

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

# JAVA WEB SERVICES



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

*που εκπονήθηκε από την φοιτήτρια:*

ΣΩΤΗΡΙΑ ΚΑΡΑΘΑΝΟΥ

ΤΟΥ ΑΝΔΡΕΑ

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΔΡ. ΕΥΚΛΕΙΔΗΣ ΚΕΡΑΜΟΠΟΥΛΟΣ**

ΘΕΣΣΑΛΟΝΙΚΗ 2010

## Πρόλογος

Η Τεχνολογία των web services κατέχει μία πολύ σημαντική θέση στην ανάπτυξη client-server εφαρμογών και αποτελεί κυρίαρχο κομμάτι της λειτουργίας των σύγχρονων επιχειρήσεων που προσφέρουν τις υπηρεσίες τους μέσω του διαδικτύου.

Οι κυριότερες τεχνολογίες που συμμετέχουν στην ανάπτυξη των web services είναι η XML ( eXtensible Markup Language ), το SOAP ( Simple Object Access Protocol ), η WSDL ( Web Services Description Protocol ) και το UDDI ( Universal Description, Discovery and Integration ). Η Τεχνολογία XML καλύπτει το κομμάτι της περιγραφής των δεδομένων που ανταλλάσσονται μέσω των web services, το πρωτόκολλο SOAP προσδιορίζει την μορφή της επικοινωνίας που επιτυγχάνεται κατά την κλήση ενός web service και συγκεκριμένα προσδιορίζει την μορφή των μηνυμάτων και τον τρόπο ανταλλαγής τους, η γλώσσα WSDL περιγράφει τα web services, συνιστώντας κατά κάποιο τρόπο την γραμματική των web services και η τεχνολογία UDDI αναφέρεται στο κομμάτι της καταχώρησης στο web επιχειρήσεων και των web services που προσφέρουν.

Η χρήση της Java ως γλώσσας υλοποίησης των web services έχει ως αποτέλεσμα την ανάπτυξη εφαρμογών με απεριόριστες δυνατότητες και πλεονεκτήματα, χωρίς περιορισμούς στην ανάπτυξη αλλά και την χρήση τους.

## **Preface**

The Web Services Technology has a very important position in the development of client-server applications and consists great part of modern enterprises' function that offer their services through the internet.

The most important technologies that take part at the web services' development are: XML ( eXtensible Markup Language ), SOAP ( Simple Object Access Protocol ) and UDDI ( Universal Description Discover and Integration ). The XML technology covers the part of the data description which are being interchanged through web services, the SOAP protocol identifies the communication mode that is being succeeded at a web service's call and more specifically identifies the message form and the way in which they are being interchange, WSDL language describes the web services proposing in a way the grammar of web services and UDDI technology refers to the part of the enterprises' registration at the web and the web services' registration that the enterprises offer.

The use of Java as web services' materialization language has as effect the development of applications with unlimited capabilities and advantages, without restrictions at their development and use.

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Ευρετήριο Παραδειγμάτων Κώδικα .....	6
Ευρετήριο Εικόνων .....	7
1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΝΝΟΙΑ ΤΩΝ WEB SERVICES .....	8
1.1 Τί είναι οι Web Services .....	8
1.2 Προγενέστερες τεχνολογίες .....	8
1.3 Σκοπός Χρήσης.....	9
1.4 Βασικά χαρακτηριστικά των Web Services.....	10
2 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΩΝ WEB SERVICES .....	11
2.1 Βασικά στοιχεία αρχιτεκτονικής .....	11
2.2 Αρχιτεκτονική SOA .....	13
2.3 Τεχνολογίες των web services .....	14
2.4 Το πρότυπο των web services.....	15
2.5 Το Μοντέλο P2P ( Peer to Peer ) .....	17
3 SOAP, ΤΟ ΠΡΩΤΟΚΟΛΛΟ ΕΠΙΚΟΙΝΩΝΙΑΣ ΤΩΝ WEB SERVICES .....	18
3.1 Παρουσίαση του πρωτοκόλλου SOAP .....	18
3.2 Ανατομία ενός SOAP μηνύματος.....	20
3.3 Κωδικοποίηση δεδομένων SOAP .....	26
3.4 Τύποι δεδομένων SOAP .....	27
3.5 Διαδικασία ανταλλαγής SOAP μηνυμάτων.....	27
3.6 Χειρισμός σφαλμάτων με SOAP .....	27
4 ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ XML – RPC .....	34
4.1 Εισαγωγή .....	34
4.2 Ορισμός.....	34
4.3 Γιατί χρησιμοποιείται .....	34
4.4 Στοιχεία αρχιτεκτονικής XML – RPC .....	34
4.5 Μοντέλο Δεδομένων XML – RPC .....	35
4.6 Δομή αιτήσεων XML – RPC .....	39
4.7 Δομή αποκρίσεων XML – RPC.....	40
4.8 Ανάπτυξη εφαρμογών με το XML – RPC .....	40
5 WSDL, Η ΠΕΡΙΓΡΑΦΙΚΗ ΓΛΩΣΣΑ ΤΩΝ WEB SERVICES .....	41
5.1 Εισαγωγή – Ορισμός.....	41
5.2 Η δομή ενός απλού WSDL εγγράφου. ....	41
5.3 Χαρακτηριστικά ενός εγγράφου WSDL.....	47
6 Η ΓΛΩΣΣΑ ΣΗΜΑΝΣΗΣ XML .....	49
6.1 Ιστορία της XML .....	49
6.2 Λόγοι χρήσης της XML .....	49
6.3 Πλεονεκτήματα .....	50
6.4 Μειονεκτήματα .....	50
6.5 Σύνταξη της XML.....	51
6.6 Σήμανσεις.....	52
6.7 XML Document Type Definition ( DTD ) .....	53
6.8 XML και προγενέστερες γλώσσες σήμανσης ( markup languages ).....	54
7 UDDI ΚΑΤΑΧΩΡΗΣΗ ΤΩΝ WEB SERVICES .....	56
7.1 Τί είναι μία καταχώρηση Web Service.....	56
7.2 Ιστορικά στοιχεία.....	56

7.3	Πώς χρησιμοποιείται.....	57
7.4	Τεχνικά Χαρακτηριστικά.....	57
7.5	Το Μοντέλο Δεδομένων UDDI .....	58
7.6	Ιδιαίτερα χαρακτηριστικά του UDDI .....	58
8	<b>JAVA SERVLETS, ΜΙΑ ΠΡΟΓΕΝΕΣΤΕΡΗ ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ WEB SERVICES.....</b>	<b>60</b>
8.1	Ορισμός.....	60
8.2	JAVA Servlets, μία προγενέστερη Τεχνολογία των Web Services.....	61
8.3	Ανατομία ενός JAVA Servlet .....	62
8.4	Χειρισμός δεδομένων φόρμας με servlets .....	66
8.5	J2EE και Web Services.....	73
9	<b>ΑΣΦΑΛΕΙΑ ΤΩΝ WEB SERVICES.....</b>	<b>75</b>
9.1	Θέματα Ασφάλειας των Web Services .....	75
9.2	Επίπεδα ασφάλειας των web services.....	76
9.3	Προδιαγραφές αντιμετώπισης των θεμάτων ασφαλείας.....	76
9.4	XML ψηφιακές υπογραφές και κρυπτογράφηση δεδομένων .....	76
9.5	Εργαλεία ανάπτυξης των Web Services .....	77
10	<b>ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ.....</b>	<b>79</b>
	<b>ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ.....</b>	<b>91</b>
	<b>ΠΑΡΑΡΤΗΜΑ.....</b>	<b>93</b>
	<b>ΕΠΕΞΗΓΗΣΗ ΑΓΓΛΙΚΩΝ ΟΡΩΝ .....</b>	<b>108</b>

## Ευρετήριο Παραδειγμάτων Κώδικα

<i>Κώδικας</i>	<i>Σελ</i>
Κώδικας 3.1 - Το στοιχείο SOAP Envelope .....	23
Κώδικας 3.2 - Το στοιχείο SOAP Header.....	23
Κώδικας 3.3 - Το στοιχείο SOAP Body .....	23
Κώδικας 3.4 - Το στοιχείο SOAP Fault .....	23
Κώδικας 3.5 – Απλοί τύποι δεδομένων SOAP .....	26
Κώδικας 3.6 – Σύνθεση τύπων δεδομένων SOAP .....	26
Κώδικας 3.7 – Δομές δεδομένων, ένας σύνθετος τύπος στο SOAP .....	26
Κώδικας 3.8 – Πίνακες, ένας σύνθετος τύπος στο SOAP .....	27
Κώδικας 3.9 – Ένα μήνυμα σφάλματος του SOAP .....	28
Κώδικας 3.10 – Ένα απλό παράδειγμα χειρισμού σφαλμάτων με SOAP Faults .....	30
Κώδικας 3.11 – Η ιδιότητα mustUnderstand στα SOAP Faults.....	31
Κώδικας 3.12 – Μορφή του SOAP Envelope με τον ορισμό της ιδιότητας mustUnderstand.....	31
Κώδικας 3.13 – Μήνυμα σφάλματος λόγω αδυναμίας αναγνώρισης του στοιχείου MessageHeader .....	32
Κώδικας 3.14 – Ένα JAVA Servlet για τον χειρισμό σφαλμάτων SOAP .....	33
Κώδικας 4.15 – Η μορφή των δεδομένων XML-RPC .....	35
Κώδικας 4.16 – Απλοί τύποι δεδομένων XML-RPC .....	37
Κώδικας 4.17 – Σύνθετοι τύποι δεδομένων XML-RPC, μονοδιαστατοι πίνακες .....	37
Κώδικας 4.18 – Σύνθετοι τύποι δεδομένων XML-RPC – πολυδιάστατοι πίνακες.....	37
Κώδικας 4.19 – Σύνθετοι τύποι δεδομένων XML-RPC, δομές δεδομένων ....	38
Κώδικας 4.20 – Σύνθετες δομές δεδομένων XML-RPC .....	39
Κώδικας 4.21 – Μορφή αίτησης XML-RPC.....	40
Κώδικας 4.22 – Μορφή απάντησης XML-RPC .....	40
Κώδικας 5.23 – Η δομή ενός απλού WSDL εγγράφου.....	41
Κώδικας 5.24 – Ένα παράδειγμα WSDL εγγράφου .....	44
Κώδικας 5.25 – Το στοιχείο definitions σε ένα WSDL έγγραφο .....	45
Κώδικας 5.26 – Το στοιχείο message σε ένα WSDL έγγραφο .....	45
Κώδικας 5.27 – Το στοιχείο portType σε ένα WSDL έγγραφο .....	46
Κώδικας 5.28 – Το στοιχείο binding σε ένα WSDL έγγραφο .....	46
Κώδικας 5.29 – Το στοιχείο service σε ένα WSDL έγγραφο .....	47
Κώδικας 5.30 – Ονοματοδοσία και σύνδεσμοι σε ένα WSDL έγγραφο .....	47
Κώδικας 5.31 – Authorization υπηρεσιών σε ένα WSDL έγγραφο .....	48
Κώδικας 6.32 – Ένα παράδειγμα XML εγγράφου .....	51
Κώδικας 6.33 – Η σύνταξη σε ένα XML έγγραφο.....	52
Κώδικας 6.34 – XML DTD.....	53
Κώδικας 6.35 - XML DTD Elements.....	53
Κώδικας 8.36 – Ένα παράδειγμα Servlet.....	63

Κώδικας 8.37 - Ένα Java Servlet που λαμβάνει απλή τιμή παραμέτρου από μία φόρμα. ....	67
Κώδικας 8.38 - Το servlet λαμβάνει την τιμή της παραμέτρου με όνομα "inputTerm" .....	69
Κώδικας 8.39 - Το servlet λαμβάνει την τιμή της παραμέτρου με όνομα "jumpMenu" η οποία προσδιορίζει μία drop – down λίστα. ....	72

## Ευρετήριο Εικόνων

<b>Εικόνα</b>	<b>Σελ</b>
<i>Εικόνα 1.1 – Εξαρτήσεις προγενέστερων των Web Services τεχνολογιών .....</i>	9
<i>Εικόνα 1.2 – Εξαρτήσεις των Web Services .....</i>	9
<i>Εικόνα 2.3 – Αρχιτεκτονική των Web Services.....</i>	11
<i>Εικόνα 2.4 – Συστατικά μέρη της αρχιτεκτονικής των Web Services.....</i>	14
<i>Εικόνα 2.5 – Το πρότυπο επικοινωνίας των Web Services.....</i>	16
<i>Εικόνα 3.6 – Πρότυπο ανταλλαγής XML μηνυμάτων.....</i>	18
<i>Εικόνα 3.7 – Μορφή μηνύματος SOAP χωρίς Attachments.....</i>	21
<i>Εικόνα 3.8 – Μορφή μηνύματος SOAP με Attachments.....</i>	22
<i>Εικόνα 5.9 – Τα επίπεδα των στοιχείων σε ένα WSDL έγγραφο .....</i>	43
<i>Εικόνα 8.10 – Επικοινωνία client/server μέσω ServletRequest και ServletResponse.....</i>	65
<i>Εικόνα 8.11 – Πώς πραγματοποιείται η επικοινωνία μέσα στο J2EE .....</i>	74

# 1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΝΝΟΙΑ ΤΩΝ WEB SERVICES

## 1.1 *Τί είναι οι Web Services*

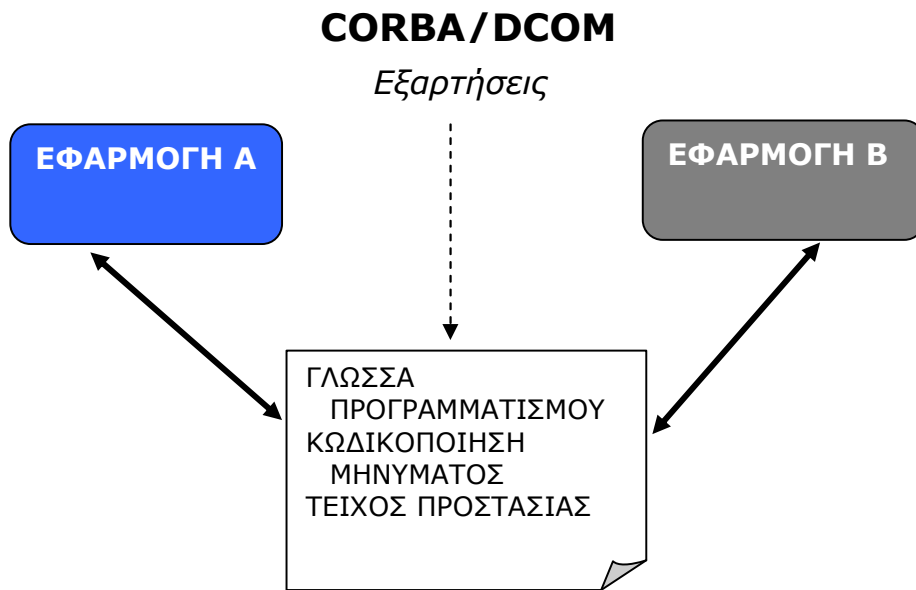
Web Service είναι μια εφαρμογή λογισμικού για την παροχή υπηρεσιών μέσω του διαδικτύου. Οι χρήστες / πελάτες μπορούν να έχουν πρόσβαση σε μια web υπηρεσία μέσω του url που αντιστοιχεί σε αυτή. Είναι πλέον ο πιο διαδεδομένος τρόπος για να παρέχουν οι επιχειρήσεις, τις υπηρεσίες τους στους πελάτες, μέσω του διαδικτύου. Όπως και κάθε τοποθεσία στο Internet, οι Web Services είναι προσβάσιμες μέσω πρωτοκόλλων επικοινωνίας όπως το HTTP, SMTP, κ.α. Η δημιουργία του γραφικού περιβάλλοντος καθώς και του συνόλου της λειτουργικότητάς τους, βασίζεται στην τυποποιημένη γλώσσα XML.

## 1.2 *Προγενέστερες τεχνολογίες*

Οι Web Services, όπως θα αναλυθεί και παρακάτω, υποστηρίζονται από την SOA Αρχιτεκτονική ( Αρχιτεκτονική προσανατολισμένη στις υπηρεσίες ). Η δομή της εφαρμογής που περιγράφει η αρχιτεκτονική αυτή ήταν σε χρήση για πολλά χρόνια. Η CORBA ( Common Object Request Broker Architecture ) αποτελεί χαρακτηριστικό παράδειγμα μιας τέτοιας αρχιτεκτονικής, που ήταν σε χρήση μέχρι τα τέλη της δεκαετίας του 1980. Ένα ακόμη παράδειγμα αποτελεί η DCOM ( Distributed Component Object Model ) της Microsoft.

Σε αντίθεση με τις δύο αυτές αρχιτεκτονικές, η SOA που βασίζεται στις Web Services, συνιστά ένα μοντέλο χαμηλής σύζευξης και έτσι, δηλαδή αλλαγές που μπορεί να γίνουν σε ένα ή περισσότερα κομμάτια μιας εφαρμογής δεν επηρεάζουν ολόκληρη την εφαρμογή και είναι ανεξάρτητο από γλώσσες προγραμματισμού, λειτουργικά συστήματα, κωδικοποίηση μηνυμάτων, κ.τ.λ. , όπως περιγράφεται από το παρακάτω σχήμα.





Εικόνα 1.1 – Εξαρτήσεις προγενέστερων των Web Services τεχνολογιών



Εικόνα 1.2 – Εξαρτήσεις των Web Services

### 1.3 Σκοπός Χρήσης

Σκοπός των Web Services είναι να δημιουργήσουν ένα κατακευματισμένο περιβάλλον που να περιλαμβάνει ένα σύνολο εφαρμογών και συστατικά εφαρμογών, παρέχοντας την δυνατότητα επικοινωνίας μεταξύ τους, αλλά και μεταξύ των επιχειρήσεων που παρέχουν τις υπηρεσίες τους μέσω Internet, χρησιμοποιώντας ένα ενιαίο πρωτόκολλο επικοινωνίας, χωρίς περιορισμούς.

## **1.4 Βασικά χαρακτηριστικά των Web Services**

Οι Web Services μπορούν να ανταποκριθούν στις απαιτήσεις των σύγχρονων επιχειρήσεων, οι οποίες προκειμένου να πραγματοποιήσουν τις συναλλαγές τους, απαιτείται η διασύνδεση των υπηρεσιών που παρέχουν μέσω του παγκόσμιου ιστού. Επιπλέον, οι Web Services επιτρέπουν στους συναλλασσόμενους από όλο τον κόσμο να ανακαλύπτουν ο ένας τον άλλο μέσω του διαδικτύου, να συνδέονται δυναμικά μεταξύ τους αλλά και με τις απαιτούμενες εφαρμογές που υποστηρίζουν την πραγματοποίηση των συναλλαγών και να πραγματοποιούν τις συναλλαγές τους σε πραγματικό χρόνο με την ελάχιστη ανθρώπινη παρέμβαση. Τα βασικά χαρακτηριστικά τα οποία αποτελούν ταυτόχρονα πλεονεκτήματα των Web Services είναι τα εξής:

Οι Web Services βασίζονται στην ανταλλαγή μηνυμάτων XML ( XML Messaging ), πράγμα που σημαίνει ότι τα δεδομένα τα οποία ανταλλάσσονται μεταξύ ενός παρόχου μιας Web υπηρεσίας και ενός χρήστη / πελάτη, είναι κωδικοποιημένα σε μορφή XML.

Οι προγραμματιστές, προκειμένου να αναπτύξουν μια Web Service, μπορούν να χρησιμοποιήσουν οποιαδήποτε γλώσσα προγραμματισμού, όπως JAVA, C, C++, C# και τα συστατικά αυτών.

Βασικό χαρακτηριστικό είναι η ανεξαρτησία τους από γλώσσες προγραμματισμού και λειτουργικά συστήματα, πράγμα που επιτρέπει την αλληλεπίδραση μεταξύ εφαρμογών στο διαδίκτυο.

## 2 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΩΝ WEB SERVICES

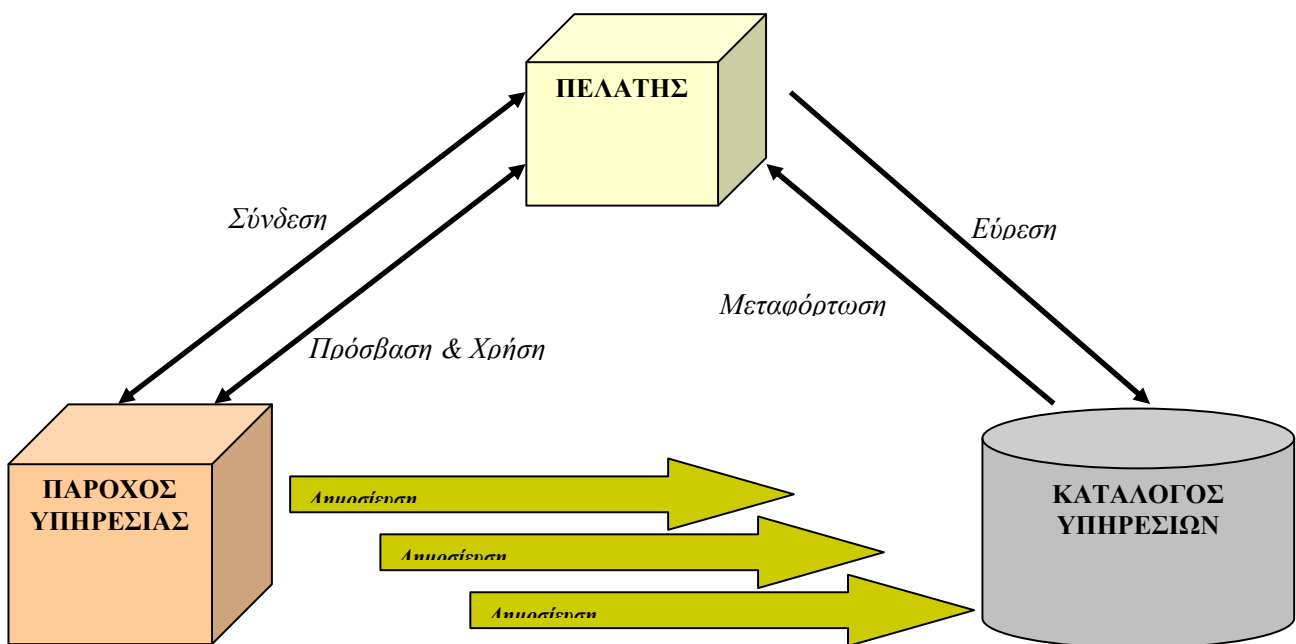
### 2.1 Βασικά στοιχεία αρχιτεκτονικής

Η τεχνολογία των web services ακολουθεί τη ροή λειτουργιών: δημοσίευση, εύρεση, σύνδεση ( publish, find, bind ).

α) δημοσίευση ( publish ) : Κατά το στάδιο αυτό, ο πάροχος της υπηρεσίας δημοσιεύει την υπηρεσία και την περιγραφή της υπηρεσίας σε έναν κατάλογο υπηρεσιών.

β) εύρεση ( find ) : Ο πελάτης ο οποίος προσδιορίζει μια υπηρεσία η οποία ικανοποιεί τις απαιτήσεις του, αναζητά την υπηρεσία αυτή στον κατάλογο υπηρεσιών.

γ) σύνδεση ( bind ) : Το αποτέλεσμα της αναζήτησής του είναι ένα σύνολο υπηρεσιών οι οποίες είναι πιο κοντά στις ανάγκες του και από αυτές επιλέγει αυτήν που προτιμά. Στην συνέχεια, μεταφορτώνει την περιγραφή της υπηρεσίας και συνδέεται με αυτήν ώστε να πραγματοποιηθεί η κλήση και η εκτέλεση της υπηρεσίας.



Εικόνα 2.3 – Αρχιτεκτονική των Web Services

## **Συστατικά στοιχεία της αρχιτεκτονικής – Ρόλοι**

### **Ο πάροχος της υπηρεσίας**

Είναι αυτός που παρέχει την web υπηρεσία. Υλοποιεί την υπηρεσία και την κάνει διαθέσιμη μέσω του Internet.

### **Ο πελάτης – Αιτώντος Υπηρεσίας**

Είναι ένας οποιοσδήποτε πελάτης μιας web υπηρεσίας ο οποίος χρησιμοποιεί μια υπάρχουσα web service πραγματοποιώντας μια σύνδεση στο διαδίκτυο και στέλνοντας ένα XML αίτημα.

### **Κατάλογος Υπηρεσιών**

Είναι ένας λογικά δομημένος κατάλογος υπηρεσιών. Προσφέρει έναν κεντρικό χώρο όπου οι προγραμματιστές έχουν την δυνατότητα να δημοσιεύσουν νέες υπηρεσίες ή να βρουν ήδη υπάρχουσες υπηρεσίες.

Ο ρόλος του καταλόγου υπηρεσιών είναι ένας προαιρετικός ρόλος στην αρχιτεκτονική των web services καθότι ένας πάροχος υπηρεσίας μπορεί να αποστείλει την περιγραφή μιας υπηρεσίας απευθείας στον πελάτη. Επίσης, οι πελάτες μπορούν να ανακτήσουν περιγραφές υπηρεσιών από άλλες πηγές, όπως ένα τοπικό αρχείο, μια τοποθεσία FTP, ή ένας δικτυακός τόπος.

## **Αντικείμενα**

### **Υπηρεσία ( Service )**

Πρόκειται για μία λειτουργία που παρέχεται από μία επιχείρηση μέσω του διαδικτύου και η οποία περιλαμβάνει μία ή περισσότερες μεθόδους που εκτελούν μία συναλλαγή μεταξύ του πελάτη και του πάροχου της υπηρεσίας. Η πρόσβαση σε μία υπηρεσία, επιτυγχάνεται μέσω μηνυμάτων σε μορφή XML που αποστέλλει ο πελάτης στον server, ο οποίος με την σειρά του αποστέλλει απαντητικά μηνύματα, επίσης σε XML μορφή και επιστρέφει το αποτέλεσμα της κλήσης μιας μεθόδου του web service.

### **Περιγραφή της υπηρεσίας ( Service Description )**

Πρόκειται για ένα σύνολο πληροφοριών που περιέχονται μέσα σε έγγραφα γραμμένα σε WSDL, την περιγραφική γλώσσα των web services. Οι πληροφορίες αυτές, προσδιορίζουν το όνομα της επιχείρησης που παρέχει ένα web service, τις λειτουργίες που εκτελεί το web service καθώς και το πού τοποθετείται ένα web service στο διαδίκτυο.

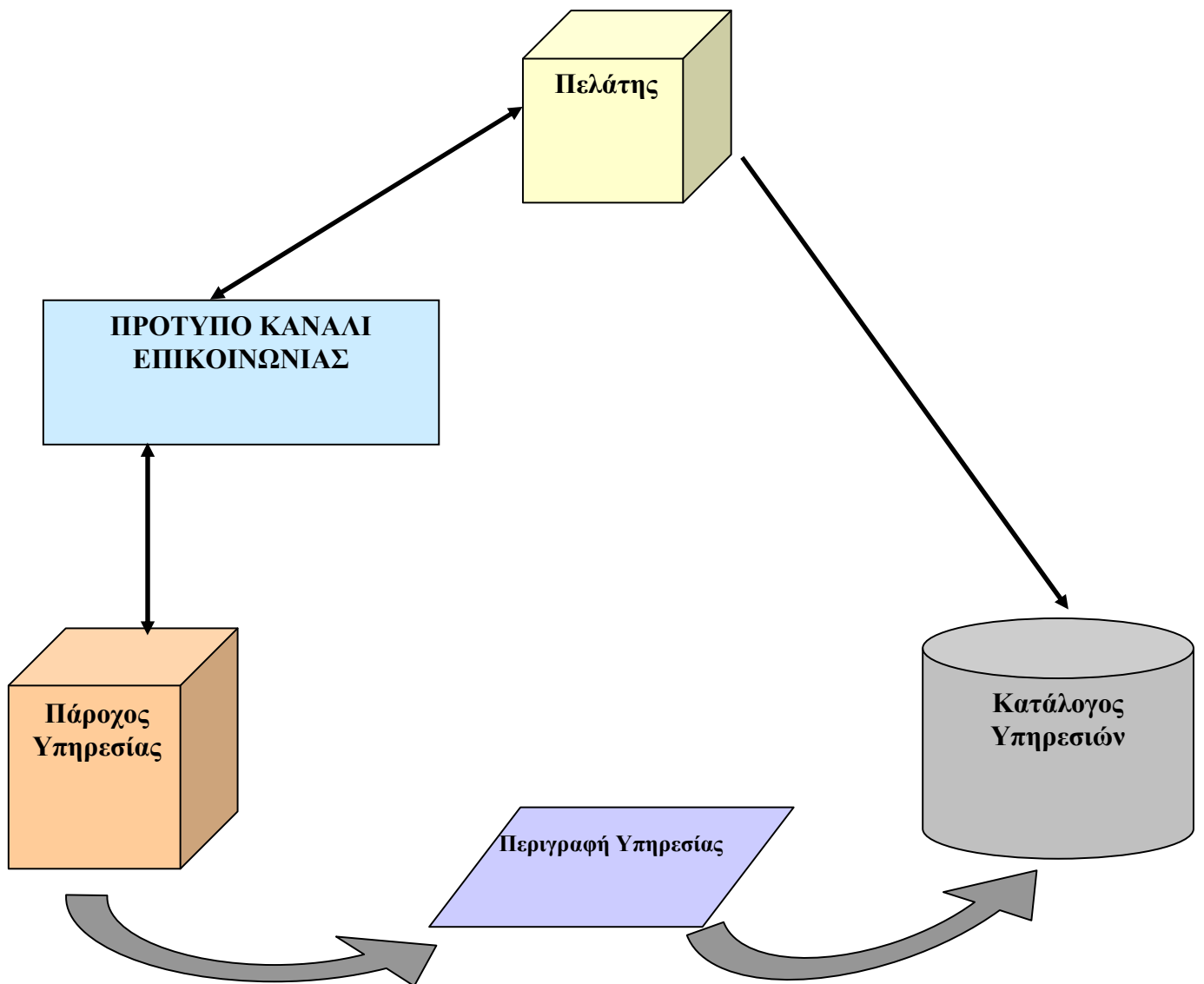
## 2.2 Αρχιτεκτονική SOA

Η αρχιτεκτονική των web services συνιστά μια αρχιτεκτονική προσανατολισμένη στις υπηρεσίες ( SOA ). Μια αρχιτεκτονική προσανατολισμένη στις υπηρεσίες, ορίζει τις μεθόδους για την ανάπτυξη συστημάτων, τα οποία ομαδοποιούν την λειτουργικότητα των επιχειρησιακών λειτουργιών σε ένα σύνολο διαλειτουργικών υπηρεσιών.

Μια SOA αρχιτεκτονική επιτρέπει την ανταλλαγή δεδομένων μεταξύ διαφορετικών εφαρμογών καθώς αυτές μετέχουν σε επιχειρησιακές λειτουργίες. Μια υπηρεσία αυτής της μορφής, στοχεύει στην χαμηλή συνδεσιμότητα των υπηρεσιών με λειτουργικά συστήματα, γλώσσες προγραμματισμού και άλλες τεχνολογίες που υποστηρίζουν διάφορων ειδών εφαρμογές. Η SOA διαχωρίζει τις λειτουργίες σε διακριτές μονάδες, ή υπηρεσίες, τις οποίες οι προγραμματιστές κάνουν προσβάσιμες μέσω του διαδικτύου με σκοπό οι χρήστες να μπορούν να τις συνδυάζουν και να τις επαναχρησιμοποιούν στην παραγωγή των επιχειρησιακών εφαρμογών. Αυτές οι υπηρεσίες επικοινωνούν μεταξύ τους με ανταλλαγή δεδομένων η μία στην άλλη ή συντονίζοντας μια δραστηριότητα μεταξύ δύο ή περισσότερων υπηρεσιών.

