



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη Java εφαρμογών με την χρήση του εργαλείου Apache Maven

Του φοιτητή

ΣΕΡΙΣΤΑΤΙΔΗ ΧΑΡΑΛΑΜΠΟΥ

Αρ. Μητρώου: 04/2505

Επιβλέπων καθηγητής

ΠΑΝ. ΣΦΕΤΣΟΣ

Θεσσαλονίκη 2013

ΠΡΟΛΟΓΟΣ

Στο σημερινό κόσμο η ανάπτυξη εφαρμογών μέσω κάποιας γλώσσας προγραμματισμού έχει διαδοθεί αρκετά σε σχέση με μια δεκαετία πριν. Σε αυτό βοήθησε η ανάπτυξη του ίντερνετ καθώς και απλοποίηση των ίδιων των γλωσσών προγραμματισμού. Ο καθένας μπορεί πλέον να δημιουργήσει μια μικρή εφαρμογή, αλλά όταν μιλάμε για μεγάλα έργα προγραμματισμού τότε τα πράγματα περιπλέκονται λίγο.

Εδώ έρχεται να βοηθήσει το λογισμικό που ονομάζεται Apache Maven. Το λογισμικό αυτό, όπως θα δούμε βοηθάει όχι στην συγγραφή κώδικα αλλά στην οργάνωση των πηγαίων αρχείων ενός έργου και στο συμμάζεμα του μαζί με εξωτερικές βιβλιοθήκες σε ένα αρχείο για εύκολη μεταφορά. Η σχετικά απλή εγκατάστασή του και η απλούστατη χρήση του συμβάλουν στην εύκολη εκμάθησή του από απλούς χρήστες πόσο μάλλον από βετεράνους του προγραμματισμού.

ΠΕΡΙΛΗΨΗ

Το λογισμικό Apache Maven λειτουργεί σε φάσεις. Η κάθε φάση είναι σχεδιασμένη να εκτελεί κάποια λειτουργία. Η Maven εκτελεί τις φάσεις σειριακά και το μόνο που έχει να κάνει ο χρήστης είναι να επιλέξει φάση, να την εκτελέσει και να αφήσει τη Maven να κάνει όλη τη δουλειά, εκτελώντας τις φάσεις μια – μια μέχρι να φτάσει στην φάση που εκτέλεσε ο χρήστης.

Οι φάσεις πραγματοποιούνται μέσω πρόσθετων. Το κάθε πρόσθετο περιέχει στόχους ο κάθε ένας από τους οποίους πραγματοποιεί μια διαφορετική εργασία. Ο χρήστης επιλέγει στόχο μαζί με τη φάση ή αφήνει τον προεπιλεγμένο στόχο της φάσης.

Μερικά από τα πρόσθετα της Maven είναι clean, compiler, deploy, failsafe, install, resources, site, surefire ή verifier. Τα πρόσθετα αυτά καθώς και πολλά άλλα θα αναλυθούν παρακάτω με περισσότερες λεπτομέρειες.

Για την ευκολία των χρηστών η Maven υποστηρίζεται από τα προγράμματα IDE Eclipse και Netbeans. Έτσι ο χρήστης κερδίζει και την ευκολία συγγραφής κώδικα και την ευκολία κατασκευής έργου και μεταφορά του σε άλλα συστήματα.

Ο χρήστης έχει τη δυνατότητα να συγγράψει τα δικά του πρόσθετα και να τα εγκαταστήσει στο σύστημά του ή να τα μοιράσει για να τα χρησιμοποιήσουν και άλλοι χρήστες της Maven.

Τέλος η Maven αν και μοιάζει με την Apache Ant δεν είναι. Οι διαφορές θα περιγραφούν και θα αναλυθούν σε επόμενο κεφάλαιο.

ABSTRACT

The software called Apache Maven works in phases. Each phase is designed to execute some function. Maven executes all the phases in an order and the only thing a user has to do is choose a phase, execute it and let Maven do all the work, executing all the phases one by one until it reaches the phase the user executed.

The phases work through plugins. Each plugin contains goals, each of them does something different. The user can select the goal along with the phase or they can leave the default goal of the phase.

Some of Maven's plugins are clean, compiler, deploy, failsafe, install, resources, site, surefire or verifier. These plugins as well as many others will be described in detail further below.

For the user's ease Maven supports IDE programs Eclipse and Netbeans. That way, the user gets help writing code and has ease in constructing a project and moves it to other systems.

The user has the ability to write his own plugins and install them in his system or distribute them so they can be used by other Maven users.

Finally although Maven looks like Ant it isn't. The differences will be listed and described in another chapter.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	2
ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	4
ΠΕΡΙΕΧΟΜΕΝΑ.....	5
ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ	7
ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ	8
ΕΙΣΑΓΩΓΗ.....	9
ΚΕΦΑΛΑΙΟ 1	11
Παρουσίαση της τεχνολογίας Apache Maven.....	11
ΕΙΣΑΓΩΓΗ	11
ΥΠΟΚΕΦΑΛΑΙΟ 1.1	12
ΥΠΟΚΕΦΑΛΑΙΟ 1.2	13
ΥΠΟΚΕΦΑΛΑΙΟ 1.3	16
ΥΠΟΚΕΦΑΛΑΙΟ 1.4	18
ΚΕΦΑΛΑΙΟ 2	24
Παρουσίαση των πρόσθετων της Maven (Maven Plugins).....	24
ΕΙΣΑΓΩΓΗ	24
ΥΠΟΚΕΦΑΛΑΙΟ 2.1	25
ΥΠΟΚΕΦΑΛΑΙΟ 2.2	27
ΚΕΦΑΛΑΙΟ 3	48
Η Maven στα IDE Eclipse και NetBeans.....	48
ΕΙΣΑΓΩΓΗ	48
ΥΠΟΚΕΦΑΛΑΙΟ 3.1	49

ΥΠΟΚΕΦΑΛΑΙΟ 3.2	55
ΚΕΦΑΛΑΙΟ 4	60
Συγγραφή προσθέτων	60
ΕΙΣΑΓΩΓΗ	60
ΥΠΟΚΕΦΑΛΑΙΟ 4.1	61
ΥΠΟΚΕΦΑΛΑΙΟ 4.2	66
ΚΕΦΑΛΑΙΟ 5	71
Σύγκριση Apache Maven με Apache Ant.....	71
ΕΙΣΑΓΩΓΗ	71
ΥΠΟΚΕΦΑΛΑΙΟ 5.1	72
ΥΠΟΚΕΦΑΛΑΙΟ 5.2	74
ΥΠΟΚΕΦΑΛΑΙΟ 5.3	78
ΚΕΦΑΛΑΙΟ 6	80
Ιστοδελίδα.....	80
ΕΙΣΑΓΩΓΗ	80
ΥΠΟΚΕΦΑΛΑΙΟ 6.1	81
ΥΠΟΚΕΦΑΛΑΙΟ 6.2	86
ΥΠΟΚΕΦΑΛΑΙΟ 6.3	96
ΥΠΟΚΕΦΑΛΑΙΟ 6.4	101

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

ΕΙΚΟΝΑ 1	16
ΕΙΚΟΝΑ 2	21
ΕΙΚΟΝΑ 3	22
ΕΙΚΟΝΑ 4	24
ΕΙΚΟΝΑ 5	30
ΕΙΚΟΝΑ 6	35
ΕΙΚΟΝΑ 7	35
ΕΙΚΟΝΑ 8	36
ΕΙΚΟΝΑ 9	38
ΕΙΚΟΝΑ 10	42
ΕΙΚΟΝΑ 11	48
ΕΙΚΟΝΑ 12	49
ΕΙΚΟΝΑ 13	49
ΕΙΚΟΝΑ 14	50
ΕΙΚΟΝΑ 15	51
ΕΙΚΟΝΑ 16	52
ΕΙΚΟΝΑ 17	53
ΕΙΚΟΝΑ 18	55
ΕΙΚΟΝΑ 19	56
ΕΙΚΟΝΑ 20	57
ΕΙΚΟΝΑ 21	81
ΕΙΚΟΝΑ 22	82
ΕΙΚΟΝΑ 23	85

EIKONA 24	86
EIKONA 25	88
EIKONA 26	89
EIKONA 27	91
EIKONA 28	92
EIKONA 29	93
EIKONA 30	96
EIKONA 31	97
EIKONA 32	98

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ 1	19
-----------------	----

ΕΙΣΑΓΩΓΗ

Η παρούσα πτυχιακή έχει ως στόχο την παρουσίαση και την περιγραφή του πλαισίου ανάπτυξης Java εφαρμογών με χρήση της τεχνολογίας Apache MAVEN. Σκοπός είναι ο αναγνώστης να αποκτήσει μια πρώτη γεύση της τεχνολογίας, να εισηγηθεί στην χρήση αυτής και να δει τα πλεονεκτήματα της καθώς και τα όποια μειονεκτήματα της.

Θα αναφερθούμε επίσης στα πρόσθετα (plugins) της τεχνολογίας Maven. Θα περιγράψουμε τα σημαντικότερα από αυτά και θα περιγράψουμε την συγγραφή και χρήση αυτών.

Θα δημιουργήσουμε και θα εκτελέσουμε έργα μόνο με χρήση της τεχνολογίας Maven καθώς και με τη βοήθεια εργαλείων όπως Eclipse και NetBeans τα οποία ενσωματώνουν την τεχνολογία Maven και βοηθούν το χρήστη. Επίσης θα συγκρίνουμε την Maven με την τεχνολογία Apache Ant, μία βιβλιοθήκη java και εργαλείο εντολών (Java library and command-line tool).

Στο πρώτο κεφάλαιο θα παρουσιάσουμε το ίδιο το πλαίσιο. Το δεύτερο κεφάλαιο θα περιέχει την παρουσίαση των java-βιβλιοθηκών και των Maven plugins ενώ στο τρίτο θα παρουσιαστεί ο τρόπος εκτέλεσης έργων Maven μέσα από τα εργαλεία Eclipse και NetBeans. Στο τέταρτο κεφάλαιο θα περιγράψουμε την διαδικασία συγγραφής προσθέτων ενώ το πέμπτο θα συγκρίνει τη Apache Maven με την Apache Ant. Τέλος θα αναπτυχθεί ένα μεγάλο έργο σε γλώσσα προγραμματισμού Java με την χρήση Ευέλικτων Πρακτικών όπως TDD, TFD+Refactorings και JUNIT

ΚΕΦΑΛΑΙΟ 1

Παρουσίαση της τεχνολογίας Apache Maven.

ΕΙΣΑΓΩΓΗ

Η τεχνολογία Apache Maven πρώτα ξεκίνησε το έτος 2002 ως μια προσπάθεια απλοποίησης των διαδικασιών κατασκευής στο έργο Jakarta-Turbine. Αρκετά έργα είχαν δικά τους αρχεία κατασκευής ANT το καθένα διαφορετικό με τα άλλα. Η Maven εισήγαγε έναν καθιερωμένο τρόπο κατασκευής έργων μια σαφή περιγραφή του τι είναι ένα έργο, έναν εύκολο τρόπο δημοσίευσης πληροφοριών έργου και διαμοίραση αρχείων jar σε διάφορα έργα.

Η Maven μπορεί να κατασκευάσει και να διαχειριστεί έργα σε γλώσσα προγραμματισμού Java διευκολύνοντας την καθημερινή εργασία των προγραμματιστών Java και γενικά να βοηθήσει με την κατανόηση οποιουδήποτε έργου java.

ΥΠΟΚΕΦΑΛΑΙΟ 1.1

Ξεκινώντας πρέπει να αναφερθούμε σε μια περιγραφή του τι είναι η Maven. Η τεχνολογία Maven είναι ένα λογισμικό διαχείρισης έργων και εργαλείο κατανόησης. Βασίζεται στην γενική ιδέα ενός έργου μοντέλου αντικειμένου (Project Object Model – POM) και μπορεί να διαχειριστεί την κατασκευή του έργου και την τεκμηρίωσή του από ένα κεντρικό κομμάτι πληροφοριών.

Η Maven παρέχει απλή ρύθμιση έργου (simple project setup) που ακολουθεί «καλές πρακτικές». Αυτό επιτρέπει την εκκίνηση ενός καινούριου έργου σε λεπτά – αν όχι δευτερόλεπτα. Η χρήση της Maven είναι ίδια σε όλα τα έργα οπότε ένας προγραμματιστής που έχει δουλέψει με Maven στο παρελθόν μπορεί να μπει κατευθείαν στο έργο χωρίς να χρειαστεί προσαρμογή. Κάθε έργο τοποθετείτε σε ξεχωριστό δικό του φάκελο οπότε η μεταφορά από έργο σε έργο είναι όσο εύκολη όσο η αλλαγή φακέλου.

Η Maven έχει επίσης τη δυνατότητα να διαχειρίζεται τις εξαρτήσεις ενός έργου όπως για παράδειγμα βιβλιοθήκες Java και να τις ανανεώνει μέσω ίντερνετ ανά πάσα στιγμή. Έχει επίσης τη δυνατότητα να παρέχει τέτοιες εξαρτήσεις μαζί με το έργο οπότε ακόμα και να μεταφερθεί το έργο σε άλλο υπολογιστή μπορεί να μεταφέρει μαζί του και τις εξαρτήσεις.

Η Maven αν και λειτουργεί με μικρά έργα εκεί που πραγματικά δείχνει τη χρησιμότητά της είναι σε μεγάλου μεγέθους έργα με πολλαπλές ομάδες, καθώς βοηθάει τα άτομα των μονάδων να συνεργαστούν εστιάζοντας στο τι απαιτεί ένα έργο από την ομάδα προγραμματισμού.

Ίσως να υπάρξουν τυχόν παραπληροφορήσεις όσον αφορά την Maven. Αν και η Maven έχει τη δυνατότητα να πραγματοποιήσει αυτές τις ενέργειες ο σκοπός της Maven δεν είναι αυτός.

- Η Maven είναι μια ιστοσελίδα και εργαλείο τεκμηρίωσης λογισμικού
- Επεκτείνει την τεχνολογία ANT επιτρέποντας τη λήψη εξαρτήσεων
- Είναι μια συλλογή επαναχρησιμοποιήσιμων δεσμών ενεργειών (scriptlet) ANT

ΥΠΟΚΕΦΑΛΑΙΟ 1.2

Η Maven για τη λειτουργία της βασίζεται, όπως αναφέρθηκε στο POM (Project Object Model). Αυτό είναι ένα αρχείο xml που τοποθετείτε στο φάκελο ενός έργου και περιέχει όλες τις πληροφορίες που αυτό απαιτεί για την λειτουργία του. Το αρχείο αυτό αναφέρει στη Maven σε τί είδους έργο ανήκει και πως να μεταβάλει την προεπιλεγμένη συμπεριφορά του ή την συμπεριφορά των προσθέτων της Maven όταν εκτελούνται στο έργο αυτό.

Το POM περιέχει τέσσερις κατηγορίες περιγραφών και διαμορφώσεων.

- Γενικές πληροφορίες έργου (general project information). Αυτή η κατηγορία περιλαμβάνει το όνομα ενός έργου, το URL του, την χρηματοδοτούσα οργάνωση, μια λίστα προγραμματιστών και συνεισφερόντων και τέλος την άδεια ενός έργου.
- Ρυθμίσεις κατασκευής (build settings). Στο κομμάτι αυτό υπάρχει η δυνατότητα προσαρμογής της προεπιλεγμένης κατασκευής ενός έργου Maven. Μπορεί να αλλάξει η τοποθεσία του πηγαίου κώδικα και των δοκιμών, μπορούν να προστεθούν νέα πρόσθετα (plugins), να επισυναφθούν στόχοι στους κύκλους κατασκευής (περισσότερα για τους κύκλους κατασκευής παρακάτω) και τέλος να προσαρμοστούν οι παράμετροι δημιουργίας ιστοσελίδας – τοποθεσίας.
- Περιβάλλον κατασκευής (build environment). Το περιβάλλον κατασκευής περιέχει προφίλ τα οποία μπορούν να ενεργοποιηθούν ανάλογα με διαφορετικά περιβάλλοντα. Παραδείγματος χάριν κατά την διάρκεια της ανάπτυξης μπορεί να χρειαστεί να αναπτυχθεί ένας διακομιστής ανάπτυξης ενώ κατά τη διάρκεια της παραγωγής ένας διακομιστής παραγωγής. Το περιβάλλον κατασκευής διαμορφώνει τις ρυθμίσεις κατασκευής για συγκεκριμένα περιβάλλοντα και συχνά συμπληρώνετε από ένα προσαρμοσμένο αρχείο ρυθμίσεων

settings.xml στο φάκελο .m2 εκεί που βρίσκεται το τοπικό αποθετήριο.

- Τέλος οι σχέσεις POM. Καθώς ένα έργο Maven σπανίως υπάρχει μόνο του και στηρίζεται σε άλλα εδώ αναφέρονται τα άλλα αυτά έργα. Επίσης κληρονομεί ρυθμίσεις POM από άλλα γονικά έργα και ορίζει δικές του συντεταγμένες (μοναδικές τοποθεσίες για κάθε έργο, GroupId, ArtifactId, Version χρησιμοποιούνται για να ονομάσουμε μοναδικά κάποιο έργο Maven), και ίσως να συμπεριλαμβάνει υπομονάδες (submodules).

Στη Maven όλα τα POM επεκτείνουν ένα γονικό Super POM το οποίο ορίζει μια συλλογή προεπιλεγμένων ρυθμίσεων οι οποίες μοιράζονται με όλα τα έργα. Το Super POM είναι μέρος της εγκατάστασης της Maven και βρίσκεται στο αρχείο maven-3.0.3-uber.jar στην τοποθεσία $\${M2_HOME}\lib$ ^[1] (περισσότερα για αυτήν την τοποθεσία παρακάτω).

Το Super POM ορίζει κάποιες τυπικές μεταβλητές ρυθμίσεων οι οποίες κληρονομούνται από όλα τα έργα. Μια έχει την ταυτότητα central και ορίζει ένα απομακρυσμένο αποθετήριο από οποία εξορισμού όλοι οι «πελάτες» Maven (Maven clients) λαμβάνουν πόρους όπως τα πρόσθετα της Maven. Κάποιο άλλο που ονομάζεται build ρυθμίζει τους προεπιλεγμένους καταλόγους στην τυπική διάταξη κατάλογων ενός έργου Maven.

Στην έκδοση 2.0.9 της Maven και μετά οι προεπιλεγμένες εκδόσεις των πρόσθετων πυρήνα παρέχονται από το Super POM. Αυτό γίνεται ώστε η Maven να παρέχει κάποια σταθερότητα στους χρήστες της Maven ο οποίοι δεν προσδιορίζουν τις εκδόσεις στα POM τους.

Το POM σαν αρχείο xml που είναι έχει μια τυπική δομή ενός τέτοιου αρχείου. Η ετικέτα <project> περιλαμβάνει το αρχείο οι ετικέτες <groupId>, <artifactId> και <version> προσδιορίζουν μοναδικά το έργο, η ετικέτα <dependencies> ορίζει την αρχή του σημείου που δίνονται όλες οι εξαρτήσεις ενός έργου ενώ η ετικέτα <dependency> βρίσκεται μεταξύ αρχής και τέλους της προηγούμενης ετικέτας και ορίζει κάθε μία εξάρτηση του έργου.

Η ετικέτα `<configuration>` ορίζει τις διάφορες ρυθμίσεις που μπορεί να αλλάξουν σε πρόσθετα. Για παράδειγμα το κομμάτι κώδικα

```
<configuration>
```

```
<source>1.5</source>
```

```
<target>1.5</target>
```

```
</configuration>
```

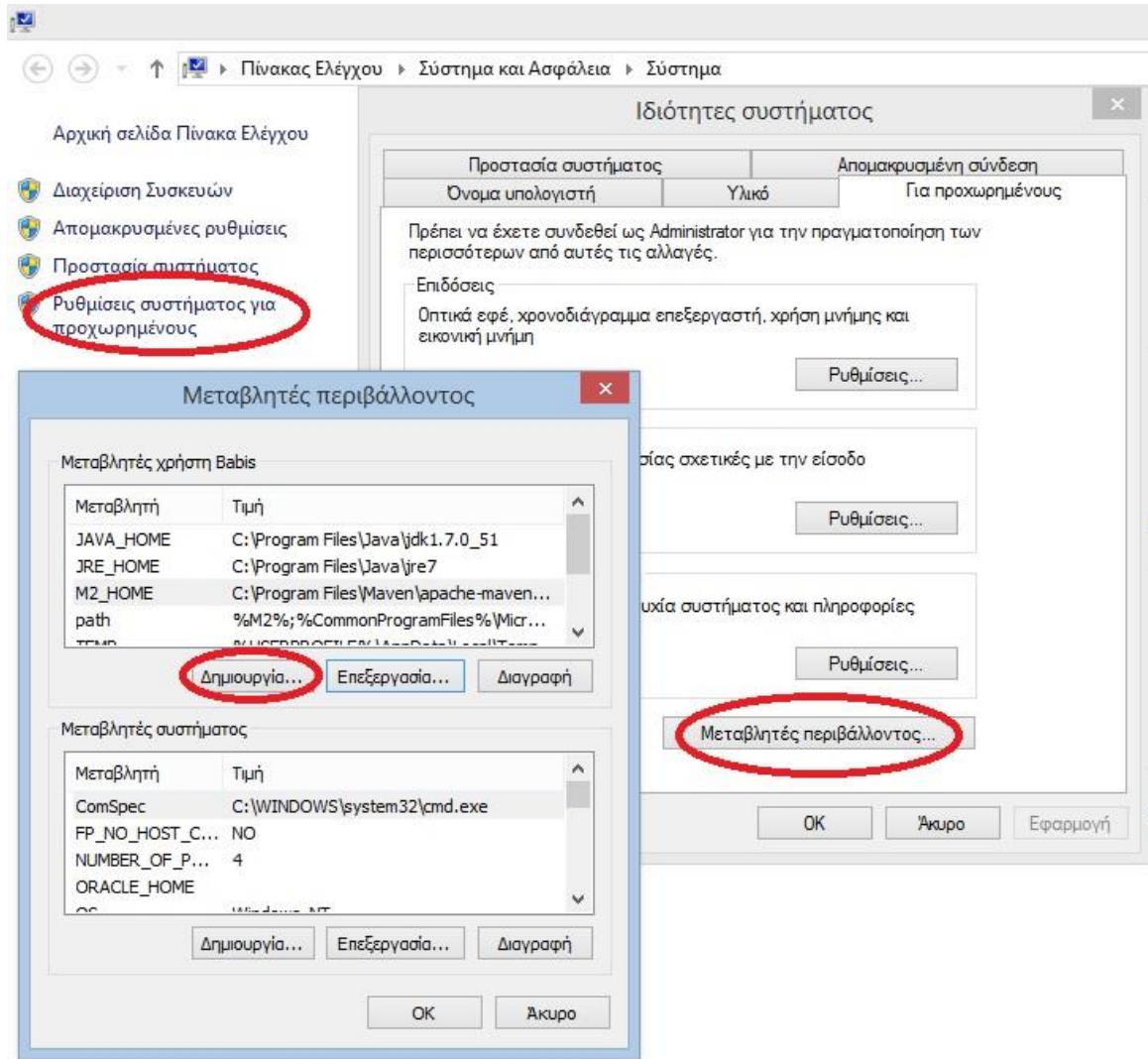
επιλέγει έκδοση του πρόσθετου μέσα στο οποίο είναι γραμμένο και συγκεκριμένα την έκδοση 1.5. Περισσότερα για τις ρυθμίσεις προσθέτων παρακάτω. Το κομμάτι αυτό πρέπει να τοποθετηθεί μέσα στην ετικέτα `<plugin>` για να ρυθμίσουμε το συγκεκριμένο πρόσθετο και αυτήν μέσα στις `<plugins>` και `<build>`.

ΥΠΟΚΕΦΑΛΑΙΟ 1.3

Η εγκατάσταση του περιβάλλοντος γίνεται απλά και μπορεί να χρησιμοποιηθεί σε οποιοδήποτε περιβάλλον συμπεριλαμβανομένου όλων των εκδόσεων Windows μετά από την έκδοση 2000, Linux/Unix ακόμα και MAC OS X^[2]. Η Maven αρκετά συχνά λαμβάνει διάφορους πόρους μέσω ίντερνετ οπότε η χρήση της Maven απαιτεί συνεχή πρόσβαση στο ίντερνετ αν και ο όγκος των πόρων δεν είναι μεγάλος και διάρκεια των λήψεων αρκετά συχνά δεν κρατάει πάνω από μερικά λεπτά. Επίσης η Maven απαιτεί την προ-εγκατάσταση της Java και συγκεκριμένα του πακέτου προγραμματιστή (Java Developer Kit).

Η εγκατάσταση της Maven ξεκινάει από την ιστοσελίδα <http://maven.apache.org/download.cgi> όπου υπάρχουν διάφορες εκδόσεις του λογισμικού που θα χρησιμοποιηθεί. Αφού κατεβάσουμε μια έκδοση σε μορφή .zip ή σε .tar.gz αποσυμπιέζουμε το αρχείο σε ένα φάκελο της επιλογής μας ο οποίος φάκελος θα είναι και ο τελικός φάκελος εγκατάστασης της Maven. Αυτό βέβαια δεν επηρεάζει την τοποθεσία των έργων που θα δουλεύουν με την Maven.

Έπειτα αυτό που απαιτείται από το λογισμικό είναι ο ορισμός μιας μεταβλητής περιβάλλοντος με όνομα M2_Home και τιμή η διαδρομή της αποσυμπίεσης που πραγματοποιήθηκε προηγουμένως για παράδειγμα σε ένα περιβάλλον windows «C:\Program Files\Maven\apache-maven-3.1.0». Η επιλογή του φακέλου Program Files δεν είναι υποχρεωτική και θα μπορούσε να είναι οπουδήποτε. Έπειτα η προσθήκη του τμήματος «;%M3_HOME%/bin» στη υπάρχουσα μεταβλητή path. Αυτή η διαδικασία είναι ίσως η πιο δύσκολη καθώς οι περισσότεροι χρήστες δεν έχει γνώση του τι είναι και τι κάνουν οι μεταβλητές περιβάλλοντος



Εικόνα 1

Για να σιγουρευτούμε ότι η εγκατάσταση έχει γίνει σωστά μπορούμε να εκτελέσουμε την εντολή «`mvn -v`» σε ένα παράθυρο εντολών (για παράδειγμα το command prompt των windows – `cmd.exe`). Αυτή η εντολή μας δείξει την έκδοση του λογισμικού, το φάκελο στον οποίο έχει γίνει η εγκατάσταση, την έκδοση της γλώσσας προγραμματισμού Java, το φάκελο εγκατάστασης αυτής και διάφορες άλλες πληροφορίες για το λειτουργικό σύστημα.

ΥΠΟΚΕΦΑΛΑΙΟ 1.4

Στη Maven όλες οι εντολές γράφονται σε ένα παράθυρο εντολών Windows (cmd.exe – command prompt) και ξεκινάνε με τα γράμματα mvn συντομογραφία φυσικά της λέξης Maven. Η σύνταξη μιας εντολής ακολουθεί τον εξής κανόνα: «mvn [options] [<goal(s)>] [<phase(s)>]».

Το κομμάτι [options] δεν είναι υποχρεωτικό. Είναι κάποιες επιλογές που λίγο πολύ έχουν όλες οι εντολές σε ένα σύστημα. Οι εντολές γράφονται με μικρά ή κεφαλαία γράμματα ανάλογα με την εντολή. Άλλες πρέπει να γραφούν με μικρά άλλες με κεφαλαία και άλλες φορές υπάρχουν άλλες εντολές για το κεφαλαίο και άλλες για το μικρό γράμμα. Η επιλογή -h μπορεί να μας βοηθήσει δίνοντας μας μια λίστα με όλες τις διαθέσιμες επιλογές. Μια άλλη σημαντική επιλογή είναι η επιλογή -d ή οποία μας βοηθάει να προσαρμόσουμε την συμπεριφορά των πρόσθετων Maven (plugins) τα οποία θα αναλυθούν περαιτέρω σε επόμενο κεφάλαιο.

Μερικές ακόμα επιλογές είναι η επιλογή -v ή οποία μας δίνει την έκδοση του λογισμικού Maven, ή επιλογή -P ενεργοποιεί ένα ή περισσότερα προφίλ κατασκευής. Τα προφίλ κατασκευής δίνουν τη δυνατότητα στο χρήστη να αλλάζει τις ρυθμίσεις ανάλογα με το περιβάλλον προγραμματισμού. Η επιλογή -o επιτρέπει στη Maven να εκτελεστεί χωρίς πρόσβαση στο ίντερνετ ενώ η επιλογή -f επιβάλλει τη χρήση ενός διαφορετικού αρχείου POM.xml – το αρχείο αυτό έχει διάφορες πληροφορίες για ένα έργο στη Maven.

Δυο πολύ σημαντικές επιλογές είναι οι -emp (encrypt master security password) και -ep (encrypt server password) οι οποίες κρυπτογραφούν κωδικούς όταν αυτοί είναι αποθηκευμένοι σε αρχεία ρυθμίσεων κάτι που ασφαλώς βοηθάει τον προγραμματιστή όταν εκείνος θέλει να κλειδώσει το λογισμικό που δημιουργεί με την Maven.

Οι εντολές -C και -c δίνουν τον έλεγχο στο χρήστη όσον αφορά τα απομακρυσμένα αποθετήρια (remote repositories) και συγκεκριμένα εάν ο

χρήστης θέλει να διακόψει το χτίσιμο ενός έργου ή να προειδοποιηθεί εάν τα αθροίσματα ελέγχου των ληφθέντων βιβλιοθηκών κλπ δεν ταιριάζουν μεταξύ τους.

Οι επιλογές `-nr`, `-cr` και `-ur` χρησιμεύουν στον έλεγχο της συμπεριφοράς της Maven όσον αφορά στα πρόσθετα (plugins), την μη ενημέρωση των πρόσθετων, τον έλεγχο των εκδόσεων των πρόσθετων και την ανανέωση τους για τις τρεις εντολές αντίστοιχα.

Οι λέξεις `goal` και `phase` της σύνταξης των εντολών συνδέονται με το λεγόμενο κύκλο κατασκευής (build lifecycle). Ο κύκλος αυτός αποτελείται από συγκεκριμένες φάσεις όπως προετοιμασία πόρων (prepare-resources), σύνταξη (compile), πακετάρισμα (packaging) και εγκατάσταση (install) μεταξύ άλλων. Υπάρχουν προ-φάσεις και μετά-φάσεις οι οποίες μπορούν να εκτελεστούν μετά από μια συγκεκριμένη φάση. Όταν χτίζουμε ένα έργο στην Maven η διαδικασία που ακολουθεί αποτελείται από μια ορισμένη ακολουθία φάσεων και εκτελέσεων στόχων (goals) που μπορεί να είναι συνδεδεμένοι με κάθε φάση.

Υπάρχουν τρεις καθιερωμένοι κύκλοι στην Maven. Ο κύκλος `clean` (καθαρισμός), `default` (προεπιλεγμένος, ορισμένες φορές ονομάζεται κύκλος κατασκευής) και `site` (τοποθεσία – ιστοσελίδα). Ο κύκλος καθαρισμού αποτελείται από τρεις φάσεις: η προ-φάση, η φάση καθαρισμού και η μετά-φάση. Η φάση καθαρισμού είναι η πιο σημαντική και ο σκοπός της είναι να καθαρίσει την παραγωγή του προεπιλεγμένου κύκλου κατασκευής ενός έργου διαγράφοντας τον τοπικό φάκελο που έχει δημιουργηθεί κατά την εκτέλεση μιας από τις φάσεις του.

Η εκτέλεση του κύκλου `clean` μπορεί να γίνει με την πληκτρολόγηση της εντολής `mvn clean` ενώ η πληκτρολόγηση `mvn clean:clean` δεν έχει νόημα καθώς το πρόσθετο που υλοποιεί τον κύκλο `clean` έχει μονάχα ένα στόχο, τον στόχο με όνομα `clean`. Ο χαρακτήρας «:» διαχωρίζει το πρόσθετο που υλοποιεί τον κύκλο από έναν στόχο ο οποίος ανήκει στο πρόσθετο.

Ο κύκλος `default` είναι ο πιο χρησιμοποιημένος από τους άλλους καθώς είναι ο υπεύθυνος για την κτίση ενός έργου. Παρακάτω δίνεται ένας πίνακας με τις φάσεις του κύκλου και μια σύντομη περιγραφή της καθεμίας.

