιέχουν μέθοδο η οποία επιστρέφει λίστα με τα δικαιώματα που έχει η οντότητα. Για παράδειγμα, η οντότητα του διαχειριστή επιστρέφει λίστα με όλα τα δικαιώματα. Κάθε μέθοδος των κλάσεων Service γνωρίζει πιο δικαίωμα απαιτείται για να ολοκληρώσει την εκτέλεσή της. Επίσης, όλες οι μέθοδοι δέχονται ως όρισμα ένα αντικείμενο τύπου AuthorizedUser. Με αυτό τον τρόπο επιτυγχάνεται ο έλεγχος των δικαιωμάτων. Έτσι, εάν θέλουμε να προσθέσουμε ένα επιπλέον δικαίωμα σε κάποιο ρόλο, το μόνο που έχουμε να κάνουμε είναι να το προσθέσουμε στη λίστα δικαιωμάτων της οντότητας.

Η κλάση που δέχεται συνδέσεις είναι κλάση που τρέχει στο βασικό thread της εφαρμογής. Για κάθε σύνδεση που γίνεται δημιουργεί ένα νήμα το οποίο θα διαχειριστεί τη σύνδεση.

Τα νήματα που διαχειρίζονται τη σύνδεση επιτελούν 4 στάδια επεξεργασίας. Το πρώτο στάδιο είναι η μετατροπή του json σε αντικείμενο. Το δεύτερο στάδιο είναι η διαχείριση του request και η δημιουργία του response αντικειμένου. Σε αυτό το στάδιο γίνεται διερεύνηση εάν πρόκειται για request που σκοπό έχει τη διαχείριση δεδομένων (SCRUD) ή request που σκοπό έχει τη σύνδεση (login/logout). Ακόμη σε αυτό το στάδιο χρησιμοποιούμε το Java Reflection API ώστε να καλέσουμε την κατάλληλη μέθοδο από την κατάλληλη κλάση Service. Αυτός είναι ο λόγος της σύμβασης με τα ονόματα των μεθόδων και κλάσεων τύπου Service. Το τρίτο στάδιο είναι η αποστολή του response στο χρήστη. Το τέταρτο στάδιο είναι το κλείσιμο της σύνδεσης με τον client και το τέλος του νήματος αυτού.

2. Τρόπος δόμησης πακέτων

Το βασικό πακέτο της εφαρμογής ακολουθεί τη μορφή του βασικού πακέτου της προηγούμενης εργασίας και είναι το "gr.unipi.mainpackage.server". Εκεί βρίσκονται τρία πακέτα το lib, το model και το service.

Το πακέτο lib περιέχει δύο κλάσεις γενικής χρήσης και επιπλέον τρία υπό-πακέτα το authority, το session και το signin. Η κλάση Encryption περιέχει μεθόδους για encryption. Ο μοναδικός τρόπος που υποστηρίζεται είναι το md5 hash. Η κλάση PropertiesUtilities είναι κλάση τύπου singleton. Αντιπροσωπεύει το αρχείο ρυθμίσεων.

Το υπό-πακέτο authority του πακέτου lib περιέχει κλάσεις οι οποίες διαχειρίζονται τα δικαιώματα χρηστών. Αυτές είναι οι Authority, AuthorityUtils, AuthorizationException και AuthorizedUser. Το

enumeration Authority περιέχει όλους τους ρόλους που υπάρχουν στο σύστημα. Η κλάση AuthorityUtils, είναι μια κλάση helper η οποία περιέχει μια μόνο μέθοδο για τη διερεύνηση εάν ένας χρήστης έχει κάποιο δικαίωμα. Η μέθοδος αυτή χρησιμοποιείται από τις μεθόδους των κλάσεων τύπου Service. Η κλάση AuthorizationException είναι μια εξαίρεση που δημιουργείτε όταν κάποιος χρήστης ζητάει πρόσβαση σε μέθοδο στην οποία δεν έχει δικαίωμα. Τέλος, το interface AuthorizedUser είναι αυτό που υλοποιούν οι οντότητες που έχουν δικαιώματα.

Το υπό-πακέτο session του πακέτου lib περιέχει δύο κλάσεις με τις οποίες γίνεται η διαχείριση των session. Η κλάση SessionCache είναι μια κλάση τύπου singleton. Το αντικείμενο αυτής της κλάσης περιέχει ένα Map στο οποίο είναι αποθηκευμένοι ανά sessionId οι χρήστες με την ημερομηνία τελευταίας σύνδεσης στο server. Ο χρήστης και η ημερομηνία πακετάρονται σε ένα μοντέλο το οποίο είναι η δεύτερη κλάση αυτού του πακέτο. Πρόκειται για την κλάση AuthorizedModel η οποία κάνει wrap τον χρήστη και την ημερομηνία για χρήση στο Map του session.