Το γεγονός ότι οι web services υποστηρίζουν μια αρχιτεκτονική προσανατολισμένη στις υπηρεσίες, θέτει ένα ερώτημα. Η εφαρμογή - πάροχος υπηρεσίας και η εφαρμογή – πελάτης μιλάνε κοινή γλώσσα μεταξύ τους. Με ποιο τρόπο όμως, οι δύο εφαρμογές μπορούν να εντοπίζουν η μία την άλλη; Για να δοθεί λύση στο πρόβλημα αυτό, η αρχιτεκτονική των web services πρέπει να παρέχει τα εξής:

- Έναν πρότυπο τρόπο επικοινωνίας
- Μια ομοιόμορφη τεχνική περιγραφής και ανταλλαγής δεδομένων
- Μια πρότυπη περιγραφική γλώσσα ( meta language ) για να περιγράψει τις υπηρεσίες που προσφέρονται
- Έναν μηχανισμό καταχώρησης και εντοπισμού των εφαρμογών που βασίζονται σε web services



Εικόνα 2.4 – Συστατικά μέρη της αρχιτεκτονικής των Web Services

## 2.3 Τεχνολογίες των web services

### XML Messaging

Η ανταλλαγή μηνυμάτων XML επιτρέπει την αποστολή και λήψη δεδομένων, χωρίς να μπαίνουν περιορισμοί όσον αφορά την πλατφόρμα του λειτουργικού συστήματος ή την γλώσσα προγραμματισμού. Η τεχνολογία αυτή προσφέρει μία πληθώρα εργαλείων, όπως οι συντάκτες κώδικα και υποστηρίζει τις περισσότερες γλώσσες προγραμματισμού, μεταξύ των οποίων είναι η Java, η Perl, η C και η C++. Είναι πολύ σημαντικός ο ρόλος της στην ανάπτυξη εφαρμογών web services. Χρησιμοποιεί δύο βασικά συστατικά της τεχνολογίας XML Messaging για την ανταλλαγή μηνυμάτων, το SOAP και το XML-RPC.

## SOAP - Simple Object Access Protocol

Αποτελεί το βασικό πρωτόκολλο επικοινωνίας μεταξύ υπολογιστών. Μολονότι, μπορεί να χρησιμοποιηθεί από εφαρμογές που χρησιμοποιούν οποιοδήποτε σύστημα που χειρίζεται την αποστολή των μηνυμάτων και το ότι υποστηρίζει την ανταλλαγή μηνυμάτων μεταξύ υπολογιστών που χρησιμοποιούν διαφορετικά πρωτόκολλα μεταφοράς μηνυμάτων, το SOAP χρησιμοποιεί το RPC για την μεταφορά των μηνυμάτων, σε αντίθεση με την μέθοδο POST του πρωτοκόλλου HTTP. Έτσι, όπως και η τεχνολογία XML-RPC εξαρτάται από τις πλατφόρμες του λειτουργικού συστήματος και τον βαθμό διαφοροποίησης που εμφανίζουν μεταξύ τους οι εφαρμογές που επικοινωνούν.

## WSDL - Web Service Description Language

Είναι η περιγραφική γλώσσα των web services, με την έννοια ότι προσδιορίζει την δημόσια διασύνδεσή τους η οποία περιλαμβάνει πληροφορίες για τις μεθόδους που περιέχει, πληροφορίες για τους τύπους των δεδομένων, πληροφορίες για το πρωτόκολλο ανταλλαγής μηνυμάτων που χρησιμοποιείται και τέλος, πληροφορίες σχετικά με το πού τοποθετείται μία υπηρεσία στο διαδίκτυο.

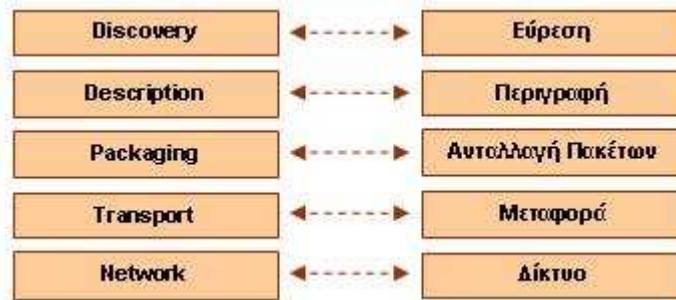
## UDDI - Universal Description, Discovery and Integration

Η Τεχνολογία UDDI αφορά το κομμάτι της καταχώρησης των web services .

Ορίζει δηλαδή τους καταλόγους στους οποίους είναι καταχωρημένες, επιχειρήσεις και υπηρεσίες. Οι αποθηκευμένες πληροφορίες δεν είναι τίποτε άλλο, παρά δεδομένα σε XML μορφή.

## **2.4 Το πρότυπο των web services**

Ο στόχος του προτύπου αυτού είναι η δημιουργία τυποποίησης έτσι ώστε να διευκολύνεται η παροχή υπηρεσιών μέσω του διαδικτύου καθώς αυτή στηρίζεται στην επικοινωνία και την ανταλλαγή πληροφοριών, έτσι πρέπει να υπάρχει τυποποίηση για την ομαλή λειτουργία τους. Το πρότυπο μέχρι στιγμής αποτελείται από πέντε επίπεδα, καθώς αυτό βρίσκεται ακόμη υπό ανάπτυξη.



Εικόνα 2.5 – Το πρότυπο επικοινωνίας των Web Services

### **Service Discovery ( Εύρεση Υπηρεσίας )**

Το επίπεδο αυτό είναι υπεύθυνο για την οργάνωση των υπηρεσιών μέσα σε ένα κοινό αρχείο, ώστε να διευκολύνεται η δημοσίευση ( ή καταχώρηση ) και η εύρεσή τους. Υποστηρίζεται από το πρωτόκολλο UDDI.

### **Service Description ( Περιγραφή Υπηρεσίας )**

Ασχολείται με την περιγραφή της διεπαφής της παρεχόμενης προς τους χρήστες web υπηρεσίας. Στο επίπεδο αυτό χρησιμοποιείται το πρωτόκολλο WSDL.

### **XML Messaging ( Ανταλλαγή XML Μηνυμάτων )**

Ασχολείται με την κωδικοποίηση των μηνυμάτων ( ή πληροφοριών ), που ανταλλάσσονται μεταξύ υπολογιστικών συστημάτων στο διαδίκτυο, σε XMLμορφή, ώστε να είναι κατανοητά από όλους. Το επίπεδο υποστηρίζεται από τα πρωτόκολλα: XML – RPC και το SOAP.

### **Service Transport ( Μεταφορά Υπηρεσίας )**

Ασχολείται με την μεταφορά πληροφοριών μεταξύ των εφαρμογών. Τα πρωτόκολλα που χρησιμοποιούνται στο επίπεδο αυτό είναι: το HTTP ( HyperText Transfer Protocol ), το SMTP ( Simple Mail Transfer Protocol ), το FTP ( File Transfer Protocol ) και ένα καινούριο πρωτόκολλο, το Beep ( Blocks Extensible Exchange Protocol ).

### **Service Network ( Δίκτυο Υπηρεσίας )**

Το επίπεδο δικτύου στο πρότυπο της Τεχνολογίας των web services είναι ανάλογο του επιπέδου δικτύου στο μοντέλο του πρωτοκόλλου TCP. Παρέχει τις βασικές ικανότητες επικοινωνίας, διευθυνσιοδότησης και δρομολόγησης μηνυμάτων και πακέτων.



## **2.5 Το Μοντέλο P2P ( Peer to Peer )**

### **Ορισμός**

Το μοντέλο Peer to Peer περιγράφει ένα δίκτυο υπολογιστών των οποίων οι πόροι μπορεί να διαμοιράζονται ισοδύναμα. Στους πόρους αυτούς περιλαμβάνονται, η επεξεργαστική ισχύς, η μνήμη και το εύρος ζώνης. Κάθε κόμβος έχει πρόσβαση στις πληροφορίες οποιουδήποτε άλλου κόμβου.

### **Κατηγορίες Peer-to-peer δικτύων**

#### Συγκεντρωτικά

Στο δίκτυο αυτό υπάρχει ένας κεντρικός Server στον οποίο είναι αποθηκευμένες οι πληροφορίες των περιεχομένων καταλόγων τους οποίους διαμοιράζονται οι κόμβοι του δικτύου. Η πρόσβαση στα περιεχόμενα αυτά γίνεται με την βοήθεια κατάλληλων προγραμμάτων – πελάτη.

#### Αποκεντρωτικά

Εδώ κάθε κόμβος του δικτύου αποτελεί ταυτόχρονα client και server . Η μεταφορά των αρχείων γίνεται με τρόπο ανάλογο του συγκεντρωτικού δικτύου.

#### Τρίτης Γενιάς

Τα δίκτυα αυτά διαθέτουν χαρακτηριστικά ανωνυμίας. Είναι αποκεντρωτικού τύπου και τα χαρακτηριστικά τους, εκτός από την ανωνυμία, είναι η μεγάλη διάρκεια ζωής, ο συνεχής διαμοιρασμός των αρχείων και η κωδικοποίηση των αρχείων έτσι ώστε να προστατεύονται από μη εξουσιοδοτημένη άσκηση ελέγχου πάνω τους.

## 3 SOAP, ΤΟ ΠΡΩΤΟΚΟΛΛΟ ΕΠΙΚΟΙΝΩΝΙΑΣ ΤΩΝ WEB SERVICES

### 3.1 Παρουσίαση του πρωτοκόλλου SOAP

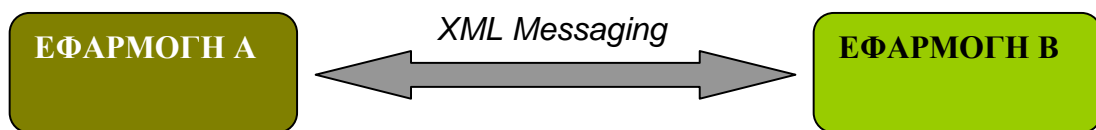
Το SOAP είναι το πρωτόκολλο που χρησιμοποιείται από τις Υπηρεσίες Ιστού για την ανταλλαγή μηνυμάτων μεταξύ των εφαρμογών. Αποτελεί μια ενθυλάκωση των πληροφοριών, η οποία βασίζεται στην XML και ένα σύνολο ρόλων για την μετάφραση των εφαρμογών και την κωδικοποίηση τύπων δεδομένων από διαφορετικές πλατφόρμες σε μορφή XML.

#### Η σχέση μεταξύ του SOAP και της XML

Το SOAP για τον ορισμό και την λειτουργία του βασίζεται σε πρότυπα XML όπως το XML σχήμα και τους XML χώρους ονομάτων.

#### XML Messaging

Η XML Messaging χρησιμοποιείται για την ανταλλαγή μηνυμάτων μεταξύ των εφαρμογών, οι οποίες επικοινωνούν ανταλλάσσοντας XML έγγραφα. Αποτελεί την βάση για την λειτουργία του SOAP ως πρωτοκόλλου επικοινωνίας των Υπηρεσιών Ιστού. Επειδή η XML είναι ανεξάρτητη από Λειτουργικά Συστήματα και Γλώσσες Προγραμματισμού, μπορεί να χρησιμοποιηθεί σε οποιοδήποτε περιβάλλον. Η Κεντρική ιδέα είναι, δύο εφαρμογές οι οποίες δεν εξαρτώνται από Λειτουργικά Συστήματα και Γλώσσες Προγραμματισμού ή άλλα τεχνικά χαρακτηριστικά, να μπορούν να ανταλλάξουν ελεύθερα πληροφορίες, χρησιμοποιώντας τίποτε παραπάνω από απλά μηνύματα, κωδικοποιημένα σε μία μορφή, που να μπορούν να καταλάβουν και οι δύο εφαρμογές. Το SOAP προσφέρει έναν πρότυπο τρόπο δημιουργίας XML μηνυμάτων.



Εικόνα 3.6 – Πρότυπο ανταλλαγής XML μηνυμάτων

#### Η Ιστορία του SOAP

Το SOAP αναπτύχθηκε από την InvelopMentor Inc ως ένα ανεξάρτητο μοντέλο πρωτοκόλλου για την πρόσβαση σε υπηρεσίες του διαδικτύου, αντικείμενα μεταξύ εφαρμογών και εξυπηρετητές ( Servers ), επικοινωνώντας χρησιμοποιώντας το πρωτόκολλο HTTP.

Το 1999 η έκδοση 1.0 του SOAP έγινε διαθέσιμη δημοσίως και υποστηρίχθηκε από RogueWave, IONA, ObjectSpace, Digital Creations, UserLand, Microsoft, και την DevelopMentor. Αργότερα η έκδοση αυτή στην κατηγορία των W3C τεχνολογιών με πρόσθετες λειτουργίες από την IBM και την LotusCorporation υποστηρίζοντας ένα μεγάλο σύνολο συστημάτων και μοντέλα επικοινωνίας όπως RPC και Messaging.

Σήμερα, η πρόσφατη έκδοση 1.2 του SOAP είναι μέρος της ομάδας πρωτοκόλλων XML W3C, η οποία ηγήθηκε από τις Sun Microsystems, IBM, HP, BEA, Microsoft και Oracle.

### **Άλλα πρωτόκολλα που βασίζονται σε XML**

Όπως έχει προαναφερθεί, η επικοινωνία μεταξύ ενός παρόχου υπηρεσίας και ενός πελάτη πραγματοποιείται με χρήση XML πρωτοκόλλου ( όπως το SOAP ). Τα στηριζόμενα σε XML πρωτόκολλα, έχουν χρησιμοποιηθεί από επιχειρήσεις, έως και σήμερα, όμως μερικά από αυτά δεν έγιναν αποδεκτά από την βιομηχανία για διάφορους λόγους. Μερικά από τα δημοφιλέστερα πρωτόκολλα είναι τα ακόλουθα:

### **CORBA ( Common Object Request Broker Architecture )**

Το πρωτόκολλο αυτό δημιουργήθηκε από την ομάδα OMG ( Object Management Group ) στην προσπάθεια σχεδιασμού ενός πρωτοκόλλου το οποίο είναι ικανό να προσφέρει την δυνατότητα επικοινωνίας μεταξύ αντικειμένων τα οποία ανήκουν σε διαφορετικά λειτουργικά συστήματα και υποστηρίζουν διαφορετικές γλώσσες προγραμματισμού. Βασίζεται στην γλώσσα IDL ( Interface Description Language ) η οποία καθορίζει την διεπαφή ενός αντικειμένου σε μια μορφή που δεν εξαρτάται από γλώσσες προγραμματισμού.

### **COM και DCOM**

Αποτελεί τεχνολογία της Microsoft και χρησιμοποιείται για την επικοινωνία μεταξύ τμημάτων λογισμικού μέσα σε ένα δίκτυο υπολογιστών. Αρχικά λεγόταν COM αλλά αργότερα έγινε η προσθήκη του "D" λόγω της εκτεταμένης χρήσης του DCE ( Distributed Computing Environment ), ενός συστήματος για την απομακρυσμένη κλήση λειτουργιών.

### **Πλεονεκτήματα SOAP**

Είναι ανεξάρτητο από πλατφόρμες και γλώσσες προγραμματισμού: Χρησιμοποιείται ευρύτατα για την επικοινωνία μεταξύ εφαρμογών οι οποίες δομούνται σε διαφορετικές πλατφόρμες ή είναι γραμμένες σε διαφορετικές γλώσσες προγραμματισμού.

Έχει την δυνατότητα επέκτασης: Δίνει την δυνατότητα σε άλλα πρότυπα να το επεκτείνουν, με την προσθήκη υπηρεσιών οι οποίες λείπουν από αυτό, παρόλο που δεν προσφέρει τόσες πολλές λειτουργίες όσο άλλες τεχνολογίες, όπως οι CORBA και DCOM. Αυτό το πλεονέκτημα του SOAP έναντι των άλλων τεχνολογιών, θεωρείται και το σημαντικότερο καθώς επάνω σε αυτό στηρίζονται πολυάριθμες αναπτυσσόμενες τεχνολογίες των web services που.

Είναι ευέλικτο: Για την επικοινωνία και την μεταφορά δεδομένων ή πληροφοριών χρησιμοποιεί πρότυπα πρωτόκολλα επικοινωνίας όπως το HTTP και το SMTP, οπότε μπορεί να χρησιμοποιηθεί στο διαδίκτυο διαπερνώντας τείχη προστασίας, αλλά χωρίς να θέτει σε περιορισμό την ασφάλεια της υποδομής μιας επιχείρησης.

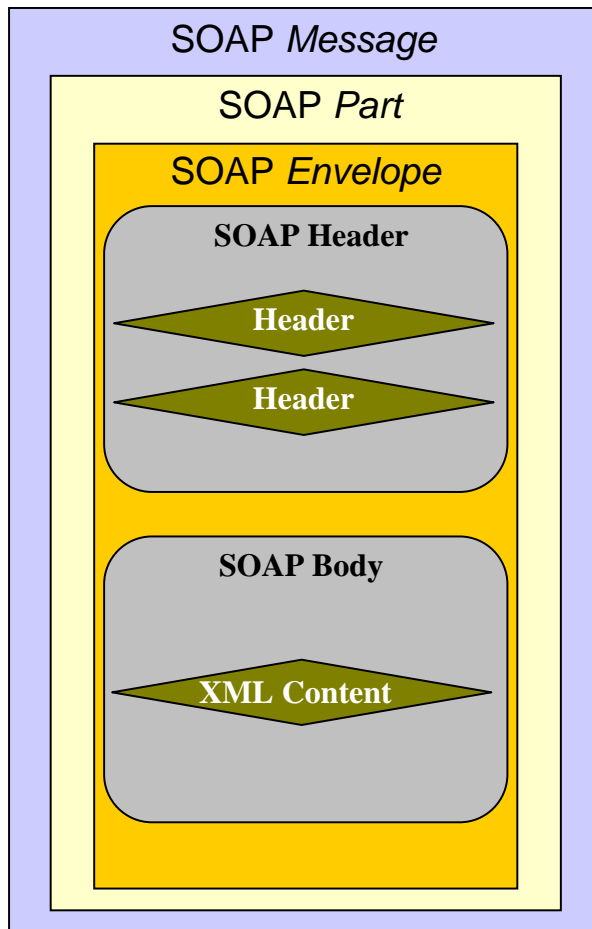
### **3.2 Ανατομία ενός SOAP μηνύματος**

Ξεκινώντας την ανάλυση της ανατομίας ενός SOAP μηνύματος είναι σκόπιμο να αναφερθεί πως τα μηνύματα SOAP διακρίνονται σε δύο κατηγορίες:

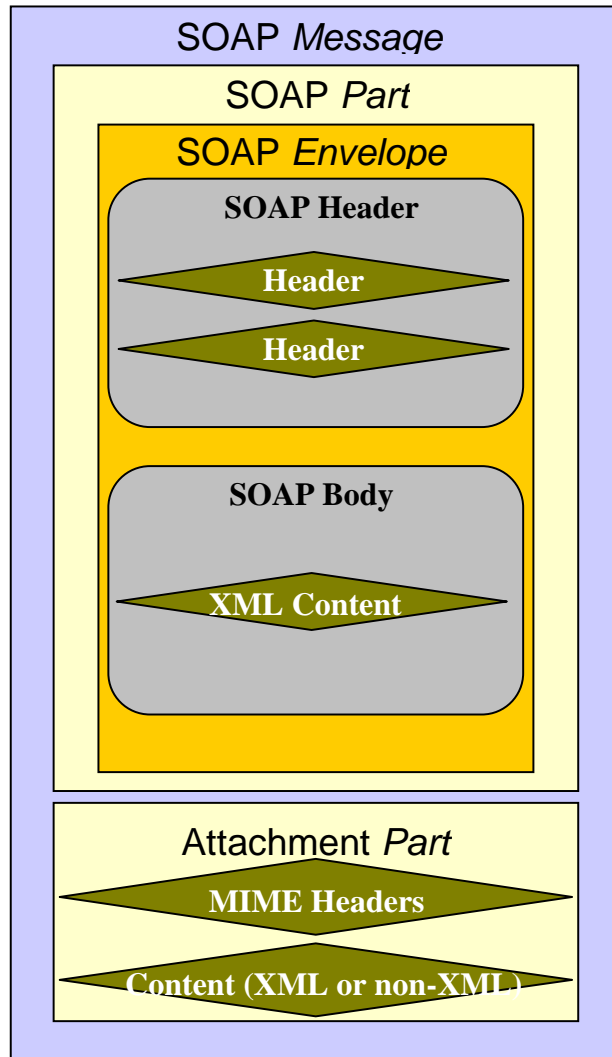
- SOAP μηνύματα χωρίς attachments
- SOAP μηνύματα με attachments

Το SOAP μήνυμα μπορεί να αποτελεί ένα μήνυμα αίτησης υπηρεσίας από ένα πελάτη ή ένα μήνυμα απόκρισης ενός εξυπηρετητή. Τα βασικά μέρη από τα οποία αποτελείται ένα SOAP μήνυμα είναι τα εξής:

- Ένα υποχρεωτικό στοιχείο Envelope από το οποίο το έγγραφο XML αναγνωρίζεται ως μήνυμα SOAP.
- Ένα υποχρεωτικό στοιχείο Header το οποίο παρέχει βοηθητικές λειτουργίες.
- Ένα υποχρεωτικό στοιχείο Body το οποίο περιλαμβάνει την πληροφορία του SOAP μηνύματος
- Ένα προαιρετικό στοιχείο Fault το οποίο παρέχει πληροφορίες για τυχόν σφάλματα που συμβαίνουν κατά την επεξεργασία του SOAP μηνύματος



Εικόνα 3.7 – Μορφή μηνύματος SOAP χωρίς Attachments



Εικόνα 3.8 – Μορφή μηνύματος SOAP με Attachments

## SOAP Envelope

```
<?xml version '1.0' encoding = 'UTF-8' ?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3c.org/2009/XMLSchema-Instance"
xmlns:xsd="http://www.w3c.org/2009/XMLSchema">
SOAP-ENV:
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</SOAP-ENV:Envelope>
```

### Κώδικας 3.1 - Το στοιχείο SOAP Envelope

## SOAP Header

```
<?xml version '1.0' encoding = 'UTF-8' ?>
<SOAP-ENV:Header>
<comer:Transaction
xmlns:comer=http://edu.comer.com/2007/netbooks
SOAP-ENV:mustUnderstand="1">
<keyValue>5</keyValue>
</comer:Transaction>
</SOAP-ENV:Header>
```

### Κώδικας 3.2 - Το στοιχείο SOAP Header

## SOAP Body

```
<?xml version '1.0' encoding = 'UTF-8' ?>
<SOAP-ENV:Body>
<m:GetBookPrice
xmlns:m="http://www.comer.com/networkTCP.library.bookPrice/">
<bookname xsi:type='xsd:string'>
Networks with TCP/IP
</bookname>
</m:GetBookPrice>
</SOAP-ENV:Body>
```

### Κώδικας 3.3 - Το στοιχείο SOAP Body

## SOAP Fault

```
<?xml version '1.0' encoding = 'UTF-8' ?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3c.org/2009/XMLSchema-Instance"
xmlns:xsd="http://www.w3c.org/2009/XMLSchema">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
<faultcode xsi:type="xsi:string">SOAP-ENV:Client</faultcode>
<faultstring xsi:type="xsi:string">
.....
</faultstring>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP:ENV:Envelope>
```

### Κώδικας 3.4 - Το στοιχείο SOAP Fault

## SOAP ENVELOPE

Το SOAP Envelope είναι το στοιχείο που σε ένα έγγραφο XML περικλύει όλα τα υπόλοιπα στοιχεία. Είναι το στοιχείο ρίζα και τοποθετείται στην κορυφή του εγγράφου δηλώνοντας ότι πρόκειται για ένα XML έγγραφο.

Τα βασικά χαρακτηριστικά του SOAP Envelope είναι τα εξής:

- Το όνομα του στοιχείου είναι υποχρεωτικά Envelope.
- Το όνομα του namespace του είναι <http://www.w3c.org/2003/05/soap-envelope>.
- Περιέχει μηδέν ή περισσότερες ιδιότητες ορισμένες με namespace.
- Περιέχει ένα ή περισσότερα στοιχεία παιδιά SOAP Header και SOAP Body.

## SOAP HEADER

Το SOAP Header παρέχει τον μηχανισμό για την επέκταση ενός SOAP μηνύματος με αποκεντρωμένο και δομημένο τρόπο.

- Το όνομα του στοιχείου είναι υποχρεωτικά Header.
- Το όνομα του namespace του είναι <http://www.w3c.org/2003/05/soap-envelope>.
- Περιέχει μηδέν ή περισσότερες ιδιότητες ορισμένες με namespace.
- Περιέχει μηδέν ή περισσότερα στοιχεία - παιδιά ορισμένα με namespace.
- Κάθε στοιχείο – παιδί του SOAP Header ονομάζεται SOAP Header Block.

Το πρωτόκολλο SOAP ορίζει δύο βασικές ιδιότητες για το στοιχείο Header:

- *Actor Attribute*

Το SOAP ορίζει μια διαδρομή μηνύματος ως μια λίστα από κόμβους υπηρεσίας. Κάθε κόμβος αφού εκτελέσει μια λειτουργία, στην συνέχεια προωθεί το μήνυμα στους υπόλοιπους κόμβους. Με την χρήση της ιδιότητας Actor προσδιορίζεται ο παραλήπτης του μηνύματος.

- *MustUnderstand Attribute*

Η ιδιότητα MustUnderstand δηλώνει αν το στοιχείο Header είναι υποχρεωτικό ή προαιρετικό σε ένα SOAP μήνυμα. Είναι μια μεταβλητή που λαμβάνει δύο τιμές, true ή 1 στην περίπτωση που το στοιχείο Header είναι υποχρεωτικό και ο παραλήπτης πρέπει να το κατανοήσει και να το επεξεργαστεί και false ή 0 σε αντίθετη περίπτωση.

## SOAP BODY

Το SOAP Body παρέχει τον μηχανισμό για την αποστολή πληροφοριών στον τελικό παραλήπτη του SOAP μηνύματος.

Τα βασικά χαρακτηριστικά του SOAP Body είναι τα εξής:

- Το όνομα του στοιχείου είναι υποχρεωτικά Body.
- Το όνομα του namespace του είναι <http://www.w3c.org/2003/05/soap-envelope>.
- Περιέχει μηδέν ή περισσότερες ιδιότητες ορισμένες με namespace.
- Περιέχει μηδέν ή περισσότερα στοιχεία - παιδιά ορισμένα με namespace.



- Μπορεί να περιέχει μηδέν ή περισσότερους κόμβους – παιδιά χαρακτήρων.
- Κάθε στοιχείο – παιδί του SOAP Header ονομάζεται SOAP Body Child Element

### **SOAP FAULT**

Το SOAP Fault παρέχει πληροφορίες για σφάλματα μέσα ένα SOAP μήνυμα.

Τα βασικά χαρακτηριστικά του SOAP Fault είναι τα εξής:

- Το όνομα του στοιχείου είναι υποχρεωτικά Fault.
- Το όνομα του namespace του είναι <http://www.w3c.org/2003/05/soap-envelope>.
- Πρέπει να περιέχει ένα ή περισσότερα στοιχεία παιδιά με την εξής σειρά:
  - Ένα υποχρεωτικό στοιχείο Code.
  - Ένα υποχρεωτικό στοιχείο Reason.
  - Ένα προαιρετικό στοιχείο Node.
  - Ένα προαιρετικό στοιχείο Role.
  - Ένα προαιρετικό στοιχείο Detail.
- Για να αναγνωρισθεί ότι ένα SOAP μήνυμα περιέχει πληροφορίες σφάλματος, θα πρέπει να περιλαμβάνει ένα στοιχείο Fault σαν το μοναδικό στοιχείο – παιδί του SOAP Body.

### 3.3 Κωδικοποίηση δεδομένων SOAP

Το πρωτόκολλο SOAP περιλαμβάνει ένα δομημένο σύνολο κανόνων για την κωδικοποίηση τύπων δεδομένων. Αυτό επιτρέπει την δήλωση εξειδικευμένων τύπων δεδομένων σε ένα μήνυμα SOAP, τύποι όπως, ακέραιοι, δεκαδικοί ή πίνακες. Η κωδικοποίηση που γίνεται από το SOAP δεν είναι ορατή στον χρήστη, αλλά μπορεί ο χρήστης να επιλέξει μεταξύ των διαφορετικών σχημάτων κωδικοποίησης που παρέχει το W3C.

Δύο έννοιες που παίζουν σημαντικό ρόλο στην διαδικασία της κωδικοποίησης SOAP είναι οι όροι "value" και "accessor". Ο όρος value αναφέρεται είτε σε απλές μονάδες δεδομένων, είτε σε σύνολα μονάδων δεδομένων. Αυτά μπορεί να είναι το όνομα ενός ατόμου, η τιμή ενός βιβλίου, η θερμοκρασία, κλπ.

Ο όρος accessor αναφέρεται σε ένα στοιχείο που περιέχει ή επιτρέπει την πρόσβαση σε μία τιμή μονάδας.

<email> info@techedu.gr </email>

#### Κώδικας 3.5 – Απλοί τύποι δεδομένων SOAP

Μία σύνθετη μορφή του στοιχείου value παριστάνει μία σύνθεση από δύο ή περισσότερα στοιχεία accessor που δομούνται σε μία ιεραρχημένη δομή, ως παιδιά ενός απλού accessor.

```
<location>
  <country> ... </country>
  <city> ... </city>
  <address> ... </address>
</location>
```

#### Κώδικας 3.6 – Σύνθεση τύπων δεδομένων SOAP

Υπάρχουν δύο είδη σύνθετων τιμών, οι δομές δεδομένων και οι πίνακες. Μια δομή είναι μια σύνθετη τιμή στην οποία κάθε accessor έχει διαφορετικό όνομα. Ο πίνακας είναι μία σύνθετη τιμή, στην οποία ο accessor έχει το ίδιο όνομα.

```
<location>
  <country> ... </country>
  <city> ... </city>
  <address> ... </address>
</location>
```

#### Κώδικας 3.7 – Δομές δεδομένων, ένας σύνθετος τύπος στο SOAP

```
<booklist>
  <booktitle> ... </booktitle>
```

```
<booktitle> ... </booktitle>  
</booklist>
```

### **Κώδικας 3.8 – Πίνακες, ένας σύνθετος τύπος στο SOAP**

Έχοντας ως κριτήριο την χρήση ή όχι των ιδιοτήτων id και href οι accessor διακρίνονται σε δύο κατηγορίες: απλής-αναφοράς( single-referenced accessors) ή πολλαπλής-αναφοράς( multireferenced accessors ).

Ο single-referenced accessor δεν έχει άλλο αναγνωριστικό εκτός από το στοιχείο-παιδί ενός στοιχείου-γονέα.

Ο multireferenced accessor χρησιμοποιεί την ιδιότητα id για να δώσει ένα αναγνωριστικό σε ένα στοιχείο. Οι υπόλοιποι accessors χρησιμοποιούν την ιδιότητα href για να προσπελάσουν ή να αναφερθούν στο στοιχείο αυτό.

### **3.4 Τύποι δεδομένων SOAP**

Οι τύποι δεδομένων που υποστηρίζονται από την μέθοδο κωδικοποίησης SOAP είναι τύποι δεδομένων που ορίζονται από το XML Schema.

Η κωδικοποίηση SOAP παρέχει δύο εναλλακτικούς τρόπους παρουσίασης τύπων δεδομένων μέσα στο σώμα ενός SOAP μηνύματος.