Πίνακας 1

ΦΑΣΕΙΣ ΚΥΚΛΟΥ	ΠΕΡΙΓΡΑΦΗ
Επικύρωση (validate)	Επικύρωση της ορθότητας του έργου και την διαθεσιμότητα όλων των απαραίτητων πληροφοριών για την ολοκλήρωση της κτίσης
Παραγωγή πηγαίου κώδικα (generate-sources)	Παραγωγή τυχόν πηγαίου κώδικα για να συμπεριληφθεί στη μεταγλώττιση.
Κατεργασία – κώδικα (process-sources)	Κατεργασία του πηγαίου κώδικα, με σκοπό παράδειγμα για το φιλτράρισμα τιμών
Παραγωγή πόρων (generate-resources)	Παραγωγή πόρων για συμπερίληψη στο πακέτο (π.χ. αρχείο .jar)
Κατεργασία πόρων (process-resources)	Κατεργασία πόρων και αντιγραφή τους στο φάκελο προορισμού, για πακετάρισμα
Μεταγλώττιση (compile)	Μεταγλώττιση του πηγαίου κώδικα του έργου
Κατεργασία κλάσεων (process-classes)	Ύστερη επεξεργασία των δημιουργημένων αρχείων από τη μεταγλώττιση
Παραγωγή – πηγαίου κώδικα δοκιμής (generate-test-sources)	Παραγωγή τυχόν πηγαίου κώδικα δοκιμής για συμπερίληψη στην μεταγλώττιση
Κατεργασία – πηγαίου κώδικα δοκιμής (process-test-sources)	Κατεργασία του πηγαίου κώδικα δοκιμής
Παραγωγή – πόρων δοκιμής (generate-test-resources)	Δημιουργία πόρων για δοκιμή
Κατεργασία – πόρων δοκιμής (process-test-resources)	Κατεργασία και αντιγραφή των πόρων στο φάκελο προορισμού δοκιμής
Δοκιμή – μεταγλώττιση (test-compile)	Μεταγλώττιση πηγαίου κώδικα δοκιμής και τοποθέτηση του στο φάκελο προορισμού δοκιμής

ΦΑΣΕΙΣ ΚΥΚΛΟΥ	ΠΕΡΙΓΡΑΦΗ
Δοκιμή (test)	Εκτέλεση δοκιμών με χρήση ενός κατάλληλου πλαισίου δοκιμή. Αυτές οι δοκιμές δεν θα πρέπει να απαιτούν το πακετάρισμα του κώδικα ή την ανάπτυξή του (deploy).
Προετοιμασία πακέτου (prepare-package)	Εκτέλεση τυχόν λειτουργιών για την προετοιμασία του πακέτου πριν την πραγματική δημιουργία του. Αυτό συχνά οδηγεί σε μια μη-πακεταρισμένη έκδοση του πακέτου.
Δημιουργία πακέτου (package)	Πακετάρισμα του μεταγλωττισμένου κώδικα στην διανεμητέα μορφή του όπως jar war ή ear.
Δοκιμή προ-ενσωμάτωσης (pre-integration-test)	Εκτέλεση ενεργειών που απαιτούνται πριν τις δοκιμές ενσωματώσεων. Αυτό μπορεί να περιλαμβάνει δραστηριότητες όπως ρύθμιση του απαιτούμενου περιβάλλοντος.
Δοκιμή ενσωμάτωσης (integration-test)	Κατεργασία και ανάπτυξη του πακέτου εάν αυτό είναι αναγκαίο σε ένα περιβάλλον όπου δοκιμές ενσωμάτωσης μπορούν να εκτελεστούν.
Δοκιμή μετά-ενσωμάτωσης (post-integration-test)	Εκτέλεση ενεργειών έπειτα από τις δοκιμές ενσωμάτωσης. Αυτό μπορεί να συμπεριλαμβάνει καθαρισμό του περιβάλλοντος.
Επαλήθευση (verify)	Εκτέλεση ελέγχων για να επαληθευτεί ότι το πακέτο πληροί κριτήρια ποιότητας. Είναι αυτή που θα πρέπει να εκτελεστεί (οπότε και οι 3 προηγούμενες) για μια Δοκιμή Ενσωμάτωσης.
Εγκατάσταση (install)	Εγκατάσταση του πακέτου στο τοπικό αποθετήριο για να χρησιμοποιηθεί σε άλλα τοπικά έργα.
Ανάπτυξη (deploy)	Αντιγραφή του τελικού πακέτου σε απομακρυσμένα αποθετήρια για διαμοίραση με

	άλλα έργα προγραμματιστών.
--	----------------------------

Ο επόμενος κύκλος κατασκευής, ο κύκλος τοποθεσίας – ιστοσελίδας (site) είναι υπεύθυνος για την τεκμηρίωση του έργου (project documentation). Περιέχει τέσσερις φάσεις:

- προ-κύκλου κατασκευής
- κύκλος κατασκευής
- μετά-κύκλου
- ανάπτυξη τοποθεσίας

Οι προεπιλεγμένοι στόχοι (goals) αυτού του κύκλου είναι: site και site-deploy^[3]. Ο τύπος πακέτου συχνά δεν μεταβάλλει τον κύκλο αυτόν αφού οι τύποι πακέτων αφορούν κυρίως την δημιουργία αντικειμένου (artifact) όχι τον τύπο ιστοσελίδας που παράχθηκε. Το πρόσθετο (plugins) τοποθεσία – ιστοσελίδα δίνει το έναυσμα για την δημιουργία ενός έγγραφο Doxia. Η εντολή που δημιουργεί μια τοποθεσία είναι η mvn site και εκτελεί όλο τον κύκλο κατασκευής.

Στις επόμενες εικόνες δίνεται ένα παράδειγμα εκτέλεσης στόχων.

```

C:\Users\Babis\first_project>mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building first_project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ first_project ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\Babis\first_project\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ first_project ---
[INFO] Compiling 1 source file to C:\Users\Babis\first_project\target\classes
[INFO]
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.048s
[INFO] Finished at: Sun May 04 22:48:40 EEST 2014
[INFO] Final Memory: 11M/110M
[INFO] -----
C:\Users\Babis\first_project>
    
```

Εικόνα 2

```

C:\Users\Babis\first_project>mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building first_project 1.0-SNAPSHOT
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ first_project ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\Babis\first_project\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ first_project ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ first_project ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\Babis\first_project\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ first_project ---
[INFO] Compiling 1 source file to C:\Users\Babis\first_project\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ first_project ---
[INFO] Surefire report directory: C:\Users\Babis\first_project\target\surefire-reports

-----
T E S T S
-----
Running com.example.website.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ first_project ---
[INFO] Building jar: C:\Users\Babis\first_project\target\first_project-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ first_project ---
[INFO] Installing C:\Users\Babis\first_project\target\first_project-1.0-SNAPSHOT.jar to C:\Users\Babis\.m2\repository\com\example\website\first_project\1.0-SNAPSHOT\first_project-1.0-SNAPSHOT.jar
[INFO] Installing C:\Users\Babis\first_project\pom.xml to C:\Users\Babis\.m2\repository\com\example\website\first_project\1.0-SNAPSHOT\first_project-1.0-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.942s
[INFO] Finished at: Sun May 04 22:49:56 EEST 2014
[INFO] Final Memory: 12M/114M
[INFO] -----

```

Εικόνα 3

Στην εικόνα αυτή βλέπουμε πως η Maven δεν ξανακάνει compile καθώς το πρόγραμμα έχει ήδη μεταγλωττιστεί.

ΚΕΦΑΛΑΙΟ 2

Παρουσίαση των πρόσθετων της Maven (Maven Plugins)

ΕΙΣΑΓΩΓΗ

Το λογισμικό της τεχνολογίας Maven από μόνο του δεν μπορεί να κάνει και πολλά. Στην πραγματικότητα είναι αρκετά «χαζό». Το μόνο που ξέρει να κάνει είναι επεξεργάζεται/αναλύει αρχεία τύπου xml και να παρακολουθεί ένα κύκλο κατασκευής και λίγα πρόσθετα. Η Maven είναι σχεδιασμένη να αναθέτει εργασίες σε μια συλλογή προσθέτων τα οποία πραγματοποιούν τις περισσότερες εργασίες και προσφέρουν πρόσβαση σε στόχους (goals).

Κάποιες από αυτές τις εργασίες είναι να μεταγλωττίζουν πηγαίο κώδικα, να δημιουργούν το πακέτο μεταγλωττισμένου κώδικα (package bytecode) δημοσιεύουν τοποθεσίες – ιστοσελίδες (sites) και οτιδήποτε άλλο που χρειάζεται να γίνει. Τέτοια πρόσθετα λαμβάνονται από τη Maven αυτόματα από μια κεντρική τοποθεσία αποθετηρίου Maven χωρίς ο χρήστης να κάνει το οτιδήποτε εκτός από το να αποκτήσει πρόσβαση στο Ίντερνετ.

Έτσι τα πρόσθετα χρησιμοποιούνται ξανά και ξανά σε κάθε χρήση της Maven αποτελώντας το απόλυτο επαναχρησιμοποιήσιμο λογισμικό. Υπάρχει επίσης η δυνατότητα για αναβάθμιση των προσθέτων από το κεντρικό αποθετήριο προσθέτων αυξάνοντας έτσι τις λειτουργίες των έργων που δημιουργούνται/διαχειρίζονται από τη Maven.

ΥΠΟΚΕΦΑΛΑΙΟ 2.1

Τα πρόσθετα, καθώς και όποιες εξαρτήσεις των έργων της Maven όταν αυτά χρειαστούν λαμβάνονται σε ένα σύστημα και αποθηκεύονται σε έναν φάκελο του συστήματος. Ο φάκελος αυτός (εάν δεν έχει αλλαχτεί στις ρυθμίσεις των προσθέτων) μπορεί να βρεθεί στην διαδρομή `#{M2_HOME}\\.m2` και συγκεκριμένα

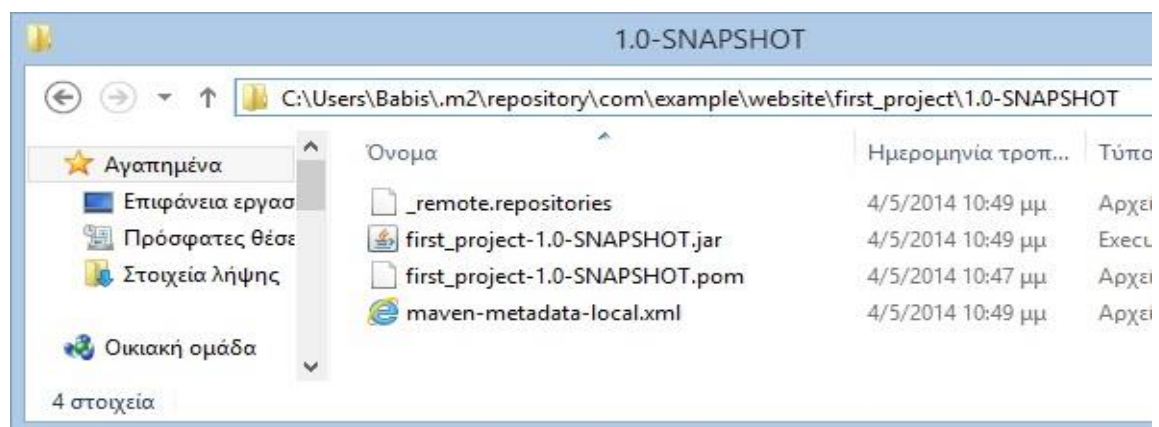
«C:\Users\{username}\\.m2» ή

«C:\Documents and Settings\{your-username}\\.m2»

του λειτουργικού Windows ενώ σε Unix/Linux

«/home/{username}/.m2/repository».

Στην εικόνα βλέπουμε το έργο από τις προηγούμενες εικόνες. Μετά την εκτέλεση της εντολής `mvn install` το έργο αυτό τοποθετήθηκε στην διαδρομή που βλέπουμε στην παρακάτω εικόνα. Το κομμάτι της διαδρομής μετά από τον φάκελο `repository` έχει να κάνει με το `GroupId` του έργου καθώς και το `ArtifactId`.



Εικόνα 4

Στον φάκελο αυτό τοποθετούνται όλες οι εξαρτήσεις (dependencies) και πρόσθετα (plugins) καθώς είναι η προεπιλεγμένη διαδρομή της Maven. Τα πρόσθετα τοποθετούνται στην διαδρομή «org\apache\maven\plugins» μέσα στον παραπάνω φάκελο.

Ένα παράδειγμα εξάρτησης ενός έργου θα ήταν με τη χρήση της εντολής import. Αυτό που ακολουθεί εάν δεν υπάρχει στο τοπικό σύστημα αναζητείται στο απομακρυσμένο αποθετήριο της Maven. Η τοποθεσία αυτή ακολουθεί την καθιερωμένη ονοματολογία ενός πακέτου (package) της Java.

Η πλήρης λίστα προσθέτων της Maven, τουλάχιστον αυτά που προσφέρονται από το λογισμικό μπορεί να βρεθεί στην ιστοσελίδα <http://maven.apache.org/plugins/index.html> καθώς και πληροφορίες του τύπου σε ποιά έκδοση της Maven προστέθηκαν, ποιά ημερομηνία μια σύντομη περιγραφή κ.α. ενώ κατά την επιλογή ενός από αυτά με το ποντίκι μπορούμε να δούμε αναλυτικές πληροφορίες για το καθένα όπως περιγραφή, επισκόπηση στόχων και παραδείγματα:

ΥΠΟΚΕΦΑΛΑΙΟ 2.2

Υπάρχουν πέντε κατηγορίες προσθέτων. Τα πρόσθετα πυρήνα (core plugins), τα πρόσθετα δημιουργίας πακέτων (packaging plugins), τα πρόσθετα αναφοράς (reporting plugins) τα εργαλεία (tools) και τέλος τα πρόσθετα Ολοκληρωμένου περιβάλλοντος ανάπτυξης^[4].

Στην πρώτη κατηγορία ανήκουν τα πρόσθετα που υλοποιούν φάσεις των κύκλων κατασκευής ή τους ίδιους τους κύκλους όπως clean, compiler, deploy, failsafe, install, resources, site, surefire και verifier. Το πρόσθετο clean είναι αυτό που υλοποιεί τον ομώνυμο κύκλο κατασκευής που περιγράφηκε παραπάνω.

Το πρόσθετο compiler μεταγλωττίζει τα αρχεία πηγαίου κώδικα ενός έργου σε αρχείο/α .class. Από την έκδοση 3.0 και μετά ο προεπιλεγμένος μεταγλωττιστής είναι ο javax.tools.JavaCompiler, εάν η έκδοση της Java είναι 1.6. Δίνεται η δυνατότητα στον χρήστη της Maven να εξαναγκάσει το πρόσθετο να χρησιμοποιήσει τον μεταγλωττιστή javac με διαμόρφωση της ρύθμισης του προσθέτου forceJavacCompilerUse. Επίσης υπάρχει η δυνατότητα για χρήση των μεταγλωττιστών AspectJ, .NET και C#.

Το πρόσθετο αυτό έχει δύο στόχους: compiler:compile και compiler:testCompile και είναι συνδεδεμένοι με τις αντίστοιχες φάσεις του κύκλου κατασκευής και εκτελούνται αυτόματα κατά τη διάρκεια αυτών των φάσεων.

Το πρόσθετο install εκτελείται κατά την αντίστοιχη ομώνυμη φάση και χρησιμοποιεί πληροφορίες από το αρχείο POM (groupId, artifactId και version) για να προσδιορίσει την σωστή τοποθεσία του αντικειμένου (artifact) στο τοπικό αποθετήριο.

Περιέχει τρεις στόχους: install, install-file και help. Ο πρώτος χρησιμοποιείται αυτόματα για την εγκατάσταση του κυρίου αντικειμένου (Jar, War ή Ear), το POM του και τυχόν επισυναπτόμενα αντικείμενα (πηγές, javadoc κ.α.) που έχουν παραχθεί από κάποιο έργο.

Ο δεύτερος στόχος, εγκαθιστά εξωτερικά αντικείμενα στο τοπικό αποθετήριο και τα POM τους. Τέλος, ο τρίτος στόχος εμφανίζει πληροφορίες βοήθειας για το πρόσθετο.

Το πρόσθετο deploy υλοποιεί την ομώνυμη φάση του προεπιλεγμένου κύκλου. Διαμοιράζεται ένα αντικείμενο (artifact) με ένα απομακρυσμένο αποθετήριο και περιέχει δύο στόχους. Τον ομώνυμο στόχο deploy και τον στόχο deploy-file. Ο πρώτος στόχος χρησιμοποιείται αυτόματα^[5] και εγκαθιστά ένα αντικείμενο, το POM του και όποια επισυναπτόμενα αντικείμενα από το έργο στο απομακρυσμένο αποθετήριο ενώ ο δεύτερος εγκαθιστά ένα αντικείμενο μαζί με το POM του.

Το πρόσθετο failsafe είναι υπεύθυνο για την εκτέλεση δοκιμών ενσωμάτωσης/ένταξης του πλαισίου (framework) JUnit σε απομονωμένη σειρά κλάσεων. Αυτό εκτελείται κατά τις φάσεις Δοκιμή ενσωμάτωσης και Επαλήθευση. Αυτό το πρόσθετο δεν θα αποτύχει κατά την φάση Δοκιμή Ενσωμάτωσης οπότε επιτρέπει την εκτέλεση της φάση Δοκιμή μετά-ενσωμάτωσης.

Αυτό το πρόσθετο διαθέτει δύο στόχους. Τον στόχο integration-test και verify. Ο πρώτος εκτελεί τις δοκιμές ενσωμάτωσης ενός έργου ενώ ο δεύτερος στόχος επαληθεύει ότι οι δοκιμές ενσωμάτωσης ήταν επιτυχείς.

Το πρόσθετο resources είναι υπεύθυνο για την φάση process-resources του παραπάνω πίνακα του προεπιλεγμένου κύκλου. Είναι υπεύθυνο για την αντιγραφή πόρων στον κατάλογο προορισμού. Υπάρχουν δυο είδη πόρων. Οι κύριοι πόροι (main resources) και πόροι δοκιμής (test resources). Οι μεν είναι συνδεδεμένοι με τον κυρίως πηγαίο κώδικα (main source code) ενώ οι δε είναι συνδεδεμένοι με τον πηγαίο κώδικα δοκιμής (test source code).

Υπάρχουν τρεις στόχοι σε αυτό το πρόσθετο. Ο στόχος resources αντιγράφει τους πόρους του κυρίως πηγαίου κώδικα στο κυρίως κατάλογο προορισμού. Αυτός ο στόχος συνήθως εκτελείται αυτόματα καθώς είναι συνδεδεμένος με την φάση του κύκλου ζωής process-resources^[6]. Χρησιμοποιεί πάντα το στοιχείο (element) project.build.resources για να καθορίσει τους πόρους και τον κατάλογο προορισμού project.build.outputDirectory για να καθορίσει τον προορισμό αντιγραφής.

Ο δεύτερος στόχος είναι ο `testResources` και πραγματοποιεί το δεύτερο κομμάτι – αντιγραφή των πόρων πηγαίου κώδικα δοκιμής στον κατάλογο προορισμού δοκιμής. Επίσης όπως και ο πρώτος στόχος συνήθως εκτελείται αυτόματα και χρησιμοποιεί το στοιχείο `project.build.testResources` για να καθορίσει τους πόρους και τον κατάλογο προορισμού `project.build.testOutputDirectory` για να καθορίσει τον προορισμό αντιγραφής.

Ο τρίτος και τελευταίος στόχος του προσθέτου είναι ο `copy-resources` ο οποίος αντιγράφει τους πόρους σε ένα κατάλογο προορισμού. Αυτός ο στόχος απαιτεί την διαμόρφωση των πόρων προς αντιγραφή και τον καθορισμό ενός καταλόγου προορισμού.

Το πρόσθετο `site` έχει στην πραγματικότητα δύο εκδόσεις. Μια για την έκδοση 2.2.0 (και μετά) της Maven και μια για την έκδοση 3. Βέβαια αυτές οι δυο εκδόσεις είναι παρόμοιες και έχουν τους ίδιους στόχους και πραγματοποιούν τις ίδιες λειτουργίες. Το πρόσθετο είναι υπεύθυνο για την δημιουργία μιας ιστοσελίδας για ένα έργο. Η ιστοσελίδα συμπεριλαμβάνει τις αναφορές ενός έργου οι οποίες είναι διαμορφωμένες στην ετικέτα `<reporting>` του POM του έργου.

Το πρόσθετο έχει οκτώ στόχους. Ο πρώτος έχει την ίδια ονομασία με το πρόσθετο, `site`, και δημιουργεί την ιστοσελίδα για ένα έργο. Ο στόχος `deploy` είναι αυτός που φορτώνει τα αρχεία της ιστοσελίδας με πρωτόκολλα `scp` ή `file` στο URL της ιστοσελίδας όπως αυτό είναι ρυθμισμένο στο τμήμα `<distributionManagement>` του POM.

Ο επόμενος στόχος είναι ο `run` και είναι αυτός που ξεκινάει την ιστοσελίδα οπτικοποιώντας (`rendering`) τα έγγραφα όπως αυτά ζητήθηκαν για γρηγορότερη επεξεργασία. Αυτό επίσης χρησιμοποιεί το `jetty` ως διακομιστή ιστού^[7]. Ο στόχος `stage` δημιουργεί μια ιστοσελίδα σε ένα τοπικό ή ψεύτικο κατάλογο βασισμένο στο URL της ιστοσελίδας ενώ ο στόχος `stage-deploy` είναι αυτός που το αναπτύσσει.

Ο στόχος `attach-descriptor` προσθέτει μια περιγραφή της ιστοσελίδας (για παράδειγμα `site.xml`) στην λίστα αρχείων που πρόκειται να εγκατασταθούν/αναπτυχθούν. Ο στόχος `jar` μαζεύει την ιστοσελίδα σε ένα αρχείο

jar για ανάπτυξη στο αποθετήριο. Τέλος ο στόχος effective-site υπολογίζει την πραγματική περιγραφή της ιστοσελίδας έπειτα από κληρονομιά (inheritance) και παρεμβολή (interpolation).

Το πρόσθετο surefire χρησιμοποιείται κατά τη διάρκεια της φάσης δοκιμή του κύκλου και εκτελεί δοκιμές μονάδων (unit tests) ενός έργου. Οι αναφορές του είναι δυο ειδών: απλού αρχείου κειμένου (txt) και xml. Αυτές οι αναφορές τοποθετούνται στον φάκελο `\target\surefire-reports` του έργου εκτός εάν ρυθμιστεί κάτι διαφορετικό. Το πρόσθετο αυτό έχει μόνο ένα στόχο. Τον στόχο test ο οποίος εκτελεί τις δοκιμές μονάδων ενός έργου.

Τέλος το πρόσθετο verifier είναι αυτό που επαληθεύει την ύπαρξη κάποιων συνθηκών για τις δοκιμές ενσωμάτωσης. Έχει ένα στόχο με την ονομασία verify και επαληθεύει την ύπαρξη ή μη αρχείων ή καταλόγων και προαιρετικά ελέγχει τα περιεχόμενα αρχείων με μια κανονική έκφραση (regular expression).

Μερικά πρόσθετα δημιουργίας πακέτων είναι αυτά που δημιουργούν αρχεία τύπου ear, ejb, jar, rar, war. Το ear δημιουργεί ένα αρχείο J2EE Enterprise Archive (EAR) και περιέχει δυο στόχους, ear και generate-application-xml ο πρώτος δημιουργεί το αρχείο ενώ ο δεύτερος την ανάπτυξη αρχείων περιγραφέα (descriptor). Το ejb πρόσθετο δημιουργεί ένα αρχείο J2EE Enterprise Javabeen (EJB) και έχει ένα μόνο ομώνυμο στόχο.

Το πρόσθετο jar δημιουργεί και υπογράφει αρχεία jar και έχει έναν στόχο jar και έναν στόχο test-jar. Ένας είναι για τις πηγές του έργου και ο δεύτερος στόχος για τα αρχεία class των δοκιμών του έργου. Το rar πρόσθετο (resource adapter) είναι ένας οδηγός λογισμικού του συστήματος ο οποίος χρησιμοποιείται από τη java για την σύνδεση σε ένα σύστημα EIS (Enterprise Information System)^[8]. Έχει την δυνατότητα αποθήκευσης των προσαρμογών πόρων (resource adapters) σε ένα αρχείο τύπου archive το οποίο μπορεί να αναπτυχθεί σε ένα διακομιστή J2EE. Επίσης έχει ένα στόχο, τον rar.

Στην εικόνα που ακολουθεί βλέπουμε ένα παράδειγμα αυτόματης εκτέλεσης του προσθέτου jar. Αυτή η εικόνα είναι στην πραγματικότητα κομμάτι προηγούμενης εικόνας όπου εκτελέσαμε τον στόχο install.

```

T E S T S
-----
Running com.example.website.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ first_project ---
[INFO] Building jar: C:\Users\Babis\first_project\target\first_project-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ first_project ---
[INFO] Installing C:\Users\Babis\first_project\target\first_project-1.0-SNAPSHOT.jar to C:\Users\Babis\.m2\repository\com\example\website\first_project\1.0-SNAPSHOT\first_project-1.0-SNAPSHOT.jar
[INFO] Installing C:\Users\Babis\first_project\pom.xml to C:\Users\Babis\.m2\repository\com\example\website\first_project\1.0-SNAPSHOT\first_project-1.0-SNAPSHOT.pom
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.942s
[INFO] Finished at: Sun May 04 22:49:56 EEST 2014
[INFO] Final Memory: 12M/114M
[INFO]

```

Εικόνα 5

Το επόμενο πρόσθετο, το `war`, είναι αυτό που δημιουργεί ένα πακέτο με εξαρτήσεις, αρχεία κλάσεων και πόρους μιας εφαρμογής ιστού σε ένα αρχείο τύπου `war`. Έχει τέσσερις στόχους: έναν στόχο `war` ο οποίος είναι ο προεπιλεγμένος στόχος που εκτελείται κατά τη διάρκεια της φάσης Δημιουργία Πακέτου (`package`), έναν στόχο `exploded` ο οποίος επιταχύνει την δοκιμή κατά τη διάρκεια της ανάπτυξης του έργου δημιουργώντας μια εφαρμογή ιστού `exploded` (`exploded webapp`) σε ένα συγκεκριμένο κατάλογο. Ένας άλλος στόχος με όνομα `inplace` είναι μια παραλλαγή του προηγούμενου στόχου όπου η εφαρμογή ιστού δημιουργείται στο πηγαίο κατάλογο της εφαρμογής, η προεπιλεγμένη διαδρομή του οποίου είναι: `src/main/webapp`. Τέλος ο στόχος `manifest` δημιουργεί ένα αρχείο `manifest` για την εφαρμογή ιστού τοποθετείται στον πηγαίο κατάλογο της εφαρμογής.

Το πρόσθετο `ACR` δημιουργεί ένα αρχείο `J2EE Application Client` και έχει ένα στόχο με το ίδιο όνομα. Χρησιμοποιείται σε έργα με τύπο πακέτου `app-client`. Τέλος το πρόσθετο `shade` δημιουργεί ένα υπέρ-`jar` αρχείο με το ισχύων έργο συμπεριλαμβανομένου και τις ίδιες τις εξαρτήσεις του, ενώ κανονικά θα είχε μόνο τις πληροφορίες για να λάβει τις εξαρτήσεις μέσω ίντερνετ αλλά όχι τις ίδιες τις εξαρτήσεις (`dependencies`). Έχει μόνο ένα ομώνυμο στόχο οποίος είναι

συνδεδεμένος με την φάση Δημιουργία Πακέτου και χρησιμοποιείται για τη δημιουργία αρχείου jar τύπου shade.

Τα πρόσθετα αναφοράς είναι αυτά που δημιουργούν αναφορές, είναι ρυθμισμένα σαν αναφορές στο αρχείο POM και εκτελούνται κατά τον κύκλο δημιουργίας site (ο τρίτος κύκλος που αναφέρθηκε)

Κάποια από αυτά είναι τα changelog, changes, checkstyle, doap, docck, javadoc, jxr, linkcheck, pmd, project-info-reports και surefire-report.

Το πρόσθετο changelog παράγει, όπως θα περίμενε κανείς, μια λίστα με τις τελευταίες αλλαγές στο λογισμικό διαχείρισης διαμόρφωσης (Software Configuration Management). Έχει τρεις στόχους: changelog – δημιουργεί την αναφορά, dev-activity – δημιουργεί την αναφορά δραστηριότητας προγραμματιστή και τέλος file-activity – δημιουργεί την αναφορά δραστηριότητας αρχείου.

Το πρόσθετο changes δημιουργεί μια αναφορά με έναν ανιχνευτή ζητήματος (issue tracker) ή ένα μεταβεβλημένο αρχείο. Περιέχει οκτώ στόχους: announcement-mail – στέλνει μια ανακοίνωση νέας έκδοσης μέσω ηλεκτρονικής αλληλογραφίας, announcement-generate – δημιουργεί μια ανακοίνωση νέας έκδοσης, changes-check – ελέγχει ότι το αρχείο changes.xml περιέχει μια έγκυρη ημερομηνία, changes-report – δημιουργεί μια αναφορά δείχνοντας όλες τις αλλαγές που έχει υποστεί το έργο, changes-validate – επικυρώνει το αρχείο changes.xml, jira-report – δημιουργεί μια αναφορά με ζητήματα που έχουν ληφθεί από το λογισμικό JIRA, trac-report – δημιουργεί μια αναφορά με ζητήματα που έχουν ληφθεί από το λογισμικό Trac και τέλος github-report – δημιουργεί μια αναφορά με ζητήματα που έχουν ληφθεί από το λογισμικό GitHub.

Το plugin ckeckstyle δημιουργεί μια ομώνυμη αναφορά σχετικά με το στυλ κώδικα που έχει χρησιμοποιηθεί από τους προγραμματιστές χρησιμοποιώντας το επίσης ομώνυμο εξωτερικό εργαλείο προγραμματισμού Checkstyle 5.6. Έχει τρεις στόχους. Ο πρώτος, ομώνυμος με το πρόσθετο πραγματοποιεί μια ανάλυση τύπου Checkstyle και δημιουργεί μια αναφορά με παραβιάσεις. Ο δεύτερος στόχος checkstyle-aggregate πραγματοποιεί τις ίδιες ενέργειες αλλά επιπλέον συναθροίζει το αποτέλεσμα στο έργο το οποίο εκκίνησε τον στόχο. Ο

τρίτος και τελευταίος στόχος είναι ο check ο οποίος πραγματοποιεί έναν έλεγχο παραβίασης συγκρίνοντας τον με την τελευταία ανάλυση τύπου Checkstyle για να ελέγξει τυχόν παραβιάσεις. Διαβάζει το αρχείο εξόδου Checkstyle μετράει τον αριθμό των παραβιάσεων και τον προβάλλει στην κονσόλα εντολών.

Το πρόσθετο DOAP (Description of a Project) δημιουργεί μια υποχρεωτική περιγραφή του έργου χρησιμοποιώντας το αρχείο POM.xml. Ο σκοπός του προσθέτου είναι να παρέχει αρχεία τύπου DOAP για τον συστήματα σημασιολογικού ιστού τα οποία τα χρησιμοποιούν σαν κύρια είσοδο αλλά που επίσης θα ελαφρύνουν το φορτίο της διατήρησης δύο συλλογές από μεταδεδομένων. Έχει ένα και μόνο στόχο με το όνομα generate.

Τα Docck και Javadoc σχετίζονται με την δημιουργία τεκμηρίωσης (documentation) το δεύτερο δημιουργώντας την και το πρώτο ελέγχοντάς την. Το πρόσθετο Docck έχει ένα στόχο με το όνομα check. Αυτό που ελέγχει είναι να συμμορφώνεται με τα ελάχιστα της καθιερωμένης τεκμηρίωσης προσθέτων (plugin documentation standard). Το Javadoc έχει δέκα στόχους.

- javadoc: δημιουργεί τα αρχεία Javadoc για το έργο. Εκτελεί το εργαλείο Javadoc και υποστηρίζει τις παραμέτρους που χρησιμοποιήθηκαν από το εργαλείο.
- test-javadoc: δημιουργεί τα αρχεία δοκιμής Javadoc για το έργο. Εκτελεί το εργαλείο Javadoc και υποστηρίζει τις παραμέτρους που χρησιμοποιήθηκαν από το εργαλείο.
- aggregate: δημιουργεί τα αρχεία Javadoc για ένα έργο τύπου aggregator. Εκτελεί το εργαλείο Javadoc και υποστηρίζει τις παραμέτρους που χρησιμοποιήθηκαν από το εργαλείο.
- test-aggregate: δημιουργεί τα αρχεία δοκιμής Javadoc για ένα έργο τύπου aggregator. Εκτελεί το εργαλείο Javadoc και υποστηρίζει τις παραμέτρους που χρησιμοποιήθηκαν από το εργαλείο.
- jar: δημιουργεί ένα αρχείο archive των δημιουργημένων αρχείων javadoc. Δημιουργεί ένα αντικείμενο Javadoc το οποίο μεταφορτώνεται στο απομακρυσμένο αποθετήριο μαζί με το μεταγλωττισμένο αρχείο του έργου και το πηγαίο αρχείο archive.

- `test-jar`: δημιουργεί ένα αρχείο `archive` με τα δημιουργημένα αρχεία δοκιμής `Javadoc`.
- `aggregate-jar`: δημιουργεί ένα αρχείο `archive` με τα δημιουργημένα αρχεία `Javadoc` για ένα έργο τύπου `aggregator`.
- `test-aggregate-jar`: δημιουργεί ένα αρχείο `archive` με τα δημιουργημένα αρχεία δοκιμής `Javadoc` για ένα έργο τύπου `aggregator`.
- `fix`: αυτός είναι ένας διαδραστικός στόχος ο οποίος καθορίζει την τεκμηρίωση `Javadoc` και τις ετικέτες για τα αρχεία `java`.
- `test-fix`: είναι ένας διαδραστικός στόχος ο οποίος καθορίζει την τεκμηρίωση `Javadoc` και τις ετικέτες για τα αρχεία δοκιμής `java`.

Το `jar` δημιουργεί μια διασταυρούμενη αναφορά (`cross-reference`) του κώδικα του έργου η οποία διευκολύνει τον χρήστη να παραπέμψει ή να αναζητήσει συγκεκριμένες γραμμές στον κώδικα. Έχει τέσσερις στόχους: έναν ομώνυμο στόχο που δημιουργεί μια σελίδα διασταυρούμενης αναφοράς των κυρίως πηγών του έργου. Έναν στόχο `aggregate` ο οποίος δημιουργεί μια συγκεντρωτική σελίδα διασταυρούμενης αναφοράς των κυρίως πηγών του έργου. Έναν στόχο `test-jar` ο οποίος δημιουργεί μια σελίδα διασταυρούμενης αναφοράς των πηγών δοκιμής του έργου. Τέλος έναν στόχο `test-aggregate` ο οποίος δημιουργεί μια συγκεντρωτική σελίδα διασταυρούμενης αναφοράς των πηγών δοκιμής του έργου.