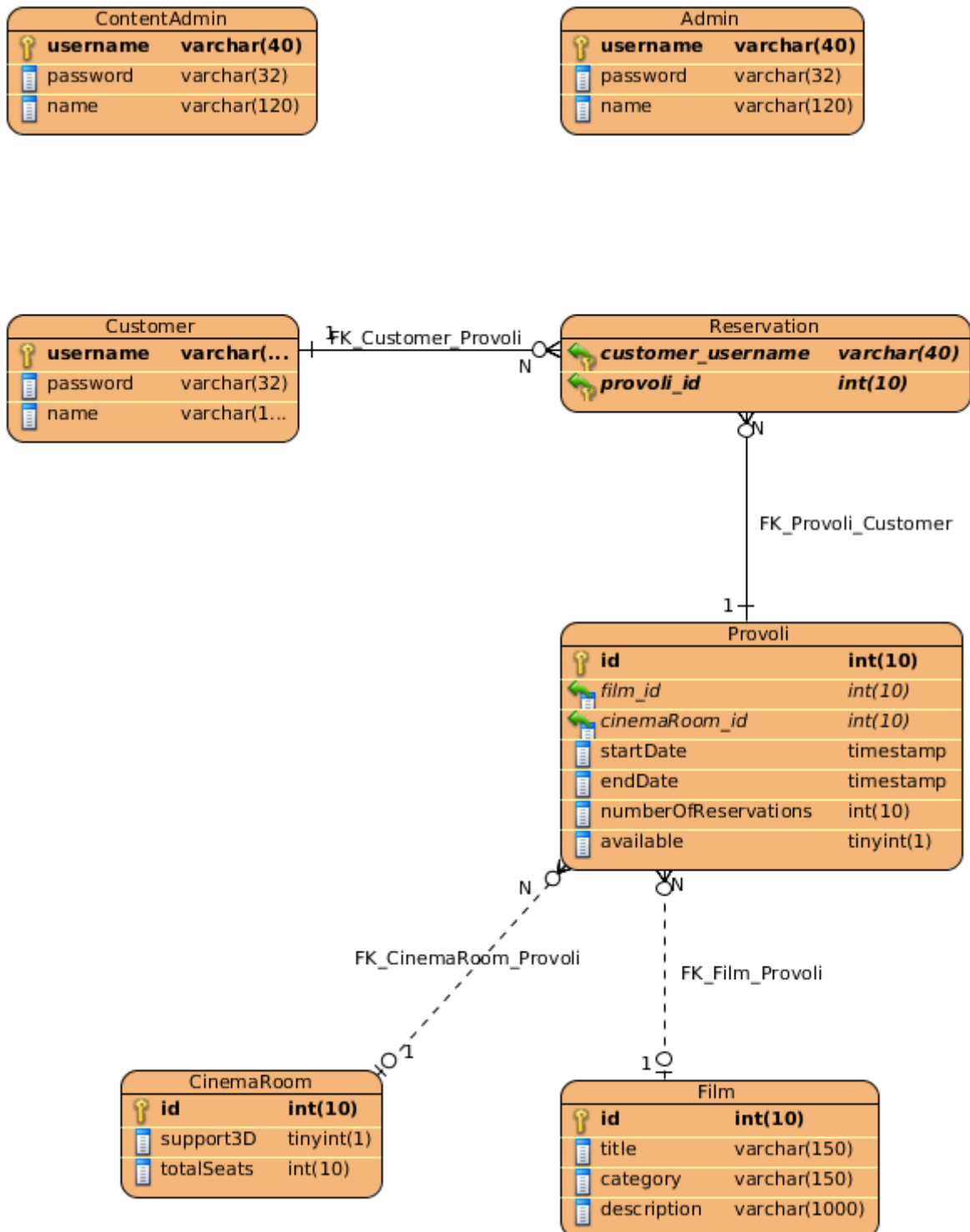
Το υπό-πακέτο signin του πακέτου lib περιέχει δύο interfaces. Το interface SignInAbleUser υλοποιούν οι οντότητες που έχουν δυνατότητα login. Το interface SignInAbleService περιέχει τις μεθόδους για τον επιτυχή έλεγχο ενός login/logout. Τα δύο interfaces δεν έχουν κάποια ιδιαίτερη χρησιμότητα στο λογισμικό. Τα δημιουργήσαμε μόνο για το λόγο ότι ορίζουν τύπους.

Το πακέτο model περιέχει όλα τα model τύπου Java Beans της εφαρμογής. Αυτά είναι δύο κατηγοριών, τα μοντέλα δεδομένων και τα μοντέλα επικοινωνίας με τον client. Έτσι, το υπο-πακέτο data περιέχει όλα τα μοντέλα δεδομένων, δηλαδή όλες τις οντότητες της εφαρμογής. Το υπο-πακέτο server περιέχει τις κλάσεις RequestMethod, RequestModel, ResponseErrorModel και ResponseModel. Το enum RequestMethod περιέχει όλες τις μεθόδους επικοινωνίας που προαναφέρθηκαν. Οι κλάσεις RequestModel, ResponseModel και ResponseErrorModel είναι μοντέλα που δέχεται και στέλνει ο server με τον client κατά τη διάρκεια της επικοινωνίας τους.

Το πακέτο service περιέχει τη λειτουργικότητα της εφαρμογής με τρία υποπακέτα. Το πρώτο υπο-πακέτο data περιέχει όλες τις κλάσεις Service που ακολουθούν σύμβαση με τα ονόματα κλάσεων και μεθόδων. Το δεύτερο υπο-πακέτο fileManager περιέχει όλες τις κλάσεις τύπου DbFileManager καθώς και το interface που αυτές υλοποιούν. Τέλος, το υπο-πακέτο server περιέχει δύο κλάσεις MainServer και ServerManager. Η κλάση MainServer είναι αυτή που ακούει στην πόρτα και για κάθε σύνδεση δημιουργεί ένα νήμα που θα τη διαχειριστεί. Η κλάση ServerManager είναι η κλάση-νήμα που διαχειρίζεται τις συνδέσεις με τρόπο που περιγράφηκε παραπάνω.

3. Σχέδιο βάσης δεδομένων

Παρακάτω το σχέδιο της βάσης δεδομένων όπως το κατασκευάσαμε με χρήση του προγράμματος Visual Paradigm For Uml.



4. Αναφορές

1. <http://www.debian.org/>
2. <https://el.libreoffice.org/>
3. <https://netbeans.org/>
4. <https://maven.apache.org>
5. <http://www.visual-paradigm.com/>
6. http://en.wikipedia.org/wiki/Create,_read,_update_and_delete
7. <https://code.google.com/p/google-gson/>