### **3.5 Διαδικασία ανταλλαγής SOAP μηνυμάτων**

Για την πραγματοποίηση της ανταλλαγής μηνυμάτων SOAP ακολουθούνται τα εξής βήματα:

- Δημιουργία της σύνδεσης SOAP
- Δημιουργία του μηνύματος SOAP
- Προσθήκη δεδομένων στο μήνυμα
- Αποστολή του μηνύματος
- Λήψη του μηνύματος

### **3.6 Χειρισμός σφαλμάτων με SOAP**

Τα σφάλματα που συμβαίνουν κατά την ανταλλαγή μηνυμάτων SOAP χειριζόμαστε με τα Fault Envelope. Στην περίπτωση που προκύψει ένα σφάλμα την στιγμή που ο server επεξεργάζεται ένα SOAP μήνυμα, δημιουργεί ένα SOAP Fault και το στέλνει πίσω στον client.

```

<?xml version='1.0' encoding='UTF-8'>
<SOAP-ENV:Envelope>
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
  <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Server</faultcode>
    <faultstring>This is a SOAP message fault</faultstring>
    <faultactor>/soap/servlet/rpcrouter/</faultactor>
    <detail>
      <stackTrace>[SOAPException:faultcode=SOAP-ENV:Server;
        message=This is a SOAP message fault]
        at InfoGradesService.getGrades(InfoGradesService.java:13)
        at java.lang.reflect.Method.invoke(NativeMethod)
        at org.apache.soap.server.RPCRouter.invoke
        (RPCRouter.java:163)
        at java.lang.Thread.run(Thread.java:500)
      </stackTrace>
    </detail>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Κώδικας 3.9 – Ένα μήνυμα σφάλματος του SOAP

## SOAP Fault, faultcode, faultstring, detail και faultactor

Το στοιχείο SOAP Fault είναι ένα ειδικό στοιχείο το οποίο πρέπει να εμφανίζεται ως στοιχείο-παιδί μέσα στο στοιχείο SOAP Body. Τα στοιχεία <faultcode> και <faultstring> πρέπει να υπάρχουν απαραίτητα. Τα στοιχεία <faultactor> και <detail> είναι προαιρετικά. Παρακάτω παρουσιάζεται ένας πίνακας πιθανών τιμών του στοιχείου <faultcode> και η περιγραφή τους.

Faultcode	Περιγραφή
Client	Αναφέρεται σε λανθασμένο ή κακώς διαμορφωμένο περιεχόμενο μηνύματος που δημιουργήθηκε από τον Client. Η επαναποστολή αυτών των δεδομένων θα οδηγήσει πιθανώς στο ίδιο σφάλμα.
Server	Το περιεχόμενο που στέλνεται από τον Client είναι αποδεκτό από τον χειριστή του SOAP μηνύματος αλλά δεν είναι σε θέση να το επεξεργαστεί. Πιθανώς λόγω μιας υπηρεσίας που δεν μπορεί να υποστηρίξει. Η επαναποστολή του ίδιου μηνύματος είναι δυνατό να έχει επιτυχή έκβαση.
DataEncodingUnknown	Αναφέρεται σε δεδομένα μη αναγνωρίσιμα ή σε κωδικοποίησή τους που δεν υποστηρίζεται.
VersionMismatch	Ο κόμβος SOAP που χειρίζεται το αίτημα αντιμετώπισε πρόβλημα ασυμβατότητας στην έκδοση του πρωτοκόλλου
MustUnderstand	Ένα στοιχείο-παιδί του SOAP Header περιέχει την ιδιότητα MustUnderstand η οποία λαμβάνει την τιμή 1 ή true σε περίπτωση επιτυχούς λήψης του μηνύματος.
DTDNotSupported	Συμπεριλαμβάνεται στην έκδοση 1.2 του SOAP. Το σφάλμα αναφέρεται στην αδυναμία υποστήριξης ενός DTD ( Document Type Definition ).
ProcedureNotPresent	Ο server αδυνατεί να εντοπίσει μια συγκεκριμένη λειτουργία.
BadArguments	Ο server δεν μπορεί να πάρει τα ορίσματα για την κλήση μιας λειτουργίας.

**Πίνακας 3.1 – Στοιχεία SOAP Fault**

Τα στοιχεία Body και Fault μπορούν να προσαρμοστούν στον χώρο ονομάτων ( namespace ) του Envelope χρησιμοποιώντας την σύνταξη:

<SOAP:ENV:Body> και <SOAP-ENV-Fault>

Το στοιχείο <faultcode> χρησιμοποιεί το τοπικό namespace και η τιμή της ιδιότητας <faultcode> αποτελεί ένα όνομα συμβατό με το namespace του Envelope.

Υποστοιχεία Fault	Περιγραφή
faultcode	Ο κώδικας για τον προσδιορισμό του σφάλματος
faultstring	Μια αναγνώσιμη επεξήγηση του σφάλματος
faultactor	Πληροφορίες για το ποιός προκάλεσε το σφάλμα
detail	Περιέχει πληροφορίες για το σφάλμα που σχετίζονται με το στοιχείο Body

---

### Πίνακας 3.2 – Υποστοιχεία SOAP Fault

Ο χειρισμός σφαλμάτων γίνεται με το SOAP Fault και τον χειριστή εξαιρέσεων που ορίζεται στην υπηρεσία InfoGradesService:

```
public class InfoGradesService{
    public double getGrades(String Id)throws org.apache.soap.SOAPException{
        double grade = (Double)product.get(id);
        if(Id.equalsIgnoreCase("fail"))
            throw new org.apache.soap.SOAPException
                (org.apache.soap.Constants.FAULT_CODE_SERVER,
                 "This is a SOAP message Fault");
        Return grade.doubleValue();
    }
}
```

### Κώδικας 3.10 – Ένα απλό παράδειγμα χειρισμού σφαλμάτων με SOAP Faults

Αυτός ο κώδικας επιστρέφει ένα μήνυμα σφάλματος στον αποστολέα του μηνύματος.

### Η ιδιότητα mustUnderstand και ο ρόλος της στα σφάλματα SOAP

Η ιδιότητα mustUnderstand ( ή misunderstood ) τοποθετείται μέσα στο στοιχείο κεφαλίδα. Όταν έχει τιμή 1 ή true, σημαίνει πως το περιεχόμενο του μηνύματος μπορεί να αναγνωρισθεί από τον παραλήπτη του. Σε περίπτωση που ο χειριστής SOAP δεν μπορεί να αναγνωρίσει, ούτε να επεξεργαστεί το στοιχείο header, δημιουργεί σφάλμα. Ο κώδικας που πρέπει να προστεθεί στον SOAP client είναι ο εξής:

```

org.w3c.dom.Element headerElement =
    Doc.createElementNS(URI,"jaws:MessageHeader");
headerElement.setAttributeNS(URI,"SOAP-ENV:mustUnderstand","1");
org.w3c.dom.Element element = doc.createElement("From");
org.w3c.dom.Text text = doc.createTextNode("Me");

```

### Κώδικας 3.11 – Η ιδιότητα mustUnderstand στα SOAP Faults

Αυτός ο κώδικας δημιουργεί το παρακάτω SOAP Envelope:

```

<?xml version='1.0' encoding='UTF-8'>
<SOAP-ENV:Envelope>
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Header>
  <jaws:MessageHeader xmlns:jaws="urn:http://localhost:5056/examplesSOAP"
  SOAP-ENV:MustUnderstand="1">
  <FROM>Me</FROM>
  <TO>You</TO>
  .
  .
  .
  </jaws:MessageHeader>
</SOAP-ENV:Header>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Κώδικας 3.12 – Μορφή του SOAP Envelope με τον ορισμό της ιδιότητας mustUnderstand

Σε περίπτωση που ο server δεν μπορεί να καταλάβει το στοιχείο <MessageHeader> τότε επιστρέφει το εξής SOAP μήνυμα σφάλματος:

```
<?xml version='1.0' encoding='UTF-8'>
<SOAP-ENV:Envelope>
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
  <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Mustunderstand</faultcode>
    <faultstring>Unsupported header:jawsMessageHeader</faultstring>
    <faultactor>/examplesSOAP/servletExamples/servlet/FaultServlet</faultactor>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### **Κώδικας 3.13 – Μήνυμα σφάλματος λόγω αδυναμίας αναγνώρισης του στοιχείου MessageHeader**

Ο κώδικας που χρησιμοποιείται για τον χειρισμό του σφάλματος, ορίζεται μέσα στο παρακάτω servlet ( *Για τα servlets θα μιλήσουμε αναλυτικά σε επόμενο κεφάλαιο* ):



```

public class FaultServlet extends HttpServlet{
    .
    public void doPost(HttpServletRequest req, HttpServletResponse res)throws
        IOException, ServletException{

        Header header = env.getHeader();
        java.util.Vector headerEntries = header.getHeaderEntries();
        screenwriter.write("\nHeader\n");

        for(java.util.Enumeration e = headerEntries.elements();e.hasMoreElements();)
        {
            org.w3c.dom.Elementelement=(org.w3c.dom.Element)e.nextElement(
);
            org.w3c.apache.soap.util.xml.DOM2WriterserializeAsXML(
                (org.w3c.dom.Node)element,screenwriter);

            String mustUnderstand = element.getAttributeNS(Constands.
                NS_URI_SOAP_ENV,"mustUnderstand");
            screenWriter.write("\nmustUnderstand=>" +mustUnderstand);
            String tagName = element.getTagName();
            screenWriter.write("\ntagName=>" +tagName);
            if(tagName.equalsIgnoreCase("\nOnlyUnderstandThis"))
            {
                screenwriter.write("Unsupported Header:" +tagName);
                SOAPException exception = new SOAPException(
                    Constands.FAULT_CODE_MUST_UNDERSTAND,
                    "Unsupported header = " +tagName);

                Fault fault = new Fault(exception);
                fault.setFaultActorURI(req.getRequestURI());
                String respEncStyle = Constand.NS_URI_SOAP_ENC;
            }
            org.apache.soap.rpc.Response soapResponse = new org.apache.soap.rpc.
                Response(null,null,fault,null,null,resEncStyle,null);

            Envelope faultEnvelope = soapResponse.buildEnvelope();
            org.apache.soap.encoding.SOAPMappingRegistry map_registry =
                new org.apache.soap.encoding.SOAPMappingRegistry();
            PrintWriter out = res.getWriter();
            faultEnvelope.marshall(out,map_registry,soapResponse.getSOAPContext());
            res.setContentType(req.getContentType());
            res.setStatus(res.SC_INTERNAL_SERVER_ERROR);
        }
    }
}

```

**Κώδικας 3.14 – Ένα JAVA Servlet για τον χειρισμό σφαλμάτων SOAP**

## 4 ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ XML – RPC

### 4.1 Εισαγωγή

Το XML–RPC παρέχει έναν μηχανισμό που βασίζεται στην XML και το πρωτόκολλο επικοινωνίας HTTP για την πραγματοποίηση κλήσεων μεθόδων ή συναρτήσεων μέσω ενός δικτύου. Προσφέρει ένα απλό και εύκολο στην χρήση σύνολο εργαλείων για την διασύνδεση ανόμοιων συστημάτων και την ανταλλαγή πληροφοριών.

### 4.2 Ορισμός

Το XML–RPC επιτρέπει στα προγράμματα να πραγματοποιούν κλήσεις συναρτήσεων ή διαδικασιών μέσω ενός δικτύου. Χρησιμοποιεί το πρωτόκολλο HTTP για την μεταφορά πληροφοριών μεταξύ ενός υπολογιστή πελάτη και ενός εξυπηρετητή περιγράφοντας την φύση των αιτημάτων και αποκρίσεων με την χρήση των κανόνων γραμματικής της XML. Οι πελάτες προσδιορίζουν ένα όνομα μεθόδου και τις παραμέτρους σε μία XML αίτηση ( XML Request ) και ο server επιστρέφει είτε ένα μήνυμα σφάλματος, είτε μία απάντηση ( XML Response ). Οι τύποι των παραμέτρων XML–RPC, περιλαμβάνουν απλούς τύπους, αλλά και σύνθετους, όπως δομές δεδομένων και πίνακες.

### 4.3 Γιατί χρησιμοποιείται

Σε ένα περιβάλλον προγραμματισμού στο οποίο κατέχουν σημαντική θέση τα αντικείμενα, το XML–RPC φαίνεται να έχει περιορισμούς για την ανάπτυξη πολλών εφαρμογών. Παρά το γεγονός αυτό, πλεονεκτεί στο ότι δίνει την δυνατότητα στους προγραμματιστές να συνδέουν με εύκολο τρόπο συστήματα διαφορετικών τύπων. Οι αναλυτές συστημάτων και οι προγραμματιστές, για την κατασκευή κατανεμημένων συστημάτων, χρησιμοποιούν το XML–RPC ως συνδετικό κώδικα, ο οποίος συνδέει ανόμοια τμήματα μέσα σε ένα ιδιωτικό δίκτυο συστημάτων. Με το XML–RPC οι προγραμματιστές εστιάζουν στην διεπαφή που παρεμβάλλεται μεταξύ των συστημάτων και όχι στο πρωτόκολλο που χρησιμοποιείται για την διασύνδεση αυτών των διεπαφών μεταξύ τους. Ακόμη, οι προγραμματιστές που δημιουργούν υπηρεσίες για δημόσια χρήση, έχουν την δυνατότητα, μέσω της χρήσης του XML–RPC να ορίσουν και να υλοποιήσουν ένα interface στην γλώσσα που επιθυμούν. Από την μία, οι πελάτες μπορούν μέσω μηνυμάτων XML–RPC να καλέσουν και να χρησιμοποιήσουν τις υπηρεσίες αυτές μέσω του διαδικτύου και από την άλλη οι προγραμματιστές έχουν την δυνατότητα να χρησιμοποιήσουν αυτές τις υπηρεσίες σε άλλες εφαρμογές που αναπτύσσουν.

### 4.4 Στοιχεία αρχιτεκτονικής XML – RPC

- Μοντέλο Δεδομένων ( Data Model )

Ένα σύνολο τύπων που περιλαμβάνει τους τύπους των παραμέτρων που ορίζονται στην δήλωση των μεθόδων, τους τύπους των τιμών που επιστρέφονται και τους τύπους των μηνυμάτων σφαλμάτων.

- Δομή Αιτήσεων ( Request Structure )

Ένα αίτημα HTTP που περιλαμβάνει πληροφορίες περί των παραμέτρων και των μεθόδων.

- Δομή Αποκρίσεων ( Response Structure )

Μία απόκριση HTTP που περιλαμβάνει πληροφορίες περί των επιστρεφόμενων τιμών και των μηνυμάτων σφαλμάτων.

## 4.5 Μοντέλο Δεδομένων XML – RPC

Στο σύνολο κανόνων σύνταξης XML–RPC ορίζονται έξι βασικοί τύποι και δύο σύνθετους τύπους που αποτελούν συνδυασμό απλών τύπων. Παρά το γεγονός ότι το σύνολο τύπων της τεχνολογίας XML–RPC είναι αρκετά περιορισμένο, ωστόσο καλύπτει ένα ευρύ φάσμα πληροφοριών και καλύπτει τις απαιτήσεις του κατανεμημένου προγραμματισμού. Όλοι οι βασικοί τύποι ορίζονται από στοιχεία XML των οποίων το περιεχόμενο ορίζει τις τιμές. Για παράδειγμα, αν θέλουμε να ορίσουμε μία συμβολοσειρά με τιμή “Hello” θα ακολουθήσουμε τον εξής κανόνα σύνταξης:

```
<string>Hello</string>
```

Κώδικας 4.15 – Η μορφή των δεδομένων XML–RPC

Τύποι Java	Τύποι XML	Περιγραφή
Integer	l4, int	Μία 32-μπιτη, non-null ακέραια τιμή.
Boolean	boolean	Μία τιμή true/false ή 0/1
String	string	Μία non-null συμβολοσειρά
Double	double	Μία 64-μπιτη, non-null, δεκαδική τιμή
Java.util.Date	dateTime.iso8601	Μία τιμή που υπακούει στην κωδικοποίηση iso8601 και δίνει πληροφορίες ώρας.
byte[]	base64	Ένας πίνακας από bytes.
Java.util.Map	struct	Ζευγάρι δύο τιμών, μία συμβολοσειρά και μία τιμή οποιουδήποτε τύπου.
Object[] Java.util.List	array	Ένας πίνακας δεδομένων, τα στοιχεία του οποίου μπορεί να είναι οποιουδήποτε τύπου, συμπεριλαμβανομένου και του πίνακα.

Πίνακας 4.1 – Τύποι Δεδομένων XML–RPC

Όταν η ιδιότητα `enabledForExtensions` ορίζεται `true`, τότε ισχύουν και οι παρακάτω τύποι δεδομένων:

Τύποι Java	Τύποι XML	Περιγραφή
None	ex:nil	Μία κενή τιμή.
Byte	ex:i1	Μία 8-μπιτη, non-null, ακέραια τιμή
Float	ex:float	Μία non-null, δεκαδική τιμή.
Long	ex:i8	Μία 64-μπιτη, non-null, ακέραια τιμή.
org.w3c.dom.Node	ex:dom	Ένας dom κόμβος που μεταφέρεται ως ένα ενσωματωμένο κομμάτι της XML.
Short	ex:i2	Μία 16-μπιτη, non-null, ακέραια τιμή.
java.io.Serializable	ex:serializable	Ένα αντικείμενο που μεταφέρεται ως πίνακας από bytes της μορφής base64.
BigDecimal	ex:bigdecimal	Ένας μεγάλος δεκαδικός.
BigInteger	ex:exbiginteger	Ένας μεγάλος ακέραιος.
java.util.Calendar	ex:dateTime	

**Πίνακας 4.2 – Πρόσθετοι τύποι δεδομένων XML-RPC**

Οι βασικοί τύποι δεδομένων του XML-RPC ορίζονται μέσα στα στοιχεία `value` μέσα σε ένα XML έγγραφο. Οι πίνακες, οι οποίοι ανήκουν στους σύνθετους τύπους προσδιορίζονται με το στοιχείο `array` το οποίο περιλαμβάνει ένα στοιχείο `data` το οποίο περιέχει μία λίστα των τιμών. Το στοιχείο `array` εσωκλείεται μέσα σε ένα στοιχείο `value`. Για παράδειγμα, ο παρακάτω πίνακας αποτελείται από τέσσερις αριθμούς `double` που αποτελούν τις διαστάσεις ενός παραλληλογράμμου.

```

<value>
  <array>
    <data>
      <value><double>12.3</double></value>
      <value><double>25.6</double></value>
      <value><double>12.3</double></value>
      <value><double>25.6</double></value>
    </data>
  </array>
</value>

```

#### Κώδικας 4.16 – Απλοί τύποι δεδομένων XML-RPC

Ένα στοιχείο `array` είναι δυνατό να περιλαμβάνει και διαφορετικούς μεταξύ τους τύπους, όπως για παράδειγμα ο πίνακας στον παρακάτω κώδικα:

```

<value>
  <array>
    <data>
      <value><string>Hello</string></value>
      <value><double>25.6</double></value>
      <value><int>3</int></value>
      <value><boolean>0</boolean></value>
    </data>
  </array>
</value>

```

#### Κώδικας 4.17 – Σύνθετοι τύποι δεδομένων XML-RPC, μονοδιάστατοι πίνακες

Επίσης, υποστηρίζεται η δημιουργία πολυδιάστατων πινάκων, όπως στο παράδειγμα:

```

<value>
  <array>
    <data>
      <value><string>Hello</string></value>
      <value><double>25.6</double></value>
      <value><int>3</int></value>
      <value><boolean>0</boolean></value>
    </data>
  </array>
</value>

```

#### Κώδικας 4.18 – Σύνθετοι τύποι δεδομένων XML-RPC – πολυδιάστατοι πίνακες

Οι δομές δεδομένων περιέχουν μη ταξινομημένο περιεχόμενο και έχουν ως αναγνωριστικό το όνομα. Τα ονόματα πρέπει να είναι συμβολοσειρές που σημαίνει πως πρέπει να δηλώνονται μέσα στα σε στοιχεία string. Κάθε δομή δηλώνεται με το στοιχείο struct και κάθε στοιχείο struct περιέχει μία λίστα από στοιχεία member. Κάθε στοιχείο member περιέχει ένα στοιχείο name και ένα value. Η τοποθέτηση των στοιχείων member δεν είναι υποχρεωτική.

```
<value>
  <struct>
    <member>
      <name>BookTitle</name>
      <value>Network Systems Designed Using Network Processors</value>
    </member>
    <member>
      <name>Writer</name>
      <value>Daglas Comer</value>
    </member>
    <member>
      <name>ISBN</name>
      <value>0-13-41792-4</value>
    </member>
  </struct>
</value>
```

#### **Κώδικας 4.19 – Σύνθετοι τύποι δεδομένων XML-RPC, δομές δεδομένων**

Οι δομές δεδομένων μπορούν να περιέχουν άλλες δομές, όπως επίσης και πίνακες:

```

<value>
  < struct >
    <member>
      <name>Tutorial Books</name>
      <value>
        <struct>
          <member>
            <name>Title</name>
            <value>Programming web services with XML-RPC</value>
          </member>
          <member>
            <name>Writer</name>
            <value>Chris Boar</value>
          </member>
        </struct>
      </value>
    </member>
    <member>
      <name>AsimpleMessage</name>
      <value>
        <array>
          <data>
            <value><string>This is</string></value>
            <value><string>a multiple struct</string></value>
          </data>
        </array>
      </value>
    </member>
  </struct>
</value>

```

**Κώδικας 4.20 – Σύνθετες δομές δεδομένων XML-RPC**

## **4.6 Δομή αιτήσεων XML – RPC**

Οι αιτήσεις XML – RPC αποτελούν συνδυασμό, στοιχείων XML και HTTP headers. Το περιεχόμενο XML καθορίζει τον τύπο δεδομένων για το πέρασμα παραμέτρων σε μεθόδους και προσδιορίζει ποιές μέθοδοι καλούνται, ενώ τα HTTP headers καθορίζουν την μορφή με την οποία οι αιτήσεις θα σταλούν μέσω του Web.

Κάθε αίτηση περιλαμβάνει ένα απλό έγγραφο XML, του οποίου στοιχείο ρίζας είναι το στοιχείο `methodCall`. Κάθε στοιχείο `methodCall` περιέχει ένα στοιχείο `methodName` και ένα στοιχείο `params`. Το στοιχείο `methodName` προσδιορίζει το όνομα της μεθόδου που καλείται, ενώ το στοιχείο `params` περιλαμβάνει μία λίστα με όλες τις παραμέτρους της μεθόδου, καθώς και τις τιμές τους.

```
<xml version "1.1"?>
<methodCall>
  <methodName>sayHello</methodName>
  <params>
    <param>
      <value><string>Hello</string></value>
    </param>
  </params>
</methodCall>
```

**Κώδικας 4.21 – Μορφή αίτησης XML-RPC**

#### **4.7 Δομή αποκρίσεων XML – RPC**

Οι αποκρίσεις XML – RPC είναι όμοιες με τις αιτήσεις, παρουσιάζοντας όμως και κάποιες διαφοροποιήσεις. Όταν η απόκριση είναι επιτυχής, που σημαίνει ότι, η μέθοδος έχει εντοπιστεί, έχει εκτελεστεί σωστά η λειτουργία της και έχουν επιστραφεί αποτελέσματα, τότε αποστέλλεται ένα μήνυμα όμοιο με την αίτηση, με την διαφορά ότι το στοιχείο `methodCall` αντικαθίσταται από το στοιχείο `methodResponse` και δεν υπάρχει στοιχείο `methodName`.

```
<xml version "1.1"?>
<methodResponse>
  <params>
    <param>
      <value><string>Hello</string></value>
    </param>
  </params>
</methodResponse>
```

**Κώδικας 4.22 – Μορφή απόκρισης XML-RPC**

#### **4.8 Ανάπτυξη εφαρμογών με το XML – RPC**

Η ανάπτυξη μιας εφαρμογής απαιτεί την εισαγωγή στο πρόγραμμα μιας βιβλιοθήκης XML – RPC και πραγματοποίηση κλήσεων μεθόδων μέσω αυτής της βιβλιοθήκης. Είναι κανόνας πως οι μέθοδοι αυτοί θα λειτουργήσουν σωστά με το XML – RPC εφόσον ο κώδικας της εφαρμογής περιλαμβάνει μόνο τους βασικούς τύπους που το XML – RPC υποστηρίζει.



## 5 WSDL, Η ΠΕΡΙΓΡΑΦΙΚΗ ΓΛΩΣΣΑ ΤΩΝ WEB SERVICES

### 5.1 Εισαγωγή – Ορισμός

Η WSDL είναι μία πρότυπη γλώσσα η οποία βασίζεται στην XML και χρησιμοποιείται για την περιγραφή μιας υπηρεσίας Ιστού και πιο συγκεκριμένα, περιγράφει τί κάνει μια υπηρεσία, με ποιό τρόπο γίνεται η κλήση των λειτουργιών της και πώς μπορούμε να έχουμε πρόσβαση στην υπηρεσία αυτή. Η ανάπτυξή της αποτέλεσε την λύση στο πρόβλημα της επικοινωνίας δύο μηχανών στο διαδίκτυο παρέχοντας μία πρότυπη μέθοδο επικοινωνίας. Περιλαμβάνει ένα σύνολο κανόνων για τον προσδιορισμό μιας υπηρεσίας, του άκρου επικοινωνίας όπου τοποθετείται και της κωδικοποίησης που χρησιμοποιείται στα εισερχόμενα και εξερχόμενα μηνύματα της υπηρεσίας.

### 5.2 Η δομή ενός απλού WSDL εγγράφου.

Παρακάτω παροσιάζεται μέρος από ένα απλό WSDL έγγραφο με σκοπό να προσδιοριστούν τα βασικά του στοιχεία. Κάποια από αυτά τα στοιχεία ( import, message, portType, operation, service ) μπορούν να εμφανίζονται περισσότερες από μία φορά σε ένα έγγραφο.

```
<definitions>
  <import>
  <types>
    {definition of types... }
  </types>
  <message>
    {definition of a message... }
  </message>
  <portType>
    {definition of a port... }
  </portType>
  <binding>
    {definition of a binding... }
  </binding>
  <service>
    {definition of a service... }
  </service>
</definitions>
```

Κώδικας 5.23 – Η δομή ενός απλού WSDL εγγράφου

**definitions**

Αποτελεί το στοιχείο ρίζα ενός WSDL εγγράφου. Ορίζει το όνομα της υπηρεσίας και περιέχει τα στοιχεία που περιγράφουν την υπηρεσία.

**import**

Χρησιμοποιείται για τη εισαγωγή άλλων WSDL εγγράφων ή XML σχημάτων μέσα σε ένα έγγραφο. Με τον τρόπο αυτό παρέχεται μεγαλύτερη μοντελοποίηση στα έγγραφα. Το αρνητικό σημείο είναι πως δεν υποστηρίζουν όλα τα WSDL εργαλεία το στοιχείο import.

**types**

Περιγράφει όλους τους τύπους δεδομένων που χρησιμοποιούνται από client και server. Η WSDL δεν ακολουθεί ένα συγκεκριμένο σύστημα τύπων δεδομένων, αλλά χρησιμοποιεί στα έγγραφα της ορίσματα του XML σχήματος. Όταν μια υπηρεσία χρησιμοποιεί ένα σχήμα το οποίο περιλαμβάνει απλούς τύπους δεδομένων, όπως ακεραίους και συμβολοσειρές, τότε το στοιχείο Types δεν είναι υποχρεωτικό να περιέχεται μέσα στο WSDL έγγραφο.

**message**

Περιγράφει ένα μήνυμα μονής κατεύθυνσης το οποίο μπορεί να αποτελεί ένα ερώτημα ( request ) ή μια απάντηση ( response ). Προσδιορίζει το όνομα του μηνύματος και περιέχει μηδέν ή περισσότερα στοιχεία που αποτελούν παραμέτρους ή επιστρεφόμενες τιμές του μηνύματος.

**portType**

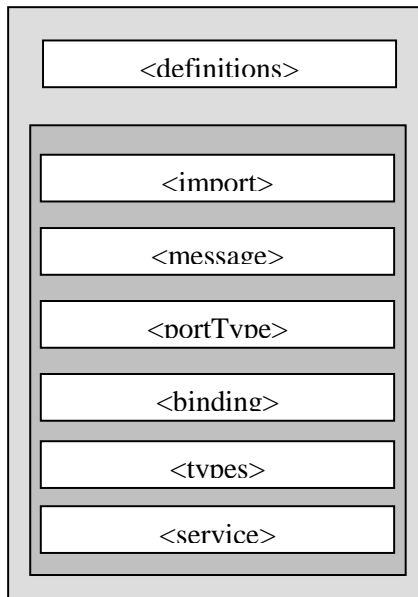
Το στοιχείο αυτό συνδυάζει πολλαπλά στοιχεία message σε μια μονόδρομη λειτουργία. Για παράδειγμα μπορεί να συνδυάσει ένα μήνυμα request και ένα μήνυμα response συνιστώντας μία λειτουργία request/response μια μορφή που χρησιμοποιείται από το SOAP πρωτόκολλο.

**binding**

Δίνει τα τεχνικά στοιχεία που προσδιορίζουν το πώς υλοποιείται ένα web service. Για το σκοπό αυτό, χρησιμοποιεί την περιγραφική γλώσσα WSDL για την περιγραφή των SOAP υπηρεσιών.

**service**

Προσδιορίζει την διεύθυνση όπου μπορούμε να βρούμε μια υπηρεσία και πιο συγκεκριμένα ορίζει μια URL διεύθυνση ώστε να έχουμε πρόσβαση σε αυτή.



*Εικόνα 5.9 – Τα επίπεδα των στοιχείων σε ένα WSDL έγγραφο*

### **Ένα παράδειγμα WSDL εγγράφου**

Ακολουθεί μια επισκόπηση των βασικών στοιχείων που συνιστούν ένα WSDL έγγραφο με βάση έγγραφο που περιγράφει μια υπηρεσία ανταλλαγής μηνυμάτων για την ενημέρωση της βαθμολογίας μαθήματος.

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions name="InfoGradesService"
target namespace="http://www.it.teithe.gr/StudentGrades/wsd/InfoGradesService.wsd/
xmlns=http://schemas.xmlsoap.org/wsd/
xmlns:soap=http://schemas.xmlsoap.org/wsd/soap/
xmlns:tns=http://www.it.teithe.gr/StudentGrades/wsd/InfoGradesService.wsd/
xmlns:xsd=http://www.w3c.org/2008/XMLSchema>
  <message name="AskLessonGrade">
    <part name="ask" type="xsd:string"/>
  </message>
  <message name="ReportLessonGrade">
    <part name="report" type="xsd:float"/>
  </message>
  <portType name="Grade_PortType">
    <operation name="reportGrade">
      <input message="reportGradeRequest"/>
      <output message="reportGradeResponse"/>
    </operation>
  </portType>

  <import namespace="http://example.gr/gradesInformation/definitions"
location="http://example.gr/gradesInformation/InfoGradesService.wsd"/>

  <binding name="Grade_Binding" type="tns:Grade_PortType">
    <soap:binding style="rpc"
Transport=http://schemas.xmlsoap.org/soap/http/>
    <operation name="reportGrade">
      <soap:operation soapAction="reportGrade"/>
      <input>
        <soap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:reports:infoGradesService" use="encoded"/>
      </input>
      <output>
        <soap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:reports:infoGradesService" use="encoded"/>
      </output>
    </operation>
  </binding>
  <service name="InfoGrade_Service">
    <documentation>This WSDL file describes the service InfoGradeService</documentation>
    <port binding="tns:InfoGrade_Binding" name="InfoGrade_Port">
      <soap:address location="http://localhost:8084/soap/servletrpcrouter"/>
    </port>
  </service>
</definitions>

```

## Κώδικας 5.24 – Ένα παράδειγμα WSDL εγγράφου

## definitions

Το στοιχείο `definitions` ορίζει ότι το παραπάνω έγγραφο είναι το *InfoGradesService*. Επίσης, προσδιορίζει πολυάριθμα namespaces τα οποία χρησιμοποιούνται στο υπόλοιπο έγγραφο. Η χρήση των namespaces είναι σημαντική για την διαφοροποίηση του των στοιχείων του εγγράφου, επιτρέποντάς του να κάνει αναφορά σε πολυάριθμες, εξωτερικές προδιαγραφές, όπως τις προδιαγραφές WSDL, SOAP και XML Schema.

```
<definitions name="InfoGradesService"
target namespace="http://www.it.teithe.gr/StudentGrades/wsd/InfoGradesService.wsd"
xmlns=http://schemas.xmlsoap.org/wsd/
xmlns:soap=http://schemas.xmlsoap.org/wsd/soap/
xmlns:tns=http://www.it.teithe.gr/StudentGrades/wsd/InfoGradesService.wsd
xmlns:xsd=http://www.w3c.org/2008/XMLSchema>
```

### Κώδικας 5.25 – Το στοιχείο `definitions` σε ένα WSDL έγγραφο

## message

Στο παράδειγμά μας ορίζονται δύο στοιχεία `message`, που αντιστοιχούν σε δύο μηνύματα, ένα μήνυμα υποβολής ερωτήματος *AskLessonGrade* και ένα μήνυμα απόκρισης *ReportLessonGrade*.

```
<message name="AskLessonGrade">
  <part name="ask" type="xsd:string"/>
</message>
<message name="ReportLessonGrade">
  <part name="report" type="xsd:float"/>
</message>
```

### Κώδικας 5.26 – Το στοιχείο `message` σε ένα WSDL έγγραφο

## portType

Το στοιχείο `portType` ορίζει μια απλή μέθοδο ( λειτουργία ) με το όνομα `reportGrade`. Η ίδια η μέθοδος περικλύει επιμέρους στοιχεία, τα *input message* και *output message*.

```

<portType name="Grade_PortType">
  <operation name="reportGrade">
    <input message="reportGradeRequest"/>
    <output message="reportGradeResponse"/>
  </operation>
</portType>

```

### Κώδικας 5.27 – Το στοιχείο portType σε ένα WSDL έγγραφο

#### binding

```

<binding name="Grade_Binding" type="tns:Grade_PortType">
  <soap:binding style="rpc"
  Transport=http://schemas.xmlsoap.org/soap/http/>"
  <operation name="reportGrade">
    <soap:operation soapAction="reportGrade"/>
    <input>
      <soap:body
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="urn:reports:infoGradesService" use="encoded"/>
    </input>
    <output>
      <soap:body
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="urn:reports:infoGradesService" use="encoded"/>
    </output>
  </operation>
</binding>

```

### Κώδικας 5.28 – Το στοιχείο binding σε ένα WSDL έγγραφο

#### service

Το στοιχείο service την τοποθεσία της web υπηρεσίας. Λόγω της χρήσης του SOAP ως πρωτόκολλο επικοινωνίας, στο παράδειγμα, γίνεται η χρήση του στοιχείου soap:address.

```
<service name="InfoGrade_Service">
<documentation>This WSDL file describes the service InfoGradeService</documentation>
<port binding="tns:InfoGrade_Binding" name="InfoGrade_Port">
<soap:address location="http://localhost:5018/soap/servletrpcrouter"/>
</port>
</service>
```

**Κώδικας 5.29 – Το στοιχείο service σε ένα WSDL έγγραφο**

### **5.3 Χαρακτηριστικά ενός εγγράφου WSDL**

#### **Ονοματοδοσία και σύνδεσμοι**

Σε κάθε έγγραφο WSDL μπορεί να οριστεί προαιρετικά ένα όνομα στην ιδιότητα name του τύπου NCNAME που προσφέρει ένα είδος τεκμηρίωσης. Προαιρετικά μπορεί επίσης να ονομαστεί η ιδιότητα targetNamespace του τύπου URI.