Το πρόσθετο `linkcheck` δημιουργεί μια αναφορά `Linkcheck` της τεκμηρίωσης του έργου και περιέχει μόνο έναν ομώνυμο στόχο. Ο στόχος αυτός χρησιμοποιεί ένα εξωτερικό εργαλείο με όνομα `Doxia Linkcheck Tool`.

Ένα άλλο πρόσθετο είναι το `pmf` το οποίο επίσης δημιουργεί μια αναφορά τύπου `pmf` και έχει τέσσερις στόχους. Έναν ομώνυμο στόχο ο οποίος δημιουργεί μια αναφορά `PMD` της ιστοσελίδας βασισμένη σε κανόνες και στη διαμόρφωση του προσθέτου. Επίσης μπορεί να δημιουργήσει ένα αρχείο εξόδου τύπου `PMD` εκτός από την αναφορά της ιστοσελίδας σε μορφή `xml`, `csv` ή `txt`. Ένας άλλος στόχος είναι ο `cpd` ο οποίος δημιουργεί μια αναφορά για το εργαλείο ανιχνευτή αντιγραφής/επικόλλησης (`Copy/Paste Detector`) του `PMD`. Μπορεί επίσης να δημιουργήσει ένα αρχείο με αποτελέσματα τύπου `cpd` σε μορφή `xml`, `csv` ή `txt`.

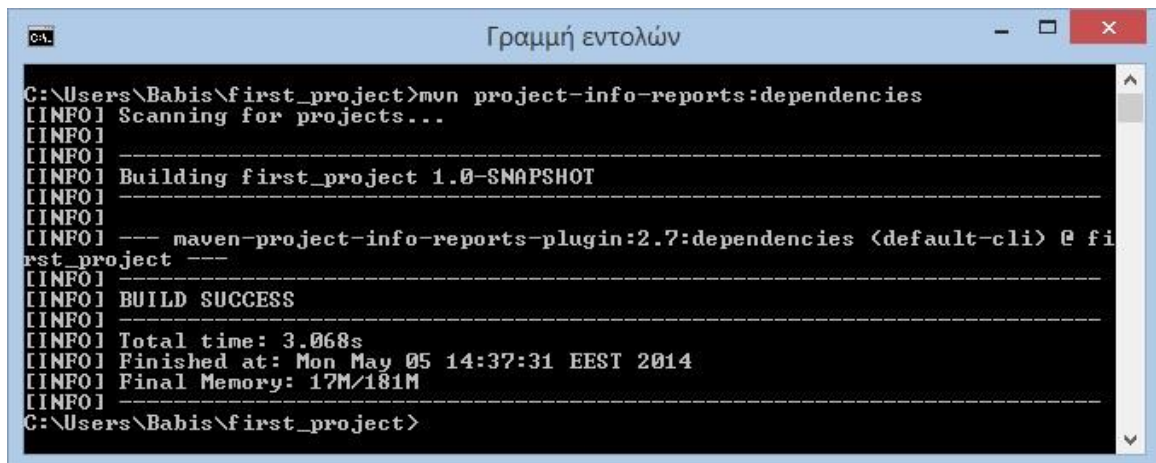
Ο επόμενος στόχος είναι ο `check` και επαληθεύει ότι η αναφορά `PMD` είναι άδεια και εάν δεν είναι αποτυγχάνει την κατασκευή του έργου. Αυτός ο στόχος εκτελείται από προεπιλογής όταν ο εκτελείται στόχος `rmc:rmc`. Τέλος ο στόχος `cpd-check` επαληθεύει ότι η αναφορά `CPD` είναι άδεια και εάν δεν είναι αποτυγχάνει την κατασκευή του έργου. Αυτός ο στόχος εκτελείται από προεπιλογής όταν ο εκτελείται στόχος `rmc:cpd`.

Το `project-info-reports` δημιουργεί διάφορες αναφορές με πληροφορίες του έργου. Επίσης απαιτεί την έκδοση 2.1 και μετά του προσθέτου `site`. Έχει τους εξής στόχους:

- `aim`: χρησιμοποιείται για να δημιουργήσει την αναφορά του έργου συνεχής ολοκλήρωσης συστήματος.
- `Dependencies`: χρησιμοποιείται για να δημιουργήσει την αναφορά των εξαρτήσεων του έργου.
- `dependency-convergence`: χρησιμοποιείται για να δημιουργήσει την αναφορά σύγκλιση εξαρτήσεων του έργου.
- `dependency-info`: χρησιμοποιείται για να δημιουργήσει αποσπάσματα κώδικα για να προστεθούν στα εργαλεία κατασκευής.
- `dependency-management`: χρησιμοποιείται για να δημιουργήσει την αναφορά διαχείριση εξαρτήσεων του έργου.
- `distribution-management`: χρησιμοποιείται για να δημιουργήσει την αναφορά διανομή εξαρτήσεων του έργου.
- `help`: χρησιμοποιείται για να προβάλει πληροφορίες βοήθειας για το πρόσθετο αναφορές πληροφοριών έργου.
- `index`: δημιουργεί την σελίδα δείκτες έργου.
- `issue-tracking`: δημιουργεί την αναφορά παρακολούθηση ζητημάτων έργου.
- `license`: χρησιμοποιείται για να δημιουργήσει την αναφορά άδεια έργου.
- `mailing-list`: χρησιμοποιείται για να δημιουργήσει την αναφορά ηλεκτρονικές ταχυδρομικές λίστες έργου.

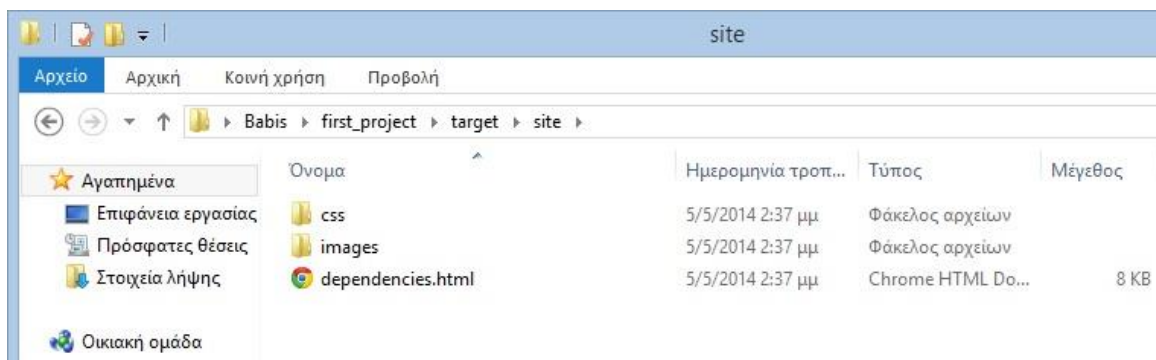
- `modules`: χρησιμοποιείται για να δημιουργήσει την αναφορά ενότητες έργου.
- `plugin-management`: χρησιμοποιείται για να δημιουργήσει την αναφορά διαχείριση προσθέτων έργου.
- `plugins`: χρησιμοποιείται για να δημιουργήσει την αναφορά προσθέτων έργου.
- `project-team`: χρησιμοποιείται για να δημιουργήσει την αναφορά ομάδα έργου.
- `scm`: χρησιμοποιείται για να δημιουργήσει την αναφορά διαχείριση πηγαίου κώδικα του έργου.
- `summary`: χρησιμοποιείται για να δημιουργήσει το σύνολο αναφορών πληροφοριών έργου.

Στις παρακάτω εικόνες βλέπουμε το αποτέλεσμα της εκτέλεσης του προσθέτου `project-info-reports` και του στόχου του `Dependencies`.




```
C:\Users\Babis\first_project>mvn project-info-reports:dependencies
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building first_project 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-project-info-reports-plugin:2.7:dependencies (default-cli) @ fi
rst_project ---
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 3.068s
[INFO] Finished at: Mon May 05 14:37:31 EEST 2014
[INFO] Final Memory: 17M/181M
[INFO] -----
C:\Users\Babis\first_project>
```

Εικόνα 6



Εικόνα 7

Last Published: 2014-05-05 | Version: 1.0-SNAPSHOT



Project Dependencies

test

The following is a list of test dependencies for this project. These dependencies are only required to compile and run unit tests for the application:

GroupId	ArtifactId	Version	Type	License
junit	junit ↗	3.8.1	jar	Common Public License Version 1.0 ↗

Project Transitive Dependencies

No transitive dependencies are required for this project.

Project Dependency Graph

Dependency Tree

- com.example.website:first_project:jar:1.0-SNAPSHOT [↓](#)
 - junit:junit:jar:3.8.1 (test) [↓](#)

Licenses

Unknown: first_project

Common Public License Version 1.0: JUnit

Dependency File Details

Filename	Size	Entries	Classes	Packages	JDK Rev	Debug
junit-3.8.1.jar	118,23 kB	119	100	6	1.1	debug
Total	Size	Entries	Classes	Packages	JDK Rev	Debug

Εικόνα 8

Το πρόσθετο surefire-report αναλύει τα αρχεία TEST-*.xml στον κατάλογο `${basedir}\target\surefire-reports` και τα παρέχει στο εργαλείο DOXIA το οποίο δημιουργεί μια διεπαφή ιστού των αποτελεσμάτων της δοκιμής

Τα πρόσθετα που θα αναφέρουμε παρακάτω είναι διαθέσιμα με τη Maven και πραγματοποιούν διάφορες ενέργειες. Αυτά είναι τα: ant, antrun, archetype, assembly, dependency, enforcer, gpg, help, invoker, jarsigner, patch, pdf,

plugin , release, reactor, remote-resources, repository, scm, scm-publish,source, stage και toolchains.

Το ant δημιουργεί αρχεία κατασκευής Ant του ισχύων έργου. Έχει δύο στόχους τους ant και clean. Ο στόχος ant δημιουργεί τα αρχεία κατασκευής Ant ενώ ο δεύτερος διαγράφει όλα τα αρχεία κατασκευής Ant.

Το antrun εκτελεί μια σειρά εργασιών Ant μιας φάσης της κατασκευής μέσω της Maven. Έχει ένα στόχο με την ονομασία run ο οποίος τρέχει εργασίες Ant (Ant tasks) για λόγο της Maven.

Το πρόσθετο archetype χρησιμοποιείται στη δημιουργία έργων χρησιμοποιώντας ως πρότυπο ένα έργο από μια συλλογή έργων τα οποία ονομάζονται αρχέτυποι (archetype). Το αντίθετο είναι επίσης πιθανό, δηλαδή την δημιουργία αρχέτυπου έργου χρησιμοποιώντας το ισχύων έργο Maven. Έχει επτά στόχους.

Τον στόχο create ο οποίος είναι ένας παρωχημένος στόχος και χρησιμοποιεί την συμπεριφορά της έκδοσης 1.0-alpha-7 του πρόσθετου archetype. Ο στόχος generate ο οποίος δημιουργεί ένα έργο από μια λίστα αρχέτυπων έργων δίνοντας μια λίστα με όλα τα αρχέτυπα διαθέσιμα έργα και ζητάει από το χρήστη την επιλογή ενός από αυτά και μετά τα λαμβάνει μέσω ίντερνετ από το απομακρυσμένο αποθετήριο.

Ο στόχος create-from-project υλοποιεί την δημιουργία αρχέτυπου έργου από ένα υπάρχον έργο. Τέλος ο στόχος crawl αναζητάει ένα αποθετήριο για αρχέτυπα έργα και ενημερώνει μια λίστα. Τρεις ακόμα στόχοι είναι δεσμευμένοι με τον προεπιλεγμένο κύκλο κατασκευής. Ο στόχος jar είναι δεσμευμένος με την φάση δημιουργία πακέτου και χρησιμοποιείται για την κατασκευή του αντικειμένου jar του αρχέτυπου. Ο στόχος integration-test είναι δεσμευμένος με την φάση δοκιμή ενσωμάτωσης και χρησιμοποιείται για την εκτέλεση δοκιμές ενσωμάτωσης του αρχέτυπου. Τέλος ο στόχος update-local-catalog, δεσμευμένος με τη φάση εγκατάσταση και ενημερώνει τον τοπικό κατάλογο.

```

C:\Users\Babis>mvn archetype:generate
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----
[INFO] >>> maven-archetype-plugin:2.2:generate (default-cli) @ standalone-pom >>>
>
[INFO] <<< maven-archetype-plugin:2.2:generate (default-cli) @ standalone-pom <<<
<
[INFO] --- maven-archetype-plugin:2.2:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Interactive mode
-

C:\Users\Babis>mvn archetype:generate
986: remote -> ru.stqa.selenium:webdriver-java-archetype (Archetype for a Maven
project intended to develop tests with Selenium WebDriver and JUnit/TestNG)
987: remote -> ru.stqa.selenium:webdriver-junit-archetype (Archetype for a Maven
project intended to develop tests with Selenium WebDriver and JUnit)
988: remote -> ru.stqa.selenium:webdriver-testng-archetype (Archetype for a Maven
project intended to develop tests with Selenium WebDriver and TestNG)
989: remote -> se.vgregion.javg.maven.archetypes:javg-minimal-archetype (-)
990: remote -> sk.seges.sesam:sesam-annotation-archetype (-)
991: remote -> tk.skuro:clojure-maven-archetype (A simple Maven archetype for Clojure)
992: remote -> tr.com.lucidcode:kite-archetype (A Maven Archetype that allows users
to create a Fresh Kite project)
993: remote -> uk.ac.rdg.resc.edal-ncwms-based-webapp (-)
Choose a number or apply filter (format: [groupId:artifactId, case sensitive contains]: 387:
Choose org.apache.maven.archetypes:maven-archetype-quickstart version:
1: 1.0-alpha-1
2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
Choose a number: 6:
Define value for property 'groupId': : com.example.website
Define value for property 'artifactId': : first_project
Define value for property 'version': : 1.0-SNAPSHOT: :
Define value for property 'package': : com.example.website: :
Confirm properties configuration:
groupId: com.example.website
artifactId: first_project
version: 1.0-SNAPSHOT
package: com.example.website
ψ: :
  
```

Εικόνα 9

Παράδειγμα εκτέλεσης του προσθέτου archetype, του στόχου generate.

Το assembly δημιουργεί μια διανομή πόρων και/ή εκτελέσιμα του έργου σε μορφή αρχείου archive. Οι διανομές μπορεί να έχουν τις μορφές: zip, tar, tar.gz,

tar.bz2, jar, dir, war και για οποιαδήποτε άλλη μορφή ο διαχειριστής archive είναι διαμορφωμένο. Ο μόνος στόχος του πρόσθετου αυτού είναι ο στόχος single. Άλλοι στόχοι είναι παρωχημένοι και θα αφαιρεθούν από μελλοντικές εκδόσεις της Maven (όπως υποστηρίζεται στην ιστοσελίδα του προσθέτου assembly της Maven <http://maven.apache.org/plugins/maven-assembly-plugin/>). Αυτοί παρωχημένοι στόχοι είναι οι assembly, attached, directory, directory-inline, single-directory και τέλος unpack.

Το πρόσθετο dependency είναι αυτό που παρέχει την δυνατότητα χειρισμού των αντικείμενων εξαρτήσεων, αντιγράφοντας ή αποσυμπιέζοντας των από ένα τοπικό ή απομακρυσμένο αποθετήριο σε μια συγκεκριμένη τοποθεσία. Περιέχει τους εξής στόχους:

- analyze: αναλύει τις εξαρτήσεις του έργου και καθορίζει ποιές χρησιμοποιούνται και δηλώνονται, ποιές χρησιμοποιούνται και δε δηλώνονται και ποιές δε χρησιμοποιούνται και δηλώνονται.
- Analyze-dep-mgt: αναλύει τις εξαρτήσεις του έργου και αναφέρει αναντιστοιχίες μεταξύ επιλυμένες εξαρτήσεις και εκείνες που αναφέρονται στο τμήμα διαχείριση εξαρτήσεων.
- analyze-only: αυτός ο στόχος είναι ίδιος με τον analyze αλλά συνδέεται με το POM. Δεν διακλαδώνει την κατασκευή και την εκτέλεση δοκιμών μεταγλώττισης.
- analyze-report: αναλύει τις εξαρτήσεις του έργου και παράγει μια αναφορά συνοψίζοντας ποιές χρησιμοποιούνται και δηλώνονται, ποιές χρησιμοποιούνται και δε δηλώνονται και ποιές δε χρησιμοποιούνται και δηλώνονται.
- analyze-duplicate: αναλύει τις εξαρτήσεις και την ετικέτα διαχείριση εξαρτήσεων του POM και προσδιορίζει τις διπλά δηλωμένες εξαρτήσεις.
- build-classpath: λέει στη Maven να αποδώσει την διαδρομή των εξαρτήσεων από το τοπικό αποθετήριο σε μια μορφή classpath για χρήση από java – cp. Το αρχείο classpath μπορεί να συνδεθεί και εγκατασταθεί μαζί με το κυρίως αντικείμενο.

- `copy`: παίρνει μια λίστα των αντικειμένων που είναι δηλωμένα στο τμήμα διαμόρφωση προσθέτων και τα αντιγράφει σε μια συγκεκριμένη τοποθεσία και αφαιρώντας την έκδοση εάν αυτό είναι επιθυμητό. Ο στόχος αυτός μπορεί να επιλύσει τα αντικείμενα από το απομακρυσμένο αποθετήριο εάν δεν υπάρχουν στο τοπικό.
- `copy-dependencies`: Παίρνει την λίστα των άμεσων εξαρτήσεων του έργου και προαιρετικά τις μεταβατικές (transitive) εξαρτήσεις και τις αντιγράφει σε μια συγκεκριμένη τοποθεσία αφαιρώντας την έκδοση εάν αυτό είναι επιθυμητό.
- `get`: επιλύει ένα μονό αντικείμενο από ένα συγκεκριμένο αποθετήριο.
- `go-offline`: η Maven επιλύει τα πάντα στα οποία το έργο βασίζεται (εξαρτήσεις, πρόσθετα, αναφορές) προς προετοιμασία για εργασία χωρίς ίντερνετ.
- `list`: εναλλακτικός στόχος για την επίλυση των εξαρτήσεων του έργου.
- `properties`: ρύθμιση μιας ιδιότητας για κάθε εξάρτηση που περιέχει το έργο επιδεικνύοντας το αρχείο της στο σύστημα αρχείων.
- `purge-local-repository`: λέει στη Maven να διαγράψει τα αρχεία εξαρτήσεων από το τοπικό αποθετήριο και προαιρετικά να τα επιλύσει και πάλι.
- `resolve`: λέει στη Maven να επιλύσει όλες τις εξαρτήσεις και να εμφανίσει την έκδοση των.
- `resolve-plugins`: λέει στη Maven να επιλύσει τα πρόσθετα και τις εξαρτήσεις των.
- `sources`: λέει στη Maven να επιλύσει όλες τις εξαρτήσεις, τους συνημμένους πόρους των και να εμφανίσει την έκδοση των.
- `tree`: εμφανίζει το δέντρο εξαρτήσεων ενός έργου.
- `unpack`: αντίθετο με τον στόχο `copy`.
- `unpack-dependencies`: αντίθετο με τον στόχο `copy-dependencies`.

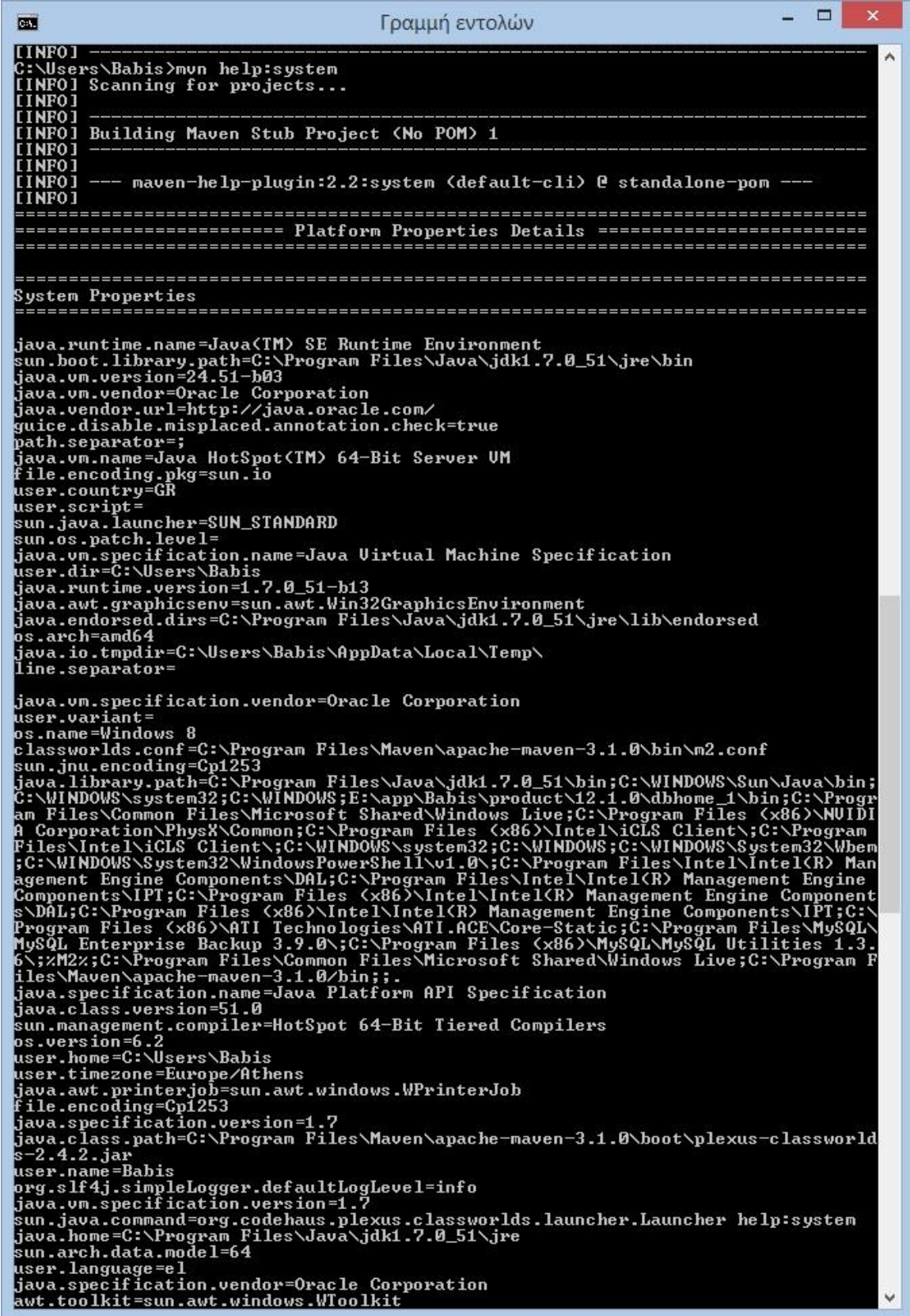
Το επόμενο πρόσθετο, το `enforcer` παρέχει στόχους για τον έλεγχο κάποιων περιβαλλοντικών περιορισμών όπως την έκδοση της Maven, του JDK ή τον τύπο λειτουργικού συστήματος. Έχει δύο στόχους, τον στόχο `enforce` ο

οποίος εκτελεί τους κανόνες για κάθε έργο σε μια κατασκευή πολλών έργων και τον στόχο `display-info` ο οποίος εμφανίζει πληροφορίες όπως αυτές έχουν ανιχνευτεί από τους κανόνες.

Το πρόσθετο `gpg` υπογράφει όλα τα αντικείμενα συνδεδεμένα με το έργο με `Gnu Privacy Guard` και έχει δύο στόχους: τον στόχο `sign` ο οποίος υπογράφει το αντικείμενο του έργου, το `POM` και τα επισυναπτόμενα αντικείμενα, και τον στόχο `sign-and-deploy-file` ο οποίος υπογράφει τα αντικείμενα και εγκαθιστά το αντικείμενο στο απομακρυσμένο αποθετήριο.

Το `help` βοηθάει το χρήστη όσον αφορά πληροφορίες για το έργο ή το σύστημα, πληροφορίες για κάποιο πρόσθετο, το `POM` (`Project Object Model`) και ενεργές ρυθμίσεις της ισχύουσας κατασκευής. Έχει τους εξής οκτώ στόχους:

- `active-profiles`: δίνει τα προφίλ τα οποία είναι ενεργά στο ισχύων έργο.
- `all-profiles`: δίνει τα διαθέσιμα προφίλ στο τρέχων έργο.
- `describe`: περιγράφει τις ιδιότητες ενός προσθέτου ή/και ενός `Mojo` (`Maven plain Old Java Object`).
- `effective-pom`: εμφανίζει το ενεργό `POM` σε μορφή `xml` για το τρέχων έργο με τα ενεργά προφίλ συνυπολογισμένα.
- `effective-settings`: εμφανίζει τις υπολογισμένες ρυθμίσεις σε μορφή `xml` για το έργο, δεδομένου τις βελτιώσεις προφίλ και την κληρονομικότητα των καθολικών ρυθμίσεων στις ρυθμίσεις χρήστη.
- `evaluate`: αξιολογεί τις εκφράσεις της `Maven` δοσμένες από τον χρήστη σε μια διαδραστική λειτουργία.
- `expressions`: εμφανίζει τις υποστηριζόμενες εκφράσεις πρόσθετων που χρησιμοποιούνται από τη `Maven`
- `system`: εμφανίζει μια λίστα λεπτομερειών πλατφόρμας όπως ιδιότητες συστήματος και μεταβλητές περιβάλλοντος.



```

[INFO] -----
C:\Users\Babis>mvn help:system
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----
[INFO]
[INFO] --- maven-help-plugin:2.2:system (default-cli) @ standalone-pom ---
[INFO]
===== Platform Properties Details =====
=====
System Properties
=====
java.runtime.name=Java(TM) SE Runtime Environment
sun.boot.library.path=C:\Program Files\Java\jdk1.7.0_51\jre\bin
java.vm.version=24.51-b03
java.vm.vendor=Oracle Corporation
java.vendor.url=http://java.oracle.com/
guice.disable.misplaced.annotation.check=true
path.separator=;
java.vm.name=Java HotSpot(TM) 64-Bit Server VM
file.encoding.pkg=sun.io
user.country=GR
user.script=
sun.java.launcher=SUN_STANDARD
sun.os.patch.level=
java.vm.specification.name=Java Virtual Machine Specification
user.dir=C:\Users\Babis
java.runtime.version=1.7.0_51-b13
java.awt.graphicsenv=sun.awt.Win32GraphicsEnvironment
java.endorsed.dirs=C:\Program Files\Java\jdk1.7.0_51\jre\lib\endorsed
os.arch=amd64
java.io.tmpdir=C:\Users\Babis\AppData\Local\Temp\
line.separator=

java.vm.specification.vendor=Oracle Corporation
user.variant=
os.name=Windows 8
classworlds.conf=C:\Program Files\Maven\apache-maven-3.1.0\bin\m2.conf
sun.jnu.encoding=Cp1253
java.library.path=C:\Program Files\Java\jdk1.7.0_51\bin;C:\WINDOWS\Sun\Java\bin;
C:\WINDOWS\system32;C:\WINDOWS;E:\app\Babis\product\12.1.0\dbhome_1\bin;C:\Progr
am Files\Common Files\Microsoft Shared\Windows Live;C:\Program Files (x86)\NVIDI
A Corporation\PhysX\Common;C:\Program Files (x86)\Intel\iCLS Client\;C:\Program
Files\Intel\iCLS Client\;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem
;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\Program Files\Intel\Intel(R) Man
agement Engine Components\DAL;C:\Program Files\Intel\Intel(R) Management Engine
Components\IPT;C:\Program Files (x86)\Intel\Intel(R) Management Engine Component
s\DAL;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\IPT;C:\
Program Files (x86)\ATI Technologies\ATI.ACE\Core-Static;C:\Program Files\MySQL\
MySQL Enterprise Backup 3.9.0\;C:\Program Files (x86)\MySQL\MySQL Utilities 1.3.
6\;%M2%;C:\Program Files\Common Files\Microsoft Shared\Windows Live;C:\Program F
iles\Maven\apache-maven-3.1.0\bin;;.
java.specification.name=Java Platform API Specification
java.class.version=51.0
sun.management.compiler=HotSpot 64-Bit Tiered Compilers
os.version=6.2
user.home=C:\Users\Babis
user.timezone=Europe/Athens
java.awt.printerjob=sun.awt.windows.WPrinterJob
file.encoding=Cp1253
java.specification.version=1.7
java.class.path=C:\Program Files\Maven\apache-maven-3.1.0\boot\plexus-classworld
s-2.4.2.jar
user.name=Babis
org.slf4j.simpleLogger.defaultLogLevel=info
java.vm.specification.version=1.7
sun.java.command=org.codehaus.plexus.classworlds.launcher.Launcher help:system
java.home=C:\Program Files\Java\jdk1.7.0_51\jre
sun.arch.data.model=64
user.language=en
java.specification.vendor=Oracle Corporation
awt.toolkit=sun.awt.windows.WToolkit

```

Εικόνα 10

Παράδειγμα εκτέλεσης του στόχου system του πρόσθετου help.

Το πρόσθετο `invoke` εκτελεί μια σειρά έργων της Maven και να ελέγξει εάν η εκτέλεσή των ήταν επιτυχής και επίσης προαιρετικά να επαληθεύσει και την έξοδο (`output`) της εκτέλεσης ενός έργου. Έχει τέσσερις στόχους:

- `install`: αντιγράφει τα αντικείμενα και τις εξαρτήσεις του έργου σε ένα αφιερωμένο τοπικό αποθετήριο για την προετοιμασία της εκτέλεσης των επιλεγμένων υπό-έργων σε ένα απομονωμένο περιβάλλον.
- `integration-test`: εκτελεί ένα σετ έργων της Maven σε ένα κατάλογο
- `verify`: επαληθεύει το αποτέλεσμα του προηγούμενου στόχου
- `run`: ένας στόχος που είναι ισοδύναμος με την εκτέλεση των δυο προηγούμενων στόχων.

Το πρόσθετο `jarsigner` παρέχει τη δυνατότητα στο χρήστη να υπογράψει ή να επαληθεύσει ένα αντικείμενο του έργου και τα επισυναπτόμενα του. Έχει δύο στόχους, τον στόχο `sign` που υπογράφει ένα αντικείμενο του έργου και των επισυναπτόμενων αντικειμένων και τον στόχο `verify` όπου επαληθεύονται τα αντικείμενα του έργου και τα επισυναπτόμενα αντικείμενα.

Το πρόσθετο `patch` μπορεί να χρησιμοποιηθεί για να ενημερώσει την έκδοση του κώδικα με ένα ή πολλαπλά κομμάτια κώδικα (`patches`) στα πηγαία αρχεία. Έχει ένα μόνο στόχο τον στόχο `apply`. Ένα άλλο χρήσιμο πρόσθετο είναι το `pdf` το οποίο μας επιτρέπει να μετατρέψουμε την τεκμηρίωση ενός έργου σε αρχείο `pdf` και έχει μόνο ένα στόχο με όνομα ίδιο με το πρόσθετο.

Το πρόσθετο `plugin` μπορεί να δημιουργήσει έναν περιγραφέα (`descriptor`) προσθέτων της Maven για κάθε ένα Mojo (`Maven plain Old Java Object`) που βρέθηκε στο πηγαίο δέντρο για συμπερίληψη στο αρχείο `jar`. Έχει έξι στόχους: τον στόχο `descriptor` ο οποίος δημιουργεί έναν περιγραφέα (`descriptor`) προσθέτου, τον στόχο `report` ο οποίος δημιουργεί μια αναφορά της τεκμηρίωσης του έργου, το `updateRegistry`, το οποίο ενημερώνει το μητρώο προσθέτων χρήστη (εάν χρησιμοποιείται) για να αντιστοιχεί την έκδοση που χρησιμοποιήθηκε. Ο επόμενος στόχος είναι ο `xdoc` και δημιουργεί αρχεία `xdoc` για τους στόχους του έργου ή για τα Mojo του έργου, ο στόχος `addPluginArtifactMetadata` προσθέτει μεταδεδομένα ενός αντικειμένου ενός

προσθέτου στο αντικείμενο του έργου και τέλος ο στόχος `helpmojo` δημιουργεί ένα Mojo βοήθειας που περιγράφει όλα τα Mojo σε ένα πρόσθετο.

Το πρόσθετο `release` μπορεί να δημοσιεύσει ένα έργο της Maven, γλιτώνοντας τον χρήστη από πολύ και επαναλαμβανόμενη εργασία και επίσης ενημερώνει το POM. Έχει οκτώ στόχους. Ο στόχος `clean` καθαρίζει μετά από την προετοιμασία δημοσίευσης, ο στόχος `prepare` προετοιμάζει την δημοσίευση στο SCM, ο `prepare-with-rom` προετοιμάζει την δημοσίευση στο SCM και δημιουργεί POM δημοσίευσης που καταγράφουν πλήρων επιλυμένα έργα που χρησιμοποιήθηκαν.

Ο στόχος `rollback` επαναφέρει μια προηγούμενη δημοσιευμένη έκδοση, ο στόχος `perform` εκτελεί μια δημοσίευση από το SCM, ο στόχος `stage` εκτελεί μια δημοσίευση από το SCM σε ένα πρόχειρο φάκελο/αποθετήριο, ο στόχος `branch` δημιουργεί μια διακλάδωση του ισχύων έργου με όλες τις εκδόσεις ενημερωμένες και τέλος ο στόχος `update-versions` ενημερώνει τις εκδόσεις στο POM.

Το πρόσθετο `reactor` κατασκευάζει ένα υποσύνολο ανεξάρτητων έργων σε κάτι που ονομάζεται `reactor`. Έχει τέσσερις στόχους: ο `resume` συνεχίζει ένα `reactor` σε ένα συγκεκριμένο σημείο (παραδείγματός χάρη όταν αποτυγχάνει στη μέση), ο στόχος `make` κατασκευάζει ένα έργο και όλα τα έργα `reactor` στα οποία το έργο αυτό βασίζεται. Ο στόχος `make-dependents` κατασκευάζει ένα έργο και όλα τα έργα `reactor` τα οποία βασίζονται στο έργο αυτό (το αντίθετο από τον προηγούμενο στόχο) και τέλος τον στόχο `make-scm-changes` ο οποίος κατασκευάζει όλα τα έργα `reactor` τα οποία έχουν μεταβληθεί από το χρήστη (σύμφωνα με το SCM) και όλα τα έργα `reactor` τα οποία βασίζονται στις αλλαγές αυτές.

Το πρόσθετο `remote-resources` ανακτά απομακρυσμένους πόρους από το ίντερνετ και τους αντιγράφει στον κατάλογο προορισμού με σκοπό την συμπερίληψη του στο αντικείμενο (`artifact`). Έχει δύο μόνο στόχους, τον στόχο `bundle` ο οποίος δημιουργεί το δηλωτικό πακέτο πόρου (`resource bundle manifest`) το οποίο απαιτείται από την επεξεργασία απομακρυσμένου δηλωτικού πακέτου πόρου. Ο δεύτερος στόχος είναι ο `process` ο οποίος ανακτά τα προσδιορισμένα απομακρυσμένα δηλωτικά πακέτα πόρου, τα επεξεργάζεται και τα καθιστά διαθέσιμα στην φάση κατεργασία πόρων

Το αντίστροφο πραγματοποιείτε από το πρόσθετο repository. Αυτό δημιουργεί μια δέσμη αντικειμένων (artifacts) τα οποία μπορούν να μεταφορτωθούν στο κεντρικό απομακρυσμένο αποθετήριο της Maven. Έχει δύο στόχους τους `bundle-create` ο οποίος δημιουργεί ένα πακέτο προς μεταφόρτωση για ένα έργο της Maven και `bundle-pack` ο οποίος δημιουργεί ένα πακέτο προς μεταφόρτωση για ένα αντικείμενο το οποίο υπάρχει στο τοπικό αποθετήριο.

Το scm είναι ένα πρόσθετο ικανό να εκτελέσει εντολές scm για το τρέχων έργο. Έχει δεκαέξι στόχους για τις διάφορες εντολές. Ο `branch` διακλαδώνει το έργο, ο στόχος `validate` επαληθεύει τις πληροφορίες scm στο POM, ο στόχος `add` είναι η εντολή για προσθήκη αρχείου, ο `unedit` είναι μια εντολή που σταματά την επεξεργασία του ενεργού αντιγράφου, ο `export` είναι εντολή που αποκτά ένα καινούριο εξαγόμενο αντίγραφο, ο `bootstrap` είναι εντολή που ολοκληρώνει (`checkout`) και κατασκευάζει ένα έργο, ο στόχος `changelog` δείχνει τις αναθεωρήσεις του πηγαίου κώδικα.

Ο στόχος `list` αποκτά μια λίστα με τα αρχεία του έργου, ο στόχος `checkin` είναι μια εντολή που διαπράττει τις αλλαγές, ο στόχος `checkout` παίρνει τον πηγαίο κώδικα, `status` που δείχνει την κατάσταση του scm ενός ενεργού αντιγράφου. Ο στόχος `update` ανανεώνει το ενεργό αντίγραφο με τις τελευταίες αλλαγές, ο στόχος `diff` που δείχνει τις διαφορές ενός ενεργού αντιγράφου και ενός απομακρυσμένου, ο στόχος `update-subprojects` ανανεώνει όλα τα έργα σε μια κατασκευή πολλών έργων, ο στόχος `edit` ο οποίος ξεκινάει μια επεξεργασία ενός ενεργού αντιγράφου και τέλος ο στόχος `tag` ο οποίος σημαδεύει μια συγκεκριμένη έκδοση.

Το `scm-publish` δημοσιεύει τον δικτυακό τόπο του χρήστη σε ένα υποστηριζόμενο scm. Έχει τρεις στόχους. Τον στόχο `help` ο οποίος προβάλλει πληροφορίες βοήθειας για το ίδιο το πρόσθετο, τον στόχο `publish-scm` ο οποίος δημοσιεύει περιεχόμενο στο scm και τον στόχο `scmpublish` ο οποίος είναι ένας άδειος στόχος για να απελευθερώνει τον κύκλο κατασκευής.

Μετά έχουμε ένα πρόσθετο ονόματι `source` το οποίο κατασκευάζει ένα αρχείο `jar` με πηγές για χρήση σε πρόγραμμα τύπου IDE και διανομή στο αποθετήριο. Έχει πέντε στόχους. Ο στόχος `aggregate` συναθροίζει τους πόρους από όλα τις μονάδες σε ένα έργο τύπου `aggregator`. Ο στόχος `jar` χρησιμοποιείται

για να μαζέψει τους κύριους πόρους του έργου σε ένα αρχείο jar. Ο στόχος test-jar μαζέψει τους πόρους δοκιμής του έργου σε ένα αρχείο jar. Ο στόχος jar-no-fork είναι παρόμοιος με τον jar αλλά δεν διακλαδίζει τον κύκλο κατασκευής και ο στόχος test-jar-no-fork παρόμοιος με τον test-jar αλλά δεν διακλαδίζει τον κύκλο κατασκευής.

Έπειτα έχουμε το stage το οποίο αντιγράφει αντικείμενα από ένα αποθετήριο σε ένα άλλο με ένα στόχο τον copy και τέλος το πρόσθετο toolchains με έναν ομώνυμο στόχο το οποίο επιτρέπει την διαμοίραση ρυθμίσεων (configuration) μεταξύ προσθέτων.

Στην τελευταία κατηγορία έχουμε μονάχα το πρόσθετο eclipse το οποίο δημιουργεί ένα έργο για το εξωτερικό πρόγραμμα eclipse του τωρινού έργου στη Maven. Έπειτα το δημιουργημένο έργο μπορεί να προσπελαστεί και να δουλευτεί με το πρόγραμμα eclipse. Το πρόσθετο αυτό έχει τρεις στόχους, τον configure-workspace ο οποίος προσθέτει την μεταβλητή περιβάλλοντος M2_REPO στο eclipse η οποία επιδεικνύει το τοπικό αποθετήριο και προαιρετικά διαμορφώνει άλλα χαρακτηριστικά του χώρου εργασίας. Ο στόχος eclipse δημιουργεί τα αρχεία διαμόρφωσης για το πρόγραμμα eclipse και τέλος ο στόχος clean διαγράφει τα αρχεία που χρησιμοποιήθηκαν από το πρόγραμμα eclipse.

Εκτός από αυτά τα πρόσθετα της Maven υπάρχουν και διάφορα εξωτερικά πρόσθετα από διάφορες εταιρίες και οργανισμούς τα οποία μπορούν να χρησιμοποιηθούν με τη Maven και πρόσθετα τα οποία έχουν πλέον αποσυρθεί τα οποία μπορεί να δει ο αναγνώστης στη σελίδα της Maven

<http://maven.apache.org/plugins/index.html>

ΚΕΦΑΛΑΙΟ 3

Η Maven στα IDE Eclipse και NetBeans.

ΕΙΣΑΓΩΓΗ

Η Maven αν και είναι ένα εργαλείο κατασκευής έργων και διαχείρισης εξαρτήσεων δεν βοηθάει το χρήστη όσον αφορά στη συγγραφή κώδικα. Δύο προγράμματα τύπου IDE (Integrated Development Environment) τα Eclipse από την εταιρία Eclipse Foundation μια μη-κερδοσκοπική υποστηριζόμενη από μέλη εταιρία^[9] και το NetBeans από την Oracle, έχουν δικά τους πρόσθετα και μεταξύ άλλων επιτρέπουν τη χρήση της Maven μέσω των προγραμμάτων αυτών.

Ο χρήστης μπορεί να κατεβάσει τα προγράμματα αυτά από τις διευθύνσεις:

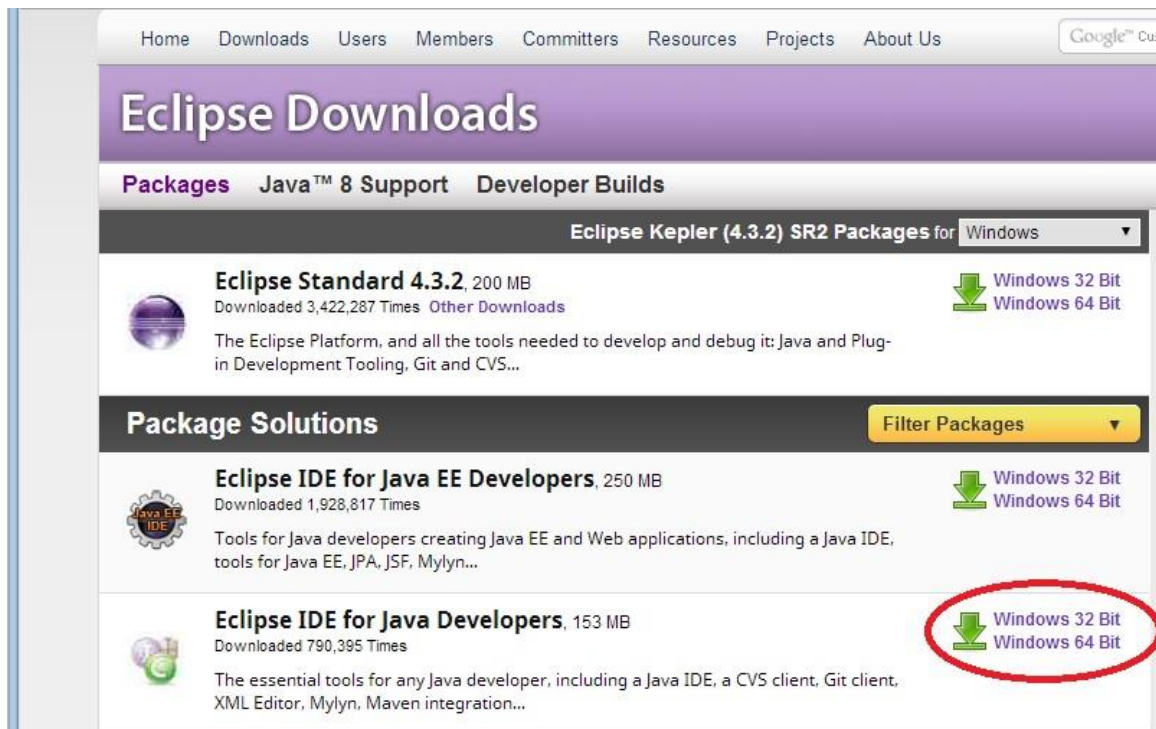
<https://netbeans.org/downloads/> και

<http://www.eclipse.org/downloads/>

Τα προγράμματα αυτά υποστηρίζουν μια συλλογή από διάφορες γλώσσες προγραμματισμού όπως Java, C, C++, HTML5, PHP κ.α.. Στα υποκεφάλαια που ακολουθούν θα περιγράψουμε πως να γίνει η εγκατάσταση των προγραμμάτων αυτών και το πρόσθετο της Maven και πως να γίνουν διάφορες ρυθμίσεις που απαιτούνται για την σωστή λειτουργία των.

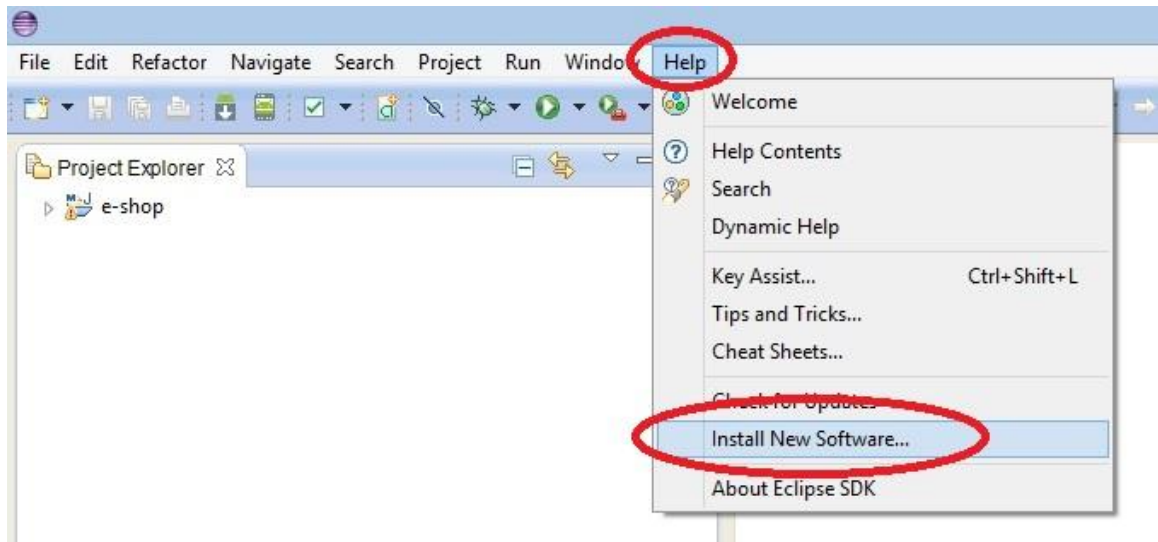
ΥΠΟΚΕΦΑΛΑΙΟ 3.1

Ξεκινώντας με το Eclipse αφού περιηγηθούμε στην παραπάνω σελίδα που αντιστοιχεί στο πρόγραμμα, κατεβάζουμε την τρίτη έκδοση (Eclipse IDE for Java Developers) και το εγκαθιστούμε στο σύστημά μας κάτι το οποίο θεωρείται απλή διαδικασία.

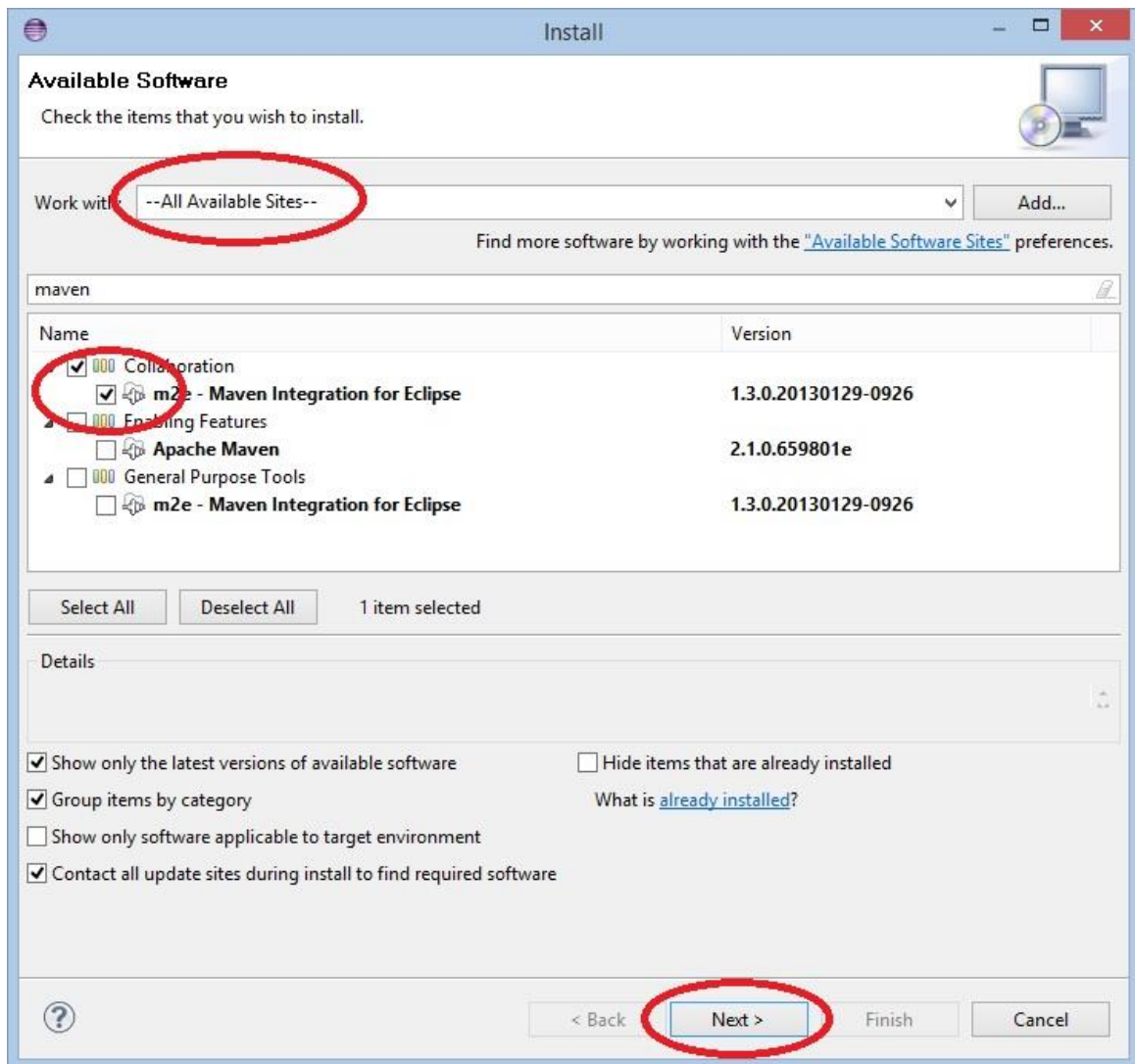


Εικόνα 11

Αφού το εγκαταστήσουμε το ξεκινάμε και επιλέγουμε HELP και Install New Software. Στο σημείο Work with: επιλέγουμε All Available Sites από την αναπτυσσόμενη λίστα. Έπειτα στο σημείο αναζήτησης πληκτρολογούμε Maven και στα αποτελέσματα τσεκάρουμε το κουτάκι Maven integration for eclipse και επιλέγουμε next. Επιλέγουμε το πρόσθετο στη λίστα που θα βγάλει, έπειτα επιλέγουμε Finish και περιμένουμε την ολοκλήρωση και εγκατάσταση της λήψης μας. Στο τέλος το πρόγραμμα θα μας ζητήσει την επανεκκίνησή του και ήμαστε έτοιμοι για να ξεκινήσουμε.

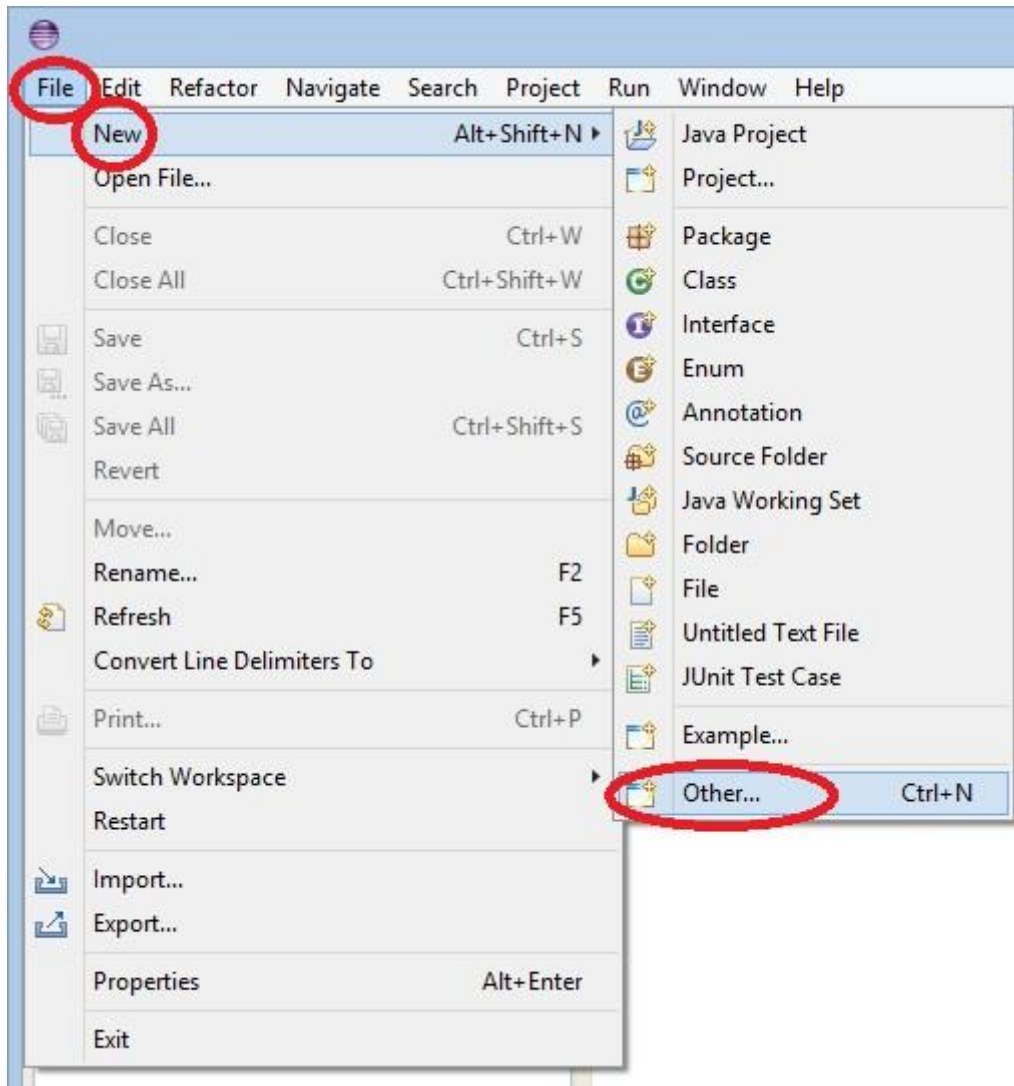


Εικόνα 12



Εικόνα 13

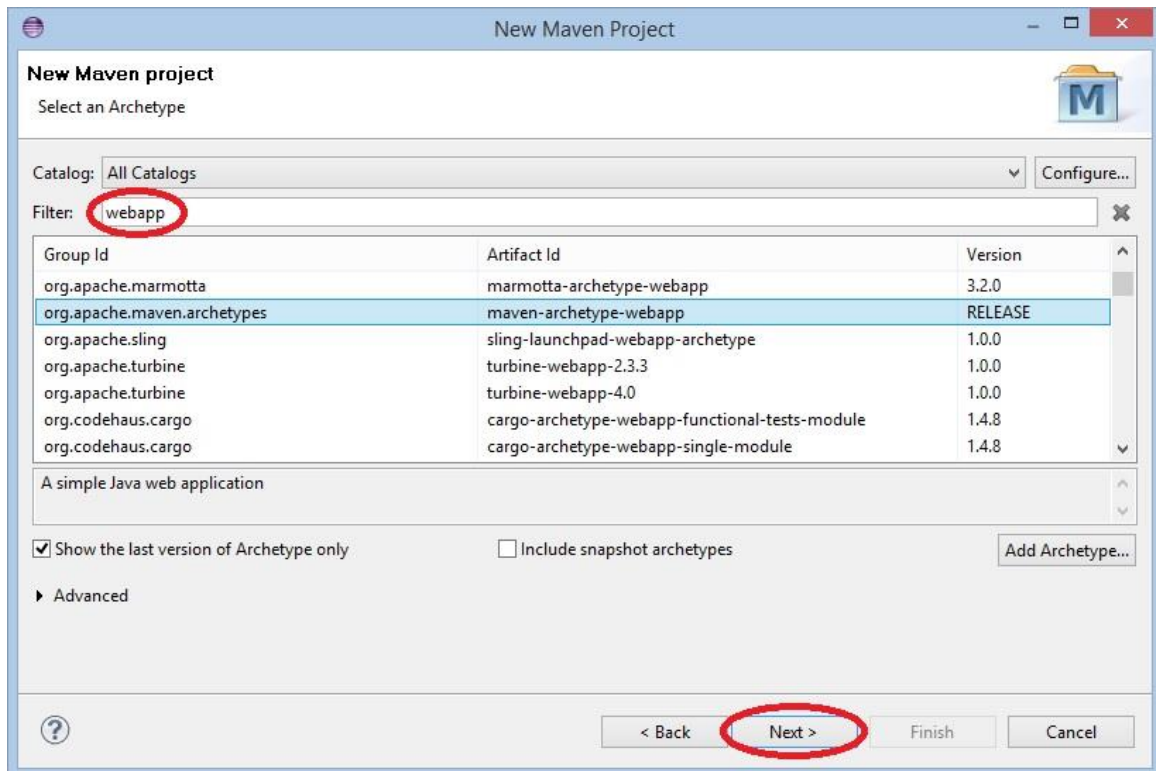
Για να ξεκινήσουμε ένα έργο Maven από το Eclipse επιλέγουμε File, New και Other στο τέλος της λίστας. Καταρρέουμε το βέλος δίπλα στην επιλογή Maven και επιλέγουμε την επιλογή Maven Project. Έπειτα πατάμε Next για να προχωρήσουμε στην επόμενη οθόνη. Η επιλογή Create Simple project μας επιτρέπει να παραλείψουμε την επιλογή αρχέτυπου έργου και δημιουργεί ένα απλό έργο.



Εικόνα 14

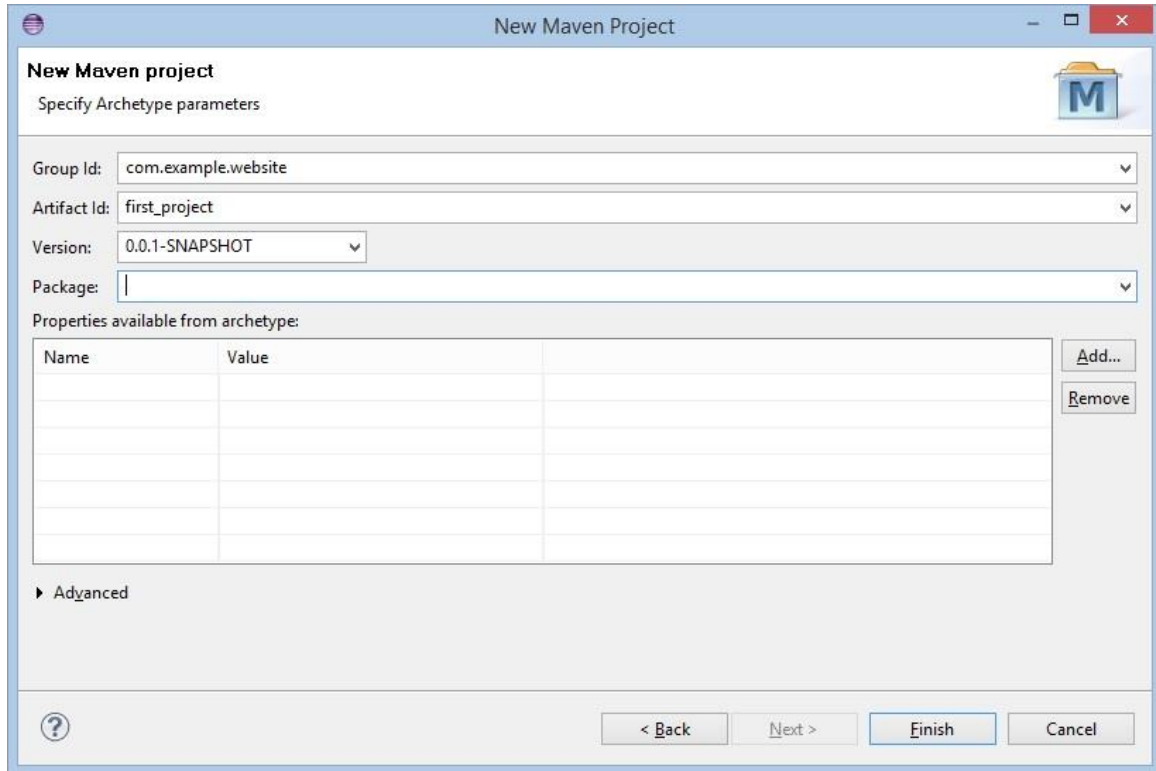
Με την επιλογή create simple project βλέπουμε μια λίστα 800+ πρότυπων έργων από τα οποία μπορούμε να φιλτράρουμε μια συγκεκριμένη κατηγορία όπως για παράδειγμα την κατηγορία webarr όπου δημιουργούμε μια εφαρμογή ίντερνετ. Με την επιλογή ενός αρχέτυπου μπορούμε σε πολλές των περιπτώσεων να δούμε μια γενική περιγραφή του αρχέτυπου και αφού αποφασίσουμε ποιο θέλουμε επιλέγουμε next. Έπειτα η Maven κατεβάζει μέσω

ίντερνετ από ένα απομακρυσμένο αποθετήριο ότι εξαρτήσεις χρειάζεται το έργο εφόσον αυτές δεν υπάρχουν στο τοπικό αποθετήριο.



Εικόνα 15

Στην επόμενη οθόνη δίνουμε πληροφορίες όπως GroupId, ArtifactId, Version και Package και βλέπουμε ή επεξεργαζόμαστε διάφορες ιδιότητες διαθέσιμες στο αρχέτυπο έργο. Αφού δώσουμε τις πληροφορίες που απαιτούνται (τουλάχιστον GroupId και ArtifactId) επιλέγουμε Finish.



Εικόνα 16

Αφού το Eclipse ολοκληρώσει την δημιουργία του έργου μπορούμε να το δούμε στο παράθυρο Package Explorer (εάν δεν υπάρχει Window, Show View, Package Explorer). Εάν καταρρεύσουμε το βέλος δίπλα στο όνομα του έργου (εδώ φαίνεται η τιμή του πεδίου artifactId ακόμα και αν έχουμε δώσει όνομα) μπορούμε να δούμε τη δομή του έργου. Τους φακέλους src, target και το POM του έργου.

Εδώ μπορούμε να ανοίξουμε το POM και να το δούμε και να το επεξεργαστούμε διαμέσου μιας γραφικής διεπαφής αντί για να το δούμε σε μορφή ενός xml αρχείου μέσω ενός επεξεργαστή κειμένου. Το Eclipse βέβαια μας δίνει τη δυνατότητα να ανοίξουμε το αρχείο με πολλούς διαφορετικούς τρόπους όπως Maven POM editor, text editor, xml editor, system editor κ.α..

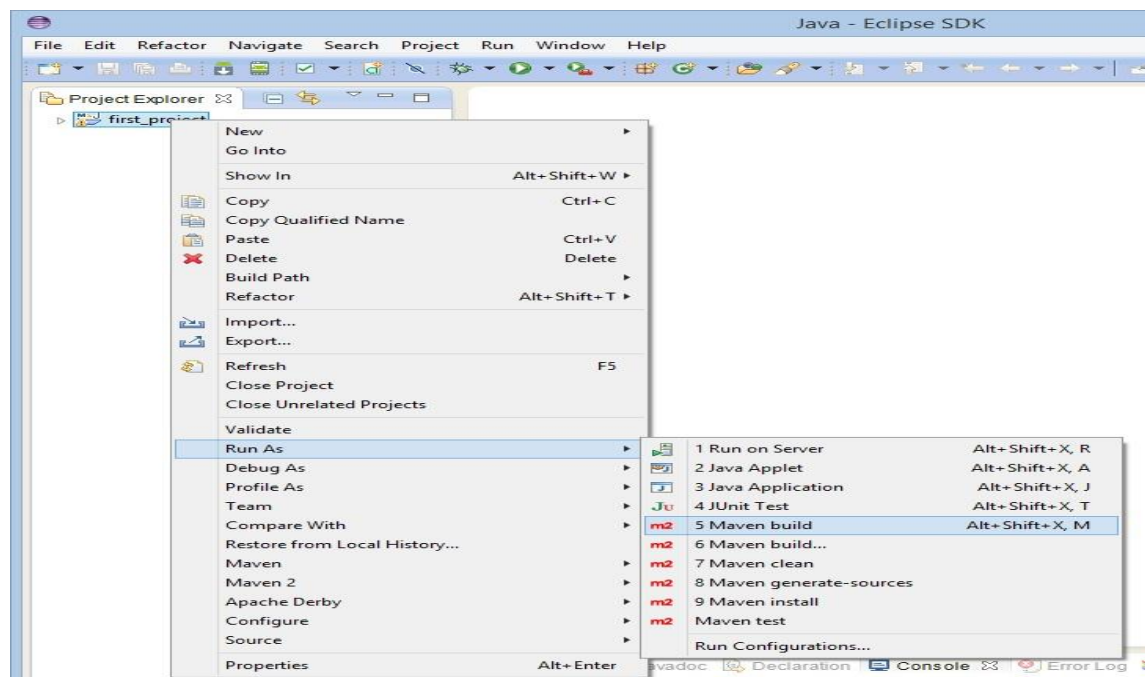
Δυο επιλογές που μας ενδιαφέρουν επίσης είναι οι επιλογές run as και Maven. Μπορούμε να τις βρούμε με δεξί κλικ στο όνομα του έργου ή δεξί κλικ στο POM. Η επιλογή Maven μας δίνει τη δυνατότητα να προσθέσουμε εξαρτήσεις στο έργο δίνοντας τις συντεταγμένες μιας εξάρτησης (groupId, artifactId και version).

Μπορούμε επίσης να προσθέσουμε και δικό μας πρόσθετο δίνοντας και πάλι τις συντεταγμένες του. Η τρίτη ενδιαφέρουσα επιλογή είναι η δυνατότητα να ξεκινήσουμε ένα νέο έργο – μονάδα (module project).

Ένα έργο μονάδα είναι σαν ένα υπό-έργο (sub-project) της Maven και μπορεί να υπάρχουν πολλαπλές μονάδες σε ένα έργο της Maven. Για ένα έργο – μονάδα πρέπει να καθορίσουμε ένα γονικό έργο.

Η επιλογή run as μας δίνει την δυνατότητα να εκτελέσουμε εντολές της Maven όπως build, clean, install, test κ.α. Για να κάνουμε μεταγλώττιση το έργο επιλέγουμε το POM ή το όνομα του έργου, επιλέγουμε run as και Maven build... (τα αποσιωπητικά σημαίνουν ότι μπορούμε να διαμορφώσουμε την εντολή που θα τρέξουμε. Στο παράθυρο που θα εμφανιστεί στο σημείο goals γράφουμε την εντολή που θέλουμε να τρέξουμε.