Στο έγγραφο WSDL μπορεί να συσχετιστεί ένα ένα στοιχείο "namespace" με ένα στοιχείο "location" του εγγράφου χρησιμοποιώντας μια δήλωση <import>

```
<definitions .... >
  <import namespace="uri" location="uri"/>
</definitions>
```

**Κώδικας 5.30 – Ονοματοδοσία και σύνδεσμοι σε ένα WSDL έγγραφο**

#### **Authoring Style**

Η χρήση του στοιχείου import επιτρέπει τον διαχωρισμό των στοιχείων ορισμού μιας υπηρεσίας σε ανεξάρτητα έγγραφα, τα οποία στην συνέχεια μπορούν να εισαχθούν αν αυτό κρίνεται αναγκαίο. Αυτή η τεχνική προσφέρει έναν πιο ξεκάθαρο ορισμό της υπηρεσίας, διαχωρίζοντας τα στοιχεία ορισμού ανάλογα με το επίπεδο αφαίρεσης. Επίσης, μεγιστοποιεί την δυνατότητα επαναχρησιμοποίησης των ορισμών κάθε είδους υπηρεσιών. Ως αποτέλεσμα, σε ένα έγγραφο WSDL το οποίο είναι δομημένο κατά αυτόν τον τρόπο, οι ορισμοί των υπηρεσιών είναι πιο εύκολο χρησιμοποιηθούν και να συντηρηθούν.

```
<import namespace="http://example.gr/gradesInformation/definitions"  
location="http://example.gr/gradesInformation/InfoGradesService.wsdl"/>
```

### **Κώδικας 5.31 – Authorization υπηρεσιών σε ένα WSDL έγγραφο**

#### **Επεκτασιμότητα και Συνοχη ( binding )**

Στην WSDL ο όρος binding αναφέρεται στην διαδικασία που συνδέει τις πληροφορίες πρωτοκόλλου ή κωδικοποίησης δεδομένων με μια αφηρημένη οντότητα, όπως είναι ένα μήνυμα, μια λειτουργία, ή ένα στοιχείο portType.

#### **Τεκμηρίωση**

Η WSDL χρησιμοποιεί το προαιρετικό στοιχείο “wsdl:document” που περιέχει μια τεκμηρίωση εύκολα αναγνώσιμη. Το περιεχόμενο αυτού του στοιχείου είναι αυθαίρετο κείμενο και στοιχεία. Τα στοιχεία τεκμηρίωσης επιτρέπονται μέσα σε κάθε στοιχείο ενός εγγράφου WSDL.



## 6 Η ΓΛΩΣΣΑ ΣΗΜΑΝΣΗΣ XML

### 6.1 Ιστορία της XML

Ο Όρος XML προέρχεται από τα αρχικά ( eXtensible Markup Language ). Πρόκειται για μία σύγχρονη γλώσσα σήμανσης η οποία αναπτύχθηκε από την ομάδα W3C το 1998 με σκοπό να καλύψει ελλειπή σημεία της HTML. Η W3C είναι ένας οργανισμός που έχει αναπτύξει τα περισσότερα πρότυπα του Παγκόσμιου Ιστού.

Η HTML είναι η δημοφιλέστερη γλώσσα σήμανσης καθώς βάσει στατιστικών υπάρχουν περισσότερες από 800 εκατομμύρια Ιστοσελίδες που βασίζονται σε αυτήν. Είναι συμβατή με ένα σύνολο χιλιάδων εφαρμογών στις οποίες περιλαμβάνονται φυλλομετρητές, επεξεργαστές κειμένου, λογισμικό ηλεκτρονικής αλληλογραφίας, βάσεις δεδομένων και πολλές άλλες.

Αρχικά, ο Παγκόσμιος Ιστός ήταν μία λύση στην δημοσίευση επιστημονικών εγγράφων. Σήμερα αποτελεί ένα διαδραστικό περιβάλλον, καθώς υποστηρίζει εφαρμογές όπως: online shops, e – banking, και forums.

Για να ανταποκριθεί σε αυτή την πρωτοφανή δημοτικότητα η HTML έχει επεκταθεί με το πέρασμα των χρόνων με την προσθήκη νέων ετικετών.

Υπάρχει ένα μεγάλο σύνολο τεχνολογιών που υποστηρίζονται από την HTML, μεταξύ των οποίων συμπεριλαμβάνονται Javascript, Java, Flash, CGI, ASP, streaming media, MP3 και ακόμη περισσότερες. Κάποιες από τις τεχνολογίες αυτές αναπτύχθηκαν από την W3C όμως άλλες έχουν αναπτυχθεί από άλλους δημιουργούς.

Ωστόσο όμως, δεν είναι όλα εύκολα με την HTML λόγω του ότι έχει εξελιχθεί σε μία περίπλοκη γλώσσα η οποία περιλαμβάνει τουλάχιστον 100 ετικέτες. Οι συνδυασμοί των tags είναι αμέτρητοι και το αποτέλεσμα κάποιου συνδυασμού από tags είναι πιθανό να διαφέρει μεταξύ φυλλομετρητών.

Ένα ακόμη πρόβλημα είναι, πως χρειάζονται πολλές ετικέτες για την μορφοποίηση ακόμη και μιας απλής Ιστοσελίδας σε σημείο που να βλέπουμε ιστοσελίδες με περισσότερα tags από ότι περιεχόμενο. Και το αποτέλεσμα αυτών των σελίδων είναι η μεγάλη καθυστέρηση στην φόρτωση και εμφάνισή τους.

Συνοψίζοντας, παρόλο που η HTML είναι μια δημοφιλής και πετυχημένη γλώσσα σήμανσης, παρουσιάζει κάποια αρνητικά σημεία. Για την αντιμετώπιση αυτών λοιπόν, έχει αναπτυχθεί η XML. Ο λόγος εμφάνισής της ήταν για να συμπληρώσει την HTML και όχι να την αντικαταστήσει και οφείλει την ύπαρξή της στην επιτυχία της HTML.

### 6.2 Λόγοι χρήσης της XML

- Η XML μπορεί να χρησιμοποιηθεί για την περιγραφή και την αναπαράσταση πληροφοριών με ακρίβεια και σαφήνεια και με τέτοιο τρόπο ώστε οι υπολογιστές να μπορούν να προγραμματιστούν ώστε να αντιληφθούν αυτές τις πληροφορίες.
- Λόγω του τυποποιημένου τρόπου περιγραφής των δεδομένων που ακολουθεί, επιτρέπει σε έγγραφα και δεδομένα του ίδιου τύπου να χειρίζονται με συνέπεια και χωρίς δομικά σφάλματα.

- Παρέχει ένα σταθερό και ευέλικτο σχήμα για την αποθήκευση και την μετάδοση πληροφοριών, σταθερό επειδή βασίζεται στην τυποποίηση και ευέλικτο επειδή χρησιμοποιεί αρχεία απλών κειμένων τα οποία μπορούν να προσπελαστούν από δυαδικά.
- Παρέχει ένα ενιαίο συντακτικό στα συστήματα ανταλλαγής μηνυμάτων για την ανταλλαγή πληροφοριών μεταξύ διαφορετικών εφαρμογών. Πριν, το κάθε σύστημα ανταλλαγής μηνυμάτων, είχε τους δικούς του κανόνες, την δική του γραμμή πάνω στην οποία γινόταν η ανταλλαγή των πληροφοριών και για το λόγο αυτό, η διαδικασία της ανταλλαγής μηνυμάτων ήταν ιδιαίτερα περίπλοκη και είχε υψηλό κόστος. Η χρήση όμως, ενιαίου συντακτικού κατέστησε τα συστήματα αυτά γρηγορότερα και περισσότερο αξιόπιστα.
- Μπορεί να χρησιμοποιηθεί ελεύθερα από οποιονδήποτε χωρίς να απαιτείται αντίτιμο, καθώς αποτελεί μια πλατφόρμα ελεύθερου λογισμικού που δεν ανήκει σε κανέναν.
- Επιτρέπει την διάκριση μεταξύ μορφοποίησης και περιεχομένου. Το XML έγγραφο περιλαμβάνει μόνο την πληροφορία ( κείμενο και δεδομένα ), ενώ τα στοιχεία μορφοποίησης τοποθετούνται σε ξεχωριστά αρχεία μορφοποίησης ( stylesheets ).

### **6.3 Πλεονεκτήματα**

- Μπορεί να χρησιμοποιηθεί για την παρουσίαση δεδομένων στον Παγκόσμιο Ιστό με την σύνδεση των XML εγγράφων με μια « γλώσσα διάταξης » όπως η CSS ή η νέα XSL.
- Με την χρήση της XSL την οποία υποστηρίζει, μπορούν να παρουσιαστούν δεδομένα σε πίνακες, ταξινομημένα και οργανωμένα σε επιμέρους ομάδες δεδομένων.
- Υποστηρίζει την SMIL η οποία χρησιμοποιείται για την παρουσίαση δεδομένων σε μορφή πολυμέσων.
- Υποστηρίζει τις XLink και MathML οι οποίες χρησιμοποιούνται για την δημιουργία και προσθήκη υπερσυνδέσμων και μαθηματικών τύπων αντίστοιχα.
- Μπορεί να διαβαστεί και από μηχανές, αλλά και από τον άνθρωπο.
- Διαχωρίζει τα δεδομένα από τον τρόπο εμφάνισής τους και τον χειρισμό τους.
- Τα δεδομένα που παρουσιάζει παρέχουν την πληροφορία και δεν ενδιαφέρει ο τρόπος εμφάνισής τους.
- Επιτρέπει την εύκολη και αποδοτική επεξεργασία των δεδομένων.
- Βασίζεται σε διεθνή πρότυπα.

### **6.4 Μειονεκτήματα**

- Μόνο οι νεότεροι φυλλομετρητές Ιστού ( δηλαδή ο Internet Explorer 5 ή νεότερος και ο Netscape 6 ή νεότερος ) είναι συμβατοί με την XML.
- Για την εφαρμογή στυλ και διάταξης σε Ιστοσελίδες είναι απαραίτητη η γνώση των CSS και XSL και ειδικότερα η νεότερη γλώσσα XSL δεν υποστηρίζεται αρκετά.
- Αρκετές πρόσθετες γλώσσες που δημιουργήθηκαν για να χρησιμοποιούνται μαζί με την XML βρίσκονται ακόμη υπό ανάπτυξη.

## 6.5 Σύνταξη της XML

Με μία πρώτη ματιά διαπιστώνουμε πως ένα XML έγγραφο δεν παρουσιάζει μεγάλες διαφορές από ένα HTML έγγραφο. Η βασική διαφορά είναι πως οι ετικέτες στην XML προσδιορίζουν τα ίδια τα δεδομένα τα οποία περικλείουν, ενώ οι ετικέτες στην HTML καθορίζουν τον τρόπο εμφάνισής τους μέσα στο HTML έγγραφο. Ενώ για παράδειγμα η ετικέτα: `< font size = "10" color = "red" > ... < /font >` ορίζει ότι για το κείμενο μέσα στις σημάνσεις, το μέγεθος της γραμματοσειράς είναι 10 και το χρώμα της γραμματοσειράς είναι κόκκινο, μία ετικέτα στην XML λειτουργεί ως όνομα πεδίου. Τοποθετείται ως ετικέτα σε ένα κομμάτι δεδομένων και προσδιορίζει τί ακριβώς είναι αυτά τα δεδομένα. Για παράδειγμα η ετικέτα `< bookTitle > ... < /bookTitle >` προσδιορίζει ότι τα δεδομένα που περικλύει αποτελούν τον τίτλο βιβλίου. Ακολουθεί ένα απλό παράδειγμα ενός XML εγγράφου που εμφανίζει τον τίτλο, τον κωδικό ISBN και την τιμή δύο βιβλίων:

```
< ? xml version="1.0" standalone="yes" encoding="UTF-8" ? >
< ! DOCTYPE bookPrice SYSTEM "bookPrice.dtd" >
< bookPrice >
  < book >
    < title >Learning XML< /title >
    < ISBN >8355< /ISBN >
    < price >44.18< /price >
  < /book >
  < book >
    < title >Java 2.0< /title >
    < ISBN >1333< /ISBN >
    < price >35.53< /price >
  < /book >
< /bookPrice >
```

### Κώδικας 6.32 – Ένα παράδειγμα XML εγγράφου

Όπως βλέπουμε στο παράδειγμα, ένα XML, στην σήμανση ρίζας ορίζεται ότι το XML έγγραφο ακολουθεί την έκδοση 1.0 της XML και την κωδικοποίηση των 8 μπιτ. Τα στοιχεία του εγγράφου είναι φωλευμένα και οι ιδιότητες μπορούν να είναι ενσωματωμένες σε αυτά. Οι τιμές των ιδιοτήτων εσωκλείονται σε εισαγωγικά και οι κενές ετικέτες ( ετικέτες που δεν περιλαμβάνουν στοιχεία ), μπορούν να κλείνουν με `/>`.

## 6.6 Σημάνσεις

Η XML δεν αποτελεί από μόνη της μια γλώσσα σήμανσης αλλά είναι ένα σύνολο κανόνων για την ανάπτυξη των γλωσσών σήμανσης. Σήμανση είναι μία πληροφορία που τοποθετείται σε ένα έγγραφο για να δώσει επιπλέον πληροφορία, προσδιορίζει ποιά είναι τα δεδομένα του εγγράφου και πώς αυτά σχετίζονται μεταξύ τους. Μια γλώσσα σήμανσης είναι ένα σύνολο στοιχείων που τοποθετούνται μέσα σε ένα έγγραφο για να προσθέσουν μια ετικέτα στα δεδομένα του εγγράφου.

Οι σημάνσεις βρίσκουν εφαρμογή σε έγγραφα ηλεκτρονικής μορφής καθώς αυτά επεξεργάζονται από προγράμματα υπολογιστικών συστημάτων. Όταν ένα έγγραφο δεν περιέχει ετικέτες, τότε το πρόγραμμα δεν είναι σε θέση να χειριστεί κατάλληλα κάθε δεδομένο και κάθε ομάδα δεδομένων του εγγράφου. Έτσι, δουλεύει με όλο το έγγραφο ως ένα ενιαίο σύνολο χωρίς να δίνεται η δυνατότητα να δουλέψουμε με μεμονωμένα, ενδιαφέροντα δεδομένα. Τη λύση σε αυτό το πρόβλημα έδωσαν οι γλώσσες σήμανσης.

Το παρακάτω παράδειγμα που αναφέρεται σε ένα μήνυμα ηλεκτρονικού ταχυδρομίου αποτελεί ένα δείγμα τοποθέτησης σημάνσεων σε ένα τμήμα κειμένου:

```
< message >
  < to >publication@infostudy.org< /to >
  < from >asmith@cambridge.com< /from >
  < subject >Technology news paper< /subject >
    < text >I have already finished the article. I am sending you a pattern< /text >
    < attached fileref="article.docx" / >
< /message >
```

### Κώδικας 6.33 – Η σύνταξη σε ένα XML έγγραφο

Το απόσπασμα περιλαμβάνει τις εξής ετικέτες ( ή tags ):

- Οι σημάνσεις < message >και < /message > ορίζουν την αρχή και το τέλος του μηνύματος.
- Οι σημάνσεις < to >και < /to > προσδιορίζουν ποιός είναι ο παραλήπτης του μηνύματος.
- Οι σημάνσεις < from >και < /from > προσδιορίζουν ποιός είναι ο αποστολέας του μηνύματος.
- Οι σημάνσεις < subject >και < /subject > περιλαμβάνουν ένα κείμενο που αποτελεί το θέμα του μηνύματος.
- Οι σημάνσεις < text >και < /text > περιλαμβάνουν ένα κείμενο που αποτελεί το περιεχόμενο του μηνύματος.

- Οι σημάνσεις < attached > και < /attached > ορίζουν έναν σύνδεσμο σε αρχείο που επισυνάπτεται στο μήνυμα.

## 6.7 XML Document Type Definition ( DTD )

Η XML χρησιμοποιεί το DTD ως μοντέλο για την περιγραφή των δεδομένων. Είναι ένα σύνολο κανόνων που αφορούν στις ετικέτες της XML. Πιο συγκεκριμένα, δίνει την δυνατότητα στους χρήστες να προσδιορίσουν τις ετικέτες, την σειρά εμφάνισής τους στα XML έγγραφα και τις ιδιότητες που σχετίζονται με αυτές. Ένα « καλής – μορφής » XML έγγραφο που είναι προσαρμοσμένο στο αντίστοιχο DTD, χαρακτηρίζεται ως έγκυρο. Το παρακάτω παράδειγμα παρουσιάζει ένα απλό DTD για το παράδειγμα με τις τιμές των βιβλίων. Το DTD μπορεί να τοποθετηθεί μέσα στο XML έγγραφο, αλλά επίσης μπορεί να περιληφθεί σε ξεχωριστό αρχείο. Στην περίπτωση αυτή, το αρχείο θα πρέπει να έχει κατάληξη .dtd και στο XML έγγραφο να γίνεται αναφορά στο αρχείο ως εξής: < ! DOCTYPE Document SYSTEM "bookPrice.dtd" >. Ακόμη, μπορεί να γίνει αναφορά σε ένα εξωτερικό .dtd αρχείο χρησιμοποιώντας ένα URL.

```
< ! ELEMENT bookPrice (book) + >
< ! ELEMENT book (title, ISBN, price) >
< ! ELEMENT title (# PCDATA) >
< ! ELEMENT ISBN (# PCDATA) >
< ! ELEMENT price (# PCDATA) >
```

### Κώδικας 6.34 – XML DTD

#### DTD Elements

Χρησιμοποιούνται για να παραστήσουν τις ετικέτες ή την γλώσσα που χρησιμοποιείται. Για να ορίσουμε ένα στοιχείο DTD ακολουθούμε την παρακάτω σύνταξη:

```
< ! ELEMENT name type >
```

### Κώδικας 6.35 - XML DTD Elements

- Name: Το όνομα του στοιχείου που ορίζουμε.
- Type: Ο τύπος του περιεχομένου που περιλαμβάνει το στοιχείο. Μπορεί να είναι κείμενο, άλλο στοιχείο, ή συνδυασμός και των δύο.

## **6.8 XML και προγενέστερες γλώσσες σήμανσης ( markup languages )**

Μια γλώσσα σήμανσης είναι μια οργάνωση κώδικα που περιέχει εντολές οι οποίες περιγράφουν την δομή ενός κειμένου ή καθορίζουν το πώς αυτό θα εμφανίζεται. Ένα παράδειγμα μιας γνωστής και ευρύτατα χρησιμοποιούμενης γλώσσας σήμανσης στον παγκόσμιο ιστό ( World Wide Web – WEB ) είναι η HTML.

### **Ιστορικά στοιχεία**

Ο όρος “markup” προέρχεται από μία παλιά συνήθη διαδικασία “marking up” ( σήμανσης) χειρογράφων, κατά την οποία τοποθετούνταν συμβολικές οδηγίες εκτύπωσης στα περιθώρια ενός χειρόγραφου. Για αρκετούς αιώνες, αυτή η εργασία ήταν στην αποκλειστικότητα ικανών τυπογράφων, γνωστοί με τον όρο “markup men” που σημείωναν ένα κείμενο για να δηλώσουν τί μορφή, ύφος και μέγεθος θα έπρεπε να εφαρμοστεί σε κάθε κομμάτι του κειμένου. Η τεχνική σήμανσης χρησιμοποιήθηκε επίσης από συντάκτες, διορθωτές τυπογραφιών δοκιμιών και σχεδιαστές γραφικών.

### **GML και SGML**

Πρωτοπόρος των γλωσσών σήμανσης υπήρξε ο ερευνητής της IBM Charles Goldfarb ο οποίος ανακάλυψε την IBM GML και η οποία δημοσιεύθηκε το 1973. Κατόπιν από την GML δημιουργήθηκε η SGML η οποία γνωστοποιήθηκε από την IBM το 1986. Η SGML επέτρεπε στους χρήστες να δημιουργούν και να χρησιμοποιούν οποιαδήποτε σήμανση επιθυμούσαν, διαλέγοντας σημάνσεις της προτίμησής τους και τις ονόμαζαν σε φυσική γλώσσα. Έτσι, η SGML είναι μια κατάλληλη μεταγλώσσα και πολλές γλώσσες σήμανσης προέρχονται από αυτήν. Από τα τέλη της δεκαετίας του 1980, πολλές σημαντικές γλώσσες σήμανσης είχαν ως κύρια βάση την SGML, όπως για παράδειγμα η TEI και η DocBook. Η SGML κηρύχθηκε διεθνές πρότυπο γλώσσας σήμανσης από την IBM το 1986.

### **HTML**

Το 1991 χρησιμοποιώντας την σύνταξη της SGML δημιουργήθηκε η HTML. Η HTML παρουσιάζει ομοιότητες με πολλές γλώσσες σήμανσης που έχουν ως βάση την SGML, όμως αυτή είναι απλούστερη από τις περισσότερες. Η χρήση της ως περιγραφική γλώσσα σήμανσης ήταν ευρύτατα διαδεδομένη στον παγκόσμιο ιστό λόγω της ευελιξίας και της επεκτασιμότητάς της. Σήμερα, η HTML είναι η γλώσσα σήμανσης που χρησιμοποιείται περισσότερο.

### **XML**

Η XML είναι μια μεταγλώσσα η οποία χρησιμοποιείται περισσότερο σήμερα. Ο κύριος λόγος της δημιουργίας της ήταν να απλοποιήσει την SGML στην περιγραφή εγγράφων που στέλνονται στο διαδίκτυο. Η XML είναι μία μεταγλώσσα όπως η SGML, η οποία επιτρέπει στους χρήστες να δημιουργούν τις δικές τους σημάνσεις. Αυτή είναι και η βασική της διαφορά από την HTML, όπως επίσης και το ότι είναι πιο « αυστηρή ». Ακόμη, η HTML έχει ως σκοπό την διατύπωση και εμφάνιση των δεδομένων, ενώ η XML την αναπαράσταση της έννοιας των δεδομένων.

## **XHTML**

Το 2000 με βάση την XML και την HTML δημιουργήθηκε η XHTML ( Extensible Hyper Text Markup Language ). Η ιδιαιτερότητα της γλώσσας ήταν ότι απαιτούσε τα XHTML WEB έγγραφα να είναι καλώς-διαμορφωμένα XML έγγραφα. Μια από τις κυριότερες διαφορές μεταξύ της HTML και της XHTML είναι ότι όλες οι σημάνσεις πρέπει να κλείνουν. Σημάνσεις της HTML οι οποίες δεν περιλαμβάνουν σημάνσεις κλεισίματος, όπως η <hr> πρέπει να κλείνουν με τις ανάλογες σημάνσεις κλεισίματος ( end tags ) ή να έχουν την μορφή </hr>. Μια ακόμη διαφορά είναι πως όλες οι τιμές ιδιοτήτων κάθε σήμανσης πρέπει να αναφέρονται. Και τέλος, όλες οι σημάνσεις και τα ονόματα ιδιοτήτων πρέπει να είναι με κεφαλαία προκειμένου να είναι έγκυρα.

## **Χαρακτηριστικά**

Το κοινό χαρακτηριστικό των περισσότερων γλωσσών σήμανσης είναι ότι αναμειγνύουν το κείμενο ενός εγγράφου με σημάνσεις στο ίδιο ρεύμα δεδομένων ή αρχείο. Αυτό δεν είναι απαραίτητο. Είναι δυνατό να απομονωθούν οι σημάνσεις από το περιεχόμενο του κειμένου, με την χρήση δεικτών, αναγνωριστικών, ή άλλων μεθόδων ώστε να απομονωθούν αυτά τα δύο.

## **Εναλλακτική Χρήση**

Παρόλο που έννοια των γλωσσών σήμανσης έχει συνδεθεί με κείμενα εγγράφων, η χρήση τους δεν περιορίζεται μόνο εκεί, αλλά χρησιμοποιούνται σε εργασίες που έχουν ως αντικείμενο την παρουσίαση κάθε τύπου πληροφορίας, όπως λίστες αναπαραγωγής, υπηρεσίες διαδικτύου και διεπαφές χρηστών. Ο λόγος που οι περισσότερες εφαρμογές στηρίζονται στην XML είναι πως πρόκειται για μια γλώσσα καλώς ορισμένη και επεκτάσιμη.

## 7 UDDI ΚΑΤΑΧΩΡΗΣΗ ΤΩΝ WEB SERVICES

### 7.1 Τί είναι μία καταχώρηση Web Service

Στον πυρήνα του το UDDI δομείται από δύο μέρη. Το πρώτο μέρος αποτελεί μία τεχνική προδιαγραφή για την δημιουργία ενός διαμοιραζόμενου στο διαδίκτυο καταλόγου από επιχειρήσεις και web services. Όλα τα δεδομένα αποθηκεύονται σε αυτόν τον κατάλογο σε μία ειδική XML μορφή και επίσης η προδιαγραφή UDDI περιλαμβάνει όλες τις απαραίτητες λεπτομέρειες χρήσης της UDDI διεπαφής προγραμματισμού εφαρμογών για την αναζήτηση των δεδομένων καταλόγου, καθώς και την καταχώρηση νέων δεδομένων σε αυτόν. Από την άλλη πλευρά, η UDDI Καταχώρηση Επιχειρήσεων που συχνά αναφέρεται με τον όρο “ cloud services ”, αποτελεί μία πλήρη λειτουργική υλοποίηση της UDDI προδιαγραφής. Το κομμάτι αυτό της τεχνολογίας UDDI αναπτύχθηκε τον Μάιο του 2001 από τις εταιρείες Microsoft και IBM και επιτρέπει, την αναζήτηση οποιονδήποτε δεδομένων είναι καταχωρημένα σε μια UDDI βάση, καθώς και την καταχώρηση επιχειρήσεων και των υπηρεσιών που παρέχουν.

Μία επιχείρηση μπορεί να καταχωρήσει τρεις τύπους πληροφορίας σε μία βάση UDDI :

- **White Pages** : Περιλαμβάνουν βασικές πληροφορίες για την επικοινωνία με μία εταιρεία, όπως, το όνομά της, την διεύθυνσή της, την διεύθυνση ηλεκτρονικού ταχυδρομείου και άλλα στοιχεία επικοινωνίας, καθώς και μοναδικά αναγνωριστικά όπως οι ταξικές διευθύνσεις IP.
- **Yellow Pages** : Περιλαμβάνουν πληροφορίες που περιγράφουν μια web service και την τοποθετούν σε μία ανάλογη κατηγορία. Αυτές οι πληροφορίες επιτρέπουν σε κάποιον να εντοπίσει μια υπηρεσία με βάση την κατηγορία στην οποία ανήκει.
- **Green Pages** : Περιλαμβάνουν τεχνικές πληροφορίες που περιγράφουν την λειτουργικότητα που παρέχει μια web υπηρεσία υποστηριζόμενη από μία εταιρεία. Επίσης, οι πληροφορίες αυτές περιλαμβάνουν δείκτες προς τα σημεία τοποθέτησης των υπηρεσιών, ώστε να διευκολύνεται ο εντοπισμός τους.

### 7.2 Ιστορικά στοιχεία

Η τεχνολογία UDDI αναπτύχθηκε τον Μάιο του 2001 από τις εταιρείες Microsoft και IBM ωστόσο την ίδια χρονιά παρουσιάστηκε μία νέα έκδοση της τεχνολογίας, η έκδοση 2.0, η οποία αποτέλεσε μία βελτιωμένη έκδοχή της προηγούμενης, περιλαμβάνοντας τα εξής επιπλέον χαρακτηριστικά :



- Δυνατότητα διεθνοποίησης των web services καθώς επέτρεψε την περιγραφή τους από τις εταιρείες τους σε διαφορετικές γλώσσες.
- Υποστηρίζει την περιγραφή υπηρεσιών πολύπλοκων οργανισμών, δίνοντας την δυνατότητα σε μία εταιρεία να δημοσιεύσει ένα ολόκληρο σύνολο απαρτιζόμενο από πολλά διαφορετικά τμήματά της και μάλιστα τοποθετημένα σε διαφορετικά σημεία του κόσμου.
- Περιλαμβάνει πιο εξειδικευμένες και βελτιωμένες δυνατότητες αναζήτησης.

### **7.3 Πώς χρησιμοποιείται**

Θα εξετάσουμε την τεχνολογία του UDDI από δύο διαφορετικές σκοπιές, του αναλυτή και του προγραμματιστή. Και οι δύο είναι αυτοί που το χρησιμοποιούν. Από την πλευρά του αναλυτή της επιχείρησης, ένας κατάλογος επιχειρήσεων UDDI είναι κάτι αντίστοιχο με μία μηχανή αναζήτησης. Μία τυπική μηχανή αναζήτησης, όπως το Google ή το Ask, οργανώνει και δημιουργεί δείκτες προς τις URL διευθύνσεις των Web Sites. Είναι σημαντικό όμως πως, μία επιχείρηση η οποία παρέχει ένα Web Service, κρίνεται απαραίτητο να δημοσιεύει πολύ περισσότερα στοιχεία από μία URL διεύθυνση. Από την άλλη πλευρά, αυτή του προγραμματιστή, οι προγραμματιστές λογισμικού χρησιμοποιούν την διεπαφή του UDDI για να δημοσιεύσουν πληροφορίες, αλλά και να δημιουργήσουν ερωτήματα στην βάση δεδομένων των υπηρεσιών, ώστε να γίνεται η εύρεσή τους με βάση πολυάριθμα κριτήρια. Γίνεται αντιληπτό το γεγονός πως το λογισμικό στο τέλος θα εντοπίσει την υπηρεσία δυναμικά και θα την χρησιμοποιήσει χωρίς να είναι απαραίτητη η ανθρώπινη αλληλεπίδραση.

### **7.4 Τεχνικά Χαρακτηριστικά**

Από τεχνική άποψη, το UDDI αποτελείται από τρία μέρη:

- Το Μοντέλο Δεδομένων UDDI

Είναι ένα XML σχήμα που χρησιμοποιείται για την περιγραφή επιχειρήσεων και των web services που παρέχουν.