Ένα παράδειγμα εντολής είναι: `compiler:compile` όπου εκτελείται το πρόσθετο `compiler` και ο στόχος του `compile` ή απλά `package` για να εκτελέσουμε το πρόσθετο χωρίς να επιλέξουμε στόχο. Εάν θέλουμε να ξαναεκτελέσουμε την ίδια εντολή μπορούμε να επιλέξουμε με δεξί κλικ, run as, maven build χωρίς τα αποσιωπητικά αυτή τη φορά και να επιλέξουμε την εντολή από τη λίστα εντολών που έχουμε ξαναεκτελέσει.



Εικόνα 17

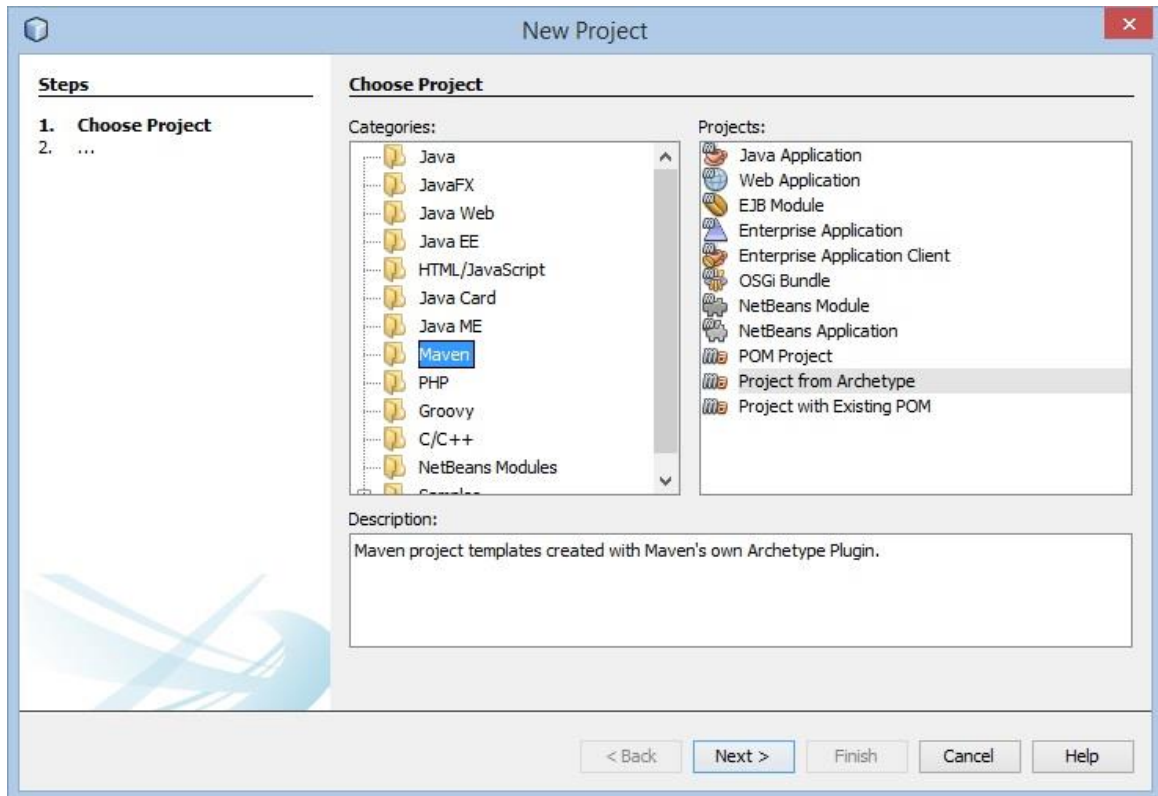
ΥΠΟΚΕΦΑΛΑΙΟ 3.2

Το NetBeans είναι ένα άλλο πρόγραμμα που υποστηρίζει την χρήση της Maven. Κατεβάζουμε την έκδοση 7.2 και πάνω του NetBeans IDE και το Java Developer Kit, έκδοση 6 και πάνω.

Το NetBeans μπορεί και αναγνωρίζει αυτόματα ένα υπάρχων έργο και μπορεί να το επεξεργαστεί^[10]. Όλα τα έργα που έχουν αναγνωριστεί σαν έργα Maven εμφανίζονται με ένα ξεχωριστό εικονίδιο ma στο παράθυρο open project. Αφού ανοίξουμε το NetBeans επιλέγουμε File, Open project (ή ctrl+shift+o για τη συντόμευση πληκτρολογίου) εντοπίζουμε το έργο στον δίσκο και επιλέγουμε open project.

Ένα μειονέκτημα είναι ότι ένα έργο του NetBeans δεν μπορεί να βασιστεί σε ένα άλλο έργο εάν εκείνο δεν έχει πρώτα μεταγλωττιστεί και πακεταριστεί. Γίνεται όμως να δημιουργηθεί μια λίστα με έργα και να πακεταριστούν σαν ένα έργο.

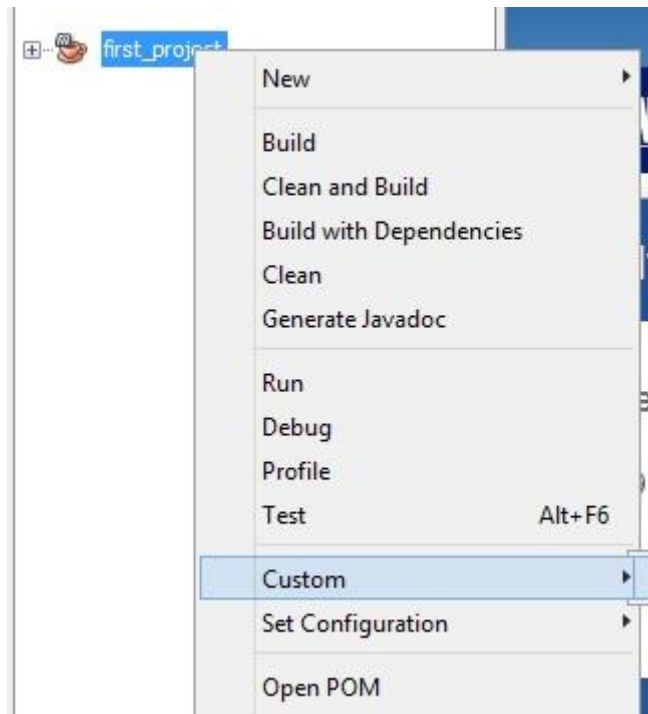
Για να ξεκινήσουμε ένα νέο έργο στο NetBeans επιλέγουμε File, New Project, αφού εντοπίσουμε το φάκελο που γράφει Maven τον επιλέγουμε και έπειτα επιλέγουμε την επιλογή Project from Archetype. Εδώ μπορούμε να δημιουργήσουμε ένα έργο επιλέγοντας ένα αρχέτυπο έργο, παρόμοιο με την εντολή archetype:generate της Maven. Στην επόμενη οθόνη επιλέγουμε ένα αρχέτυπο έργο από τη λίστα με όλους τους διαθέσιμους αρχέτυπους και επιλέγουμε επόμενο (next). Εδώ δίνουμε πληροφορίες όπως το όνομα του έργου, συντεταγμένες του έργου (groupid artifactid, version) και προαιρετικά το όνομα πακέτου και επιλέγουμε finish.



Εικόνα 18

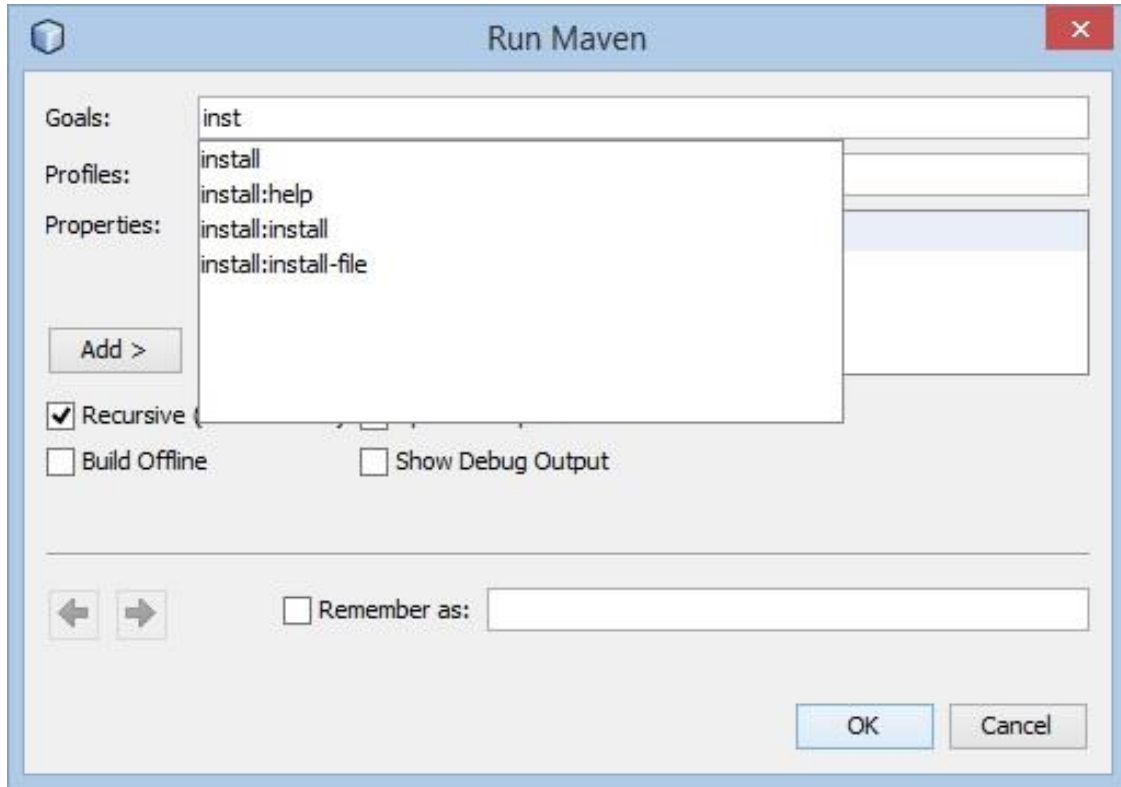
Η πρώτη επιλογή Java Application δημιουργεί ένα απλό έργο της Maven και θα έχουμε τη δυνατότητα να δώσουμε και πάλι τα στοιχεία που θέλουμε για το έργο αυτό. Το αρχείο POM.xml μπορεί να βρεθεί στο παράθυρο projects (εάν δεν υπάρχει επιλέγουμε window και projects ή πληκτρολογούμε ctrl+1 στο πληκτρολόγιο) στο φάκελο project files. Από εκεί μπορούμε να το επεξεργαστούμε με διπλό κλικ ή με δεξί κλικ και την επιλογή open.

Για να εκτελέσουμε εντολές της Maven επιλέγουμε με δεξί κλικ το όνομα του έργου. Εδώ είναι διαθέσιμες οι απλές εντολές όπως κατασκευή του έργου, εκκαθάριση (clean) εκκαθάριση και κατασκευή (clean and build) και άλλα. Για να εκτελέσουμε όποια πρόσθετα με όποιους στόχους θέλουμε επιλέγουμε με δεξί κλικ το όνομα του έργου και την επιλογή custom και goals. Το NetBeans μας βοηθάει δίνοντάς μας τους διαθέσιμους στόχους σύμφωνα με τα όσα έχουμε πληκτρολογήσει.



Εικόνα 19

Εάν για παράδειγμα πληκτρολογήσουμε το γράμμα d μας δείχνει τα πρόσθετα dependency, deploy, doap και docck και όλους τους στόχους τους. Με την πληκτρολόγηση του πλήκτρου o αφαιρεί τα πρόσθετα dependency και deploy και δείχνει μόνο τα doap και docck, κάτι που μπορεί να φανεί πολύ χρήσιμο και δεν χρειάζεται να καταφεύγουμε στην τεκμηρίωση των πρόσθετων της Maven.



Εικόνα 20

Όσον αφορά τις εξαρτήσεις το NetBeans προσθέτει ότι εξαρτήσεις απαιτεί ένα έργο αυτόματα ενώ μας δίνει τη δυνατότητα να δηλώσουμε επιπλέον δικές μας εξαρτήσεις στο POM και να αφήσουμε το NetBeans να φροντίσει για την λήψη των μέσω ίντερνετ από το απομακρυσμένο αποθετήριο. Το POM.xml μπορούμε να το βρούμε στο φάκελο Project Files στο παράθυρο projects. Ανοίγουμε το POM με διπλό κλικ του ποντικιού ή με δεξί κλικ και άνοιγμα (open). Έπειτα το NetBeans ανοίγει τον κώδικα του αρχείου POM.xml όπου και μπορούμε να προσθέσουμε τις εξαρτήσεις στο σημείο <dependencies>.

Κάθε διαφορετική εξάρτηση πρέπει να μπει στη δική της ετικέτα <dependency> μέσα στην ετικέτα <dependencies>. Εδώ πρέπει να πούμε ότι μερική γνώση της γλώσσας xml είναι απαραίτητη σε αντίθεση με το Eclipse που μας δίνει μια γραφική διεπαφή για να τροποποιήσουμε το POM, εάν το επιθυμούμε. Η προσθήκη της εξάρτησης γίνεται με την πληκτρολόγηση των συντεταγμένων της (groupid, artifactid, version) και προαιρετικά το scope το οποίο μπορεί να πάρει τις τιμές compile, provided, runtime, test, system και import. Αφού πληκτρολογήσουμε τις εξαρτήσεις που θέλουμε στο POM έπειτα επιλέγουμε δεξί κλικ στο Dependencies και download declared dependencies.

Επίσης μπορούμε να προσθέσουμε μια εξάρτηση πληκτρολογώντας τις συντεταγμένες τις (groupid, artifactid, version). Αυτό μπορεί να γίνει με δεξί κλικ στο φάκελο dependencies του παραθύρου projects και μετά add dependencies. Με την εκκίνηση της πληκτρολόγησης το NetBeans ψάχνει όλες τις διαθέσιμες εξαρτήσεις και τις φιλτράρει σύμφωνα με τα όσα έχει πληκτρολογήσει ο χρήστης και έτσι βοηθώντας τον στην αναζήτηση εξαρτήσεων. Σε αυτήν την περίπτωση το NetBeans λαμβάνει τις εξαρτήσεις από το απομακρυσμένο αποθετήριο και αλλάζει και το αρχείο POM.xml προσθέτοντας την εξάρτηση που επιλέξαμε.

ΚΕΦΑΛΑΙΟ 4

Συγγραφή προσθέτων

Εισαγωγή

Η Maven επιτρέπει στους χρήστες της να συγγράψουν και να χρησιμοποιήσουν δικά τους πρόσθετα ή να χρησιμοποιήσουν πρόσθετα δημιουργημένα από τρίτους. Αυτό γίνεται με την συγγραφή αντικειμένων τα οποία ονομάζονται Maven Old Java Object (MOJO). Κάθε MOJO είναι ένας εκτελέσιμος στόχος της Maven και κάθε πρόσθετο είναι μια διανομή ενός ή περισσότερων MOJO.

ΥΠΟΚΕΦΑΛΑΙΟ 4.1

Ξεκινώντας την συγγραφή ενός προσθέτου πρέπει πρώτα να το ονομάσουμε. Η ονομασία ενός προσθέτου πρέπει να ακολουθεί την εξής μορφή: «όνομα προσθέτου–maven–plugin»^[11]. Η Maven έχει διατηρήσει την δυνατότητα να ονομάζει τα πρόσθετά της με τη μορφή: «maven–όνομα προσθέτου–plugin» και η ονομασία προσθέτου με αυτή τη μορφή είναι μια παράβαση του σήμα κατατεθέν της Apache Maven.

Το πιο απλό πρόσθετο θα μπορούσε να αποτελείται από μια και μοναδική κλάση Java. Δεν υπάρχει απαίτηση πολλαπλών κλάσεων java σε ένα πρόσθετο. Για να αναγνωριστεί μια κλάση σαν MOJO πρέπει να υπάρχει το κομμάτι κώδικα @Mojo για παράδειγμα @Mojo(name = "helloworld") στον κώδικα της. Αυτό είναι απαραίτητο για τον έλεγχο του πως και πότε θα εκτελείται το MOJO. Για την περισσότερη υποδομή ενός MOJO μπορεί να χρησιμοποιηθεί η γραμμή κώδικα «import org.apache.maven.plugin.AbstractMojo;» εκτός από μια μέθοδο με το όνομα execute.

Το κομμάτι κώδικα: «import org.apache.maven.plugin.MojoExecutionException» μπορεί να χρησιμοποιηθεί για την περίπτωση που θα δημιουργηθεί ένα σφάλμα και δημιουργήσει ένα μήνυμα σφάλματος κατασκευής (build error message).

Κάτι άλλο που μπορούμε να κάνουμε σε ένα MOJO είναι να προσθέσουμε τη γραμμή «import org.apache.maven.plugin.MojoFailureException» για να ελέγξουμε προβλήματα όπως σφάλματα μεταγλώττισης και δημιουργεί ένα μήνυμα αποτυχίας κατασκευής (build failure message).

Τέλος υπάρχει και μια μέθοδος ορισμένη στην κλάση AbstractMojo με την ονομασία getLog(); και επιστρέφει ένα αντικείμενο τύπου logger το οποίο επιτρέπει στα πρόσθετα να δημιουργήσουν μηνύματα σε διαφορετικά επίπεδα όπως debug, info, warn και error.

Η κατασκευή του προσθέτου ξεκινάει μόνο αφού τα MOJO έχουν ολοκληρωθεί. Για να γίνει αυτό πρέπει να έχουν οριστεί οι εξής ρυθμίσεις:

- `groupId`: το `groupId` πρέπει να ορίζει μοναδικά το πρόσθετο μεταξύ όλων των προσθέτων οπότε πρέπει να οριστούν κάποιοι κανόνες. Πρέπει να ακολουθούνται οι κανόνες ονομασίας πακέτων της Java όπως ένα όνομα τομέα (`domain name`) και σε αυτό μπορούν να ανήκουν όσες υποομάδες χρειάζονται. Ένα παράδειγμα είναι `org.apache.maven`.
- `artifactId`: είναι το όνομα του πακέτου χωρίς τον αριθμό της έκδοσης. Πρέπει να περιέχει πεζά γράμματα αλλά όχι παράξενα σύμβολα.
- `version`: ο αριθμός της έκδοσης του προσθέτου μπορεί να αποτελείται από αριθμούς και τελείες. Μερικές τυπικές ονομασίες είναι 1.1, 1.0, 2.3.1 ή 2.0.1. Η ονομασία της έκδοσης δεν πρέπει να περιέχει ημερομηνίες καθώς αυτές συχνά σχετίζονται με κατασκευές τύπου SNAPSHOT που δεν έχουν κυκλοφορήσει.
- `packaging`: αυτή η ρύθμιση θα πρέπει να είναι ορισμένη ως `maven-plugin`
- `dependencies`: οι εξαρτήσεις πρέπει να δηλώνονται στο API των εργαλείων προσθέτων της Maven για επίλυση της κλάσης `AbstractMojo` και άλλων κλάσεων.

Σε ένα πρόσθετο ορίζονται μερικοί στόχοι κατασκευής με το πακέτο `maven-plugin`. Οι στόχοι `compile`, `test`, `package`, `install` και `deploy` οι οποίοι έχουν περιγραφεί προηγουμένως βοηθούν το πρόσθετο να λειτουργήσει ως μέρος ενός κύκλου κατασκευής.

Ο πιο άμεσος τρόπος εκτέλεσης ενός προσθέτου είναι η εκτέλεση ενός στόχου από την γραμμή εντολών. Αυτό γίνεται με την διαμόρφωση του προσθέτου `hello-maven-plugin` στο έργο. Επίσης πρέπει να υπάρχει ένας στόχος που πληροί τις προϋποθέσεις. Η εκτέλεση του γίνεται με την εντολή `mvn groupId:artifactId:version:goal`. Ένα παράδειγμα θα ήταν η εντολή «`mvn sample.plugin:hello-maven-plugin:1.0-SNAPSHOT:helloworld`» όπου `sample.plugin` το `groupId`, `hello-maven-plugin` το `artifactId`, `1.0-SNAPSHOT` η έκδοση του έργου και `helloworld` το σημείο `@Mojo(name = "helloworld")` στον κώδικα του MOJO ενώ η έκδοση δεν είναι απαραίτητη.

Υπάρχει η δυνατότητα να συνδεθεί ένα MOJO με ένα κύκλο κατασκευής. Αυτό γίνεται με την προσθήκη του παρακάτω κώδικα στο POM του MOJO μετά από τα groupId, artifactId και version.

```
<executions>
    <execution>
        <phase>compile</phase>
        <goals>
            <goal>helloworld</goal>
        </goals>
    </execution>
</executions>
```

Σε ένα MOJO πρέπει να προσθέσουμε παραμέτρους οι οποίες παρέχουν κάποιες πολύ σημαντικές ενέργειες. Επιτρέπουν το χρήστη να προσαρμόζει την συμπεριφορά του προσθέτου όπως αρμόζει στην κάθε περίπτωση. Επίσης παρέχει τα μέσα για την εξαγωγή τιμών των στοιχείων του POM χωρίς να το επεξεργαστεί.

Ο ορισμός μιας παραμέτρου στον κώδικα του MOJO γίνεται με την προσθήκη μιας μεταβλητής με τον εξής τρόπο:

```
@Parameter( property = "helloworld.greeting", defaultValue = "Hello World!" )
private String greeting;
```

Οι παράμετροι μπορούν να ρυθμιστούν μέσα από το POM ενός έργου το οποίο χρησιμοποιεί το πρόσθετό μας. Ο παρακάτω κώδικας τοποθετείται μετά από τα groupId, artifactId και version και αλλάζει την τιμή της παραμέτρου από «Hello World!» σε «Welcome».

```
<configuration>
    <greeting>Welcome</greeting>
```

</configuration>

Υπάρχει η δυνατότητα χρήσης παραμέτρων όλων των τύπων όπως Boolean, integer κ.α.. Η δήλωσή τους γίνεται με την προσθήκη του κώδικα @Parameter ακολουθούμενο από τις δηλώσεις «private boolean myBoolean», «private Integer myInteger», «private Double myDouble», «private Date myDate» (για δήλωση ημερομηνίας), «private File myFile» (για δήλωση αρχείου) και «private URL myURL».

Η τοποθέτηση τιμών στις παραμέτρους μέσα από το POM γίνεται με την πληκτρολόγηση της ονομασίας των παραμέτρων και την τιμή όπως και στο παραπάνω παράδειγμα με την παράμετρο greeting. Η κάθε μια παράμετρος έχει ιδιαιτερότητες όπως για παράδειγμα η Boolean που θα καταγράψει ως αληθή τιμή την λέξη true ενώ οποιαδήποτε άλλη λέξη θα καταγραφεί σαν ψευδής τιμή (false).

Η παράμετρος τύπου int/integer θα μετατρέψει το κείμενο στο POM σε αριθμό με χρήση μεθόδων όπως Integer.parseInt() και valueOf(). Αυτό σημαίνει ότι το κείμενο πρέπει να είναι μια έγκυρη ακέραια τιμή αποτελούμενη από τους αριθμούς «0-9» με την «-» προαιρετικά εάν ο αριθμός είναι αρνητικός.

Η παράμετρος τύπου date μπορεί να ακολουθεί τις μορφές: yyyy-MM-dd HH:mm:ss.S a (1994-12-01 2:22:55.1 PM) ή yyyy-MM-dd HH:mm:ssa (1994-12-01 2:22:55PM). Η μέθοδος που χρησιμοποιείται είναι η DateFormat.parse() η οποία είναι επιεικής.

Η παράμετρος τύπου file μπορεί να ακολουθεί μια απόλυτη διαδρομή στο δίσκο ή μια σχετική. Η σχετική διαδρομή θα είναι σχετική με τη διαδρομή του αρχείου POM. Η παράμετρος τύπου URL πρέπει να ακολουθεί τις κατευθυντήριες γραμμές RFC 2396 (scheme://host:port/path/to/file).

Η παράμετρος τύπου text μπορεί να είναι τύπου char/character ή string και StringBuffer. Για παραμέτρους τύπου char/character λαμβάνεται υπόψη μόνο ο πρώτος χαρακτήρας. Υπάρχει επίσης η δυνατότητα οι παράμετροι να είναι τύπου πίνακας (array). Δηλώνονται με τη γραμμή κώδικα:

@Parameter

private String[] myArray; και παίρνει τιμές από το POM ως εξής:

```
<myArray>  
<param>value1</param>  
<param>value2</param>  
</myArray>.
```

Επίσης υπάρχει και παράμετρος τύπου Collection η οποία είναι οποιαδήποτε κλάση υλοποιεί το πακέτο java.util.Collection όπως ArrayList και HashSet. Ο κώδικας είναι:

```
@Parameter  
private List myList; και στο POM  
<myList>  
<param>value1</param>  
<param>value2</param>  
</myList>.
```

ΥΠΟΚΕΦΑΛΑΙΟ 4.2

Η Maven δίνει τη δυνατότητα στους χρήστες της να δοκιμάζουν τα πρόσθετά τους με δοκιμές μονάδας (unit tests), δοκιμές ένταξης (integration tests) και λειτουργικές δοκιμές (functional tests).

Οι δοκιμές μονάδας επαληθεύουν το MOJO σαν απομονωμένη μονάδα δημιουργώντας ένα ψεύτικο περιβάλλον Maven^[12]. Μια τέτοια δοκιμή δεν είναι σχεδιασμένη για ένα πραγματικό περιβάλλον αλλά εστιάζει σε πιο άμεσες και γρήγορες εκτελέσεις.

Οι δοκιμές ένταξης/λειτουργικές δοκιμές δοκιμάζουν ένα MOJO σε ένα πραγματικό έργο Maven εκκινώντας μια πραγματική εκδοχή της Maven. Δημιουργείται ένα εικονικό έργο Maven χρησιμοποιώντας ένα πραγματικό POM. Αυτό απαιτεί την προ-εγκατάσταση του προσθέτου στο τοπικό αποθετήριο ώστε να χρησιμοποιηθεί σε μια πραγματική κατασκευή της Maven. Λειτουργικές δοκιμές εκτελούνται πιο αργά από τις δοκιμές μονάδας αλλά μπορούν να βρουν ατέλειες στο έργο οι οποίες δεν μπορούν να εντοπιστούν από τις δοκιμές μονάδας.

Οι δοκιμές μονάδας εκτελούνται με τη βοήθεια του πρόσθετου surefire. Η εκτέλεση δοκιμών γίνεται με τη Maven να τοποθετεί τις κλάσεις που εκτελούν τις δοκιμές στη διαδρομή φακέλων `src/test/java` στον ίδιο κατάλογο με το `POM.xml` και τους πόρους στην διαδρομή `src/test/resources`. Με τη δημιουργία ενός έργου η Maven φροντίζει επίσης να προσθέσει μια τυπική εξάρτηση στο POM την εξάρτηση JUnit η οποία μπορεί να χρησιμοποιηθεί για δοκιμές, με πεδίο δράσης (scope) `test` ώστε να μην πακεταριστεί με το `jar/war` κατά τη δημοσίευση του έργου.

Οι δοκιμές μονάδας ενός MOJO γράφονται σαν οποιαδήποτε άλλη δοκιμή περίπτωσης JUnit (JUnit test case). Η εκτέλεση δοκιμών με JUnit γίνεται με την εντολή `mvn test`. Όπως έχουμε δει η Maven με την εκτέλεση αυτής της εντολής θα εκτελέσει ένα σύνολο προσθέτων πριν από αυτό, όπως το `compile` για παράδειγμα.

Για την επίτευξη αυτής της μεθόδου πρέπει να γραφεί μια κλάση η οποία θα επεκτείνει την κλάση `TestCase`. Τα περισσότερα MOJO όμως θέλουν διάφορες πληροφορίες για να λειτουργήσουν σωστά. Για παράδειγμα πρέπει να εισαχθεί μια αναφορά (reference) σε ένα έργο της Maven ώστε να μπορεί το MOJO να λάβει μεταβλητές του έργου.

Η εισαγωγή μεταβλητών σε ένα MOJO μπορεί να γίνει με μερικούς τρόπους. Ο ένας είναι με χρήση του `plexus container`. Πολλά MOJO είναι σχεδιασμένα να λειτουργούν με το `plexus container` εκτελώντας ένα κύκλο κατασκευής ή δηλώνοντας διάφορες εισαγόμενες εξαρτήσεις. Εάν μόνο οι υπηρεσίες του `plexus container` είναι επιθυμητές τότε η κλάση θα επεκτείνει την κλάση `PlexusTestCase` αντί την `TestCase`.

Η βιβλιοθήκη λογισμικού (library) `maven-plugin-testing-harness` είναι ένας άλλος τρόπος διαχείρισης των δοκιμών σε κάποιο MOJO. Μπορεί για παράδειγμα να προκατασκευάσει τα εξαρτήματα `plexus` (`plexus components`) παρέχοντας στελέχη αντικειμένων (`stub projects`) για να επιτύχει έτσι λειτουργικότητα της Maven – όπως έργα ή γεμίζοντας πεδία από ένα αρχείο `xml` το οποίο είναι παρόμοιο με την διαμόρφωση των προσθέτων στο POM.

Το `maven-plugin-testing-harness` είναι σχεδιασμένο να εκτελεί δοκιμές για την υλοποίηση: `org.apache.maven.reporting.AbstractMavenReport#execute()`. Για τη χρήση του πρέπει να δηλωθεί σαν εξάρτηση στο έργο και να δημιουργηθεί μια κλάση `MojoTest` η οποία επεκτείνει την `AbstractMojoTestCase`. Ο κώδικας για την εξάρτηση να είναι ο παρακάτω:

```
<dependency>

  <groupId>org.apache.maven.shared</groupId>

  <artifactId>maven-plugin-testing-harness</artifactId>

  <version>1.0-beta-1</version>

  <scope>test</scope>

</dependency>
```

Η κλάση που θα εκτελεί την δοκιμή για κάποιο MOJO

```
public class YourMojoTest

    extends AbstractMojoTestCase

{

    /**

    * @see junit.framework.TestCase#setUp()

    */

    protected void setUp() throws Exception {

        // required for mojo lookups to work

        super.setUp();

    }

    public void testMojoGoal() throws Exception

    {

        testPom = new File( getBasedir(),

            "src/test/resources/unit/basic-test/basic-test-plugin-config.xml" );

        YourMojo mojo = (YourMojo) lookupMojo ("yourGoal", testPom);

        assertNotNull( mojo );

    }

}
```

Το `maven-verifier` είναι ένας τρόπος για εκτέλεση δοκιμών ένταξης/ενσωμάτωσης. Οι δοκιμές εκτελούνται με χρήση των JUnit ή TestNG και παρέχουν στο χρήστη μια απλή κλάση επιτρέποντας του να εκτελέσει τη Maven και να βεβαιώσει το αρχείο log και τα αντικείμενα κατασκευής. Επίσης παρέχει ένα αντικείμενο τύπου `ResourceExtractor` το οποίο εξάγει ένα έργο της Maven από τον κατάλογο `src/test/resources` σε ένα προσωρινό κατάλογο όπου ο χρήστης μπορεί να πειραματιστεί μαζί του. Το `maven-verifier` το χρησιμοποιεί και η ίδια η Maven για να εκτελέσει τις δοκιμές ένταξης.

Ένα πρόσθετο που μπορεί να χρησιμοποιηθεί για δοκιμές ένταξης σε πρόσθετα είναι το `invoker`. Η εκτέλεσή τους γίνεται με την εντολή `invoker:run`. Τέτοιες δοκιμές δεν εκτελούνται μέσω JUnit ή TestNG αλλά από την ίδια την Maven.

Αρχικά πρέπει να εκτελεστεί ο στόχος `install` του προσθέτου. Αυτός θα εκτελεστεί κατά τη διάρκεια της φάσης `pre-integration-test` και θα αντιγράψει το κυρίως αντικείμενο, καθώς και άλλα προσκολλημένα αντικείμενα στον κατάλογο `target/local-repo`. Επιπλέον κάποια τοπικά γονικά POM που θα βρεθούν θα αντιγραφούν στο προσωρινό αποθετήριο. Τέλος εάν το έργο είναι τύπου `reactor` όλες οι εξαρτήσεις θα αντιγραφούν επίσης.

Έπειτα πρέπει να εκτελεστεί ο στόχος `run` (`invoker:run`). Ο στόχος αυτός θα εκτελεστεί κατά τη φάση `integration-test` και θα χρησιμοποιήσει τα διαμορφωμένα πρότυπα συμπερίληψης / εξαίρεσης για να αναζητήσει POM τύπου IT (Integration Test) στον κατάλογο `src/it`. Για κάθε κατάλογο που θα βρεθούν IT POM θα αντιγραφούν όλα τα περιεχόμενά του στον κατάλογο `target/it`. Επιπλέον τα IT POM θα φιλτραριστούν και εκφράσεις όπως `@project.version@` θα αντικατασταθούν με τις αντίστοιχες στο POM του έργου.

Έπειτα μια κατασκευή Maven θα ξεκινήσει εκτελώντας τη φάση `package` (σύμφωνα με την προεπιλεγμένη ρύθμιση του πρόσθετου `invoker` κάτι που μπορεί να αλλάξει στις ρυθμίσεις του πρόσθετου ή στις ρυθμίσεις της δοκιμής ένταξης). Στο τέλος της δοκιμής το πρόσθετο θα ξεκινήσει μια προαιρετική δέσμη ενεργειών (`script`) μετά-κατασκευής η οποία θα ελέγξει ότι η εκτέλεση της δοκιμής ήταν επιτυχής και παράγαγε τα αποτελέσματα που έπρεπε.

Ένα άλλο είναι το `shitty-maven-plugin` (Super Helpful Integration Testing ThingY) το οποίο παρέχει ένα απλό τρόπο για εκτέλεση δοκιμών ένταξης για ένα έργο, (απλή μονάδα ή πολλαπλή μονάδα). Αυτό το πρόσθετο πραγματοποιεί παρόμοιες λειτουργίες με το πρόσθετο `invoker` αλλά έχει κάποιες λειτουργίες οι οποίες δεν υπάρχουν στο `invoker`. Παρέχει κάποια προηγμένα βήματα για αυτόματη εγκατάσταση ενός πρόσθετου χρησιμοποιώντας μια ειδική ονομασία έκδοσης (την `TESTING`) έτσι ώστε τα εικονικά έργα (που εκτελούν τις δοκιμές) να βασίζονται σε αυτήν την έκδοση αποκλειστικά.

Το πρόσθετο αυτό είναι ένα άθροισμα μερικών στόχων με μερικώς επαυξημένη συμπεριφορά προσαρμοσμένων για δοκιμές ένταξης. Αφαιρεί την παραγωγή και τις καταγραφές (logs) των μονάδων δοκιμών ένταξης. Εγκαθιστά αυτά αντικείμενα μονάδων που έχουν ονομασία έκδοσης testing έτσι ώστε να είναι σίγουρο ποια αντικείμενα χρησιμοποιούνται στις δοκιμές. Επίσης ξεκινάει παιδικές διαδικασίες της Maven (child Maven processes) για κάθε μονάδα δοκιμής ένταξης και καταγράφει τα αποτελέσματα.

Οι στόχοι του πρόσθετου ρυθμίζονται ώστε να εκτελούνται σε σχέση με τη φάση integration-test και άλλες παρόμοιες φάσεις και η εγκατάσταση αντικειμένου δοκιμής συμβαίνει πριν την κανονική φάση install και η φάση install θα διαδώσει την πραγματική έκδοση αντικειμένου στο αποθετήριο όταν όλες οι δοκιμές ένταξης μονάδων έχουν εκτελεστεί επιτυχώς. Εάν μια από αυτές αποτύχει τότε όλη η κατασκευή αποτυγχάνει μετά από την εκτέλεση όλων των δοκιμών ένταξης.

Το maven-it-plugin χρησιμοποιείται στη φάση integration-test. Δεν μπορεί να χρησιμοποιηθεί για την δοκιμή ενός πρόσθετου που χρησιμοποιείται για πρώτη φορά (χωρίς να έχει δημοσιευτεί). Σε αυτήν την περίπτωση θα ήταν δυνατό να οριστεί μια έκδοση it-snapshot (version 1.0-it-SNAPSHOT) του πρόσθετου με χρήση του πρόσθετου install κατά τη διάρκεια της φάσης pre-integration-test.

Το πρόσθετο maven-plugin-management-plugin υπάρχει για να προσθέτει/αφαιρεί ένα άλλο πρόσθετο στο τοπικό αποθετήριο για τις φάσεις pre-integration-test και post-integration-test. Σε την περίπτωση πρέπει να ρυθμιστεί το maven-plugin-test-plugin και το maven-surefire-plugin.

ΚΕΦΑΛΑΙΟ 5

Σύγκριση Apache Maven με Apache Ant

Εισαγωγή

Η Apache Ant είναι μια βιβλιοθήκη λογισμικού και ένα εργαλείο εντολών που χρησιμοποιείται για την κατασκευή εφαρμογών Java ή και άλλων γλωσσών όπως C ή C++. Η Maven είναι βασισμένη στην Ant και τα δυο εργαλεία έχουν παρόμοια χρήση. Η Ant είναι ένα σύστημα κατασκευής δημιουργημένο με βάση το λογισμικό Make με στόχους (targets όχι goals) και εξαρτήσεις.

ΥΠΟΚΕΦΑΛΑΙΟ 5.1

Η Ant λειτουργεί με ένα σύνολο εργασιών (tasks) που επιτρέπουν το χρήστη να μεταγλωττίσει, να συναρμολογήσει, να εκτελέσει εφαρμογές Java και δοκιμές για τις εφαρμογές αυτές. Η Ant είναι γραμμένη στη γλώσσα Java και οι χρήστες της μπορούν να γράψουν δικά τους «antlib» τα οποία περιέχουν εργασίες και τύπους (types) Ant. Ένας μεγάλος αριθμός εμπορικών ή ανοιχτού λογισμικού antlib είναι διαθέσιμος. Η Ant είναι πολύ ευέλικτη και δεν επιβάλλει συμβάσεις κώδικα ή διατάξεις καταλόγων στα έργα Java που θα τη χρησιμοποιήσουν σαν εργαλείο κατασκευής.

Η Ant λειτουργεί με αρχεία κατασκευής γραμμένα σε μορφή xml. Κάθε τέτοιο αρχείο περιέχει ένα έργο και τουλάχιστον ένα στόχο (target). Οι στόχοι περιέχουν στοιχεία εργασιών (task elements) κάθε ένα από τα οποία περιέχει ένα χαρακτηριστικό id και σε αυτό μπορεί αργότερα να γίνει μια αναφορά χρησιμοποιώντας το χαρακτηριστικό αυτό οπότε δημιουργείται ο κανόνας κάθε τιμή id πρέπει να είναι μοναδική. Κάθε έργο στην Ant περιέχει τρία χαρακτηριστικά. Όνομα (name), προεπιλογή (default) και βασικό κατάλογο (basedir).

Το χαρακτηριστικό «όνομα» περιέχει το όνομα του έργου αλλά δεν απαιτείται να γραφτεί. Το δεύτερο χαρακτηριστικό, το default είναι αυτό που θα χρησιμοποιηθεί εάν δεν δοθεί κανένας στόχος (target). Αυτό επίσης δεν είναι απαιτούμενο αλλά κάθε έργο συμπεριλαμβάνει ένα υπονοούμενο στόχο που περιέχει όλες τις εργασίες και τους τύπους (types). Αυτός ο στόχος θα εκτελείται πάντα σαν μέρος της αρχικοποίησης του έργου.

Το τρίτο χαρακτηριστικό, το basedir είναι ο βασικός κατάλογος από τον οποίο υπολογίζονται όλες οι διαδρομές. Αυτό το χαρακτηριστικό μπορεί να υπερσκελιστεί ρυθμίζοντας την ιδιότητα basedir εκ των προτέρων. Όταν συμβεί αυτό πρέπει να παραληφθεί στην ετικέτα του έργου. Εάν ούτε το χαρακτηριστικό ούτε η ιδιότητα δεν ρυθμιστούν τότε θα χρησιμοποιηθεί ο γονικός κατάλογος του αρχείου κατασκευής. Μια σχετική διαδρομή επιλύεται σχετικά με τον κατάλογο

που περιλαμβάνει το αρχείο κατασκευής. Αυτό το χαρακτηριστικό δεν είναι απαραίτητο.

Κάθε έργο ορίζει έναν ή περισσότερους στόχους (target). Θα μπορούσε να υπάρχει ένας στόχος για μεταγλώττιση, ένας για δημιουργία διανεμητή (distributable). Ο διανεμητής μπορεί να κατασκευαστεί μόνο όταν έχει πραγματοποιηθεί η μεταγλώττιση έτσι ο στόχος distribute θα βασίζεται στον στόχο compile. Η Ant επιλύει τέτοιες εξαρτήσεις.

Μια εργασία (task) είναι ένα κομμάτι κώδικα που μπορεί να εκτελεστεί. Μπορεί να έχει πολλαπλά χαρακτηριστικά η τιμές των οποίων μπορεί να περιέχουν αναφορές σε μια ιδιότητα. Τέτοιες αναφορές θα επιλυθούν πριν εκτελεστεί η εργασία.

Ιδιότητες είναι ένας σημαντικός τρόπος για να προσαρμοστεί η διαδικασία κατασκευής. Στην πιο απλή μορφή οι ιδιότητες ορίζονται είτε μέσα στο αρχείο κατασκευής είτε έξω από την Ant. Μια ιδιότητα έχει ένα όνομα και μια τιμή.

Ένα έργο μπορεί να έχει μια συλλογή ενδείξεων οι οποίες μπορεί να επεκτείνονται αυτόματα όταν ένα αρχείο αντιγράφεται, όταν η συμπεριφορά φιλτραρίσματος-αντιγραφής είναι επιλεγμένη στις εργασίες που την υποστηρίζουν. Τέτοιες συμπεριφορές μπορεί να είναι ρυθμιστούν στο αρχείο κατασκευής από την εργασία filter.

Μπορούν να καθοριστούν αναφορές τύπου PATH- και CLASSPATH- με τους χαρακτήρες «:» και «;» σαν χαρακτήρες διαχωρισμού. Η Ant θα μετατρέψει τους χαρακτήρες διαχωρισμού στο σωστό χαρακτήρα για κάθε λειτουργικό σύστημα.

Η Ant υποστηρίζει ένα μηχανισμό πρόσθετων για χρήση εξωτερικών ενεργειών (third party tasks). Αυτές τοποθετούνται σε σημείο που θα μπορούν να βρεθούν από την Ant και δηλώνονται. Η δήλωση μπορεί να γίνει, μια ενέργεια για κάθε εντολή, ένα σύνολο ενεργειών με ένα αρχείο ιδιοτήτων , ένα σύνολο ενεργειών με ένα αρχείο xml, ένα σύνολο ενεργειών με ένα αρχείο xml με το όνομα antlib.xml.

ΥΠΟΚΕΦΑΛΑΙΟ 5.2

Οι χρήστες που θέλουν να χρησιμοποιήσουν ένα εργαλείο κατασκευής πρέπει να επιλέξουν εάν θέλουν να υιοθετήσουν την Apache Ant ή την Apache Maven.

Η Ant είναι ένα εξαιρετική στην διαδικασία κατασκευής. Είναι βασισμένη σε στόχους (target) ο κάθε ένας από τους οποίους αποτελείται από εντολές κωδικοποιημένες σε xml. Υπάρχει η εργασία copy, η javac και η εργασία jar. Όταν κάποιος χρησιμοποιεί την Ant, την προμηθεύει με ειδικές εντολές για μεταγλώττιση και πακετάρισμα του εξαγόμενου αποτελέσματος.

Ο χρήστης πρέπει να «πει» στην Ant ακριβώς τι να κάνει. Για έναν στόχο javac ο οποίος πραγματοποιεί την μεταγλώττιση και την τοποθετεί σε ένα κατάλογο πρέπει ο χρήστης να προσδιορίσει σε ποιον κατάλογο η Ant θα βρει τον πηγαίο κώδικα, σε ποιον κατάλογο θα τοποθετηθεί ο μεταγλωττισμένος κώδικας και πως θα πακεταριστεί σε αρχείο jar. Ενώ υπάρχουν εξελίξεις στον τομέα οι οποίες θα βοηθήσουν την Ant ώστε να γίνει λιγότερο διαδικαστική, η εμπειρία του χρήστη με την Ant είναι η κωδικοποίηση μια διαδικαστικής γλώσσας γραμμένης σε κώδικα xml.

Εάν αυτό συγκριθεί με την Maven θα δούμε ότι για να γίνει το πακετάρισμα το μόνο που έχει να κάνει ο χρήστης είναι να δημιουργήσει το POM να τοποθετήσει τον πηγαίο κώδικα στον κατάλογο «src/main/java» και να εκτελέσει την εντολή mvn install από τη γραμμή εντολών. Για ένα απλό παράδειγμα χωρίς εξαρτήσεις το POM θα μπορούσε να περιέχει μόνο τα modelVersion, groupId, artifactId και version.

Η εντολή mvn install θα εκτελέσει μια σειρά άλλων απαραίτητων εντολών οπότε θα κατεργαστεί τους πόρους, θα μεταγλωττίσει τον πηγαίο κώδικα, θα εκτελέσει δοκιμές μονάδας, θα δημιουργήσει το πακέτο jar και θα το εγκαταστήσει στο τοπικό αποθετήριο για επαναχρησιμοποίηση σε άλλα έργα. Επίσης το mvn site θα δημιουργήσει ένα αρχείο index.html στον κατάλογο «target/site» ο οποίος θα περιέχει συνδέσμους προς αναφορές javadoc και κάποιες αναφορές για τον πηγαίο κώδικα.

Όλα αυτά για ένα απλό παράδειγμα και όχι ένα πραγματικό έργο. Ένα πραγματικό έργο θα ήθελε προσαρμογή έως κάποιο βαθμό. Το POM μεταβάλλεται και το μέγεθός του μεγαλώνει ανάλογα με την προσαρμογή που θέλει να του κάνει ο χρήστης. Σε μεγάλα και πολύπλοκα έργα υπάρχουν πολλαπλά και πολύπλοκα POM τα οποία περιέχουν πολλά πρόσθετα και δηλώσεις εξαρτήσεων με προσαρμοσμένη συμπεριφορά για να λειτουργούν σύμφωνα με τις ανάγκες του έργου. Αλλά ακόμα και τότε το POM διατηρεί άλλου είδους πληροφορίες για το έργο από ότι ένα έργο παρομοίου μεγέθους δημιουργημένο σε Ant.

Το POM της Maven περιέχει δηλώσεις του τύπου «Αυτό είναι ένα έργο Jar» ή «Ο πηγαίος κώδικας βρίσκεται στον κατάλογο ...». Η Ant θα είχε πληροφορίες του τύπου «Αυτό είναι ένα έργο», «Ο πηγαίος κώδικας βρίσκεται στον κατάλογο ...», «Εκτέλεσε την εντολή javac στον κατάλογο...», «Τοποθέτησε τα αποτελέσματα στον κατάλογο...», «Δημιούργησε ένα πακέτο Jar από το...» και άλλες παρόμοιες οδηγίες. Ενώ η Ant πρέπει να είναι σαφής για την διαδικασία, η Maven ξέρει που θα βρει τον πηγαίο κώδικα και πώς να τον επεξεργαστεί.

Η Ant δεν έχει επίσημες συμβάσεις όπως μια κοινή δομή έργου ή κοινή συμπεριφορά. Ο χρήστης πρέπει να πει ακριβώς που να βρει τον πηγαίο κώδικα και που να τοποθετήσει το αποτέλεσμα. Με την πάροδο του χρόνου ανεπίσημες συμβάσεις έχουν προκύψει αλλά δεν έχουν κωδικοποιηθεί στο λογισμικό. Η Maven από την άλλη ξέρει που θα βρει τον πηγαίο κώδικα εάν ο χρήστης ακολουθήσει τη σύμβαση και εάν ο χρήστης επιθυμεί μπορεί να τροποποιήσει την συμπεριφορά της Maven για να ταιριάζει στα μέτρα του. Η Maven ξέρει που να τοποθετήσει τον μεταγλωττισμένο κώδικα χωρίς να απαιτήσει από τον χρήστη να το δηλώσει πουθενά.

Η Ant είναι διαδικαστική κάτι που σημαίνει ότι ο χρήστης πρέπει να τις πει τι να κάνει, πότε να το κάνει και με ποιά σειρά να το κάνει. Πρέπει να τις πει ότι πρέπει πρώτα να μεταγλωττίσει, έπειτα να αντιγράψει τα αποτελέσματα και μετά να τα πακετάρει. Η Maven από την άλλη είναι δηλωτική. Αυτό σημαίνει ότι ο χρήστης πρέπει να δημιουργήσει το POM.xml και να τοποθετήσει τον πηγαίο

κώδικα στον προεπιλεγμένο κατάλογο. Όλη η υπόλοιπη διαδικασία ακολουθείται από τη Maven.

Τέλος η Ant δεν έχει κύκλους κατασκευής. Ο χρήστης πρέπει να ορίσει τους στόχους και τις εξαρτήσεις και να συνάψει μια σειρά εργασιών σε κάθε στόχο. Η Maven αφετέρου έχει ένα κύκλο κατασκευής ο οποίος ξεκίνησε με την εκτέλεση της εντολής `mvn install`. Η εντολή αυτή «είπε» στη Maven να εκτελέσει μια σειρά σειριακών φάσεων κύκλων κατασκευής μέχρι να φτάσει στην φάση κύκλου κατασκευής `install`. Ως αποτέλεσμα η Maven πραγματοποίησε όλη την εργασία που ο χρήστης ήθελε, μεταγλωττίζοντας τον πηγαίο κώδικα και πακετάροντας το αρχείο `Jar`.

Η Maven έχει την ευφυΐα του να εκτελεί κοινές εργασίες σε ένα έργο με τη μορφή πρόσθετων. Εάν ο χρήστης θέλει να συγγράψει και να εκτελέσει δοκιμές μονάδας το μόνο που έχει να κάνει είναι η συγγραφή των δοκιμών, η τοποθέτηση των στον προεπιλεγμένο κατάλογο «`src/test/java`» και η τοποθέτηση εξάρτησης με `scope` με τιμή `test` (`TestNG` ή `JUnit`) και η εκτέλεση της εντολής `mvn test`.

Αυτά είναι δυνατόν να γίνουν και στην Ant αλλά ο χρήστης πρέπει να γράψει τις οδηγίες από την αρχή. Στην Ant ο χρήστης πρέπει πρώτα να καταλάβει που πρέπει να τοποθετηθούν τα `JUnit` και `Jar`. Τότε πρέπει να δημιουργηθεί μια διαδρομή `class` που θα περιλαμβάνει τα αρχεία `JUnit` και `Jar`. Μετά ο χρήστης πρέπει να πει στην Ant που να ψάξει για τον κώδικα δοκιμής, να γράψει έναν στόχο ο οποίος μεταγλωττίζει τον κώδικα δοκιμής και να εκτελέσει τις δοκιμές μονάδας.

Χωρίς τη βοήθεια των `Antlib` και `Ivy` (ή ακόμα και με τη βοήθεια αυτών) η Ant μοιάζει με μια προσαρμοσμένη διαδικαστική κατασκευή. Μια συλλογή POM της Maven σε ένα έργο, το οποίο ακολουθεί τις συμβάσεις της, έχει λίγες εντολές `xml` συγκριτικά με ένα παρόμοιο στην Ant. Ένα άλλο προνόμιο που απολαμβάνουν οι χρήστες της Maven είναι το να βασίζονται σε μια μεγάλη λίστα διαμοιραζόμενων δημοφιλών πρόσθετων. Οι περισσότεροι χρήστες χρησιμοποιούν το πρόσθετο `surefire` για εκτέλεση δοκιμών μονάδας και εάν κάποιος προσθέσει υποστήριξη για ένα νέο πλαίσιο δοκιμών κάποιος χρήστης μπορεί να κερδίσει καινούριες δυνατότητες στα δικά του έργα απλά αυξάνοντας την έκδοση κάποιου πρόσθετου της Maven στο POM του έργου του.

Παρόλα αυτά η χρήση της Ant είναι μερικές φορές αναγκαία και έχει ακόμα θέση σε μια περίπλοκη κατασκευή. Εάν αυτή η κατασκευή έχει κάποιες ιδιαίτερα προσαρμοσμένες διαδικασίες ή εάν ο χρήστης έχει γράψει κάποιες δέσμες ενεργειών με την Ant για να ολοκληρώσει μια συγκεκριμένη διαδικασία με ένα συγκεκριμένο τρόπο και δεν μπορεί να μεταφερθεί σε Maven μπορεί να χρησιμοποιήσει τις ίδιες αυτές δέσμες ενεργειών Ant όπως είναι στη Maven.

Η Ant είναι διαθέσιμη σαν πρόσθετο πυρήνα της Maven. Επίσης προσαρμοσμένα πρόσθετα της Maven μπορούν να εφαρμοστούν στην Ant και έργα δημιουργημένα με Maven μπορούν να διαμορφωθούν ώστε να εκτελούν δέσμες ενεργειών Ant μέσα σε ένα κύκλο κατασκευής της Maven.

ΥΠΟΚΕΦΑΛΑΙΟ 5.3

Για την εκτέλεση εργασιών Ant μέσω Maven η διαδικασία είναι εύκολη. Πραγματοποιείται με τη χρήση δύο πρόσθετων της Maven, τα πρόσθετα ant και antrun. Το πρώτο από τα δύο πρόσθετα μπορεί να χρησιμοποιηθεί για την δημιουργία αρχείων κατασκευής Ant για ένα έργο της Maven χρησιμοποιώντας το αρχείο POM.xml. Η εντολή `mvn ant:ant` είναι αυτή που δημιουργεί τα αρχεία `build.xml`, `maven-build.xml` και `maven-build.properties`.

Το αρχείο `build.xml` είναι ένα γενικό αρχείο Ant και μπορεί να επεξεργαστεί από τον χρήστη. Περιέχει μια γραμμή στην οποία δηλώνεται το όνομα του έργου, το είδος του πακέτου (από προεπιλογής είναι `jar`) και η παράμετρος `basedir` με τιμή «..». Επίσης εισάγει το αρχείο `maven-build.xml` με την εντολή `<import file="maven-build.xml"/>` και ένα απλό στόχο (`target`) με το όνομα `help` που παροτρύνει τον χρήστη να καλέσει την εντολή βοήθειας `ant -projecthelp`.

Κατά την εκτέλεση της εντολής `mvn ant:ant` το αρχείο `build.xml` δεν θα αντικατασταθεί εάν προϋπάρχει. Για την αντικατάστασή του πρέπει να εκτελέσουμε την εντολή «`mvn ant:ant -Doverwrite=true`».

Το αρχείο `maven-build.xml` είναι ένα αρχείο Ant για το έργο. Περιέχει αρκετούς στόχους (`targets`) όπως `clean`, `test`, `compile` και `package` οι οποίοι πραγματοποιούν λειτουργίες όπως καθαρισμός, δοκιμή, μεταγλώττιση και πακετάρισμα. Κάποιοι άλλοι στόχοι δημιουργούν την τεκμηρίωση του έργου σε μορφή `Javadoc`. Με την εντολή `ant -projecthelp` μπορεί ο χρήστης να δει όλους τους διαθέσιμους στόχους της Ant. Σε αντίθεση με το αρχείο `build.xml` το `maven-build.xml` θα αντικατασταθεί από την εντολή `mvn ant:ant` οπότε η επεξεργασία του δεν έχει νόημα.

Τέλος το αρχείο `maven-build.properties` περιέχει ιδιότητες οι οποίες χρησιμοποιούνται από το αρχείο `maven-build.xml`. Αυτό το αρχείο θα αντικατασταθεί από την εντολή `mvn ant:ant` επίσης.

Το πρόσθετο `antrun` μπορεί να χρησιμοποιηθεί για εκτέλεση εργασιών Ant μέσα από τη Maven. Οι δέσμες ενεργειών Ant (`Ant scripts`) μπορούν να

ενσωματωθούν στο POM της Maven. Η Maven ενθαρρύνει την μεταφορά των δεσμών αυτών σε ένα αρχείο build.xml ώστε να μην «μολύνεται» το POM και η εκτέλεσή του μέσω του POM με την εντολή `<ant/> task`.

Ένας από τους σκοπούς αυτού του πρόσθετου είναι η διευκόλυνση της μετανάστευσης από έργα βασισμένα σε Ant σε έργα Maven. Κάποια έργα μπορεί να μην μπορούν να μεταφερθούν γιατί βασίζονται σε λειτουργικότητα προσαρμοσμένης κατασκευής την οποία η Maven δεν παρέχει.

Το πρόσθετο αυτό εκτελεί εργασίες Ant μέσω της Maven. Για να γίνει αυτό πρέπει να υπάρχει ένα υπάρχον έργο και το πρόσθετο πρέπει να έχει την ετικέτα (tag) `target` διαμορφωμένη στο POM του πρόσθετου. Οι εργασίες Ant τοποθετούνται μεταξύ των ετικετών `<target>` και `</target>`. Οι εργασίες Ant μπορούν να συνδεθούν με φάσεις κύκλων κατασκευής της Maven και να εκτελούνται κατά τις φάσεις αυτές.

Εάν οι εργασίες Ant δημιουργούν επιπλέον πηγαίο κώδικα ο οποίος πρέπει να προστεθεί στην κατασκευή τότε το πρόσθετο `build-helper-maven-plugin` είναι αυτό που μπορεί να χρησιμοποιηθεί.

Όλες οι ιδιότητες που είναι διαθέσιμες στη Maven είναι επίσης διαθέσιμες στην διαμόρφωση της ετικέτας `target`. Ωστόσο είναι καλό να κληθεί μια εξωτερική δέσμη ενέργειας κατασκευής Ant (Ant build script) χρησιμοποιώντας την εργασία `ant task`. Προς αποφυγή της σύγκρουσης ονομάτων μόνο ένα μέρος των ιδιοτήτων είναι διαθέσιμο στην εξωτερική κατασκευή Ant. Μεταξύ αυτών είναι και όλες οι ιδιότητες που είναι ορισμένες στο σημείο `properties` στο POM. Επίσης περιέχει προκαθορισμένες εκδόσεις μερικών από τις πιο συχνά χρησιμοποιούμενες ιδιότητες της Maven. Εάν η ιδιότητα που θέλει να χρησιμοποιήσει ο χρήστης δεν είναι διαθέσιμη σε εξωτερικό αρχείο τότε πρέπει να επανακαθορίσει την ιδιότητα πριν καλέσει την Ant.

Η χρήση αρκετών χαρακτηριστικών της Maven μέσω της Ant μπορεί να πραγματοποιηθεί με ένα επιπλέον λογισμικό το οποίο μπορεί να βρεθεί στην ιστοσελίδα της Maven <http://maven.apache.org/ant-tasks/index.html>. Οι εργασίες Maven Ant επιτρέπουν λειτουργίες της Maven όπως διαχείριση εξαρτήσεων

(dependency management), εγκατάσταση αντικειμένων (artifact deployment), επεξεργασία POM (POM processing).

ΚΕΦΑΛΑΙΟ 6

Ιστοσελίδα

Εισαγωγή

Για τις ανάγκες της πτυχιακής έχει δημιουργηθεί μια ιστοσελίδα τύπου JSP (Java Server Pages). Η γλώσσα προγραμματισμού συνδυάζει κώδικα html με κώδικα Java. Για την δημιουργία της τα εργαλεία που χρησιμοποιήθηκαν είναι: Eclipse, Apache Maven (η ενσωματωμένη έκδοση της στο Eclipse), Apache Tomcat καθώς είναι απαραίτητο για να φορτωθούν τα αρχεία τύπου JSP και MySql για την αποθήκευση διαφόρων στοιχείων της ιστοσελίδας. Πληροφορίες των πελατών/χρηστών της ιστοσελίδας όπως όνομα, διεύθυνση κλπ, πληροφορίες για τις παραγγελίες που αυτοί έχουν αποστείλει ή πληροφορίες του υπάλληλου (ή των υπαλλήλων) που είναι υπεύθυνος για τον έλεγχο των παραγγελιών.

ΥΠΟΚΕΦΑΛΑΙΟ 6.1

Ο κώδικας της ιστοσελίδας, ο οποίος εκτείνεται σε τριάντα διαφορετικά αρχεία, είναι γραμμένος σε γλώσσα JSP. Η γλώσσα αυτή είναι κώδικας html συνδυασμένος με κομμάτια κώδικα Java. Τα κομμάτια Java ξεχωρίζουν από τον υπόλοιπο κώδικα αφού περικλείονται από τους χαρακτήρες '<' και '>'. Έπειτα από τον αρχικό χαρακτήρα ακολουθεί το σύμβολο '%' ώστε να μπορεί ο διακομιστής να τα ξεχωρίσει.

Κάθε αίτημα (request) προς την ιστοσελίδα ξεκινάει από κάποιον περιηγητή (π.χ. Internet Explorer) ο οποίος ζητάει από τον διακομιστή κάποιο από τα αρχεία της ιστοσελίδας. Εκεί τα κομμάτια κώδικα Java μεταγλωττίζονται και στη συνέχεια εκτελούνται με χρήση του περιβάλλοντος Java. Έπειτα αφαιρούνται από τον κώδικα και το αρχείο στέλνεται σαν απάντηση (response) στον περιηγητή ο οποίος δεν θα καταλάβει ποτέ ότι η ιστοσελίδα περιείχε και κάτι άλλο που ο ίδιος δεν έλαβε ποτέ.

Βλέποντας παρακάτω στην «εικόνα 21» είναι ένα παράδειγμα κώδικα στο οποίο φαίνεται η διαφορά μεταξύ κώδικα του αρχείου και κώδικα τον οποίο βλέπει ο περιηγητής. Στην παρακάτω εικόνα βλέπουμε πως αναμειγνύονται οι κώδικες Java και Html. Το κομμάτι κώδικα αυτό ξεκινάει με Java και παρατηρούμε την εκτέλεση ενός query SQL προς τη βάση δεδομένων μέσα σε ένα "try – catch" για να πάρουμε μια λίστα με τις παραγγελίες και διάφορες πληροφορίες της παραγγελίας αυτής. Έπειτα ο κώδικας μετατρέπεται σε html όπου δημιουργούμε ένα πίνακα για να προβάλουμε τις πληροφορίες αυτές. Ο διακομιστής Apache Tomcat εκτελεί τον κώδικα, το query SQL και τοποθετεί σε κάθε ένα κελί μια εγγραφή η οποία περιέχει, για παράδειγμα, τον τύπο της παραγγελίας, το όνομα χρήστη που την υπέβαλε κ.α. Στο τέλος του try έρχεται το catch για πιθανά σφάλματα εκτέλεσης και το finally για να κλείσει τα αντικείμενα που χρησιμοποιήθηκαν για την σύνδεση με τη βάση.

Στην «εικόνα 22» μπορούμε να δούμε τον κώδικα όπως έφτασε στον περιηγητή. Τα κομμάτια Java έχουν αφαιρεθεί και στη θέση τους έχουν προστεθεί δυναμικά πληροφορίες της παραγγελίας, που είναι αποθηκευμένες στη βάση δεδομένων.

```

try{
    rs = s.executeQuery(sql_query);
    rs.next();
    %>
    <table border="3" align="center">
    <tr>
        <td>Αριθμός Παραγγελίας:</td>
        <td><%= rs.getInt("order_id") %></td>
    </tr>
    <tr>
        <td>Όνομα Χρήστη που έκανε την παραγγελία:</td>
        <td><%= rs.getString("username") %></td>
    </tr>
    <tr>
        <td>Τύπος παραγγελίας</td>
        <td><%= rs.getString("order_type") %></td>
    </tr>
    <tr>
        <td>Τεμάχια:</td>
        <td><%= rs.getString("order_quantity") %></td>
    </tr>
    <tr>
        <td>Είδος Χαρτιού:</td>
        <td><%= rs.getString("order_paper_type") %></td>
    </tr>
    <tr>
        <td>Διαστάσεις:</td>
        <td><%= rs.getString("order_dimention") %></td>
    </tr>
    <tr>
        <td>Δίπλωμα:</td>
        <td><%= rs.getString("order_folding") %></td>
    </tr>
    <tr>
        <td>Βιβλιοδεσία: </td>
        <td><%= rs.getString("order_bookbind") %></td>
    </tr>
    <tr>
        <td>Πληροφορίες Παραγγελίας: </td>
        <td><%= rs.getString("order_information") %></td>
    </tr>
    </table>
    <br>
    <%
}
catch(Exception e){e.printStackTrace();}
finally{
if(s!=null) s.close();
if(con!=null) con.close();
}

```

Εικόνα 21

```

<table border="3" align="center">
<tr>
<td>Αριθμός Παραγγελίας:</td>
<td>23</td>
</tr>
<tr>
<td>Όνομα Χρήστη που έκανε την παραγγελία:</td>
<td>babis</td>
</tr>
<tr>
<td>Τύπος παραγγελίας</td>
<td>ΕΠΙΣΤΟΛΟΧΑΡΤΑ</td>
</tr>
<tr>
<td>Τεμάχια:</td>
<td>1000</td>
</tr>
<tr>
<td>Είδος Χαρτιού:</td>
<td>Velvet</td>
</tr>
<tr>
<td>Διαστάσεις:</td>
<td>123</td>
</tr>
<tr>
<td>Δίπλωμα:</td>
<td>2πτυχο</td>
</tr>
<tr>
<td>Βιβλιοδεσία: </td>
<td>Καρφίτσα</td>
</tr>
<tr>
<td>Πληροφορίες Παραγγελίας: </td>
<td>243562</td>
</tr>
</table>

```

Εικόνα 22

Αυτό από μόνο του είναι και καλό και κακό. Εάν είναι επιθυμητή η εκτέλεση κώδικα Java έπειτα από επιλογή του χρήστη όπως για παράδειγμα κατά το κλικ με ποντίκι σε ένα κουμπί (button), τότε πρέπει το αρχείο να ζητηθεί πάλι από τον διακομιστή Apache Tomcat να ξαναεκτελεστεί ο κώδικας Java και να σταλεί το αρχείο στον περιηγητή.

Ένα παράδειγμα τέτοιου προβλήματος, όπως θα το δούμε παρακάτω είναι όταν θέλουμε να δούμε σε μια σελίδα όλες τις παραγγελίες που έχουν στείλει οι πελάτες της επιχείρησης και θέλουμε να προβάλουμε αυτές τις παραγγελίες μια –

μία. Με μια σειρά δημιουργίας αντικειμένων συνδεόμαστε με την βάση MySQL και με εκτέλεση εντολών προς την βάση (SQL Queries) παίρνουμε τις πληροφορίες της κάθε παραγγελίας. Αυτό πραγματοποιείται με χρήση τριών διαφορετικών SQL Queries. Ένα για να πάρουμε τον αριθμό παραγγελιών που υπάρχουν στη βάση. Όλες οι παραγγελίες είναι αποθηκευμένες στη βάση σε έναν πίνακα. Ο πίνακας έχει διάφορες στήλες η κάθε μια είναι μια πληροφορία της παραγγελίας (π.χ. πόσα τεμάχια θέλει ο χρήστης) και κάθε παραγγελία είναι μια σειρά στον πίνακα.