- Τη Διεπαφή Προγραμματισμού Εφαρμογών UDDI

Μια διεπαφή λογισμικού που βασίζεται στο πρωτόκολλο ανταλλαγής μηνυμάτων SOAP και χρησιμοποιείται για την δημοσίευση και την αναζήτηση δεδομένων σε μια UDDI βάση.

- Τις UDDI cloud services

Διαδικτυακοί τόποι που επιτρέπουν την υλοποίηση του UDDI συγχρονίζοντας τα δεδομένα της βάσης του UDDI

## 7.5 Το Μοντέλο Δεδομένων UDDI

Το μοντέλο του UDDI αποτελείται από ένα σχήμα XML που περιγράφει τρεις τύπους δεδομένων:

- `businessEntity`
- `businessService`
- `bindingTemplate`
- `tModel`

### ***businessEntity***

Το στοιχείο αυτό περιλαμβάνει πληροφορίες που περιγράφουν την ίδια την επιχείρηση που παρέχει την Web Service. Περιλαμβάνει πληροφορίες όπως όνομα, διεύθυνση, τηλέφωνο, αλλά και άλλα στοιχεία επικοινωνίας.

### ***businessService***

Το στοιχείο αυτό περιλαμβάνει πληροφορίες που περιγράφουν μία μεμονωμένη Web Υπηρεσίας ή συνόλο αυτών. Οι πληροφορίες αυτές είναι το όνομα, περιγραφή λειτουργίας δηλαδή τί μπορεί να προσφέρει και προαιρετικά μία λίστα από στοιχεία τύπου `bindingTemplate`.

### ***bindingTemplate***

Το στοιχείο `bindingTemplate` περιλαμβάνει πληροφορίες για το πώς και πού μπορεί κάποιος να αποκτήσει πρόσβαση σε ένα Web Service.

### ***tModel***

Ο τελευταίος τύπος δεδομένων του UDDI μοντέλου είναι το `tModel`. Το στοιχείο αυτό καλύπτει το τεχνικό μέρος του μοντέλου, και συγκεκριμένα παρέχει δείκτες προς εξωτερικές τεχνικές προδιαγραφές.

## 7.6 Ιδιαίτερα χαρακτηριστικά του UDDI

- Επαναληπτικό έγγραφο UDDI

Αυτό το έγγραφο περιγράφει τις επαναληπτικές διεργασίες που γίνονται επί των δεδομένων και το `interface` στο οποίο ο χειριστής της καταχώρησης UDDI πρέπει να προσαρμοστεί ώστε να επιτευχθεί η αναπαραγωγή των δεδομένων μεταξύ διαφορετικών ιστοτόπων.

- Τελεστές χειρισμού UDDI

Αυτό το έγγραφο περιγράφει την συμπεριφορά και τις παραμέτρους χειρισμού που απαιτούνται από τους χειριστές των UDDI κόμβων επικοινωνίας. Ορίζει τις απαιτήσεις χειρισμού των δεδομένων στις οποίες πρέπει να ανταποκρίνονται οι χειριστές. Για παράδειγμα, οι χειριστές είναι υπεύθυνοι για την διαρκή εγγραφή και επαναφορά όλων των δεδομένων, επιβεβαιώνοντας ότι κάθε εγγραφή επιχείρησης έχει μία έγκυρη διεύθυνση ηλεκτρονικού ταχυδρομίου για επικοινωνία με αυτή και εξασφαλίζοντας την ακεραιότητα των δεδομένων από μη επιθυμητή διαγραφή.

- UDDI Διεπαφή προγραμματισμού εφαρμογών

Το έγγραφο αυτό, ορίζει ένα σύνολο λειτουργιών που υποστηρίζουν όλες οι βάσεις UDDI για την κάλυψη των απαιτήσεων των υπηρεσιών που είναι καταχωρημένες και την δημοσίευση πληροφοριών για μία επιχείρηση ή υπηρεσία. Ορίζει μία σειρά από μηνύματα SOAP τα οποία ενσωματώνονται μέσα σε έγγραφα XML τα οποία μία καταχώρηση UDDI λαμβάνει, επεξεργάζεται και στα οποία απαντά. Μαζί με το UDDI XML API σχήμα δεδομένων και το μοντέλο δεδομένων UDDI δημιουργεί ένα πλήρες προγραμματιστικό interface για μία UDDI καταχώρηση.

- Μοντέλο δεδομένων UDDI

Ορίζει την μορφή των δεδομένων που περιέχονται μέσα στα SOAP μηνύματα. Ορίζει πέντε δομές δεδομένων, καθώς και τις σχέσεις μεταξύ τους.

## 8 JAVA SERVLETS, ΜΙΑ ΠΡΟΓΕΝΕΣΤΕΡΗ ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ WEB SERVICES

### 8.1 Ορισμός

Ένα servlet είναι ένα πρόγραμμα γραμμένο σε JAVA που τρέχει σε έναν διακομιστή web ή εφαρμογή διακομιστή που λειτουργεί ως γέφυρα επικοινωνίας μεταξύ ενός web server και ενός HTTP client. Παρέχει έναν δυναμικό τρόπο χειρισμού των αιτημάτων του client που πραγματοποιείται από την πλευρά του server. Τον χειρισμό αυτό, αναλαμβάνουν τα servlets, πραγματοποιώντας τις εξής εργασίες:

- Διαβάζουν τα δεδομένα που στέλνονται από έναν χρήστη μέσω μίας φόρμας ιστοσελίδας, μέσω ενός Java Applet, ή οποιασδήποτε άλλης HTTP εφαρμογής.
- Επεξεργάζονται τα δεδομένα αυτά, εργασία που πιθανώς να απαιτεί πρόσβαση σε μία βάση δεδομένων.
- Στέλνουν τα αποτελέσματα τις επεξεργασίας πίσω στον client, καθορίζοντας την μορφή του εγγράφου που θα αποσταλλεί, HTML, δυαδική ( για παράδειγμα ως εικόνα ) ή XML.

## 8.2 *JAVA Servlets, μία προγενέστερη Τεχνολογία των Web Services*

Πριν την εμφάνιση των JAVA Servlets, το πρότυπο για τον δυναμικό χειρισμό της ανταλλαγής δεδομένων μεταξύ client και server αποτελούσε το CGI ( Common Gate Interface ). Το πρότυπο CGI χρησιμοποιείται σε συστήματα www από το 1993 και συνιστά την διεπαφή για επικοινωνία μεταξύ ενός server και εξωτερικών προγραμμάτων gateways. Ο όρος πρόγραμμα gateway αναφέρεται σε μια κοινή εφαρμογή λογισμικού που δέχεται αιτήσεις για ανάκτηση μιας πληροφορίας και επιστρέφει το κατάλληλο έγγραφο ή το κατασκευάζει δυναμικά ( on-the-fly document generation ). Μέσω του CGI ο Http Server μπορεί να παρέχει πρόσβαση σε πληροφορίες οι οποίες δεν είναι σε μορφή άμεσα κατανοητή από τον χρήστη ( π.χ. πρόσβαση σε στοιχεία μιας Βάσης Δεδομένων μέσω ερωτημάτων SQL ). Το CGI λειτουργεί σαν μία πύλη επικοινωνίας ( εξού και όρος gate ) μεταξύ των δύο και παράγει κάτι που είναι κατανοητό στον client. Ο όρος common αναφέρεται στο γεγονός ότι η διεπαφή πραγματοποιείται με τρόπο ανεξάρτητο της πλατφόρμας υλοποίησης. Τα προγράμματα CGI ή scripts είναι εκτελέσιμα προγράμματα που μπορούν να λειτουργήσουν ανεξάρτητα. Μπορούν να δημιουργηθούν με την βοήθεια οποιασδήποτε γλώσσας προγραμματισμού αρκεί αυτή να επιτρέπει την παραγωγή εκτελέσιμων αρχείων. Οι πλέον δημοφιλείς γλώσσες για ανάπτυξη εφαρμογών CGI σε περιβάλλον Windows είναι οι Borland Delphi, Visual Basic, Visual C/C++ και σε UNIX οι C/C++, Perl, TCL/Tk, Bourne Shell, C Shell. Το πρότυπο αυτό όμως εμφάνιζε κάποια μειονεκτήματα, όπως για παράδειγμα το γεγονός ότι σε κάθε εκτέλεση ενός script ξεκινά μια νέα διαδικασία. Και εάν υπάρχουν πολλοί χρήστες ταυτόχρονα σε έναν Web Server και εκτελούνται πολλά scripts, τότε ο Server μπορεί να υπερφορτωθεί. Για τον λόγο αυτό, αλλά και για αρκετούς άλλους την τεχνολογία CGI ήρθε να υποκαταστήσει αυτή των Servlets.

### **Πλεονεκτήματα JAVA Servlets έναντι CGI**

- Καλύτερη απόδοση

Με τα προγράμματα CGI, για τον χειρισμό κάθε HTTP αίτησης απαιτείται μία νέα διαδικασία, με αποτέλεσμα να υπερφορτώνεται ο Web Server αν την ίδια χρονική στιγμή αποστέλλονται πολλά αιτήματα. Αντίθετα, τα Servlets χρησιμοποιούν ελαφρά νήματα της JAVA και έτσι δεν επιβαρύνεται το Λειτουργικό Σύστημα. Από την πλευρά των CGI, με κάθε αίτηση φορτώνεται εκ νέου το πρόγραμμα CGI στην μνήμη, ενώ με τα Servlets δημιουργείται ένα αντίγραφο. Με την λήξη της διεκπεραίωσης μιας αίτησης το CGI τερματίζεται και πρέπει να φορτωθεί εκ νέου στην cache ενώ τα Servlets παραμένουν αποθηκευμένα μέχρι να διεκπεραιωθεί και το τελευταίο αίτημα.

- Είναι ευέλικτα

Η τεχνολογία των JAVA Servlets πλεονεκτεί έναντι αυτής των CGI και οποιασδήποτε πλατφόρμας που χρησιμοποιείται για server-side προγραμματισμό καθώς ο κώδικας που παράγεται σε JAVA είναι περισσότερο αξιόπιστος και επαναχρησιμοποιήσιμος. Επίσης, τα Servlets διαθέτουν μια ευρεία γκάμα εργαλείων και τεχνικών για την μεταγλώττιση και αποκωδικοποίηση δεδομένων σε HTML μορφή, τον χειρισμό των cookies, τον εντοπισμό συνεδριών και πολλά ακόμη βοηθητικά προγράμματα υψηλού επιπέδου.

- Είναι Ισχυρά και Αποτελεσματικά

Τα JAVA Servlets μπορούν να έχουν άμεση επικοινωνία με έναν Web Server σε αντίθεση με τα CGI που απαιτούν την ύπαρξη ειδικής διεπαφής. Ακόμη, επιτρέπεται η ανταλλαγή δεδομένων μεταξύ πολλαπλών Servlets καθιστώντας ευκολότερη τη σύνδεση σε μία Βάση Δεδομένων και την ανάκτηση δεδομένων από αυτή. Επίσης, τα Servlets έχουν την δυνατότητα να διατηρούν στοιχεία από προηγούμενες αιτήσεις και να αποθηκεύουν προσωρινά τα αποτελέσματα της επεξεργασίας τους.

- Φορητότητα

Τα Servlets καθώς είναι γραμμένα σε JAVA, τρέχουν σε πολλούς servers όπως τους Apache, MII Server (IIS), IBM Sphere και StarNine WebStar χωρίς να υπάρχει οποιοδήποτε πρόβλημα συμβατότητας.

### **8.3 Ανατομία ενός JAVA Servlet**

Κατά την δημιουργία ενός JAVA Servlet θα πρέπει πρώτα απ'όλα να κάνουμε import δύο πακέτα, javax.servlet.\* και javax.servlet.http.\*. Η επέκταση του JAVA server που περιέχεται στην κλάση javax.servlet είναι ένας σκελετός για την δημιουργία Servlets. Περιέχει APIs για τον χειρισμό αιτημάτων του server, την αποστολή νημάτων και άλλες εργασίες του server. Το πακέτο javax.servlet.http.\* είναι μια υλοποίηση των JAVA Servlets σχεδιασμένο για Web servers και την δημιουργία σελίδων HTML.

### Ένα παράδειγμα JAVA Servlet που δημιουργεί μία HTML σελίδα με φόρμα

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class Dictionary extends HttpServlet{

    HttpServletRequest request;
    HttpServletResponse response;

    public void doGet(HttpServletRequest
req,HttpServletResponse res)throws
ServletException,IOException{

        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        String input="";
        out.println("<HTML><HEAD><TITLE>Dictionary</TITLE></HEAD>
");
        out.println("<BODY BGCOLOR=#FFFFFF><CENTER><P>");

        out.println("<form name = 'form1'>Type A
Term:<input type = 'text' name = 'field1'>");
        out.println("<input type = 'submit' value =
'Find' /></form>");
        out.println("<br>Description:<br><form name =
'form2' ><input type = 'textArea' size = '50'
readonly='readonly' /></form>");
        out.println("<BR>");
        out.println(b);
        out.println("</P></CENTER></BODY></HTML>");

        out.flush();
        out.close();
    }

    public String getServletInfo()

    {
        return "A Java Servlet";
    }
}
```

### Κώδικας 8.36 – Ένα παράδειγμα Servlet

### Οι μέθοδοι doGet() και doPost()

Ένα Servlet συνήθως κάνει υπέρβαση της μεθόδου doGet () της κλάσης HttpServlet. Η μέθοδος αυτή καλείται από τον Web server κάθε φορά που ο browser του χρήστη ζητά μέσω ενός HTTP Get Request πρόσβαση σε ένα URL που δείχνει στο servlet. Η κλάση HttpServlet ορίζει do μεθόδους για τον χειρισμό και άλλων αιτημάτων HTTP, όπως οι μέθοδοι doDelete(), doOptions(), doPost(), doPut() και doTrace(). Κάθε μια από αυτές τις μεθόδους μπορεί να τις υπερβεί ένα Servlet προκειμένου να γίνει ο χειρισμός οποιουδήποτε αιτήματος.

Οι μέθοδοι do παίρνουν δύο παραμέτρους: ένα αντικείμενο τύπου HttpServletRequest και ένα αντικείμενο τύπου HttpServletResponse. Οι δύο αυτές κλάσεις HttpServletRequest και HttpServletResponse περιλαμβάνουν μεθόδους και στατικές μεταβλητές που επιτρέπουν στο Servlet να επικοινωνήσει με τον host server του. Η τυπική υπογραφή της μεθόδου doGet () είναι η εξής: `public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {}`.

### **Η κλάση HttpServletRequest**

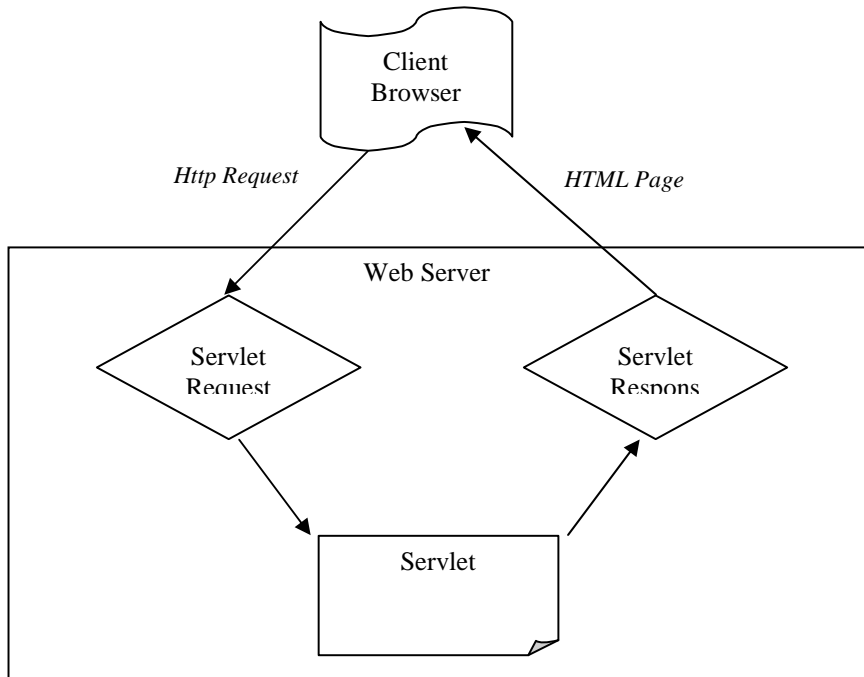
Η κλάση αυτή περιλαμβάνει όλες τις πληροφορίες που χρειάζεται το Servlet για να χειριστεί το εκάστοτε αίτημα. Έχει μεθόδους που επιστρέφουν μία λίστα με όλες τις λειτουργίες που γίνονται από τον web browser για την απάντηση σε ένα αίτημα του χρήστη.

### **Η κλάση HttpServletResponse**

Επιτρέπει στο Servlet να επιστρέψει τα αποτελέσματά του στον Web server. Η μέθοδος της κλάσης, setContentype(String) προσδιορίζει τον τύπο της πληροφορίας που επιστρέφει το Servlet. Οι επιτρεπόμενοι τύποι είναι, text/html, text/plain. Για παράδειγμα, μια υλοποίηση της μεθόδου setContentype μπορεί να είναι, setContentype ("text/html") η οποία επιστρέφει μία HTML σελίδα. Η επόμενη σημαντική μέθοδος της κλάσης HttpServletResponse είναι η getWriter() η οποία επιστρέφει ένα αντικείμενο PrintWriter που σχετίζεται με ένα ρεύμα εξόδου που το Servlet χρησιμοποιεί για να τυπώσει τα αποτελέσματά του πίσω στον Web server. Όταν απαιτείται από το Servlet η δημιουργία μιας HTML σελίδας ως απόκριση σε ένα HTTP request, τότε ο κώδικας HTML τοποθετείται παραμετρικά μέσα στο PrintWriter. Η γραμμή κώδικα: PrintWriter out = request.getWriter() ορίζει το αντικείμενο out τύπου PrintWriter.

*Πώς πραγματοποιείται η επικοινωνία client/server μέσω ServletRequest και ServletResponse*





---

*Εικόνα 8.10 – Επικοινωνία client/server μέσω ServletRequest και ServletResponse*

<i>Μέθοδοι της κλάσης HttpServlet</i>	
doGet()	Τα servlets υλοποιούν αυτή την μέθοδο προκειμένου να χειριστούν τα αιτήματα GET. Η αρχική υλοποίηση της μεθόδου επιστρέφει ένα σφάλμα τύπου BAD_REQUEST.
doPost()	Τα servlets υλοποιούν αυτή την μέθοδο προκειμένου να χειριστούν τα αιτήματα POST. Η αρχική υλοποίηση της μεθόδου επιστρέφει ένα σφάλμα τύπου BAD_REQUEST.
doDelete()	Τα servlets υλοποιούν αυτή την μέθοδο προκειμένου να χειριστούν τα αιτήματα DELETE. Η αρχική υλοποίηση της μεθόδου επιστρέφει ένα σφάλμα τύπου BAD_REQUEST.
doOptions()	Η αρχική υλοποίηση της μεθόδου αποφασίζει ποιές επιλογές υποστηρίζονται και επιστρέφει το ανάλογο μήνυμα. Αν για παράδειγμα, σε ένα servlet γίνεται υπέρβαση των μεθόδων doGet, doPost και doDelete, τότε ο browser πληροφορείται ότι υποστηρίζονται οι λειτουργίες GET, POST και DELETE.

**Πίνακας 8.1 – Μέθοδοι της κλάσης HttpServlet**

## 8.4 Χειρισμός δεδομένων φόρμας με servlets

Ένα σημαντικό κομμάτι του server – side προγραμματισμού είναι ο χειρισμός των δεδομένων που στέλνονται σε έναν server μέσω μιας html φόρμας. Τη λύση στο ζήτημα αυτό έρχονται να δώσουν τα servlets τα οποία λαμβάνουν τα δεδομένα που εισάγει ο χρήστης σε μια φόρμα, τα επεξεργάζονται και επιστρέφουν τα αποτελέσματα πίσω στον client.

Τα servlets χρησιμοποιούν δύο βασικές μεθόδους της κλάσης HttpServletRequest:

- `getParameter()`: Μέθοδος που χρησιμοποιείται για ανάγνωση απλών τιμών μιας παραμέτρου.
- `getParameterValues()`: Μέθοδος για ανάγνωση πολλαπλών τιμών.

Τα servlets έχουν την δυνατότητα να λαμβάνουν απευθείας τα δεδομένα που εισάγει ο χρήστης. Χρησιμοποιούν αρχικά την μέθοδο `getParameter()` της κλάσης `HttpServletRequest` η οποία παίρνει ένα όρισμα, το όνομα της παραμέτρου της οποίας η τιμή πρόκειται να διαβαστεί και επιστρέφει μια συμβολοσειρά που αντιστοιχεί στην τιμή αυτή. Σε περίπτωση που δεν έχει δοθεί τιμή στην παράμετρο, τότε επιστρέφεται κενό `String`. Αν δεν υπάρχει η παράμετρος, τότε επιστρέφεται `null String`.

Όταν πρόκειται να γίνει η ανάγνωση των τιμών μιας παραμέτρου η οποία παίρνει πολλαπλές τιμές, τότε καλείται η μέθοδος `getParameterValues()` η οποία παίρνει ως όρισμα το όνομα της παραμέτρου και επιστρέφει έναν πίνακα από `Strings`. Αν η παράμετρος δεν υπάρχει, τότε επιστρέφεται `null String`. Αν η παράμετρος έχει μία τιμή, τότε επιστρέφεται ένας πίνακας με ένα μόνο `String`.

Τα ονόματα των παραμέτρων είναι case sensitive, που σημαίνει ότι υπάρχει διάκριση μεταξύ κεφαλαίων και πεζών χαρακτήρων. Έτσι για παράδειγμα, οι δηλώσεις `request.getParameter("NameField")` και `request.getParameter("namefield")` αναφέρονται σε διαφορετικές παραμέτρους.

### Κώδικας 8.37 - Ένα Java Servlet που λαμβάνει απλή τιμή παραμέτρου από μία φόρμα.

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

import DataAccess.*;

public class Dictionary extends HttpServlet{

    HttpServletRequest request;
    HttpServletResponse response;
    static DataAccess data = new DataAccess();
    static String b = "";

    public void doGet(HttpServletRequest req,HttpServletResponse res)throws
    ServletException,IOException{

        res.setContentType("text/html");
        res.setCharacterEncoding("ISO-8859-7");

        PrintWriter out = res.getWriter();
        String input="";
        String a = req.getParameter("inputTerm");

        out.println("<HTML><HEAD><meta http-equiv='Content-Type'
content='text/html; Charset = ISO-8859-7' /><TITLE>Dictionary</TITLE></HEAD>");
        out.println("<BODY BGCOLOR=#FFFFFF><CENTER><P><table
width='30%' border='1' align='center' style='border-bottom:medium; border-left:medium;
border-right:medium; border-top:medium'><TR><TD><DIV style:'text-align=center'>");
        out.println(a);
        out.println("</DIV></TD></TR><TR><TD>");

        String termDescription = "";
        try
        {
            termDescription = data.getTermsDescription(a);
        }
        catch(Exception e)
        {
            System.out.println("An Error occurred searching for term "+termValue);
        }

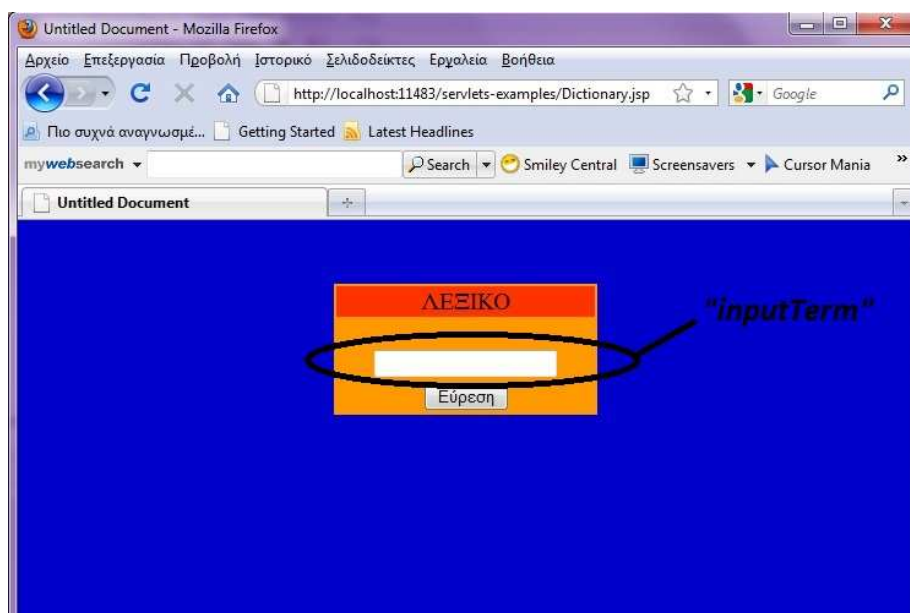
        out.println(termDescription);
    }
}
```

```
        out.println("</TD></TR></TABLE></P></CENTER></BODY></HTML>");

        out.flush();
        out.close();
    }

    public String getServletInfo()
    {
        return "A Java Servlet Written By Ria";
    }
}
```

***Dictionary.java***



### Κώδικας 8.38 - Το servlet λαμβάνει την τιμή της παραμέτρου με όνομα "inputTerm"

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>

<body bgcolor="#0000CC">
<p>&nbsp;</p>
<form name="form2" action="servlet/Dictionary">
<table width="30%" border="0" align="center" bgcolor="#FF9900" style="border-
bottom:medium; border-left:medium; border-right:medium; border-top:medium">
<tr>
<td bgcolor="#FF3300"><div style="text-align:center; font-size:20px;">ΛΕΞΙΚΟ</div></td>
</tr>
<tr>
<td>&nbsp;</td>
</tr>
<tr>
<td><div style="text-align:center"><input type="text" name="inputTerm" /></div></td>
</tr>
<tr>
<td><div style="text-align:center"><input type="submit" value="Εύρεση" /></div></td>
</tr>
</table>
</form>
<p>&nbsp;</p>
</body>
</html>

```

*Dictionary.jsp*

**Ένα Java Servlet που λαμβάνει πολλαπλές τιμές παραμέτρου από μία φόρμα.**

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
```

```
import DataAccess.*;
```

```
public class Books extends HttpServlet{
```

```
    HttpServletRequest request;
    HttpServletResponse response;
    static DataAccess data = new DataAccess();
    static String b = "";
```

```
    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws
    ServletException, IOException{
```

```
        res.setContentType("text/html");
```

```
        PrintWriter out = res.getWriter();
        String input="";
        String a = req.getParameter("inputBook");
```

```
        String dropMenu[] = req.getParameterValues("jumpMenu");
        int dropMenuFlag = -1;
```

```
//-----CHECK THE DROP LIST'S VALUE-----
```

```
for(int i = 0; i<dropMenu.length; i++)
{
    //pw.println("The selected colors are" + colors[i]+ "<br>");
    if(dropMenu[i] == "0")
    {
        dropMenuFlag = 1;
    }
    else if(dropMenu[i] == "1")
    {
        dropMenuFlag = 2;
    }
    else if(dropMenu[i] == "2")
    {
        dropMenuFlag = 3;
    }
    else if(dropMenu[i] == "3")
    {
        dropMenuFlag = 4;
    }
    else if(dropMenu[i] == "4")
    {
        dropMenuFlag = 5;
    }
}
```

```
        out.println("<HTML><HEAD><meta http-equiv='Content-Type'
content='text/html; charset=utf-8' /><TITLE>Books</TITLE></HEAD>");
        out.println("<BODY BGCOLOR=#FFFFFF><CENTER><P>");
```

```
        String bookDescription = "";
        try
        {
            bookDescription = data.getBook(a,dropMenuFlag);
        }
        catch(Exception e)
        {
            System.out.println("An Error oqqured");
        }
    }
```

```
        out.println(bookDescription);
```

```
        out.println("</P></CENTER></BODY></HTML>");
```

```
        out.flush();
        out.close();
```

```
    }
```

```
@Override
```

```
    public String getServletInfo()
```

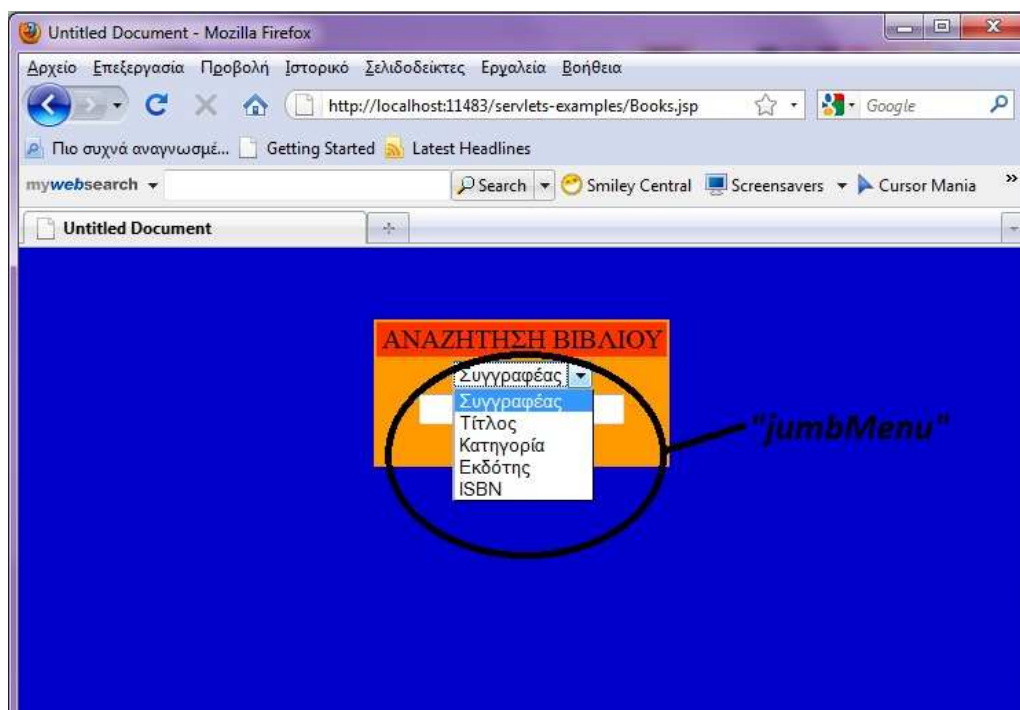
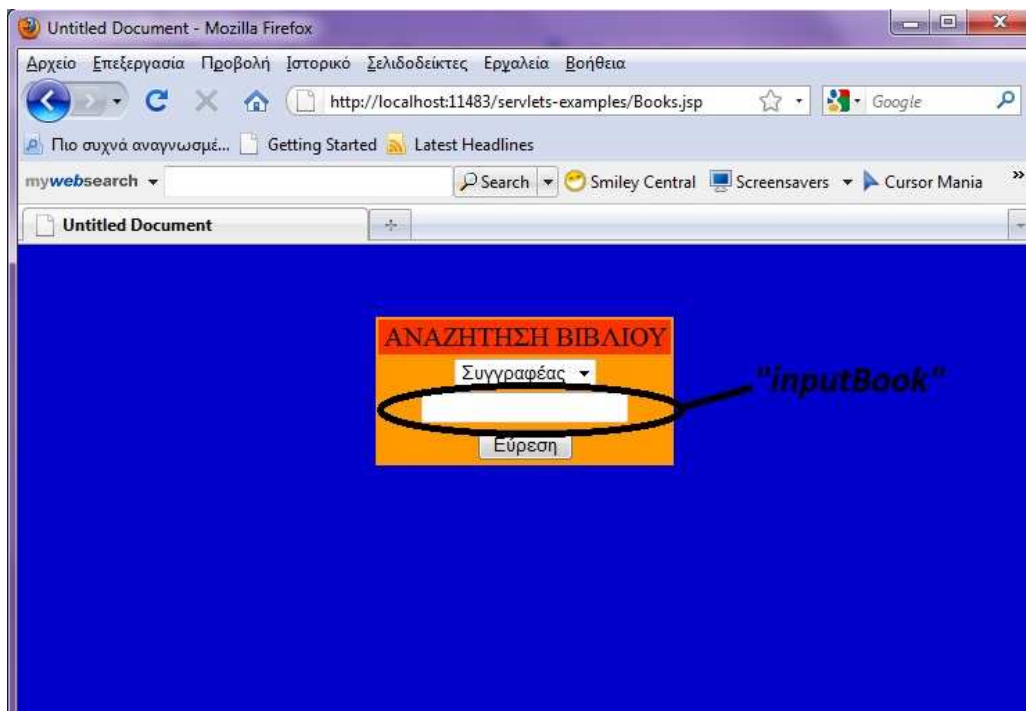
```
    {
```

```
        return "A Java Servlet Written By Ria";
```

```
    }
```

```
}
```

**Books.java**



**Κώδικας 8.39 - Το servlet λαμβάνει την τιμή της παραμέτρου με όνομα "jumpMenu" η οποία προσδιορίζει μία drop – down λίστα.**

```
<%@ page contentType="text/html; charset=utf-8" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
```



```

</head>

<body bgcolor="#0000CC">
<p>&nbsp;&nbsp;&nbsp;</p>
<form name="form1" action="servlet/Books">
<table width="30%" border="0" align="center" bgcolor="#FF9900" style="border-
bottom:medium; border-left:medium; border-right:medium; border-top:medium">
<tr>
<td bgcolor="#FF3300"><div style="text-align:center; font-size:20px;">ΑΝΑΖΗΤΗΣΗ
ΒΙΒΛΙΟΥ</div></td>
</tr>
<tr>
<td>
<div style="text-align:center">
<select name="jumpMenu" id="jumpMenu">
<option value="0">Συγγραφέας</option>
<option value="1">Τίτλος</option>
<option value="2">Κατηγορία</option>
<option value="3">Εκδότης</option>
<option value="4">ISBN</option>
</select>
</div></td>
</tr>
<tr>
<td><div style="text-align:center"><input type="text" name="inputBook"/></div></td>
</tr>
<tr>
<td><div style="text-align:center"><input type="submit" value="Εύρεση" /></div></td>
</tr>
</table>
</form>
<p>&nbsp;&nbsp;&nbsp;</p>
</body>
</html>

```

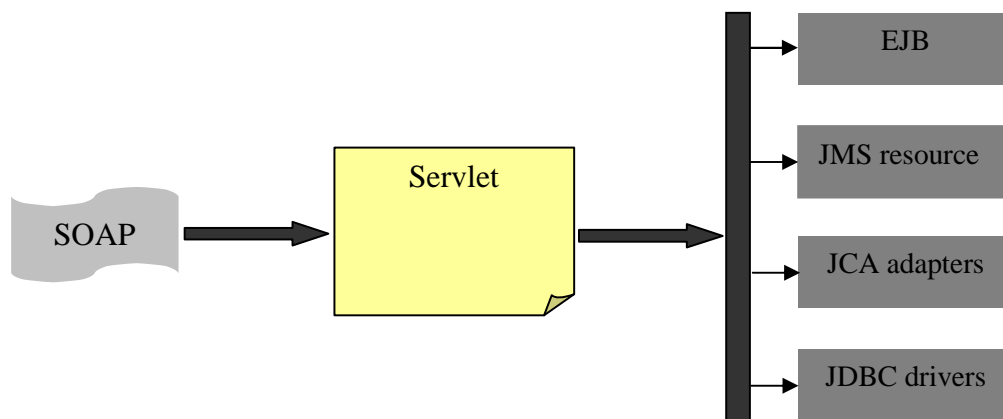
**Books.jsp**

## 8.5 J2EE και Web Services

Μια Web Service δεν αποτελεί από μόνη της μια εφαρμογή, αλλά είναι ένα κομμάτι που υλοποιείται ως μέρος μιας ολοκληρωμένης εφαρμογής η οποία χαρακτηρίζεται από αξιοπιστία, διαθεσιμότητα, ικανότητα στο να παρέχει υπηρεσίες, δυνατότητα συναλλαγών, ασφάλεια και ό,τι άλλο αποτελεί στόχο μιας Web Service. Το στάνταρ που ανταποκρίνεται σε αυτά τα χαρακτηριστικά είναι το J2EE, το οποίο συσχετίζεται με τις Web Services μέσω της Java. Από την μία πλευρά, οι Web Services μπορούν να υλοποιηθούν με Java και από την άλλη, το J2EE συνιστά μία πλατφόρμα εφαρμογών Java.