Έπειτα δημιουργείται ένας πίνακας ακεραίων αριθμών (integer array) με μήκος το αποτέλεσμα της προηγούμενης εντολής SQL. Ο πίνακας αυτός θα γεμίσει με το δεύτερο SQL Query το οποίο θα επιτρέψει όλα τα πεδία της στήλης με όνομα `order_id`. Αυτή η στήλη περιέχει ακέραιους αριθμούς, έναν για κάθε μια γραμμή διαφορετικό από όλους τους άλλους των άλλων γραμμών. Έτσι σιγουρευόμαστε ότι οι παραγγελίες δεν θα μπερδευτούν ακόμα και αν είναι ακριβώς ολόιδιες. Η στήλη `order_id` είναι ρυθμισμένη να παίρνει τιμή αυτόματα κατά την εισαγωγή της παραγγελίας στη βάση (auto increment) με σκοπό την αυτοματοποίηση της διαδικασίας χωρίς να μπλέξει ο χρήστης της σελίδας με την βάση δεδομένων.

Τέλος ένας πίνακας HTML με κατάλληλο αριθμό στηλών και γραμμών γεμίζει τα κελιά του με το τρίτο και τελευταίο SQL Query. Αυτή η εντολή SQL χρησιμοποιεί τον πίνακα με τους ακέραιους αριθμούς κάθε ένας από τους οποίους αντιπροσωπεύει μια εγγραφή στη βάση και χρησιμοποιεί τον δείκτη του πίνακα ώστε να αναφερθεί στις εγγραφές του πίνακα της βάσης. Στην πρώτη θέση του πίνακα ακεραίων αριθμών βρίσκεται το `order_id` της πρώτης παραγγελίας, στη δεύτερη θέση του πίνακα ακεραίων βρίσκεται το `order_id` της δεύτερης εγγραφής στον πίνακα στη βάση δεδομένων (άρα και η δεύτερη παραγγελία) κ.ο.κ.

Η μετάβαση από παραγγελία σε παραγγελία είναι λίγο παράξενη. Στο σημείο αυτό βρίσκεται και το πρόβλημα της αφαίρεσης του κώδικα Java από το αρχείο πριν την αποστολή του στον περιηγητή. Δεν είναι δυνατή η αύξηση του δείκτη του πίνακα ακεραίων ώστε να αναφερθούμε στην επόμενη παραγγελία χωρίς να ξαναστείλουμε αίτημα στον διακομιστή. Έτσι μια φόρμα στέλνει τον

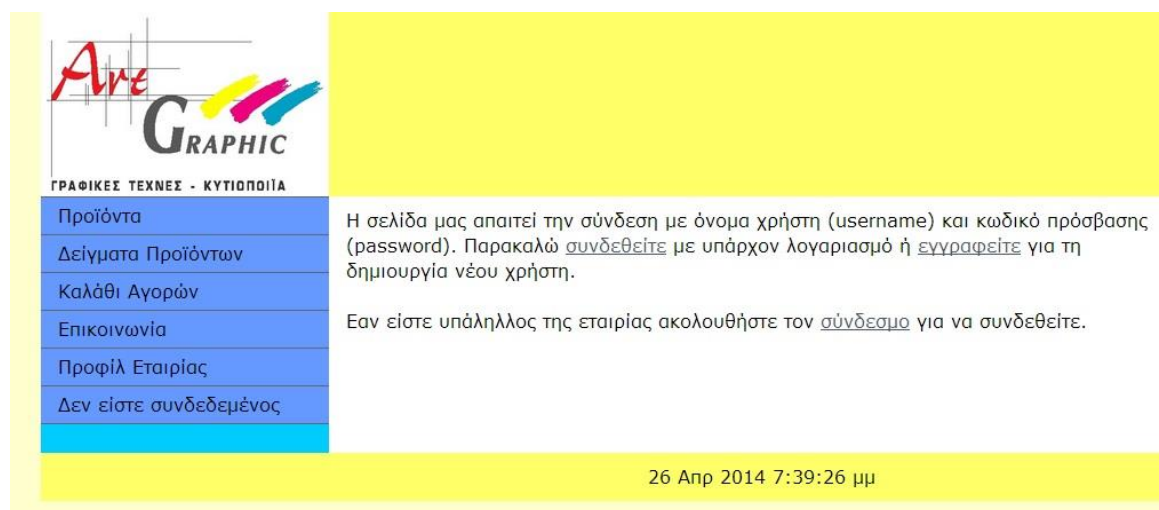
δείκτη (index) του πίνακα ακεραίων σε ένα άλλο αρχείο με χρήση της ιδιότητας action του HTML κώδικα όπου και αυξάνεται κατά ένα ή μειώνεται κατά ένα για την μετάβαση σε επόμενη ή προηγούμενη παραγγελία. Έπειτα με χρήση του κώδικα «response.sendRedirect("file.jsp");» ξαναστέλνει τον χρήστη στην προηγούμενη σελίδα όπου και ξαναεκτελείται ο κώδικας Java με τα τρία SQL Queries.

Η τεχνική αυτή χρησιμοποιείται σε τρεις διαφορετικές περιστάσεις. Όταν θέλει κάποιος χρήστης – πελάτης να δει τις παραγγελίες του, όταν θέλει κάποιος χρήστης – υπάλληλος να δει τις παραγγελίες όλων των πελατών και όταν θέλει να δει τους ίδιους τους πελάτες.

ΥΠΟΚΕΦΑΛΑΙΟ 6.2

Η ιστοσελίδα είναι σχεδιασμένη για να χειρίζεται από δυο διαφορετικές ομάδες χρηστών. Η πρώτη ομάδα είναι οι πελάτες της επιχείρησης και διάφοροι άλλοι ενδιαφερόμενοι οι οποίοι θέλουν να μάθουν σχετικά με την επιχείρηση και ίσως να γίνουν και οι ίδιοι πελάτες. Η δεύτερη ομάδα είναι οι εργαζόμενοι της επιχείρησης. Ένας τέτοιος λογαριασμός θεωρείται administrator και η δημιουργία του είναι δυνατή μόνο από άλλο εργαζόμενο. Περισσότερες πληροφορίες για αυτή την ομάδα χρήστη θα δούμε παρακάτω.

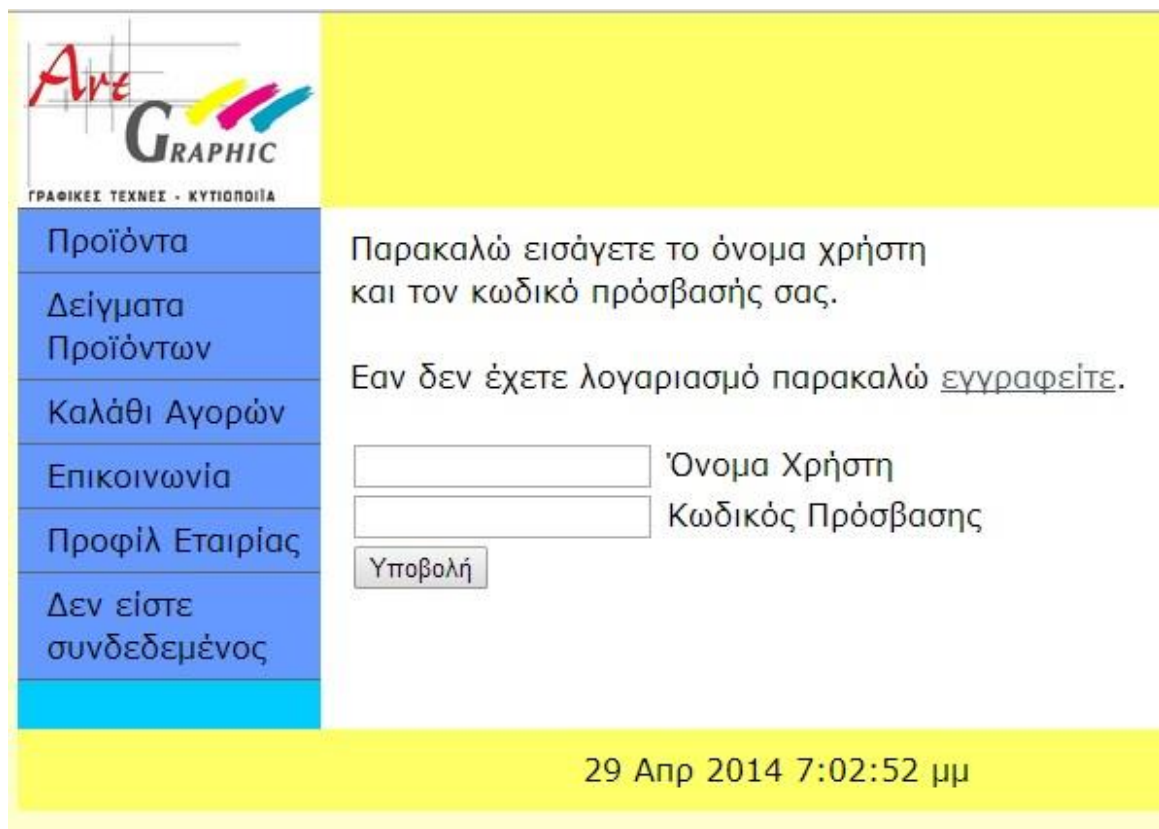
Η εισαγωγή στην ιστοσελίδα «εικόνα 23» είναι αδύνατη χωρίς τη σύνδεση με υπάρχον λογαριασμό ή τη δημιουργία νέου. Αυτό γίνεται για λόγους ασφάλειας και για να υπάρχει έλεγχος όσον αφορά το ποιοι μπορούν να έχουν πρόσβαση στην ιστοσελίδα είτε αυτοί είναι αντίπαλες εταιρίες είτε είναι συνάδελφοι ή απλοί πελάτες. Στην παρακάτω εικόνα βλέπουμε τι αντικρίζει κάποιος που μόλις έχει εισέλθει στην ιστοσελίδα.



The screenshot shows the website for 'Are GRAPHIC' with the tagline 'ΓΡΑΦΙΚΕΣ ΤΕΧΝΕΣ - ΚΥΤΙΟΠΟΙΙΑ'. On the left is a blue navigation menu with the following items: Προϊόντα, Δείγματα Προϊόντων, Καλάθι Αγορών, Επικοινωνία, Προφίλ Εταιρίας, and Δεν είστε συνδεδεμένος. The main content area has a yellow background and contains a login message: 'Η σελίδα μας απαιτεί την σύνδεση με όνομα χρήστη (username) και κωδικό πρόσβασης (password). Παρακαλώ [συνδεθείτε](#) με υπάρχον λογαριασμό ή [εγγραφείτε](#) για τη δημιουργία νέου χρήστη. Εαν είστε υπάλληλος της εταιρίας ακολουθήστε τον [σύνδεσμο](#) για να συνδεθείτε.' At the bottom right, a timestamp reads '26 Apr 2014 7:39:26 μμ'.

Εικόνα 23

Ο χρήστης είναι υποχρεωμένος να επιλέξει μεταξύ των συνδέσμων «συνδεθείτε» και «εγγραφείτε». Η σύνδεση με υπάρχον λογαριασμό φέρνει τον χρήστη στην «εικόνα 24». Εκεί καλείται να πληκτρολογήσει το όνομα χρήστη του και τον κωδικό πρόσβασης. Η πληκτρολόγηση των στοιχείων αυτών πρέπει να είναι ακριβής. Κανόνες δεν υπάρχουν αλλά τα στοιχεία πρέπει να είναι ίδια με εκείνα κατά την εγγραφή του χρήστη στην ιστοσελίδα (για οδηγίες εγγραφής νέου χρήστη βλέπε παρακάτω). Η μη σωστή πληκτρολόγηση των στοιχείων θα οδηγήσει τον χρήστη σε μια σελίδα σφάλματος, η οποία θα ενημερώσει τον χρήστη ότι υπήρξε κάποιο σφάλμα και περιέχει συνδέσμους για τις σελίδες: σύνδεση με υπάρχον λογαριασμό, εγγραφή νέου χρήστη και σύνδεση υπαλλήλου.



The screenshot shows a website interface with a yellow header and a blue sidebar. The sidebar contains a menu with the following items: Προϊόντα, Δείγματα Προϊόντων, Καλάθι Αγορών, Επικοινωνία, Προφίλ Εταιρίας, and Δεν είστε συνδεδεμένος. The main content area has a yellow background and contains the following text: 'Παρακαλώ εισάγετε το όνομα χρήστη και τον κωδικό πρόσβασής σας.' Below this, it says 'Εαν δεν έχετε λογαριασμό παρακαλώ εγγραφείτε.' There are two input fields: the first is labeled 'Όνομα Χρήστη' and the second is labeled 'Κωδικός Πρόσβασης'. Below the second field is a 'Υποβολή' button. At the bottom of the page, there is a yellow footer with the text '29 Apr 2014 7:02:52 μμ'.

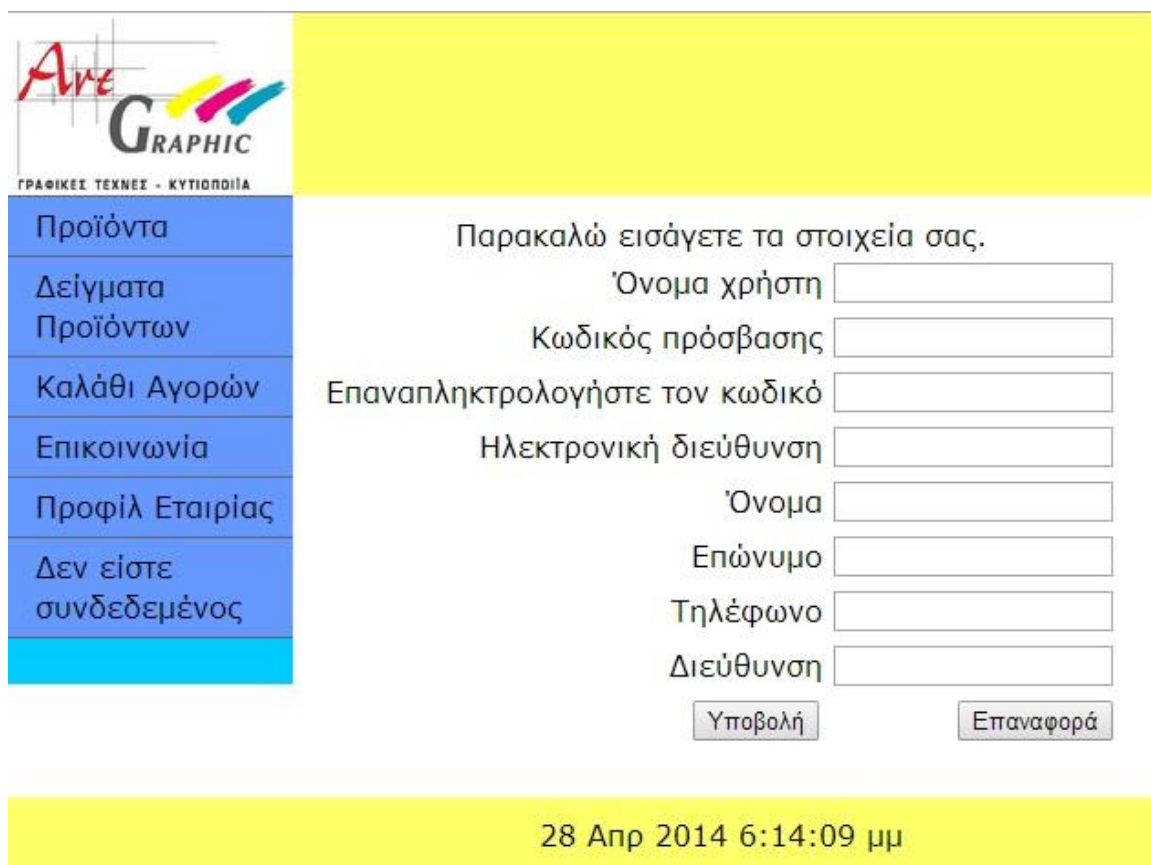
Εικόνα 24

Εάν ο χρήστης δεν έχει λογαριασμό μπορεί να δημιουργήσει νέο. Απλά πρέπει να πατήσει στο σύνδεσμο «εγγραφείτε» της παραπάνω εικόνας. Αυτό θα τον μεταφέρει στη σελίδα που βλέπουμε στην «εικόνα 25». Εκεί ο χρήστης πρέπει να πληκτρολογήσει κάθε ένα από τα στοιχεία που ονομάζονται αριστερά από τα κουτιά κειμένου. Οι χαρακτήρες με τους οποίους θα γίνει η πληκτρολόγηση (ελληνικοί ή λατινικοί) έχουν αφεθεί στην ελευθερία του χρήστη αλλά για το όνομα χρήστη και τον κωδικό πρόσβασης ο χρήστης είναι υποχρεωμένος να θυμάται τι θα γράψει λατινικούς ή ελληνικούς, κεφαλαία ή πεζά, τονισμένα ή όχι, ώστε να μπορεί να γίνει η ταυτοποίηση από το σύστημα.

Κατά την υποβολή της φόρμας ή και κατά την πληκτρολόγηση πραγματοποιούνται κάποιοι έλεγχοι από τον περιηγητή. Ελέγχονται καταρχήν εάν τα δύο πεδία του κωδικού πρόσβασης είναι ίδια. Ο έλεγχος αυτός γίνεται στο δεύτερο πεδίο και εκτελείται κατά το συμβάν onblur του πεδίου αυτού. Ο έλεγχος γίνεται μέσω μιας σύντομης μεθόδου Javascript η οποία συγκρίνει τα δύο πεδία και τα διαγράφει εάν δεν είναι ίδια. Το πεδίο Ηλεκτρονική Διεύθυνση (e-mail) ελέγχει ότι τα περιεχόμενά του έχουν σωστή μορφή Ηλεκτρονικής Διεύθυνσης δηλαδή λατινικούς χαρακτήρες, να περιέχουν τον χαρακτήρα '@' και επιπλέον λατινικούς χαρακτήρες έπειτα από αυτό. Τέλος ελέγχεται ότι κανένα πεδίο της φόρμας δεν είναι κενό.

Κατά την υποβολή της φόρμας τα στοιχεία λαμβάνονται από ένα άλλο αρχείο JSP. Αυτό το αρχείο δημιουργεί τα κατάλληλα αντικείμενα για τη σύνδεση με τη βάση SQL και αποθηκεύει τις παραμέτρους που έλαβε σε κατάλληλες μεταβλητές τύπου String και Long (για το τηλέφωνο). Έπειτα χρησιμοποιεί τις μεταβλητές αυτές για να δημιουργήσει το SQL Query που θα πραγματοποιηθεί ώστε να εγγραφούν οι πληροφορίες εγγραφής στη βάση δεδομένων.

Αφού όλα πραγματοποιηθούν σωστά και χωρίς προβλήματα επιστρέφει τον χρήστη στη σελίδα σύνδεσης με τη βάση και ένα επιπλέον μήνυμα το οποίο ενημερώνει το χρήστη ότι η εγγραφή έχει γίνει σωστά και ο χρήστης μπορεί πλέον να συνδεθεί στην σελίδα.



The screenshot shows a web application interface. On the left is a vertical navigation menu with blue buttons: 'Προϊόντα', 'Δείγματα Προϊόντων', 'Καλάθι Αγορών', 'Επικοινωνία', 'Προφίλ Εταιρίας', and 'Δεν είστε συνδεδεμένος'. The top right has a yellow header. The main content area has a yellow background and contains the text 'Παρακαλώ εισάγετε τα στοιχεία σας.' followed by a login form with fields for 'Όνομα χρήστη', 'Κωδικός πρόσβασης', 'Επαναπληκτρολογήστε τον κωδικό', and 'Ηλεκτρονική διεύθυνση'. Below these are fields for 'Όνομα', 'Επώνυμο', 'Τηλέφωνο', and 'Διεύθυνση'. At the bottom of the form are two buttons: 'Υποβολή' and 'Επαναφορά'. A yellow footer bar at the bottom of the screenshot displays the timestamp '28 Απρ 2014 6:14:09 μμ'.

Εικόνα 25

Έπειτα ο χρήστης είναι ελεύθερος να συνδεθεί στην ιστοσελίδα, καλωσορίζεται και του παρέχεται μια συνοπτική λίστα όλων των λειτουργιών της ιστοσελίδας, όπως αυτό φαίνεται στην «εικόνα 26». Τώρα ο χρήστης μπορεί να περιηγηθεί στην σελίδα, ακολουθώντας το μενού στα αριστερά. Εκεί διάφοροι σύνδεσμοι θα τον οδηγήσουν σε διάφορες σελίδες δίνοντάς του δυνατότητες όπως αποστολή νέας παραγγελίας, αντιπροσωπευτικά δείγματα προϊόντων κα. Κάθε ένας τέτοιος σύνδεσμος μεταφέρει το χρήστη σε μια άλλη σελίδα δηλαδή σε ένα άλλο αρχείο της σελίδας. Ο χρήστης μπορεί να επανέλθει στην σελίδα αυτή πατώντας με το ποντίκι του πάνω στο λογότυπο της εταιρίας.



Εικόνα 26

Ο σύνδεσμος «Προϊόντα» θα στείλει το χρήστη σε μια σελίδα η οποία θα περιέχει μια λίστα με τα προϊόντα της επιχείρησης. Στο τέλος της σελίδας αυτής υπάρχει ένας σύνδεσμος ο οποίος θα παραπέμψει το χρήστη στη σελίδα παραγγελίας μέσω της οποίας ο χρήστης μπορεί να παραγγείλει κάτι.

Η σελίδα παραγγελίας, όπως βλέπουμε στην «εικόνα 27» παρακάτω, περιέχει μια φόρμα με διάφορες επιλογές. Ο χρήστης μπορεί να επιλέξει είδος παραγγελίας όπως αφίσες, διαφημιστικά έντυπα, επαγγελματικές κάρτες, φακέλους κ.α. να πληκτρολογήσει πόσα τεμάχια θέλει για το προϊόν αυτό σε ένα text input της φόρμας και να επιλέγει από μια λίστα διαθέσιμων ειδών χαρτιού. Επόμενη επιλογή του χρήστη είναι η επιλογή διάστασης του προϊόντος.

Στο σημείο αυτό να τονίσουμε ότι ο χρήστης έχει την ελευθερία να πληκτρολογήσει τα τεμάχια και την διάσταση όπως ο ίδιος τα καταλαβαίνει αλλά και για ένα άλλο λόγο. Μερικά προϊόντα χρεώνονται ανά κιλό ενώ άλλα ανά τεμάχιο. Δίνεται η ελευθερία στο χρήστη να το πληκτρολογήσει όπως είναι σωστό για κάθε προϊόν ενώ κάποιο σφάλμα του στην πληκτρολόγηση δεν θα δημιουργήσει πρόβλημα στην αποστολή της παραγγελίας. Τα περιεχόμενα των πεδίων τεμάχια και διάσταση αποθηκεύονται σε μια στήλη σε έναν πίνακα στη

βάση και όταν κάποιος υπάλληλος συνδεθεί στην ιστοσελίδα για να ελέγξει τις παραγγελίες θα το δει και θα το ερμηνεύσει ανάλογα και εάν υπάρχει κάποιο πρόβλημα μπορεί να αναζητήσει το όνομα χρήστη του πελάτη και να επικοινωνήσει μαζί του για διευκρινήσεις.

Η επόμενη επιλογή του χρήστη είναι το δίπλωμα της παραγγελίας. Κάποιες παραγγελίες δεν μπορούν να διπλωθούν οπότε μπορεί ο χρήστης να επιλέξει και «Όχι» ώστε το προϊόν να μη διπλωθεί. Η βιβλιοδεσία της παραγγελίας διατίθεται για κάποια προϊόντα οπότε και στο σημείο αυτό ο χρήστης μπορεί να επιλέξει ένα τύπο βιβλιοδεσίας από τη λίστα ή και να επιλέξει να μην υπάρχει βιβλιοδεσία.

Τέλος υπάρχει και η δυνατότητα στο χρήστη να πληκτρολογήσει ένα σύντομο μήνυμα προς την επιχείρηση με κάποια πληροφορία που δεν του δόθηκε η ευκαιρία να επιλέξει μέχρι στιγμής ή κάτι άλλο που θέλει να πει στους υπαλλήλους της εταιρίας.

Για την αποστολή της παραγγελίας ο χρήστης μπορεί να επιλέξει «Υποβολή» ή «Επαναφορά» για τον καθαρισμό της φόρμας. Μπορεί, τέλος, να εκτυπώσει τη σελίδα με την παραγγελία του. Κατά την υποβολή της παραγγελίας τα στοιχεία στέλνονται σε ένα αρχείο μέσω της μεθόδου POST ως παράμετροι. Εκεί αποθηκεύονται τα στοιχεία αυτά στη βάση με ένα SQL insert με παρόμοια μέθοδο όπως και κατά την δημιουργία νέου χρήστη. Ταυτόχρονα στη βάση εγγράφονται και το όνομα χρήστη και ο κωδικός πρόσβασης, του χρήστη που είναι τώρα συνδεδεμένος, ώστε να γνωρίζει ο υπάλληλος (admin) που θα ελέγξει τις παραγγελίες ποιος έχει καταχωρήσει αυτήν την παραγγελία.

Έπειτα ο έλεγχος της ιστοσελίδας στέλνεται σε μια άλλη σελίδα, την οποία ο χρήστης δεν βλέπει. Εκεί χρησιμοποιούνται διάφορα αντικείμενα Java για την σύνδεση με τη βάση δεδομένων και χρησιμοποιώντας ένα SQL Query λαμβάνεται το e-mail του χρήστη που είναι συνδεδεμένος αυτή τη στιγμή. Έπειτα το πρόγραμμα στέλνει ένα μήνυμα ηλεκτρονικής αλληλογραφίας ενημερώνοντας το χρήστη ότι η παραγγελία του έχει ληφθεί σωστά από την ιστοσελίδα, δηλαδή ότι έχει αποθηκευτεί σαν εγγραφή στη βάση. Έπειτα το πρόγραμμα επιστρέφει το χρήστη στην σελίδα νέας παραγγελίας σε περίπτωση που ο χρήστης επιθυμεί να εισάγει νέα παραγγελία.

Εικόνα 27

Πατώντας στο σύνδεσμο «Καλάθι Αγορών» στο μενού αριστερά, ο χρήστης μπορεί να μεταβεί στην κατάλληλη σελίδα και δει τη λίστα με τις παραγγελίες του όπως φαίνεται στην «εικόνα 28», και μόνο αυτές οι οποίες δεν έχουν ολοκληρωθεί ακόμα. Τα κουμπιά (buttons) «Επόμενη Παραγγελία / Προηγούμενη Παραγγελία» βοηθούν το χρήστη να μεταβεί σε προηγούμενη ή επόμενη παραγγελία αντίστοιχα. Οι παραγγελίες που φαίνονται εδώ είναι αυτές που υπάρχουν ακόμα και δεν έχουν διαγραφεί από έναν υπάλληλο (λογαριασμό admin) της ιστοσελίδας.

Εδώ πρέπει να τονίσουμε ότι τα κουμπιά αυτά δεν εμφανίζονται για την πρώτη και την τελευταία παραγγελία (το κουμπί Προηγούμενη Παραγγελία στην πρώτη και το κουμπί Επόμενη Παραγγελία στην τελευταία) ώστε να γνωρίζει ο χρήστης τότε έχει δει όλες τις παραγγελίες του, αλλά και για να μη δημιουργηθεί πρόβλημα στον κώδικα (ArrayIndexOutOfBoundsException).

Οι παραγγελίες αυτές εισάγονται στην ιστοσελίδα σε πραγματικό χρόνο μέσω χρήσης εντολών SQL προς και από τη βάση δεδομένων του συστήματος και δεν είναι προγραμματισμένες στον κώδικα. Κάθε παραγγελία είναι αποθηκευμένη σαν μια εγγραφή σε ένα πίνακα με το όνομα orders στη βάση SQL

και για να προβληθούν ελέγχονται πρώτα τα στοιχεία (username – password) του χρήστη που είναι συνδεδεμένος με τα αντίστοιχα στοιχεία που αποθηκεύτηκαν μαζί με την παραγγελία στη βάση ώστε να φανούν οι σωστές παραγγελίες στους σωστούς χρήστες. Τα στοιχεία του χρήστη είναι αποθηκευμένα σαν μεταβλητές στο αντικείμενο session και καταστρέφονται κατά την αποσύνδεση του χρήστη από την ιστοσελίδα.

Αlive GRAPHIC
ΓΡΑΦΙΚΕΣ ΤΕΧΝΕΣ - ΚΥΤΙΟΠΟΙΙΑ

- Προϊόντα
- Δείγματα Προϊόντων
- Καλάθι Αγορών
- Επικοινωνία
- Προφίλ Εταιρίας
- Αποσύνδεση ως **babis**

Παραγγελίες που εκκρεμούν.

Αριθμός Παραγγελίας:	27
Όνομα Χρήστη που έκανε την παραγγελία:	babis
Τύπος παραγγελίας	ΑΦΙΣΕΣ
Τεμάχια:	1000
Είδος Χαρτιού:	Gloss
Διαστάσεις:	30 x 40
Δίπλωμα:	2πτυχο
Βιβλιοδεσία:	Όχι
Πληροφορίες Παραγγελίας:	για κάποιο πρόβλημα τηλέφωνο 2310111111

28 Apr 2014 4:39:05 μμ

Εικόνα 28

Στο μενού αριστερά στην «εικόνα 29» παρακάτω βλέπουμε σύνδεσμο με την ονομασία Επικοινωνία. Αυτός ο σύνδεσμος οδηγεί σε μια σελίδα με πληροφορίες της εταιρίας όπως όνομα επιχείρησης, τοποθεσία, διεύθυνση, ταχυδρομικό κώδικα, τηλέφωνο κ.α. Επίσης έχει ένα σύνδεσμο με δεδομένα χάρτη Google, δείχνοντας την επιχείρηση στον χάρτη της εταιρίας Google (<https://maps.google.com>). Εκεί μπορεί ο χρήστης να δει τυχόν επιπλέον πληροφορίες για την επιχείρηση και οδηγίες για την μετάβαση στην τοποθεσία αυτή.

Μια επιπλέον δυνατότητα που δίνει η ιστοσελίδα είναι η αποστολή email στην ηλεκτρονική διεύθυνση της εταιρίας. Για να πραγματοποιηθεί η λειτουργία αυτή απαιτείται η ενεργοποίηση του λογισμικού Microsoft Outlook.

Αrte GRAPHIC
ΓΡΑΦΙΚΕΣ ΤΕΧΝΕΣ - ΚΥΤΙΟΠΟΙΙΑ

Προϊόντα
Δείγματα Προϊόντων
Καλάθι Αγορών
Επικοινωνία
Προφίλ Εταιρίας
Αποσύνδεση ως babis

Γ. ΣΕΡΙΣΤΑΤΙΔΗΣ - Γ. ΛΑΜΠΡΟΠΟΥΛΟΣ Ο.Ε. | ART GRAPHIC
5ο χιλ. Παλαιάς οδού Νεάπολης - Κοζάνη και Κ. Καραμανλή 3 Νεάπολη Κοζάνης
Τ.Κ. 50001
Τηλέφωνο: 2468023218, 2468023142
Φαξ: 2468023559
Ηλεκτρονική Διεύθυνση: serilame@otenet.gr

ΣΕΡΙΣΤΑΤΙΔΗΣ, Γ., - Γ. ΛΑΜΠ...
Οδός Νεάπολης - Κοζάνης (5 χλμ.), Νεάπολη, 50001
Αποθή...
Γράψτε πρώτοι μια κριτική
Προβολή στους Χάρτες Google

Μυλιώνα Α. Σοφία
ΣΕΡΙΣΤΑΤΙΔΗΣ, Γ., - Γ. ΛΑΜΠΡΟΠΟΥΛΟΣ Ο.Ε.
ΝΕΙΑΔΗΣ ΑΘ - ΓΜΙΑΔΗΣ Ι Ο.Ε.
20
Google
©2014 Google - Δεδομένα χάρτη Όροι Χρήσης Αναφορά σφάλματος χάρτη
[Προβολή μεγαλύτερου χάρτη](#)

29 Απρ 2014 6:42:57 μμ

Εικόνα 29

Ο σύνδεσμος του μενού «Προφίλ Εταιρίας» περιέχει ένα κείμενο με σκοπό να δώσει στον αναγνώστη – επισκέπτη της ιστοσελίδας μια πρώτη γεύση για την εταιρία, τον τρόπο δημιουργίας εργασιών και ίσως να διαφημίσει την εταιρία σε κάποιον που σκέφτεται να εγγραφεί και να παραγγείλει κάτι. Είναι η μόνη σελίδα η οποία δεν απαιτεί από τον χρήστη σύνδεση με την ιστοσελίδα.

Τέλος ο σύνδεσμος «Αποσύνδεση» έχει διπλή λειτουργία. Η κάθε λειτουργία ενεργοποιείται για κάθε αντίστοιχη περίπτωση. Η πρώτη λειτουργία είναι να ενημερώσει το χρήστη ότι δεν είναι συνδεδεμένος. Κατά την επιλογή του συνδέσμου αυτού ενώ ο χρήστης δεν είναι συνδεδεμένος η ιστοσελίδα μεταφέρει το χρήστη στην σελίδα εισαγωγής για την σύνδεση ή την δημιουργία νέου λογαριασμού.

Η δεύτερη λειτουργία είναι να ενημερώσει το χρήστη ότι είναι συνδεδεμένος και με ποιο όνομα είναι συνδεδεμένος. Σε αυτήν την περίπτωση η επιλογή αυτού του συνδέσμου οδηγεί τον χρήστη στην σελίδα αποσύνδεσης, στην οποία καταστρέφονται οι όποιες μεταβλητές του αντικειμένου Session και το ίδιο το αντικείμενο Session. Κάτι τέτοιο είναι κρίσιμο και από άποψη λειτουργικότητας και από άποψη ασφάλειας καθώς οι μεταβλητές του αντικειμένου Session είναι υπεύθυνες για την σωστή λειτουργία της ιστοσελίδας και για τα ειδικά προνόμια του λογαριασμού admin, δηλαδή του υπαλλήλου.