Προκειμένου να γίνει αντιληπτό με ποιόν τρόπο σχετίζονται οι Web Services με το J2EE, πρέπει να αναλύσουμε πώς λειτουργούν μαζί το J2EE και το SOAP. Απ,ότι γνωρίζουμε, το SOAP είναι ένα ασύρματο πρωτόκολλο, το οποίο ανήκει στο επίπεδο των πρωτοκόλλων Internet, HTTP, FTP και SMTP. Το J2EE υποστηρίζει αυτά τα πρωτόκολλα μέσω των servlets. Για τον λόγο αυτό, τα servlets μαζί με την τεχνολογία JSP αποτελούν το βασικό κομμάτι της πλατφόρμας J2EE.

Μέσα στο J2EE, τα servlets, τα αρχεία JSP, EJB, JMS, οι JDBC drivers και οι J2EE CA adapters παρέχουν πρόσβαση σε οποιαδήποτε πηγή χρειάζεται μία Web Service. Τα servlets και τα αρχεία JSP είναι συμβατά με ένα πολυάριθμο σύνολο πρωτοκόλλων του Internet. Τα servlets αποτελούν ένα σημαντικό κομμάτι των Web Services και λειτουργούν ως γέφυρα επικοινωνίας μεταξύ των Web Services και των εφαρμογών και υπηρεσιών του J2EE που περιέχονται μέσα σε έναν εξυπηρετητή εφαρμογών.



*Εικόνα 8.11 – Πώς πραγματοποιείται η επικοινωνία μέσα στο J2EE*

## 9 ΑΣΦΑΛΕΙΑ ΤΩΝ WEB SERVICES

### 9.1 Θέματα Ασφάλειας των Web Services

Η ιδέα των web services βασίζεται στην μεταφορά πληροφοριών και δεν έχει σημασία ο τύπος της πληροφορίας που μεταφέρεται. Μια ασφαλής web service είναι αυτή στην οποία ο αποστολέας είναι απόλυτα βέβαιος πως ο παραλήπτης των πληροφοριών είναι κάποιος που θα είναι σε θέση να αναλάβει και τον αντίστροφο ρόλο. Επίσης, μια ασφαλής web service είναι αυτή στην οποία οι πληροφορίες παραλαμβάνονται και πρόσβαση σε αυτές έχει μόνο ο κάθε αρμόδιος παραλήπτης. Για τους λόγους αυτούς, ισχύουν δύο κανόνες:

- Πρέπει να υπάρχει ένα είδος αυθεντικοποίησης.
- Πρέπει να υπάρχει ένα είδος ασφάλειας και προστασίας της ακεραιότητας των δεδομένων.

#### **Αυθεντικοποίηση**

Ο πάροχος υπηρεσίας καλείται να αποφασίσει ποιός θα είναι αυτός που καλεί την web service. Για παράδειγμα, αν μια web service δημοσιοποιεί πληροφορίες για τα προϊόντα και τις υπηρεσίες μιας επιχείρησης, δεν είναι απαραίτητο να γνωστοποιείται ποιός καλεί την υπηρεσία. Από την άλλη πλευρά, αν πραγματοποιούνται παραγγελίες, δηλαδή πάροχος λαμβάνει προσωπικές πληροφορίες του πελάτη, τότε πρέπει να γίνεται γνωστό ποιός καλεί την υπηρεσία.

#### **Εξουσιοδότηση**

Ο πάροχος της υπηρεσίας πρέπει να αποφασίσει εάν διαφορετικοί χρήστες της υπηρεσίας θα έχουν και πρόσβαση σε διαφορετικές λειτουργίες της υπηρεσίας.

Για παράδειγμα, για μια web service η οποία είναι σχεδιασμένη για χρήση μέσα σε ένα εταιρικό δίκτυο, ένας εξουσιοδοτημένος χρήστης μπορεί να υποβάλλει αναφορές εξόδων με χρήση της web service, αλλά μόνο διευθυντές μπορούν να εγκρίνουν αυτές τις αναφορές.

#### **Ιδιωτικοποίηση**

Είναι χρήσιμο να αποφασιστεί αν τα δεδομένα που μεταφέρονται από τον πελάτη στην web service θα πρέπει να προστατεύονται από τρίτους που επιχειρούν να έχουν πρόσβαση σε αυτές τις πληροφορίες.

#### **Ακεραιότητα**

Η ακεραιότητα αφορά στο αν ένα μήνυμα θα πρέπει να προστατευθεί κατά την μεταφορά του μεταξύ client και server. Δεν είναι πάντα απαραίτητη η κρυπτογράφηση του μηνύματος, αλλά πρέπει να εξασφαλιστεί ότι το μήνυμα δεν θα αλλοιωθεί πριν φθάσει στον παραλήπτη του.

## **9.2 Επίπεδα ασφάλειας των web services**

Η Τεχνολογία των web services περιλαμβάνει μηχανισμούς για την αντιμετώπιση κινδύνων και την παροχή ασφάλειας σε ό,τι αφορά τα παρακάτω επίπεδα:

- Επίπεδο ασφάλειας μεταφοράς δεδομένων, όπως τα τείχη προστασίας ( firewalls ), τα ιδεατά ιδιωτικά δίκτυα ( VPNs ), η επικύρωση ( authentication ), η μη αποκύρση ευθύνης ( non-repudiation ) και η κρυπτογράφηση ( encryption ).
- Επίπεδο ασφάλειας μηνύματος, όπως τα στοιχεία επικύρωσης ( authentication tokens ) για την πιστοποίηση της ταυτότητας των χρηστών και οι δηλώσεις έγκρισης ( authorization assertions ) για τον έλεγχο πρόσβασης στις web services.
- Επίπεδο ασφάλειας δεδομένων όπως η κρυπτογράφηση ( encryption ) και η ηλεκτρονική υπογραφή ( digital signature ) για την εξασφάλιση της ακεραιότητας των δεδομένων.
- Επίπεδο ασφάλειας όπως η διαχείριση ( management ), η αναγραφή ( logging ) και ο έλεγχος ( auditing ) για τον προσδιορισμό των προβλημάτων που πρέπει να αντιμετωπιστούν.

## **9.3 Προδιαγραφές αντιμετώπισης των θεμάτων ασφαλείας**

- WS – Security: Ορίζει μια αρχιτεκτονική που παρέχει ασφάλεια στην επικοινωνία.
- WS – Policy: Ορίζει ένα σύνολο κανόνων αλληλεπίδρασης των web services.
- WS – Trust: Εξασφαλίζει προστασία στην πραγματοποίηση συναλλαγών.
- WS – Privacy: Εξασφαλίζει την ιδιωτικότητα των προσωπικών δεδομένων.
- WS – Secure Conversation: Παρέχει ασφάλεια στην ανταλλαγή δεδομένων μεταξύ web services με τον συνδυασμό των κανόνων που ορίζονται από τις προδιαγραφές WS – Policy, WS – Trust, WS – Privacy.
- WS – Federation: Ορίζει ένα σύνολο κανόνων όσον αφορά την ταυτότητα σε ένα καταμεμημένο περιβάλλον τεχνολογίας.
- WS – Authorization: Μηχανισμός επεξεργασίας για την επικύρωση όσον αφορά την πρόσβαση και την ανταλλαγή πληροφοριών.

## **9.4 XML ψηφιακές υπογραφές και κρυπτογράφηση δεδομένων**

Το σχήμα της XML ψηφιακής υπογραφής ορίζει έναν πρότυπο τρόπο σύνταξης για την ψηφιακή καταχώρηση δεδομένων και την κωδικοποίηση αυτής της υπογραφής ως XML. Ο σκοπός των ψηφιακών υπογραφών είναι η προστασία της ακεραιότητας των επιχειρηματικών συναλλαγών στο διαδίκτυο. Η XML κρυπτογράφηση καθορίζει το πώς τα κρυπτογραφημένα δεδομένα και οι πληροφορίες που αποκρυπτογραφούν αυτά τα δεδομένα, μπορούν να κωδικοποιηθούν σε μορφή XML. Ο σκοπός της κρυπτογράφησης είναι η πιστοποίηση της εγκυρότητας των πληροφοριών που ανταλλάσσονται μεταξύ των web services.

Μέχρι στιγμής, μόνο τα εργαλεία ανάπτυξης των web services της IBM και η τεχνολογία .NET της Microsoft υποστηρίζουν την ψηφιακή υπογραφή και κρυπτογράφηση των SOAP μηνυμάτων. Μολονότι υποστηρίζουν και οι δύο πλευρές τις XML ψηφιακές υπογραφές και την κρυπτογράφηση, ωστόσο διαφέρει ο τρόπος υλοποίησης, δηλαδή το πώς αυτά τοποθετώνται μέσα στο ανάπτυγμα ενός SOAP μηνύματος.

## **9.5 Εργαλεία ανάπτυξης των Web Services**

### **Apache AXIS**

Είναι ένα εργαλείο της εταιρείας Apache το οποίο είναι συμβατό με το SOAP 1.1. Αρχικά ονομάστηκε Apache Soap αλλά αργότερα μετεξελίχθηκε σε περιβάλλον Java με την ονομασία Apache Axis. Για να λειτουργήσει σε ένα σύστημα το Apache Axis, απαιτεί την εγκατάσταση ενός διακομιστή εφαρμογών, όπως είναι ο Apache Tomcat και ενός SOAP Parser. Αρχικά, δημιουργείται η διαδικτυακή εφαρμογή σε γλώσσα Java και γίνεται η δημοσιοποίηση των μεθόδων της εφαρμογής και η ρύθμιση παραμέτρων. Το αρχείο WSDL παράγεται αυτόματα.

### **IBM Web Services Toolkit**

Στο πακέτο είναι ενσωματωμένη μια περιορισμένη έκδοση του διακομιστή εφαρμογών WebSphere της IBM. Το εργαλείο αυτό παρέχει την λειτουργικότητα των προηγούμενων πακέτων, όπως για παράδειγμα αυτόματη ή χειρωνακτική δημιουργία των αρχείων WSDL. Αυτό που το διακρίνει από τα προηγούμενα είναι η καλή τεκμηρίωση και το πλούσιο συνοδευτικό υλικό που περιλαμβάνει έτοιμα παραδείγματα σύνθετων διαδικτυακών εφαρμογών.

### **Microsoft SOAP Toolkit 2**

Με το Toolkit 2 δίνεται η δυνατότητα στον προγραμματιστή να αξιοποιεί τις διαδικτυακές εφαρμογές σε πλατφόρμες των Windows στις οποίες δεν έχει εγκατασταθεί το .NET Framework. Μπορεί να ενσωματωθεί στο περιβάλλον Visual Studio 6 και υποστηρίζει τις εκδόσεις SOAP 1.1 ,WSDL 1.1 και το UDDI.

## **Visual Studio .NET**

Είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού, από την Microsoft. Χρησιμοποιείται για την δημιουργία του γραφικού περιβάλλοντος εφαρμογών, την ανάπτυξη διαδικτυακών εφαρμογών, ιστοσελίδων, web services. Διαθέτει ένα ολοκληρωμένο σύστημα αποσφαλμάτωσης που λειτουργεί τόσο σε επίπεδο σφαλμάτων κώδικα χρήστη, όσο και σε επίπεδο μηχανής. Υποστηρίζει αρκετές γλώσσες προγραμματισμού, μεταξύ των οποίων είναι οι, C/C++, C++, VB.NET, Python, Ruby. Ακόμη υποστηρίζει τις XML/XSLT, HTML/XHTML Javascript και CSS.

## **Netbeans**

Όπως και το Visual Studio .NET, αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού και υποστηρίζει , HTML/XHTML, XML/XSLT, Javascript και CSS. Είναι γραμμένο σε γλώσσα Java και τρέχει σε οποιοδήποτε μηχάνημα που έχει εγκατεστημένο το JVM ( Java Virtual Machine ). Για το Netbeans θα γίνει εκτενής αναφορά στο παράρτημα που παρατίθεται στο τέλος.

## 10 ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ

### Πλατφόρμες και εργαλεία ανάπτυξης κώδικα

#### Γλώσσα προγραμματισμού

Ως γλώσσα υλοποίησης της εφαρμογής χρησιμοποιήθηκε η JAVA και συγκεκριμένα το πακέτο J2EE SDK 1.4. Η γλώσσα JAVA παρέχει τα εξής πλεονεκτήματα:

- Μεταφερσιμότητα κώδικα

Ο κώδικας είναι ανεξάρτητος από αρχιτεκτονικές και πλατφόρμες Λειτουργικού Συστήματος καθώς ο κώδικας που παράγει ο μεταγλωττιστής της JAVA δεν παράγεται για την αρχιτεκτονική ενός συστήματος, αλλά για την εικονική μηχανή της JAVA ( Java Virtual Machine ). Για τον λόγο αυτό, κάθε εφαρμογή η οποία είναι υλοποιημένη σε JAVA μπορεί να τρέχει σε οποιαδήποτε πλατφόρμα.

- Είναι αντικειμενοστραφής

Αυτό σημαίνει πως ο κώδικας ο οποίος είναι γραμμένος σε JAVA είναι επαναχρησιμοποιήσιμος.

- Παρέχει αξιοπιστία

Η JAVA έχει την δυνατότητα να αποτρέπει αλλά και να αποκαθιστά τα σφάλματα προγραμματισμού τα οποία μπορεί να έχουν ως αποτέλεσμα την δημιουργία εφαρμογών που χαρακτηρίζονται από αστάθεια και αναξιπιστία. Παρέχει μηχανισμούς για τον έλεγχο και την αντιμετώπιση λαθών και σφαλμάτων χρόνου εκτέλεσης ( run – time – errors ) και διευκολύνει την διαχείριση λαθών με τον χειρισμό των εξαιρέσεων.

- Είναι δυναμική και κατανεμημένη

Είναι δυναμική, που σημαίνει ότι μία κλάση της φορτώνεται δυναμικά στον μεταγλωττιστή οποιαδήποτε χρονική στιγμή κατά την διάρκεια εκτέλεσης. Είναι κατανεμημένη, καθώς παρέχει υποστήριξη σε επίπεδο δικτύου.

Παρέχει μία πληθώρα APIs για την ανάπτυξη ολοκληρωμένων εφαρμογών

- Παρέχει τεκμηρίωση

Συνοδεύεται από τεκμηρίωση και παρέχει σύνολο βοηθημάτων που διατίθενται μέσω του Internet.

#### Δημιουργία δυναμικών ιστοσελίδων

Για την δημιουργία του client χρησιμοποιήθηκε η τεχνολογία των Java Server Pages ( JSP ) η οποία παρέχει μία πληθώρα πλεονεκτημάτων σε σύγκριση με άλλες παρόμοιες τεχνολογίες:

- Επιτρέπει στους προγραμματιστές να δημιουργήσουν δυναμικές ιστοσελίδες οι οποίες μεταφράζονται κατά την διάρκεια εκτέλεσης.
- Αποτελεί κομμάτι του κώδικα που συνοδεύει την πλατφόρμα της JAVA.
- Έχουν την δυνατότητα να χειρίζονται αλλά και να συγχρονίζουν αιτήσεις πολλών χρηστών που στέλνονται ταυτόχρονα.
- Δίνει την δυνατότητα δημιουργίας με εύκολο τρόπο, τόσο στατικών, όσο και δυναμικών οντοτήτων.
- Είναι ανεξάρτητες από το σύστημα στο οποίο εκτελούνται.

- Παρέχουν την δυνατότητα επικοινωνίας μεταξύ οντοτήτων και κλάσεων με αποτελεσματικό τρόπο.
- Τα JSPs έχουν την δυνατότητα να μεταφέρουν τις αιτήσεις των χρηστών σε άλλους servers ή σελίδες JSP.
- Υποστηρίζονται από τους περισσότερους servers που κυκλοφορούν στην αγορά.
- Συνοδεύονται από μία πλήρη τεκμηρίωση η οποία είναι διαθέσιμη μέσω του Web.

### **Εργαλείο ανάπτυξης κώδικα**

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε το NetBeans IDE. Πρόκειται για ένα περιβάλλον ανάπτυξης λογισμικού, ανοικτού κώδικα, το οποίο υποστηρίζει την ανάπτυξη όλων των τύπων εφαρμογών Java ( J2EE, J2SE, Web Applications, EJB ) αλλά υποστηρίζει και πολλές άλλες γλώσσες προγραμματισμού όπως, C, C++, PHP, Python, JAvascript και παρέχει μία πληθώρα εργαλείων, μεταξύ των οποίων, εργαλεία σχεδιασμού SOA, εργαλεία XML Schemas και μοντέλα UML. Το NetBeans τρέχει σε πολλές πλατφόρμες Λειτουργικών Συστημάτων, συμπεριλαμβανομένων των Windows, Linux, Solaris και MAC OS X και είναι αρκετά εύκολο στην εγκατάσταση και χρήση του. Ακόμη ένα από τα πλεονεκτήματά του είναι το ότι παρέχει βοηθήματα τα οποία δίνουν την δυνατότητα στους προγραμματιστές να αναπτύσσουν ευκολότερα και αποδοτικότερα τις εφαρμογές τους. Στα βοηθήματα αυτά περιλαμβάνονται, το Source Code Editor και συγκεκριμένα, επισήμανση των συντακτικών κανόνων στον κώδικα, live επισήμανση λαθών και live parsing και αναδυόμενα παράθυρα βοήθειας. Επίσης, παρέχει βοηθήματα για την κατασκευή GUI ( Graphical User Interfaces ) και διαχείριση βάσεων δεδομένων.

### **Επιλογή Συστήματος Διαχείρισης Βάσεων Δεδομένων**

Ως Σύστημα Διαχείρισης Βάσεων Δεδομένων χρησιμοποιήθηκε η Apache Derby η οποία αποτελεί ένα Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων ανοικτού κώδικα. Οι βασικοί λόγοι επιλογής του συγκεκριμένου συστήματος, είναι το γεγονός ότι μπορεί να ενσωματωθεί σε JAVA εφαρμογές και το ότι μπορεί να χρησιμοποιηθεί για την διαχείριση online συναλλαγών. Στα πλεονεκτήματα που προσφέρει συμπεριλαμβάνονται, η ευκολία εγκατάστασης, η ευκολία χρήσης, το μικρό μέγεθος και η ασφάλεια. Επίσης, προσφέρει έναν ενσωματωμένο JDBC οδηγό που επιτρέπει την σύνδεση με ένα πρόγραμμα γραμμένο σε JAVA και υποστηρίζει την ανάπτυξη client/server εφαρμογών.

### **Επιλογή Web Server**

Ως Web Server επιλέχθηκε ο Apache Tomcat 6.0 ο οποίος παρέχει υλοποίηση ανοικτού κώδικα των τεχνολογίες των Servlets και JSPs.



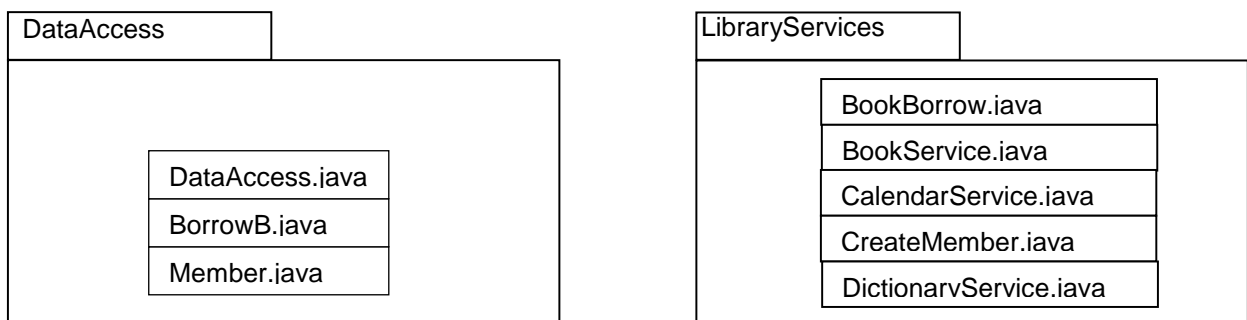
## Οργάνωση Κώδικα

### Πακέτο DataAccess

Περιέχει βοηθητικές κλάσεις και συγκεκριμένα, την κλάση DataAccess που πραγματοποιεί σύνδεση με την βάση δεδομένων και περιέχει μεθόδους που επιστρέφουν δεδομένα από αυτή και τις κλάσεις BorrowB και Member που δημιουργούν τα ανάλογα αντικείμενα κατά τον δανεισμό και την εγγραφή ενός μέλος αντίστοιχα.

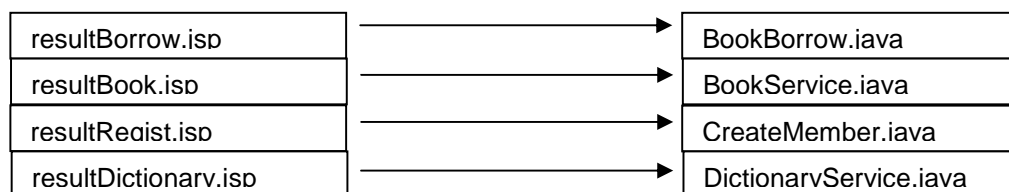
### Πακέτο LibraryServices

Περιέχει τις κλάσεις που πραγματοποιούν όλες τις λειτουργίες της εφαρμογής, χρησιμοποιώντας τις κλάσεις του παραπάνω πακέτου. Συγκεκριμένα, συνιστούν τις κλάσεις, οι οποίες υλοποιούν τα web services της εφαρμογής. Οι κλάσεις αυτές είναι: BookBorrow, BookService, CalendarService, CreateMember, DictionaryService.



### Directory Web Pages

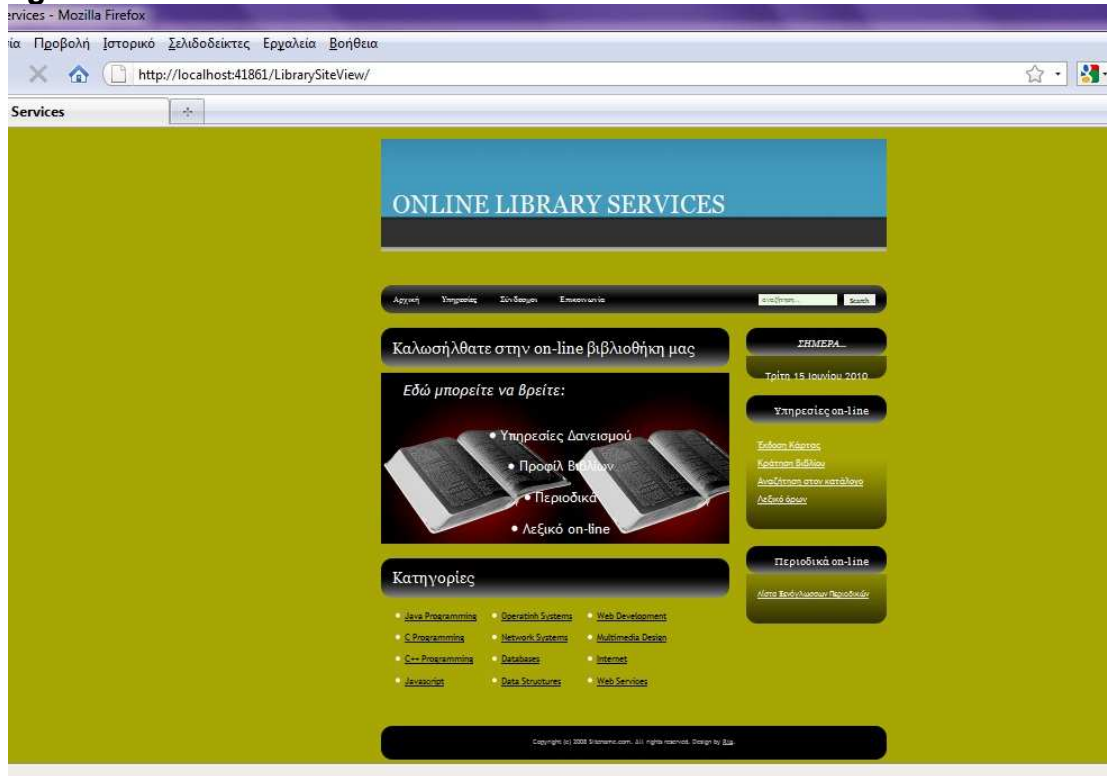
Περιέχει όλα τα αρχεία JSP για την ανάπτυξη της ιστοσελίδας. Συγκεκριμένα, περιέχει τα αρχείο index.jsp που αντιστοιχεί στην κεντρική σελίδα και τα αρχεία που καλούν τα web services.



## Εκτέλεση Εφαρμογής

Παρακάτω παρουσιάζεται η αρχική σελίδα στην οποία περιέχεται το σύνολο των web services. Αυτά είναι: Έκδοση Κάρτας, Αναζήτηση Βιβλίου, Κράτηση Βιβλίου και Λεξικό που υλοποιούνται από τις κλάσεις: CreateMember, BookBorrow, BookService και DictionaryService αντίστοιχα. Επίσης, η ημερομηνία που εμφανίζεται στην σελίδα προκύπτει ως αποτέλεσμα της κλήσης μεθόδου του web service, CalendarService που επιστρέφει την ημερομηνία στην μορφή που παρουσιάζεται παρακάτω.

## Main Page



## Index.jsp

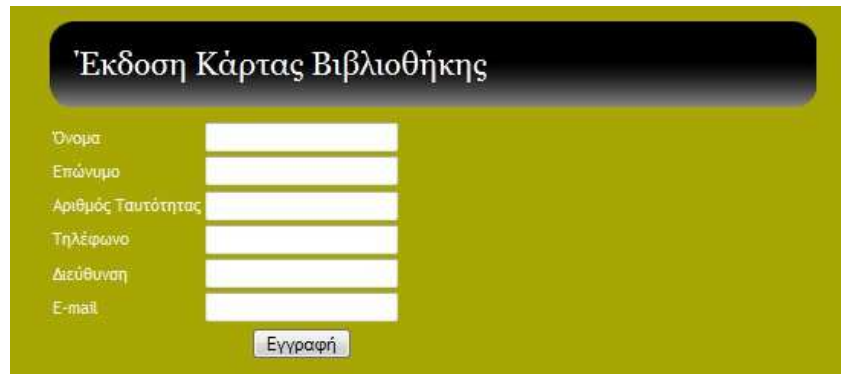
### Web Service: CalendarService



### Web Services



### Web Service: CreateMember



Έκδοση Κάρτας Βιβλιοθήκης

Όνομα

Επώνυμο

Αριθμός Ταυτότητας

Τηλέφωνο

Διεύθυνση

E-mail

Υποβολή

### Web Service: BookBorrow



Κράτηση Βιβλίου

Όνομα

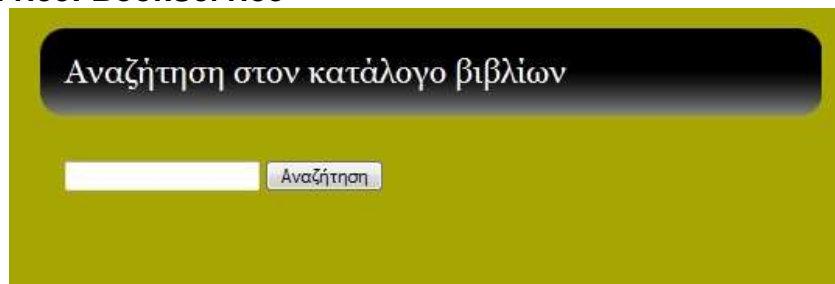
Επώνυμο

Αριθμός Ταυτότητας

ISBN Βιβλίου

Υποβολή

### Web Service: BookService



Αναζήτηση στον κατάλογο βιβλίων

Υποβολή

### Web Service: DictionaryService



Λεξικό Τεχνολογικών Όρων

ΑΒΓΔΕΕΦΓΗΙΪΚΛΜΝΟΡΡΣΤΥΥΧΥΖ

Υποβολή

Έστω ότι θέλουμε να καλέσουμε το web service, DictionaryService, το οποίο υλοποιείται με τον παρακάτω κώδικα:

```

package LibraryServices;

import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import java.util.Properties;

import DataAccess.*;

@WebService()

public class DictionaryService {

    private DataAccess data;
    private String result;
    Properties prop = new Properties();

    /**
     * Web service operation
     */
    @WebMethod(operationName = "getDescription")
    public String getDescription(@WebParam(name = "inputTerm")
    String inputTerm)

        data = new DataAccess();

        try
        {
            result = data.getTermsDescription(inputTerm);
        }
        catch(Exception e)
        {
            return null;
        }

        return result;
    }
}

```

Κατά το deployment του web service παράγεται το παρακάτω αρχείο wsdl το οποίο αποτελεί την περιγραφή του.

```

<?xml version='1.0' encoding='UTF-8?'><!-- Published by JAX-WS RI at http://jax-
ws.dev.java.net. RI's version is JAX-WS RI 2.2-hudson-752-. --><!-- Generated by JAX-WS
RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2-hudson-752-. --><definitions
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd"          xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"      xmlns:tns="http://LibraryServices/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"      targetNamespace="http://LibraryServices/"
name="DictionaryServiceService">
<types>
<xsd:schema>
<xsd:import          namespace="http://LibraryServices/"
schemaLocation="http://localhost:41861/LibrarySiteView/DictionaryServiceService?xsd=1" />
</xsd:schema>
</types>
<message name="getDescription">
<part name="parameters" element="tns:getDescription" />
</message>
<message name="getDescriptionResponse">
<part name="parameters" element="tns:getDescriptionResponse" />
</message>
<portType name="DictionaryService">
<operation name="getDescription">
<input      wsam:Action="http://LibraryServices/DictionaryService/getDescriptionRequest"
message="tns:getDescription" />
<output      wsam:Action="http://LibraryServices/DictionaryService/getDescriptionResponse"
message="tns:getDescriptionResponse" />
</operation>
</portType>
<binding name="DictionaryServicePortBinding" type="tns:DictionaryService">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
<operation name="getDescription">
<soap:operation soapAction="" />
<input>
<soap:body use="literal" />
</input>
<output>
<soap:body use="literal" />
</output>
</operation>
</binding>
<service name="DictionaryServiceService">
<port name="DictionaryServicePort" binding="tns:DictionaryServicePortBinding">
<soap:address location="http://localhost:41861/LibrarySiteView/DictionaryServiceService" />
</port>
</service>
</definitions>

```

## DictionaryService.wsdl

Επίσης παράγεται το παρακάτω configuration file του web service που ορίζει το namespace:

```
<?xml version='1.0' encoding='UTF-8'?><!-- Published by JAX-WS RI at http://jax-  
ws.dev.java.net. RI's version is JAX-WS RI 2.2-hudson-752-. --><xs:schema  
xmlns:tns="http://LibraryServices/" xmlns:xs="http://www.w3.org/2001/XMLSchema"  
version="1.0" targetNamespace="http://LibraryServices/">
```

```
<xs:element name="getDescription" type="tns:getDescription" />
```

```
<xs:element name="getDescriptionResponse" type="tns:getDescriptionResponse" />
```

```
<xs:complexType name="getDescription">
```

```
<xs:sequence>
```

```
<xs:element name="inputTerm" type="xs:string" minOccurs="0" />
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
<xs:complexType name="getDescriptionResponse">
```

```
<xs:sequence>
```

```
<xs:element name="return" type="xs:string" minOccurs="0" />
```

```
</xs:sequence>
```

```
</xs:complexType>
```

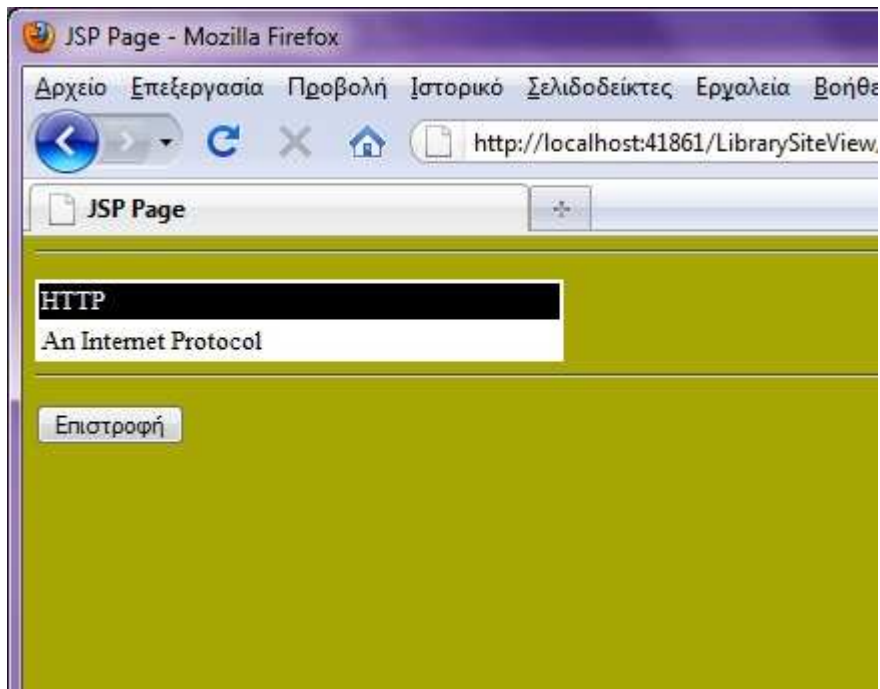
```
</xs:schema>
```

## DictionaryService.xsd

Πληκτρολογούμε έναν όρο στο πεδίο αναζήτησης και καλείται το web service DictionaryService μέσα από το αρχείο resultDictionary.jsp το οποίο μας δίνει και το αποτέλεσμα. Ο κώδικας παρουσιάζεται πιο κάτω:



The image shows a web interface with a dark green background. At the top, there is a black rounded rectangle containing the text "Λεξικό Τεχνολογικών Όρων" in white. Below this, there is a horizontal line of letters "A B C D E F G H I J K L M N O P Q R S T U V W X Y Z" in a light blue font. Underneath the letters, there is a white input field containing the text "HTTP|". To the right of the input field is a blue button with the text "Αναζήτηση" in white.



Το αποτέλεσμα της κλήσης του web service: DictionaryService

```
<%--
```

```
Document : resultDictionary
```

```
Created on : 6 IouV 2010, 12:02:53 πμ
```

```
Author : RIA
```

```
--%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>JSP Page</title>
```

```
</head>
```

```
<body bgcolor="#A8A604">
```

```
<%-- <jsp:include page="Dictionary.jsp"/>-->
```

```
<%-- start web service invocation --%>
```

```
<hr>
```

```
<p>
```

```
<%
```

```
try {
```

```
java.lang.String inputTerm = request.getParameter("inputTerm");
```

```
request.setCharacterEncoding("UTF-8");
```

```
library_clients.DictionaryServiceService service = new library_clients.DictionaryServiceService();
```

```
library_clients.DictionaryService port = service.getDictionaryServicePort();
```



```

        java.lang.String result = port.getDescription(inputTerm);

        out.println("<html><head><title></title></head><body><table bgcolor = 'white' width = '20%'><tr bgcolor
=          'black'          style='color:white;          font-
size:16px'><td>"+inputTerm+"</td></tr><tr><td>"+result+"</td></tr></table></body></html>");

    } catch (Exception ex) {

        // TODO handle custom exceptions here

    }

    %>

<%-- end web service invocation --%>

</p>

<hr>

<p align="center">

<form name="returnF" action="Dictionary.jsp">

    <input type="submit" name="submitR" value="Επιστροφή">

</form>

</body>

</html>

```

## resultDictionary.jsp

Κατά την εκτέλεση του web service που παρουσιάστηκε, γίνεται ανταλλαγή των παρακάτω SOAP μηνυμάτων μεταξύ client και server:

```

<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:getDescription xmlns:ns2="http://LibraryServices/">
      <inputTerm>HTTP</inputTerm>
    </ns2:getDescription>
  </S:Body>
</S:Envelope>