ΥΠΟΚΕΦΑΛΑΙΟ 6.3

Στην ιστοσελίδα αυτήν όπως έχει αναφερθεί πολλές φορές ήδη υπάρχουν δύο είδη λογαριασμών χρηστών. Ο χρήστης – πελάτης στον οποίο αναφερόμασταν μέχρι τώρα και ο χρήστης – υπάλληλος. Η ομάδα χρήστη «υπάλληλος» εκτελεί χρέη administrator στην σελίδα, φροντίζει για τη σωστή λειτουργία της, ελέγχει πελάτες και παραγγελίες και έχει τη δυνατότητα διαγραφής παραγγελίας (λόγω ολοκλήρωσης ή μη δυνατής ολοκλήρωσής της), διαγραφή πελάτη (για λόγους ασφαλείας) και προσθήκη ενός επιπλέον λογαριασμού υπαλλήλου – διαχειριστή.

Η εισαγωγή ενός λογαριασμού υπαλλήλου μπορεί να γίνει μόνο από έναν άλλο ήδη υπάρχον λογαριασμό υπαλλήλου ή από την απευθείας σύνδεση με τη βάση εκτός ιστοσελίδας, κάτι που φυσικά δεν απαιτείται από τους χρήστες της. Η σύνδεση ενός διαχειριστή χρήστη γίνεται από την αρχική σελίδα την οποία βλέπει ο οποιοσδήποτε δεν έχει συνδεθεί όπως βλέπουμε στην εικόνα 23 (σελίδα 86). Από εκεί ο διαχειριστής ακολουθεί τον τρίτο σύνδεσμο της σελίδας για την εισαγωγή του στο ειδικό μέρος της ιστοσελίδας σχεδιασμένο για αυτόν. Εκεί βλέπει μια παρόμοια σελίδα με αυτήν της «εικόνας 24» (σελίδα 87). Αφού πληκτρολογήσει σωστά τα στοιχεία του η ιστοσελίδα τον οδηγεί στην σελίδα όπως τη βλέπουμε στην «εικόνα 30».

Ανε GRAPHIC
ΓΡΑΦΙΚΕΣ ΤΕΧΝΕΣ - ΚΥΤΙΟΠΟΙΙΑ

Αρχική Σελίδα Υπαλλήλων

Πελάτες

Προφίλ Εταιρίας

Προσθήκη Εργαζομένου

Αποσύνδεση ως sakis

Υπάλληλος: Θανάσης Αθανασίου

Αριθμός Παραγγελίας:	27
Όνομα Χρήστη που έκανε την παραγγελία:	babis
Τύπος παραγγελίας	ΑΦΙΣΕΣ
Τεμάχια:	1000
Είδος Χαρτιού:	Gloss
Διαστάσεις:	30 x 40
Δίπλωμα:	2πτυχο
Βιβλιοδεσία:	Όχι
Πληροφορίες Παραγγελίας:	για κάποιο πρόβλημα τηλέφωνο 2310111111

Διαγραφή Παραγγελίας

Προηγούμενη Παραγγελία

29 Apr 2014 8:34:59 μμ

Εικόνα 30

Εκεί βλέπει έναν πίνακα με τις παραγγελίες που έχουν υποβάλει οι χρήστες. Με τον ίδιο τρόπο που εμφανίζεται αυτός ο πίνακας σε κάθε χρήστη για τις δικές του παραγγελίες, με τον ίδιο τρόπο εμφανίζεται και εδώ αλλά για όλες τις παραγγελίες στον πίνακα της βάσης SQL orders (εκεί που αποθηκεύονται). Και πάλι τα κουμπιά «Προηγούμενη Παραγγελία» και «Επόμενη Παραγγελία» εμφανίζονται για την μετάβαση από παραγγελία σε παραγγελία, εκτός από την πρώτη και την τελευταία όπου εμφανίζεται μόνο το ένα από τα δυο. Η επιλογή «Διαγραφή Παραγγελίας» μεταφέρει τον έλεγχο της ιστοσελίδας σε ένα άλλο αρχείο το οποίο χρησιμοποιώντας τα κατάλληλα αντικείμενα δημιουργεί σύνδεση με τη βάση και εκτελεί ένα SQL Query τύπου delete και διαγράφει την συγκεκριμένη παραγγελία χρησιμοποιώντας τον αριθμό παραγγελίας.

Αρχική Σελίδα Υπαλλήλων

Πελάτες

Προφίλ Εταιρίας

Προσθήκη Εργαζομένου

Αποσύνδεση ως sakis

Υπάλληλος: Θανάσης Αθανασίου

Όνομα χρήστη	babis
Κωδικός πρόσβασης	12345
Ηλεκτρονική διεύθυνση	babis814@yahoo.gr
Όνομα	Μπάμπης
Επώνυμο	Σεριστατίδης
Τηλέφωνο	6981931936
Διεύθυνση	Παύλου Μελα 21

Διαγραφή Πελάτη

Επόμενος Πελάτης

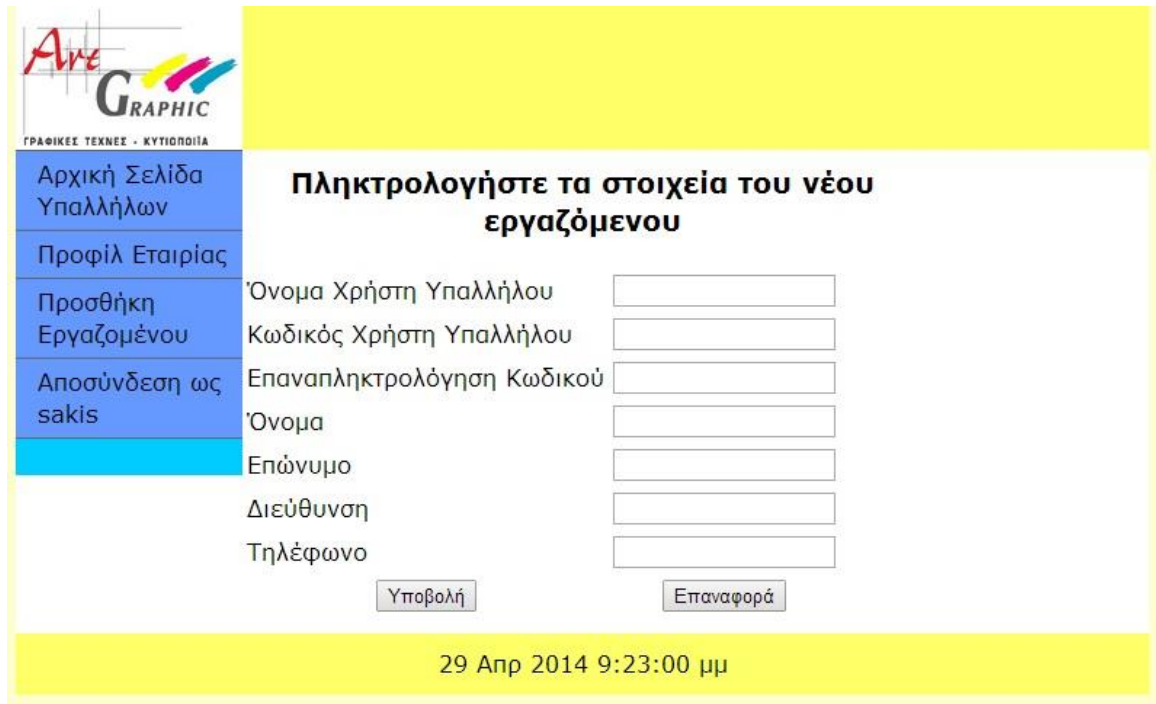
29 Απρ 2014 9:05:36 μμ

Εικόνα 31

Η επιλογή του συνδέσμου «Πελάτες» θα μεταφέρει τον χρήστη στη σελίδα που βλέπουμε στην «εικόνα 31». Εκεί μέσω σύνδεσης με τη βάση προβάλλεται ένας πίνακας με τους χρήστες – πελάτες της επιχείρησης. Με τα κουμπιά Επόμενος Πελάτης (και το κουμπί Προηγούμενος Πελάτης που θα εμφανιστεί) ο διαχειριστής μπορεί να μεταβεί την λίστα με τους χρήστες – υπάλληλους.

Αυτή η λειτουργία όπως και οι άλλες σαν αυτή είναι σχετικά περίπλοκη να υλοποιηθεί. Εξαιτίας της αφαίρεσης όλου του κώδικα Java από τον Apache Tomcat ο κώδικας που θα αλλάζει τον πίνακα (ή συγκεκριμένα θα τον αναδημιουργεί) δεν γίνεται να συνδυαστεί με ένα κουμπί HTML. Ο τρόπος που αυτό υλοποιείται είναι μέσω μιας κρυφής φόρμας στο σημείο εκείνο η οποία στέλνει μια παράμετρο σε ένα άλλο αρχείο μέσω της ιδιότητας action του κώδικα html. Το άλλο αρχείο αυξάνει μια μεταβλητή η οποία καταλήγει να είναι η στήλη id

αποθηκευμένη στη βάση μαζί με τις υπόλοιπες πληροφορίες χρήστη – πελάτη (ή παραγγελιών σε προηγούμενο κεφάλαιο). Οπότε κατά την αλλαγή αυτής της μεταβλητής αλλάζει και ο χρήστης – πελάτης ή οι πληροφορίες παραγγελίας.



The screenshot shows a web interface for adding a new user. On the left, there is a sidebar menu with the following items: Αρχική Σελίδα Υπαλλήλων, Προφίλ Εταιρίας, Προσθήκη Εργαζομένου (highlighted in blue), Αποσύνδεση ως sakis, and a blue bar. The main content area has a yellow header with the logo 'Ave GRAPHIC' and the text 'ΓΡΑΦΙΚΕΣ ΤΕΧΝΕΣ - ΧΥΤΙΟΠΟΙΙΑ'. Below the header, the title is 'Πληκτρολογήστε τα στοιχεία του νέου εργαζόμενου'. The form contains the following fields: Όνομα Χρήστη Υπαλλήλου, Κωδικός Χρήστη Υπαλλήλου, Επαναπληκτρολόγηση Κωδικού, Όνομα, Επώνυμο, Διεύθυνση, and Τηλέφωνο. At the bottom of the form are two buttons: 'Υποβολή' and 'Επαναφορά'. A yellow footer bar displays the date and time: '29 Απρ 2014 9:23:00 μμ'.

Εικόνα 32

Τέλος στην «εικόνα 32» βλέπουμε την σελίδα προσθήκης νέου λογαριασμού διαχειριστή. Ο νέος λογαριασμός που θα δημιουργηθεί θα είναι ίδιος με όλους τους άλλους λογαριασμούς διαχειριστές και δεν θα έχει μειωμένα προνόμια. Οι δυνατότητες διαγραφής παραγγελιών και διαγραφής πελάτη καθώς και επίβλεψης της σωστής λειτουργίας της σελίδας είναι ίδιες με τους άλλους λογαριασμούς διαχειριστές.

Η πληκτρολόγηση των στοιχείων ακολουθεί τους ίδιους κανόνες με την δημιουργία νέου χρήστη – πελάτη. Ελληνικοί ή λατινικοί, κεφαλαία ή πεζά γράμματα, αριθμούς και σύμβολα. Το όνομα χρήστη θα πρέπει να έχει ακριβώς την ίδια ορθογραφία κατά την επόμενη (και όλες τις επόμενες) σύνδεση του νεοδημιουργηθέν χρήστη. Ο κωδικός πρέπει να πληκτρολογηθεί δυο φορές για την αποφυγή τυχών λαθών κατά την πληκτρολόγηση ενώ η οποιαδήποτε

διαφορά των δύο κωδικών καταλήγει στην διαγραφή των. Τα υπόλοιπα στοιχεία δεν θα χρησιμοποιηθούν κατά την εισαγωγή στην ιστοσελίδα αλλά αποθηκεύονται για λόγους αρχειοθέτησης ή/και ασφάλειας.

ΥΠΟΚΕΦΑΛΑΙΟ 6.4

Η ιστοσελίδα αυτή για τη λειτουργία της απαιτεί συνεχή σύνδεση με μια βάση δεδομένων. Στη συγκεκριμένη περίπτωση έχει επιλεγεί η βάση MySQL. Η ιστοσελίδα αυτή περιέχει τρεις πίνακες που χρειάζεται για τη λειτουργία της. Ο πρώτος πίνακας έχει τα στοιχεία των χρηστών – πελατών. Έχει τις εξής 8 στήλες: id, username, password, e_mail, first_name, last_name, telephone, address. Όλες οι στήλες του πίνακα έχουν την ιδιότητα Not null οπότε η βάση δε θα δεχθεί κενή τιμή για καμία από τις στήλες κατά την εγγραφή νέου χρήστη. Αυτό ελέγχεται και πριν την υποβολή της φόρμας HTML κατά την εισαγωγή νέου χρήστη.

Η πρώτη στήλη περιέχει ακέραιους αριθμούς και είναι τύπου auto increment (αυτόματη αύξηση) καθώς και το κύριο κλειδί του πίνακα σε συνδυασμό με την στήλη username. Για κάθε καινούρια εγγραφή στον πίνακα, δηλαδή για κάθε νέο χρήστη – πελάτη που δημιουργεί νέο λογαριασμό, η στήλη αυτή αυξάνει από μόνη της, χωρίς να ορίζεται η τιμή της από τον κώδικα της ιστοσελίδας. Η νέα τιμή είναι κατά ένα μεγαλύτερη από τον μεγαλύτερο αριθμό όλων των γραμμών. Ακόμα και άμα διαγραφεί μια εγγραφή στον πίνακα ο αριθμός της στήλης id μόνο αυξάνεται. Εάν δηλαδή ο τρίτος χρήστης – πελάτης διαγραφεί ο επόμενος θα έχει τον αριθμό 4 και ο χρήστης με τον αριθμό 3 θα λείπει.

Η δεύτερη στήλη με το όνομα username κρατάει το όνομα χρήστη με το οποίο κάποιος μπορεί να συνδεθεί στην ιστοσελίδα. Η τιμή της στήλης αυτής εισάγεται από τον χρήστη κατά την εγγραφή του στην ιστοσελίδα. Μπορεί να κρατήσει μέχρι 50 χαρακτήρες (varchar(50)) και μπορεί να αποθηκεύσει επιτυχώς ελληνικούς και λατινικούς χαρακτήρες. Αυτήν και η επόμενη στήλη είναι αυτές που χρησιμοποιούνται κατά την σύνδεση ενός χρήστη με χρήση ενός SQL select. Επίσης οι δυο αυτές στήλες είναι αυτές οι οποίες αποθηκεύονται μαζί με τις διάφορες πληροφορίες παραγγελίας.

Η στήλη με το όνομα password είναι παρόμοια με την προηγούμενη στη δήλωσή της. Varchar(50), not null, αλλά όχι κύριο κλειδί, είναι αυτήν που κρατάει τον κωδικό του χρήστη. Κατά την πληκτρολόγησή του κωδικού του, ο χρήστης

καλείται να τον πληκτρολογήσει δυο φορές και έπειτα από έναν έλεγχο σύγκρισης των δύο κωδικών, ο δεύτερος είναι αυτός που εισάγεται στη βάση. Παρόλο που αποθηκεύεται σαν απλό κείμενο στη βάση, δεν φαίνεται πουθενά σαν κείμενο στην ιστοσελίδα ούτε πουθενά στον κώδικα. Αποθηκεύεται μεν σε μια μεταβλητή του αντικειμένου session ώστε να υπάρχει έλεγχος και να εγγράφεται μαζί με τις πληροφορίες παραγγελίας.

Επόμενη στήλη στον πίνακα είναι η στήλη e_mail. Ίδιο μέγιστο μήκος με τις προηγούμενες (50) είναι η στήλη που αποθηκεύει την ηλεκτρονική διεύθυνση του χρήστη. Σε αυτήν την ηλεκτρονική διεύθυνση στέλνεται μήνυμα κατά την εισαγωγή νέας παραγγελίας μέσω σύνδεσης του κώδικα με την βάση SQL και εκτέλεσης ενός SQL Query. Οπότε πρέπει ο χρήστης κατά την πληκτρολόγησή του να εισάγει μια υπάρχουσα διεύθυνση εάν θέλει να λαμβάνει ενημερωτικά μηνύματα. Ο έλεγχος που γίνεται από την ιστοσελίδα όσων αφορά το εάν είναι πράγματι μια ηλεκτρονική διεύθυνση και όχι τυχαία γράμματα πραγματοποιείται αυτόματα από τον περιηγητή. Ελέγχει ο περιηγητής εάν τα γράμματα είναι λατινικοί χαρακτήρες επειδή δεν δέχεται ελληνικούς, εάν υπάρχει το σύμβολο '@' και εάν υπάρχουν επιπλέον λατινικοί χαρακτήρες μετά το σύμβολο. Το ελάχιστον που δέχεται ο περιηγητής μοιάζει με 'a@a' και φυσικά δεν είναι μια πραγματική διεύθυνση.

Οι στήλες first_name και last_name αποθηκεύουν το όνομα και το επώνυμο του χρήστη. Αποθηκεύουν μέγιστο αριθμό 50 χαρακτήρων η καθεμία, δεν μπορούν να είναι κενές και μπορούν να αποθηκεύσουν λατινικούς και ελληνικούς χαρακτήρες. Αυτά τα στοιχεία αποθηκεύονται είναι για λόγους ασφάλειας και ίσως για λογιστικούς λόγους. Εάν για παράδειγμα ο χρήστης είναι μια επιχείρηση το όνομα και επώνυμο θα μπορούσαν να αποθηκεύουν το ονοματεπώνυμο του χρήστη και την ονομασία της επιχείρησης.

Επόμενη στήλη στον πίνακα είναι η στήλη τηλέφωνο. Τα δεδομένα αυτής της στήλης είναι bigint ή long στον κώδικα Java. Ο λόγος που δεν είναι varchar (χαρακτήρες) όπως οι προηγούμενες στήλες είναι για να μη έχει τη δυνατότητα ο χρήστης να δώσει λάθος στοιχεία. Τα δεδομένα στη στήλη θεωρούνται μια διαφορετική μορφή ακεραίου η οποία αποθηκεύει μεγαλύτερους αριθμούς. Ένα τηλέφωνο αποτελείται από δέκα ψηφία και ένας απλός ακέραιος δεν μπορεί να

αποθηκεύσει τόσο μεγάλο αριθμό, γιατί στην ουσία πρόκειται περί αριθμού. Ο μέγιστος αριθμός που μπορεί να αποθηκευτεί σε ένα απλό ακέραιο είναι δέκα ψηφίων λίγο μεγαλύτερος από 4 δισεκατομμύρια. Ο αριθμός που χρησιμοποιήθηκε μπορεί να αποθηκεύσει μέγιστο αριθμό 19 ψηφίων, υπέρ αρκετά για την αποθήκευση ενός αριθμού τηλεφώνου. Ο λόγος αποθήκευσης αυτής της πληροφορίας είναι για να μπορεί ο διαχειριστής της ιστοσελίδας να τηλεφωνεί και να ενημερώνει για την εξέλιξη των παραγγελιών που έχει αποστείλει ο χρήστης ή για την διευκρίνιση τυχών αδιευκρίνιστων λεπτομερειών της παραγγελίας.

Τέλος η τελευταία στήλη του πίνακα είναι η διεύθυνση. Για τη συγκεκριμένη πληροφορία δεν δίνεται συγκεκριμένη μορφή εάν, για παράδειγμα, θα περιέχει την διεύθυνση και τον αριθμό ή μόνο την διεύθυνση ή αν θα είναι η πόλη στην οποία ζει ο πελάτης. Έχει μέγιστο όριο 50 χαρακτήρων ελληνικούς και λατινικούς και κενά, όπως και οι προηγούμενες στήλες, και δεν υπάρχει η δυνατότητα να είναι κενή.

Ένας άλλος πίνακας στη βάση είναι ο πίνακας orders. Ο πίνακας αυτός περιέχει πληροφορίες των παραγγελιών. Η τιμές των στηλών της κάθε εγγραφής δίνονται από τον χρήστη (τεμάχια, διάσταση και άλλες πληροφορίες) ενώ όλα τα υπόλοιπα παίρνουν σαν τιμή την τιμή της ιδιότητας value του κώδικα HTML η οποία έχει δοθεί σε κάθε μια επιλογή της κάθε μια λίστας από επιλογές τις οποίες μπορεί να επιλέξει ο χρήστης.

Η διαδικασία για να συμβεί κάτι τέτοιο είναι η εξής: Πρώτα ο χρήστης επιλέγει μια επιλογή από τη λίστα. Η πρώτη εγγραφή στη λίστα είναι η προεπιλεγμένη, που σημαίνει ακόμα και αν ο χρήστης δεν επιλέξει τίποτα αυτή η επιλογή θα περάσει στη βάση. Κάθε λίστα είναι ένα αντικείμενο select της φόρμας του HTML κώδικα και κάθε μια επιλογή είναι ένα option μέσα στο προηγούμενο αντικείμενο. Κάθε ένα από αυτά περιέχει μια λέξη στην ιδιότητα value, για παράδειγμα ΑΦΙΣΕΣ.

Αφού ο χρήστης πατήσει το κουμπί υποβολή τα δεδομένα αυτά στέλνονται σε μια άλλη σελίδα. Εκεί υπάρχει μόνο κώδικας Java, οπότε ο χρήστης που βλέπει τη σελίδα δεν θα δει ποτέ τη σελίδα αυτή. Τα δεδομένα των ιδιοτήτων value έχουν σταλεί σαν παράμετροι στη σελίδα αυτή και με τη χρήση της μεθόδου

getParameter του αντικειμένου request αποθηκεύονται σε μεταβλητές τύπου String. Έπειτα με χρήση αυτών των μεταβλητών συναρμολογείται μια άλλη μεταβλητή τύπου String που στην ουσία είναι το SQL Query. Δημιουργείται η σύνδεση με τη βάση δεδομένων με χρήση των αντικειμένων Connection, Statement και σε σελίδες με Query τύπου select, αλλά όχι σε αυτήν, ένα επιπλέον αντικείμενο Resultset το οποίο περιέχει τα επιστρεφόμενα αποτελέσματα του Query.

Έπειτα από την επιτυχή εκτέλεση του Query, με χρήση της μεθόδου sendRedirect του αντικειμένου response, η σελίδα στέλνει τον χρήστη στην σελίδα email_send στην οποία στέλνεται ένα μήνυμα ηλεκτρονικής αλληλογραφίας στη διεύθυνση ηλεκτρονικής αλληλογραφίας του χρήστη ενημερώνοντάς τον για την επιτυχή αποστολή της παραγγελίας.

Ο πίνακας orders έχει στήλες με ονόματα: order_id, username, password, order_type, order_quantity, order_paper_type, order_dimension, order_folding, order_bookbind και order_information.

Η στήλη order_id αποθηκεύει έναν αύξων ακέραιο αριθμό (int auto increment) ο οποίος χαρακτηρίζει την παραγγελία και την ξεχωρίζει από τις άλλες. Ακόμα και άμα ο ίδιος πελάτης στείλει την παραγγελία δυο φορές, ο χρήστης – υπάλληλος θα μπορεί να τις ξεχωρίσει και να αποφασίσει εάν όντως οι παραγγελίες είναι δυο ή μια που κατά λάθος έχει αποσταλεί δυο φορές. Η στήλη αυτή αποτελεί το κύριο κλειδί του πίνακα και η τιμή της δεν εξαρτάται από τον κώδικα της ιστοσελίδας παρά μόνο με τη βάση. Έτσι δεν είναι δυνατόν δυο παραγγελίες να εισαχθούν με τον ίδιο αριθμό.

Οι στήλες username και password αποθηκεύουν το όνομα χρήστη και τον κωδικό πρόσβασης που είναι αποθηκευμένα στο αντικείμενο session. Έτσι ο χρήστης που έστειλε την παραγγελία είναι ο ίδιος χρήστης που είχε συνδεθεί στην ιστοσελίδα πριν την αποστολή της παραγγελίας. Έτσι ο χρήστης – υπάλληλος γνωρίζει ποιος έστειλε την παραγγελία οπότε γνωρίζει και πώς να επικοινωνήσει μαζί του ή που να αποστείλει τα προϊόντα. Η δυνατότητες αποθήκευσης δεδομένων στις δυο αυτές στήλες είναι ίδιες με τις αντίστοιχες ομώνυμες στήλες στον πίνακα login.

Η επόμενη στήλη έχει το όνομα `order_type`. Η στήλη αυτή όπως περιγράψαμε παραπάνω παίρνει την τιμή της αυτόματα από μια λίστα ανάλογα με το ποια επιλογή της λίστας έχει επιλέξει ο χρήστης. Η τιμές που παίρνει αποτελούνται από ελληνικούς κεφαλαίους χαρακτήρες ο μακρύτερος από τους οποίους αποτελείται από 42 χαρακτήρες.

Επόμενη στήλη έχει το όνομα `order_quantity`. Εκεί αποθηκεύεται η επιλογή του χρήστη για το πόσο από την παραγγελία θέλει. Τα περισσότερα προϊόντα πωλούνται ανά τεμάχιο αλλά υπάρχουν και προϊόντα που πωλούνται με το κιλό ή με το γραμμάριο. Έτσι αυτή η στήλη έχει τη δυνατότητα να αποθηκεύσει και γράμματα και όχι μόνο αριθμούς ώστε να μπορεί ο χρήστης να παραγγείλει '1000' βιβλία, '150 κιλά' κουτιά ή 'μισό τόνο' κουτιά.

Στη στήλη με το όνομα `order_paper_type` είναι αποθηκευμένο το χαρτί στο οποίο θα εκτυπωθεί η παραγγελία. Εδώ όπως και με το είδος της παραγγελίας οι τιμές είναι συγκεκριμένες. Το μέγιστο μήκος τιμής που θα αποθηκευτεί εδώ είναι 18 χαρακτήρες. Velvet, Illustration, Mat, Gloss ή Dalí είναι κάποιοι από τους 8 τύπους χαρτιού που θα αποθηκευτούν.

Στη στήλη `order_dimention` τοποθετείται η διάσταση του προϊόντος. Εδώ ο χρήστης είναι ελεύθερος να τοποθετήσει ότι θέλει όπως A4, 30 επί 40, 15 χ 40, 70*50 ή 20 εκ – 10 εκ. Ότι και να γράψει ο χρήστης εδώ τοποθετείται αυτούσιο στη βάση, ελέγχεται από τον χρήστη – υπάλληλο και εάν δεν είναι κάποια μορφή διάστασης, τότε ο διαχειριστής μπορεί να επιλέξει να διαγράψει την παραγγελία ή να επικοινωνήσει τηλεφωνικώς με τον χρήστη – πελάτη για διευκρίνιση. Εδώ αποθηκεύονται ελληνικοί ή λατινικοί χαρακτήρες, αριθμοί και σύμβολα μέγιστου μήκους 50 χαρακτήρων.

Η στήλη `order_folding` περιέχει το δίπλωμα της παραγγελίας. Πολλά προϊόντα (π.χ. διαφημιστικά φυλλάδια) είναι απαραίτητο να διπλωθούν ώστε να πάρουν τη μορφή τετραδίου. Έτσι η μια σελίδα μετατρέπεται σε δύο, οι δύο σε τέσσερις κοκ. Εδώ ο χρήστης επιλέγει από μια λίστα επιλογών οπότε και εδώ οι τιμές που μπορεί να πάρει είναι συγκεκριμένες. Μια επιπλέον επιλογή «Όχι» υπάρχει για τα προϊόντα που δεν απαιτούν δίπλωμα όπως για παράδειγμα οι Αφίσες. Δίπτυχο, τρίπτυχο, οχτασέλιδο ή δεκαεξασέλιδο είναι μερικές επιλογές του χρήστη.

Η επόμενη επιλογή του χρήστη που αποθηκεύεται στη βάση είναι η επιλογή βιβλιοδεσίας. Η επιλογή βιβλιοδεσίας για κάποια προϊόντα είναι απαραίτητη ενώ για κάποια άλλα αδύνατη. Δίνεται στο χρήστη μια λίστα με διάφορες βιβλιοδεσίες να επιλέξει. Η βάση παίρνει αυτόματα τιμή για την επιλογή αυτή από την ιδιότητα value του HTML κώδικα της κάθε μια επιλογής. Καρφίτσα, Κολλητό, Ραφτοκολλητό και Σπιράλ είναι λίγες από τις επιλογές του χρήστη. Οι επιλογές αυτές είναι γραμμένες με ελληνικούς κεφαλαίους χαρακτήρες και η επιλογή με το μεγαλύτερο μήκος έχει δώδεκα χαρακτήρες.

Τέλος η στήλη order_information έχει τη δυνατότητα να αποθηκεύσει 255 χαρακτήρες, όλων των ειδών. Χαρακτήρες ελληνικοί, λατινικοί, αριθμοί και σύμβολα δίνουν τη δυνατότητα στον χρήστη – πελάτη να δώσει ένα μήνυμα στους διαχειριστές της ιστοσελίδας – πελάτες για την παραγγελία ή να συμπληρώσει κάποια πληροφορία που δεν συμπλήρωσε μέχρι τώρα. Παίρνει την τιμή του από το HTML αντικείμενο – textarea και συγκεκριμένα από το value, όπου τοποθετούνται αυτά που πληκτρολογεί ο χρήστης.

Τέλος ο τρίτος και τελευταίος πίνακας της βάσης έχει το όνομα employees. Περιέχει στοιχεία του κάθε χρήστη – υπάλληλου και μπορεί να συμπληρωθεί μόνο από ένα άλλο χρήστη – υπάλληλο που ήδη υπάρχει στη βάση. Οπότε για την περαιτέρω συμπλήρωσή του μια εισαγωγή στη βάση είναι απαραίτητη. Ο πρώτος χρήστης – υπάλληλος θα εισαχθεί στη βάση όχι μέσω ιστοσελίδας αλλά μέσω οποιασδήποτε πρόσβασης έχει δοθεί στο χρήστη κατά την εγκατάσταση της βάσης δεδομένων.

Ο πίνακας αυτός έχει 7 στήλες: id, username, password, first_name, last_name, address, telephone. Η στήλη id αποθηκεύει έναν προσδιοριστικό αριθμό υπαλλήλου – διαχειριστή και τα δεδομένα της στήλης είναι ακέραιοι αυτο-αυξανόμενοι αριθμοί. Η τιμή για κάθε εγγραφή στη βάση αποφασίζεται από τη βάση την ίδια και όχι από τον κώδικα της ιστοσελίδας.

Οι επόμενες δύο στήλες αποθηκεύουν το όνομα χρήστη και τον κωδικό πρόσβασης του διαχειριστή. Αυτά τα δύο τα εισάγει ο υπάρχον υπάλληλος κατά την εισαγωγή νέου υπαλλήλου. Ισχύουν οι κανόνες που έχουμε περιγράψει για την εισαγωγή νέου χρήστη – πελάτη. Και το όνομα χρήστη και ο κωδικός πρόσβασης πρέπει να έχουν την ίδια ορθογραφία κατά την εισαγωγή της

εγγραφής στη βάση και για όλες τις επόμενες συνδέσεις στη βάση. Οι χαρακτήρες θα είναι είτε ελληνικοί είτε λατινικοί και μπορούν να περιέχουν αριθμούς ή γράμματα. Μέγιστο μήκος είναι οι 50 χαρακτήρες για την κάθε μια στήλη.

Επόμενες στήλες στον πίνακα είναι το ονοματεπώνυμο του χρήστη – υπάλληλου. Οι ίδιοι κανόνες ισχύουν και εδώ όπως και σε όλη τη βάση. Οι χαρακτήρες είναι ελληνικοί ή λατινικοί και αν και η βάση δέχεται και αριθμούς και σύμβολα σε ένα όνομα κάτι τέτοιο δεν χρειάζεται.

Τέλος για διάφορους λόγους αποθηκεύονται και δυο επιπλέον πληροφορίες, το τηλέφωνο και η διεύθυνση των υπαλλήλων, τουλάχιστον αυτών που έχουν δημιουργήσει λογαριασμό στη βάση. Οι κανόνες παραμένουν οι ίδιοι. Μήκος εγγραφής στήλης και για τις δυο στήλες είναι οι 50 χαρακτήρες και επιτρέπονται χαρακτήρες όλων των ειδών, ελληνικοί, λατινικοί, σύμβολα και αριθμοί.

ΑΝΑΦΟΡΕΣ

1. <http://books.sonatype.com/mvnref-book/reference/pom-relationships-sect-pom.html#pom-relationships-sect-super-pom>
2. <http://maven.apache.org/download.cgi>
3. <http://books.sonatype.com/mvnref-book/reference/lifecycle-sect-structure.html#lifecycle-sect-site>
4. <http://maven.apache.org/plugins/index.html>
5. <http://maven.apache.org/plugins/maven-deploy-plugin>
6. <http://maven.apache.org/plugins/maven-resources-plugin>
7. <http://maven.apache.org/plugins/maven-site-plugin>
8. <http://maven.apache.org/plugins/maven-rar-plugin>
9. <https://www.eclipse.org/org>
10. http://wiki.netbeans.org/MavenBestPractices#Open_existing_project
11. <http://maven.apache.org/guides/plugin/guide-java-plugin-development.html>
12. <http://maven.apache.org/plugin-developers/plugin-testing.html>

Βιβλιογραφία

<http://books.sonatype.com/mvnref-book/reference/index.html>

<http://maven.apache.org/>

<http://khmarbaise.github.io/maui/index.html>

<http://www.maestrodev.com/wp-content/uploads/2012/03/betterbuildswithmaven-2008.pdf>