```

## SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getDescriptionResponse xmlns:ns2="http://LibraryServices/">
      <return>An Internet Protocol</return>
    </ns2:getDescriptionResponse>
  </S:Body>
</S:Envelope>
```

## SOAP Response

## BIBΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

- 1) *Programming Web Services with SOAP*, Doug Tidwell, James Snell, Pavel Kulchenko – O' Reilly ( December 2001, ISBN: 0-596-00095-2 )
- 2) *Web Services Essentials*, Ethan Cerami – O' Reilly ( February 2002, ISBN: 0-596-00224-6 )
- 3) *Web Services and Service Oriented Architectures*, Gerhard Wiehler - Siemens
- 4) *Developer's Guide to Building XML-based Web Services with the Java 2 Platform Enterprise edition ( J2EE )*, James Cao – The Middleware company ( June 2001 )
- 5) *Core Servlets and Java Server Pages*, Marty Hall – Sun microsystems
- 6) *Java & XML*, Brett McLaughlin – O' Reilly ( September 2001 )
- 7) *JavaServer Pages*, Hans Bergsten – O' Reilly( December 2000 )
- 8) *JavaServlet Programming*, Jason Hunter with William Grawford – O' Reilly ( October 1998 )
- 9) *XML Schema*, Eric van der Vlist – O' Reilly ( June 2002 )
- 10) *Learning XML*, Erik T. Ray – O' Reilly (January 2001, ISBN: 0-59600-046-4 )
- 11) *XML Programming in Java*, Doug Tidwell – developerWorks XML Team ( September 1999 )
- 12) *Programming Web Services with XML – RPC*, Simon St. Laurent, Joe Johnston, Edd Dumbill – O' Reilly ( June 2001, ISBN: 0-596-00119-3 )
- 13) *XML By Example*, Benoit Marchal – John Pierce ( Copyright: 2000, ISBN: 0-7897-2242-9 )
- 14) *XML Web Services in the Organization*, Chris Boar – Microsoft Corporation ( Copyright: 2003, ISBN: 0-7356-1882-8 )
- 15) *Developing XML Solutions*, Jake Sturm – Microsoft Corporation ( Copyright: 2000, ISBN: 0-7356-0796-6 )
- 16) *XML in Action Web Technology*, William J. Pardi – Microsoft Corporation ( Copyright: 1999, ISBN: 0-7356-0562-9 )
- 17) *XML Programming*, Brad Leupen – Microsoft Corporation ( Copyright: 2002, ISBN: 0-7356-1185-8 )
- 18) *Real World XML*, Steven Holzner – New Riders Publishing ( January 16, 2003, ISBN: 0-7357-1286-7 )
- 19) *Designing Enterprise Applications with the Java™ 2 Platform, Enterprise Edition*, Nicholas Kassem and the Enterprise Team – Sun Microsystems ( October 3, 2000 )
- 20) *Java™ 2 Platform Enterprise Edition Specification v14*, Bill Shannon – Sun Microsystems ( November 24, 2003 )
- 21) *330 Java Tips*, <http://www.JavaFAQ.nu>
- 22) *Extending Your Markup: An XML Tutorial*, André Bergholz – Standford University
- 23) *Working with XML The Java API for XML Parsing ( JAXP ) Tutorial*, Eric Armstrong – Sun Microsystems ( August 21, 2001 download site: <http://java.sun.com/xml/jaxp-1.1/docs/tutorial/index.html> )
- 24) *The J2EE™ Tutorial*, Stephanie Bodoff, Dale Green, Kim Haase, Eric Jendrock, Monica Pawlan, Beth Stearns – Sun Microsystems ( 2002 )
- 25) *Advanced Techniques for Java Developers*, Daniel J. Berg, J. Steven Fritzing – WILEY ( 1997 )

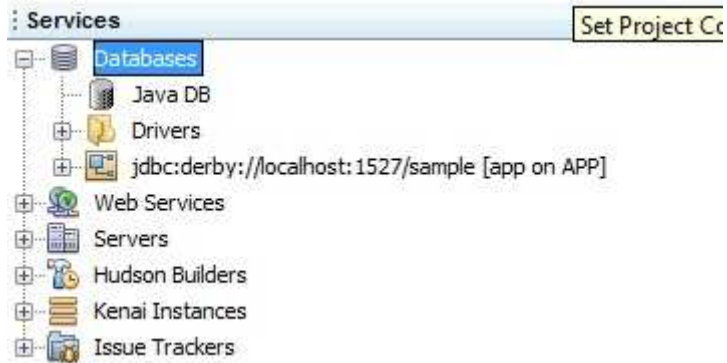
### Πηγές Ιστοσελίδων

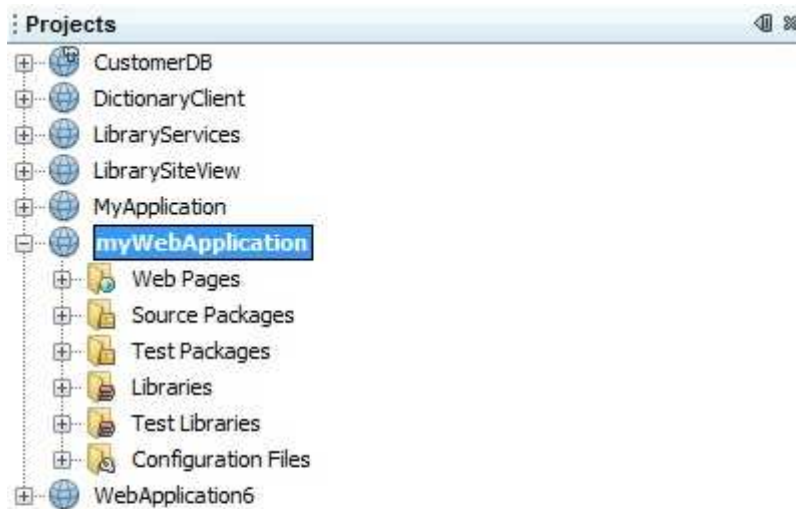
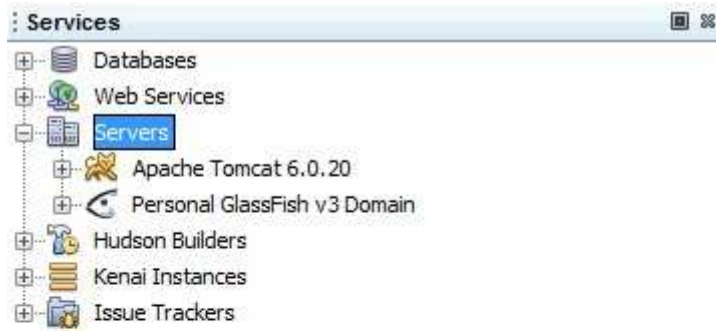
- I. <http://java.sun.com>
- II. <http://ws.apache.org>

- III. <http://www.amazon.com>
- IV. <http://www.brics.com>
- V. <http://www.go-online.gr>
- VI. <http://www.w3.org>
- VII. <http://www.java.net>
- VIII. <http://www.xmethods.com>
- IX. <http://www.xmlrpc.com>
- X. <http://xml.coverpages.org>
- XI. <http://wikipedia.org>

# ΠΑΡΑΡΤΗΜΑ

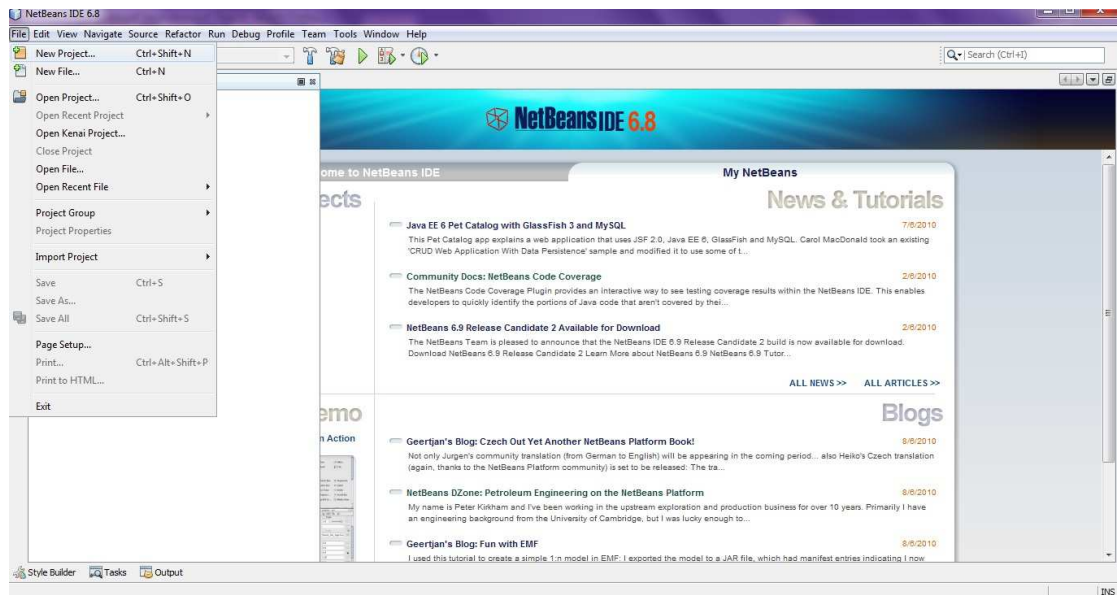
## Επίδειξη Δημιουργίας Web Service με το NetBeans



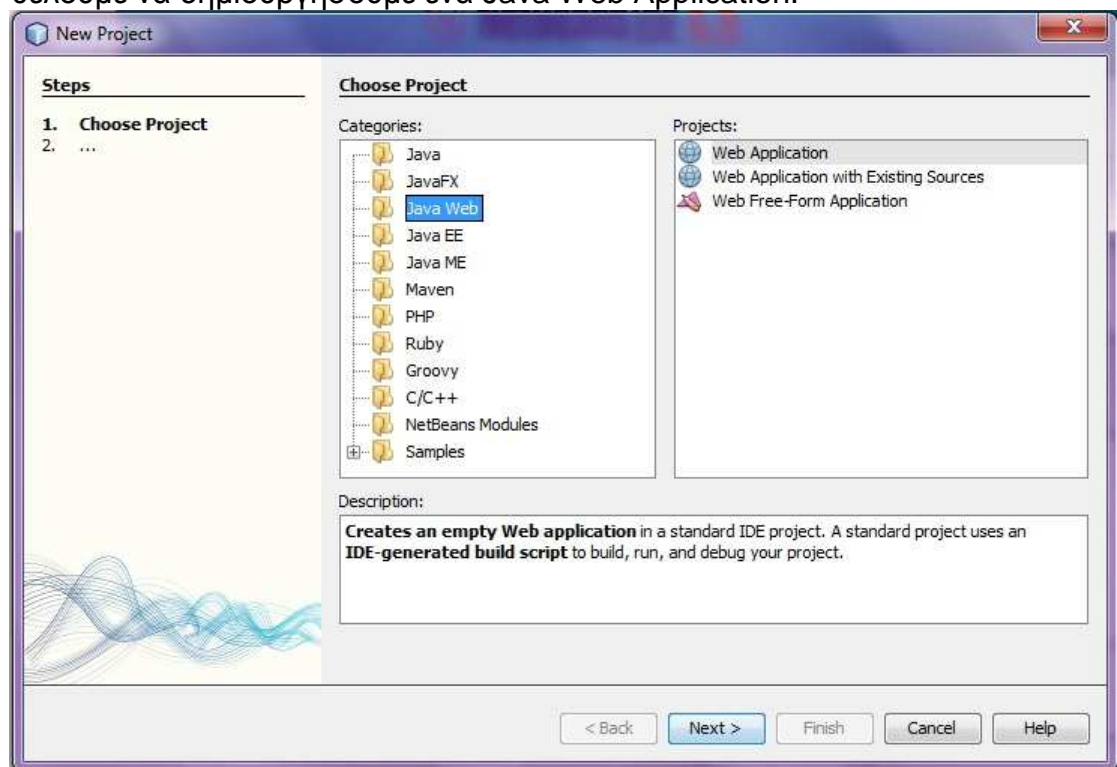


## Δημιουργία Web Service με το NetBeans

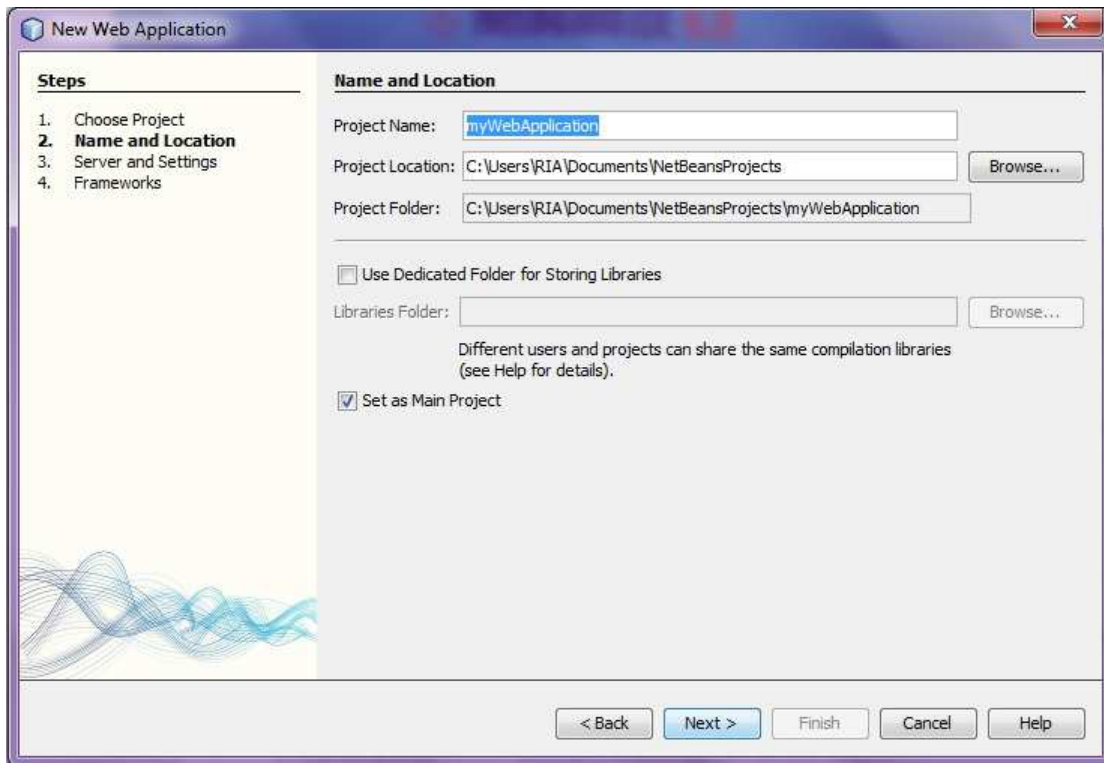
Για την δημιουργία νέου Project αρχικά, επιλέγουμε από το μενού: Αρχείο > Νέο Project.



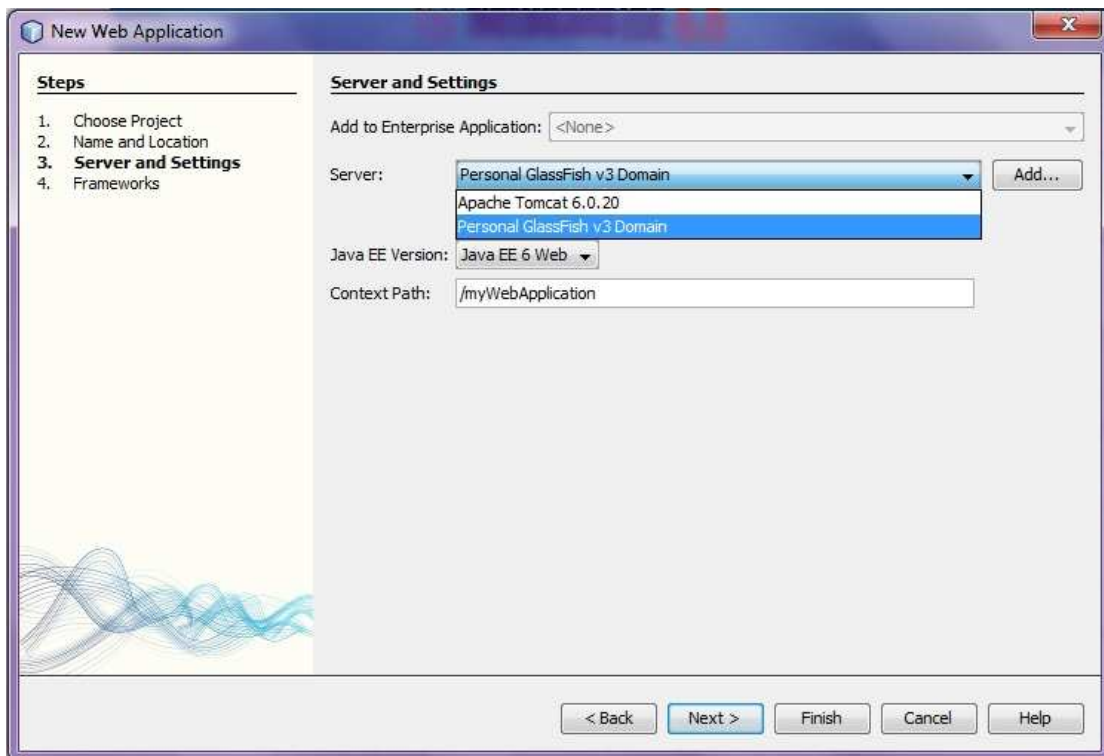
Στην συνέχεια, από το παράθυρο δημιουργίας νέου Project επιλέγουμε την Κατηγορία στην οποία επιθυμούμε να ανήκει. Στην συγκεκριμένη περίπτωση θέλουμε να δημιουργήσουμε ένα Java Web Application.



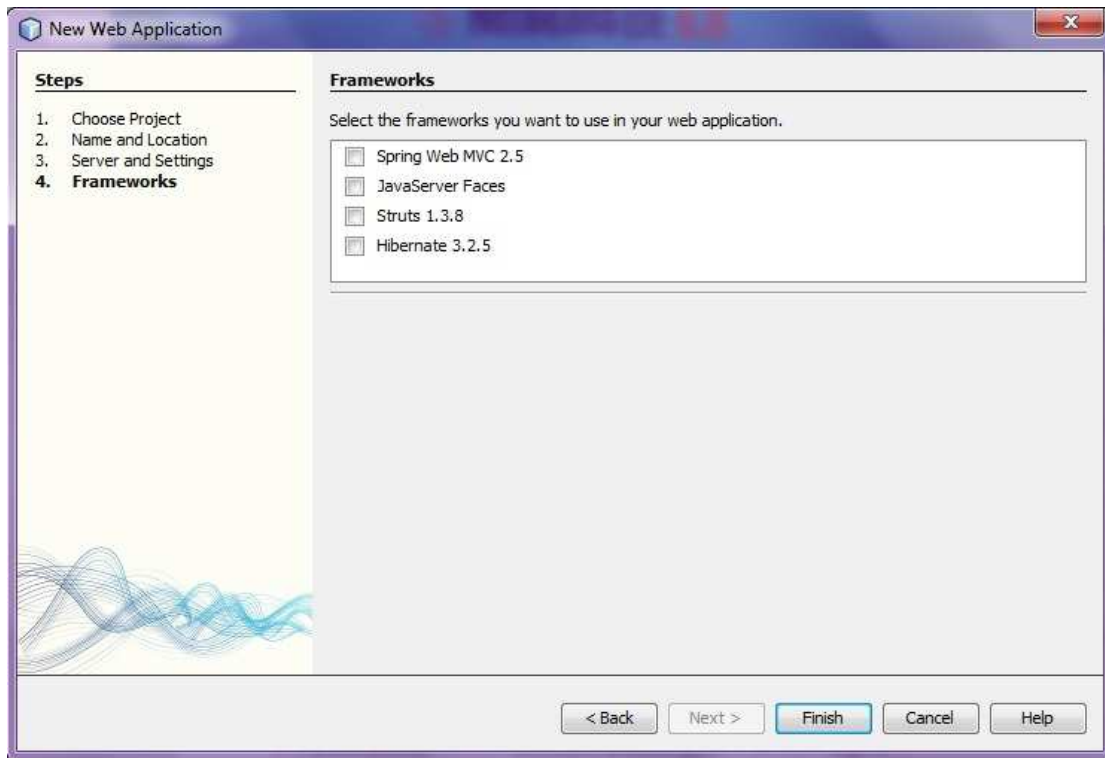
Στο επόμενο βήμα, πληκτρολογούμε το όνομα του Project και ελέγχουμε την τοποθεσία στην οποία θα αποθηκευθούν τα αρχεία του. Προτείνεται να διατηρούμε τα προεπιλεγμένα Locations και Folders.



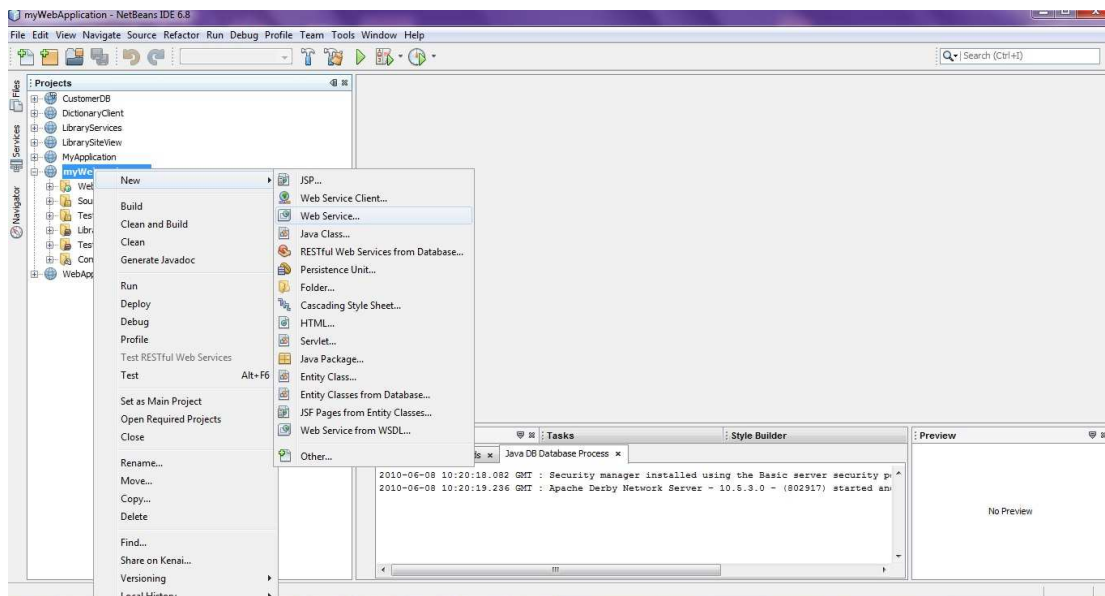
Επιλέγουμε τον Server στον οποίο θα ανεβάσουμε και θα τρέξουμε την εφαρμογή. Εδώ, δίνονται δύο επιλογές, των εγκατεστημένων servers Apache Tomcat και GlassFish.



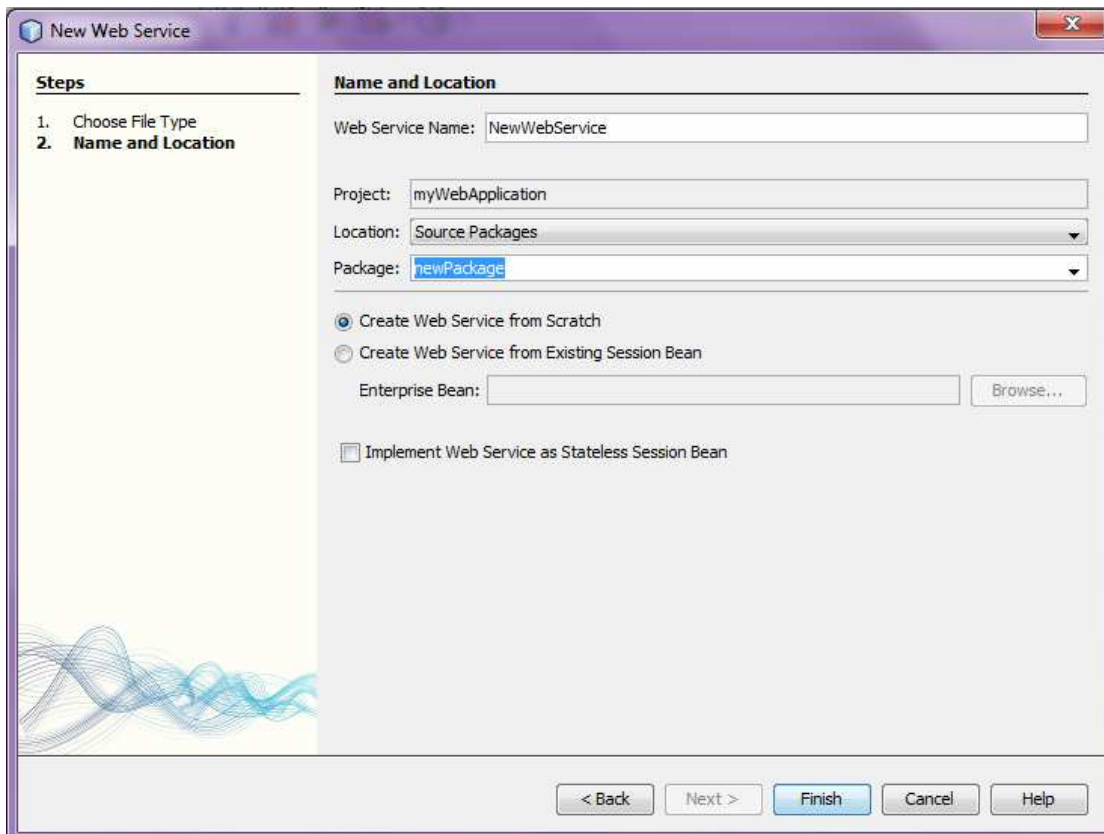
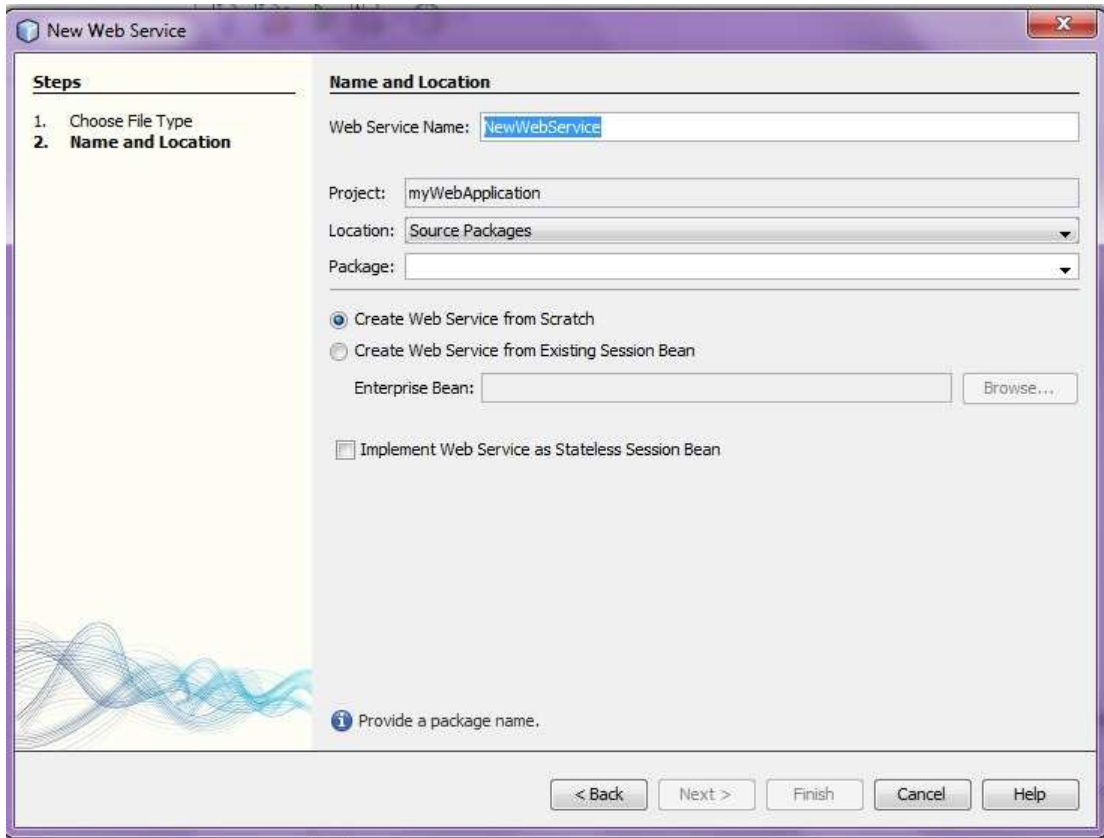




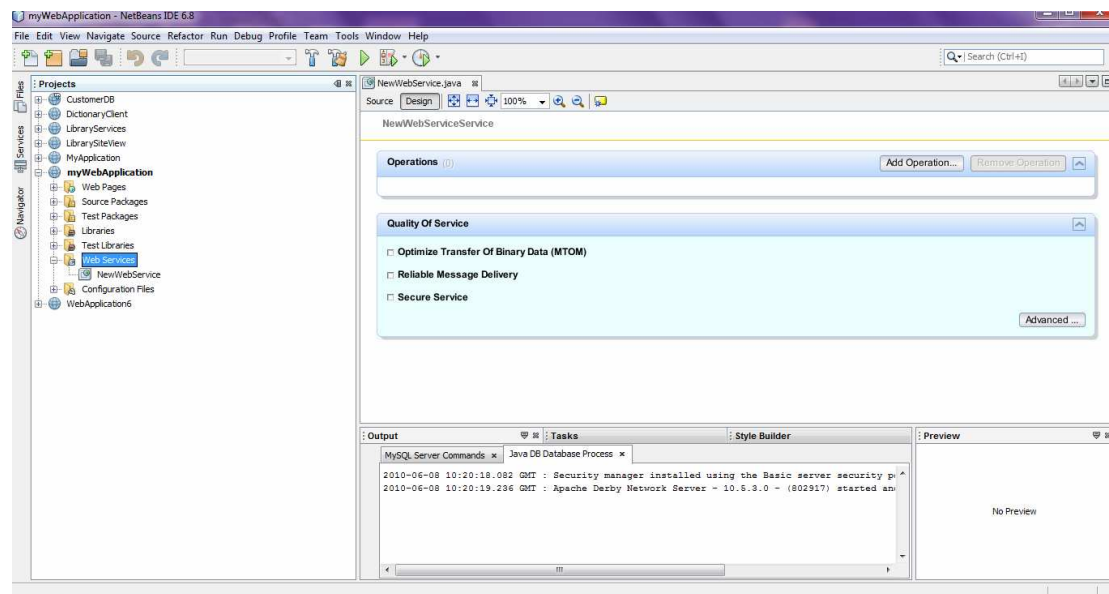
Το επόμενο βήμα μας είναι να δημιουργήσουμε ένα web service. Από το projects panel, κάνουμε δεξιά κλικ επάνω στο όνομα του Project που μόλις πριν δημιουργήσαμε. Επιλέγουμε New > Web Service.



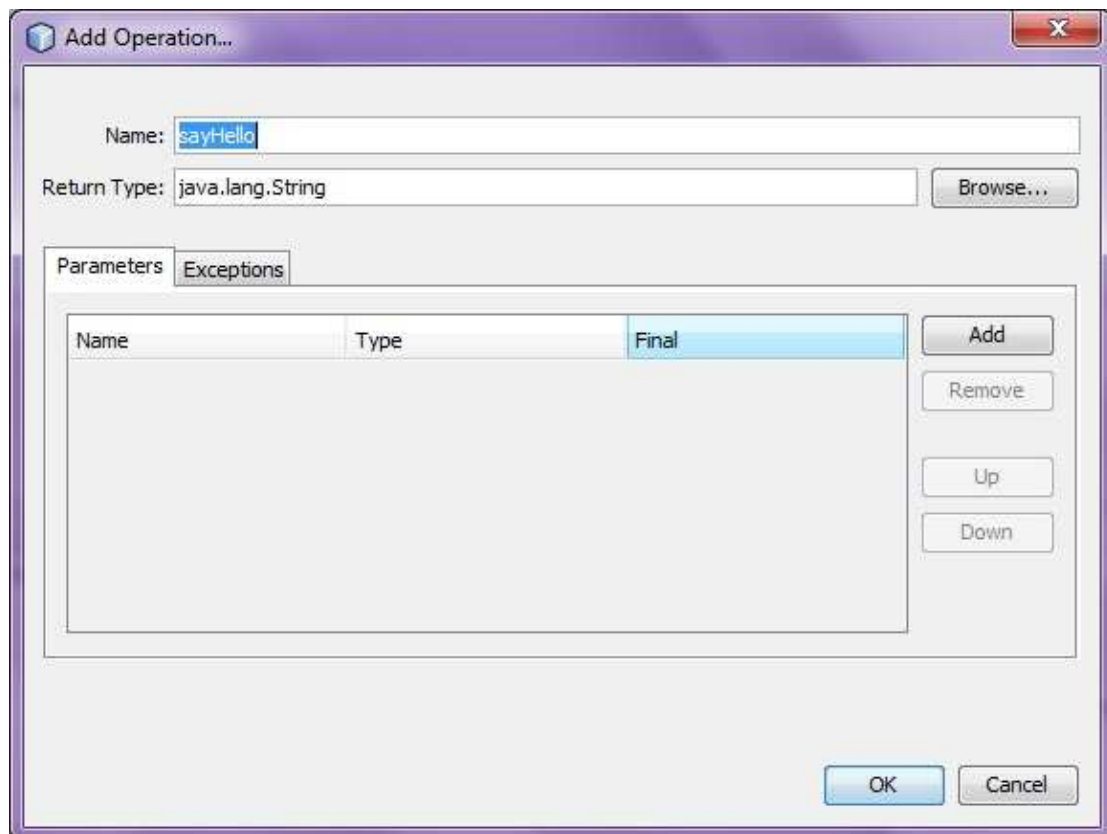
Προσδιορίζουμε το όνομα του web service και το όνομα του πακέτου στο οποίο θα τοποθετηθεί.



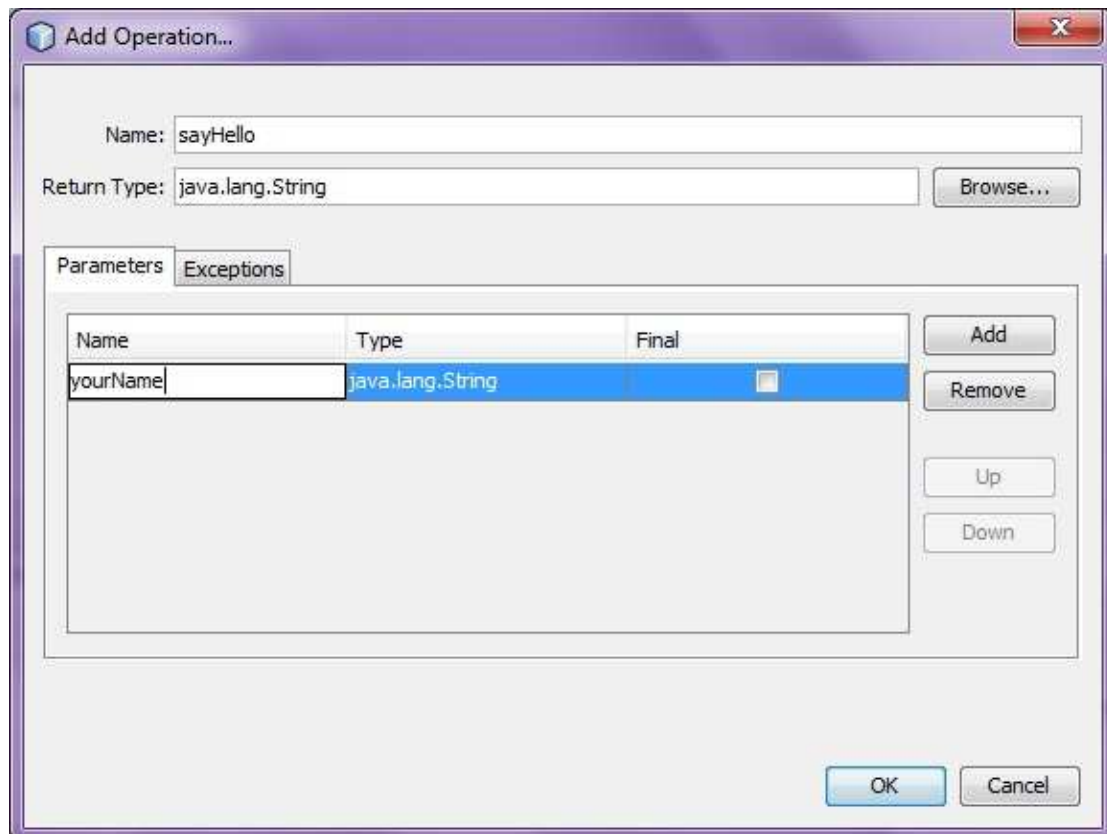
Παρακάτω, παρουσιάζεται το panel του web service που μόλις δημιουργήσαμε, σε δύο όψεις: σχεδίασης και κώδικα.



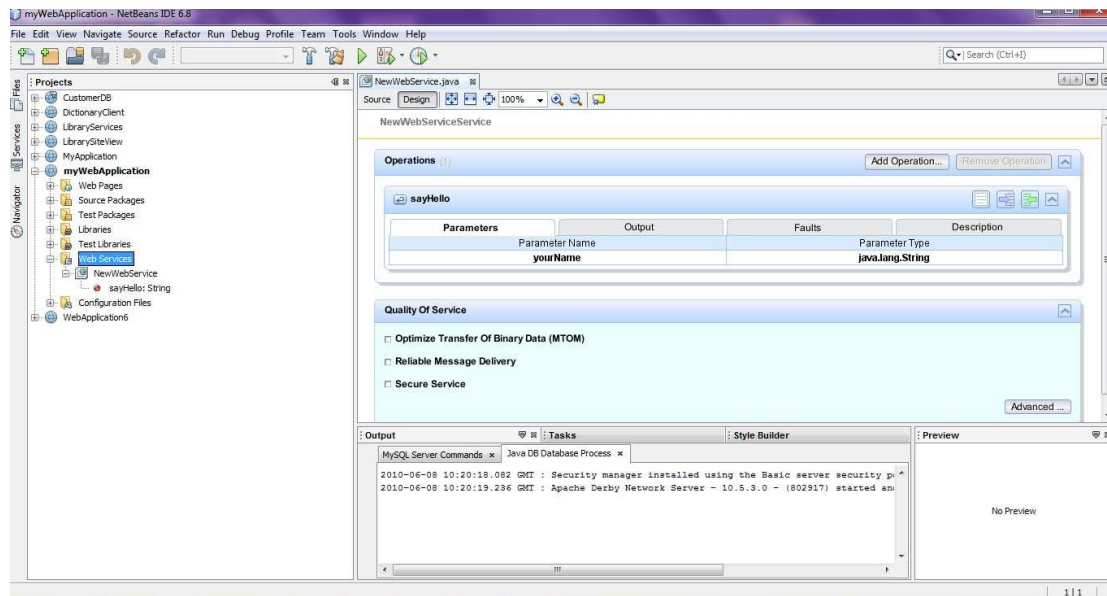
Στην συνέχεια, έστω ότι θέλουμε να προσθέσουμε μία μέθοδο στο web service. Από το panel στην σχεδιαστική του όψη, επιλέγουμε: Προσθήκη Operation, προσδιορίζουμε το όνομα της μεθόδου, καθώς και τον τύπο της επιστρεφόμενης τιμής.



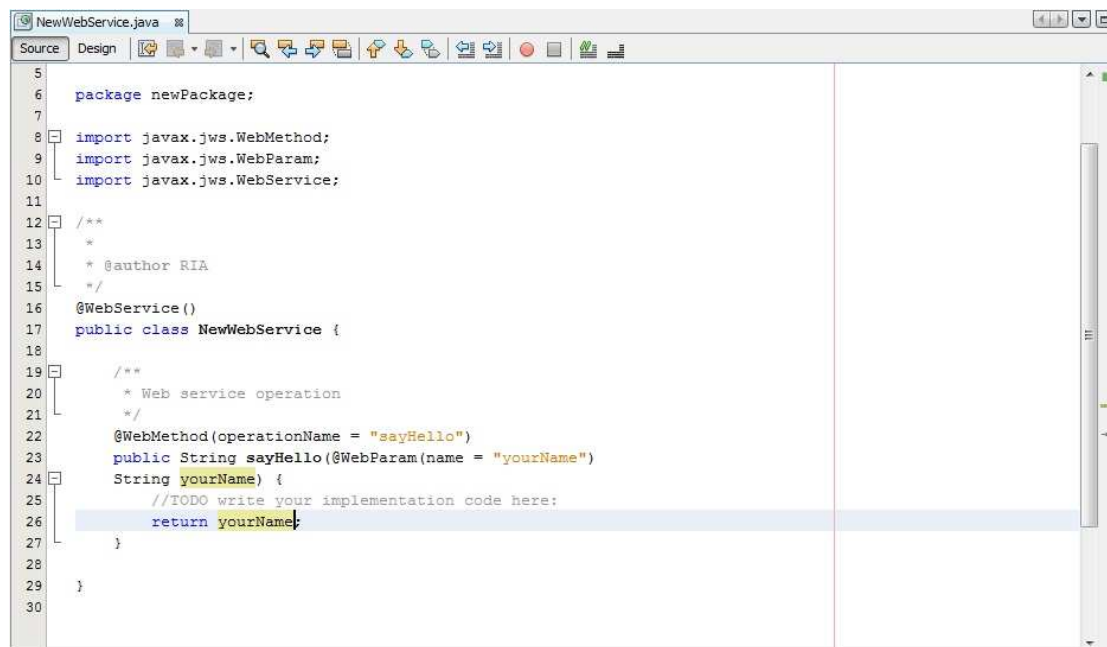
Ορίζουμε τις παραμέτρους της μεθόδου και προσδιορίζουμε τον τύπο τους.



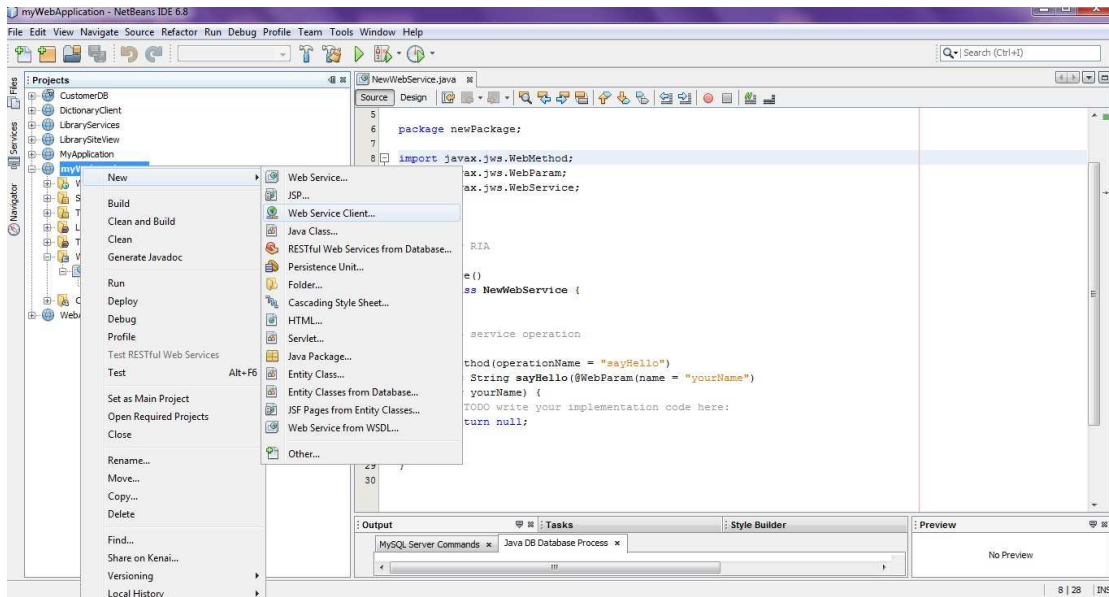
Μόλις ορίσαμε την μέθοδο sayHello() του web service NewWebService() η οποία δέχεται μία παράμετρο, yourName τύπου String και επιστρέφει μία συμβολοσειρά.



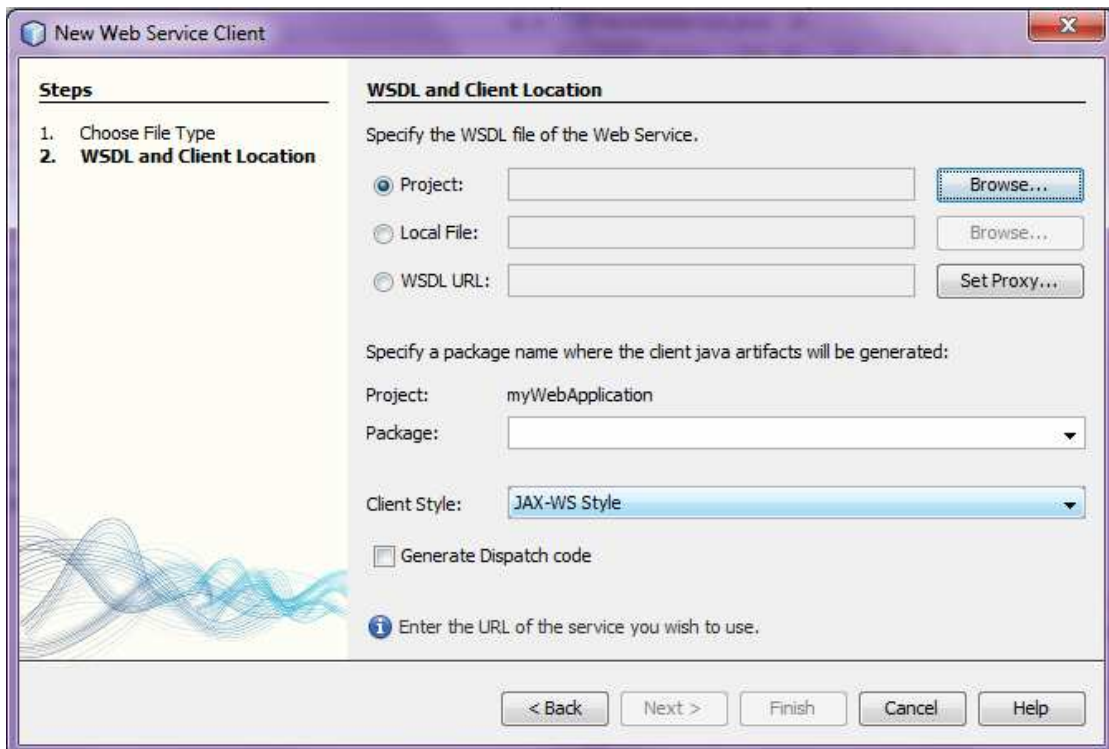
Παρακάτω παρουσιάζεται ο κώδικας του web service που υλοποιείται με την κλάση NewWebService.

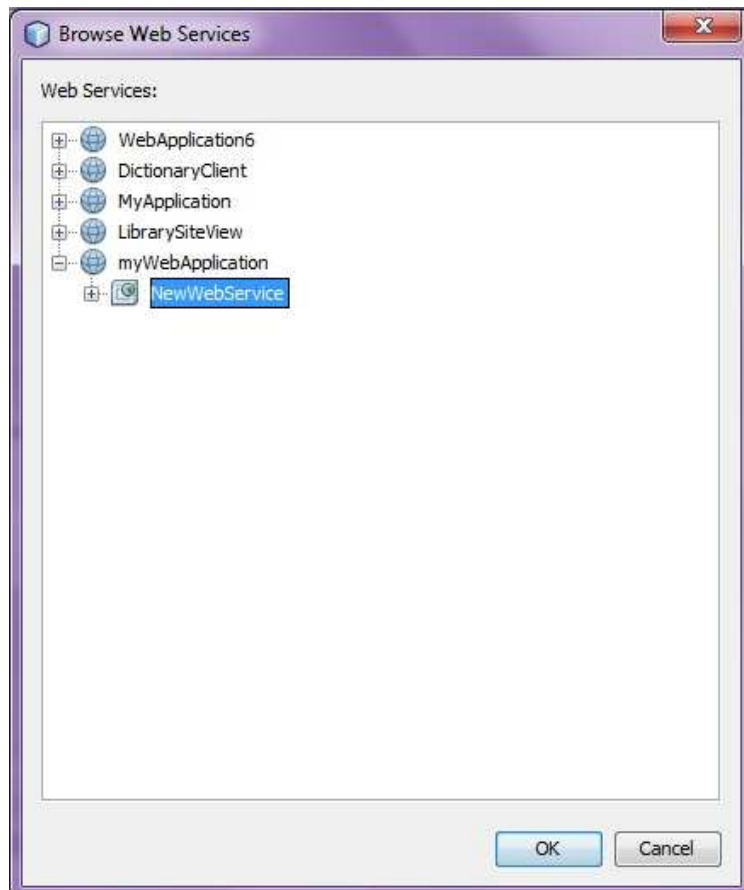


Παρακάτω δημιουργούμε τον client του web service ώστε να το κάνουμε deploy:

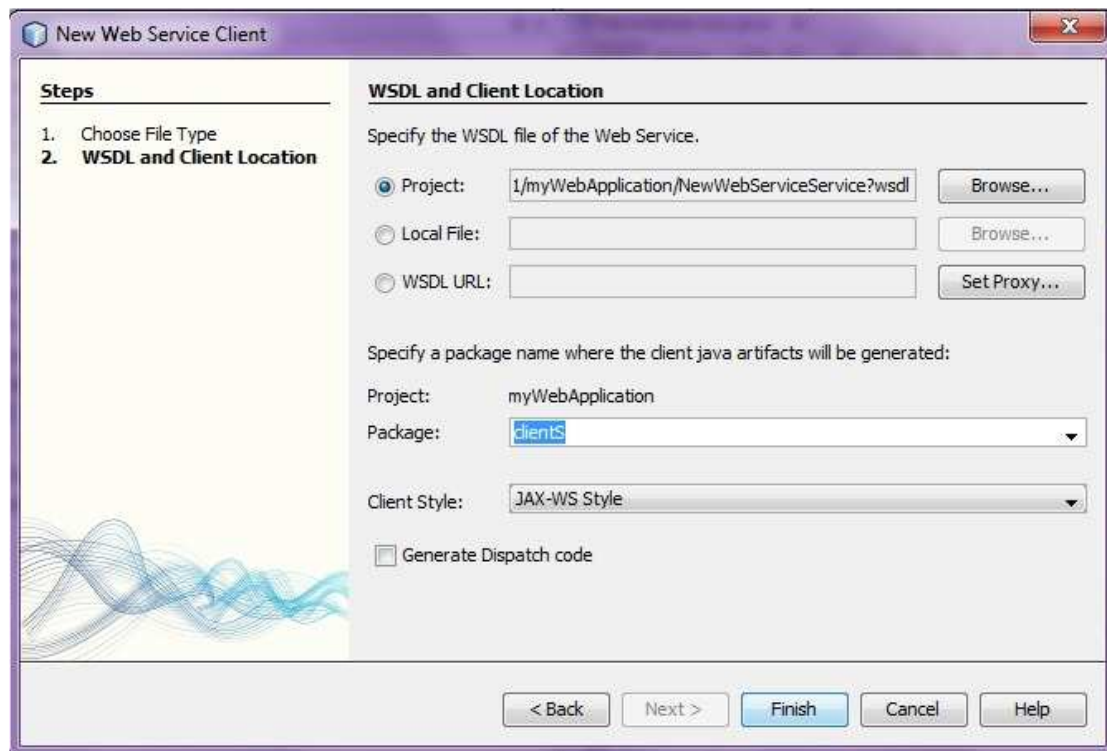


Από το project, επιλέγουμε το web service για το οποίο θέλουμε να δημιουργήσουμε τον client:

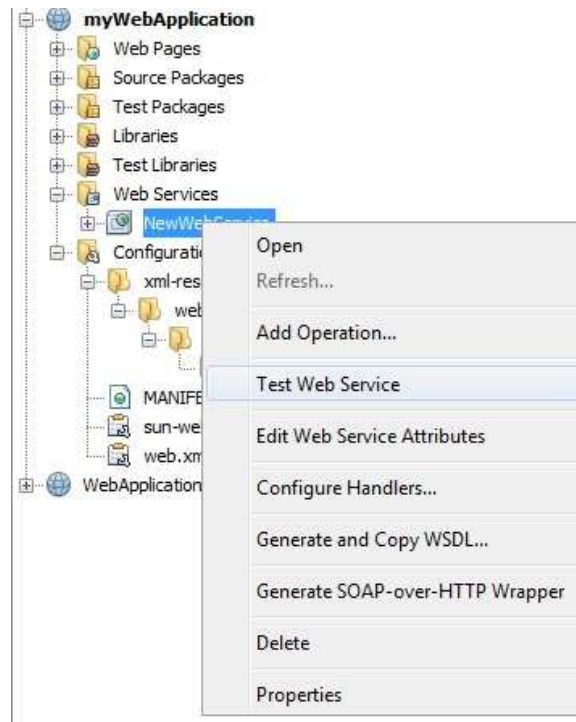




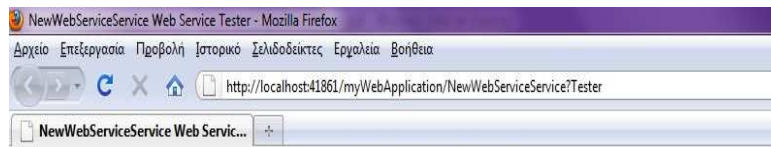
Προσδιορίζουμε το όνομα του πακέτου του project στο οποίο θα περιέχεται ο client:



Τέλος, κάνουμε δοκιμή εκτέλεσης του web service, μια λειτουργία που μας παρέχει το NetBeans:







## NewWebServiceService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

---

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

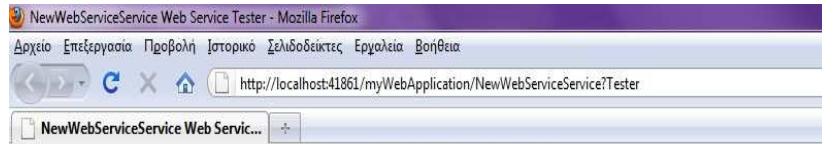
### Methods :

```
public abstract java.lang.String newpackage.NewWebService.sayHello(java.lang.String)
```

sayHello (  )

---

Ολοκληρώθηκε



## NewWebServiceService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

### Methods :

```
public abstract java.lang.String newpackage.NewWebService.sayHello(java.lang.String)
```

sayHello (  )

Ολοκληρώθηκε

Method invocation trace

### sayHello Method invocation

---

#### Method parameter(s)

Type	Value
java.lang.String	Freddy

---

#### Method returned

java.lang.String : "Freddy"

---

#### SOAP Request

---

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:sayHello xmlns:ns2="http://newPackage/">
      <yourName>Freddy</yourName>
    </ns2:sayHello>
  </S:Body>
</S:Envelope>
```

---

#### SOAP Response

---

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:sayHelloResponse xmlns:ns2="http://newPackage/">
      <return>Freddy</return>
    </ns2:sayHelloResponse>
  </S:Body>
</S:Envelope>
```

---

Ολοκληρώθηκε

## ΕΠΕΞΗΓΗΣΗ ΟΡΩΝ

### A

**Apache SOAP** – Μία υλοποίηση του πρωτοκόλλου SOAP σε πλατφόρμα ανοικτού κώδικα JAVA.

**API ( Application Program Interface )** – Μια διεπαφή εφαρμογής που παρέχεται στους προγραμματιστές ώστε να αλληλεπιδρούν με ένα δίκτυο ή λειτουργικό σύστημα.

### B

**B2B ( Business to Business )** – Αρχιτεκτονική που επιτρέπει την ανταλλαγή δεδομένων και υπηρεσιών μεταξύ επιχειρήσεων.

**Binding** – Μια λειτουργία των web services η οποία περιλαμβάνει μια διεπαφή, μια έγκυρη κωδικοποίηση και ένα πρωτόκολλο για την ομαλή μεταφορά μηνυμάτων.

**binding Template** – Ένα UDDI XML στοιχείο, που περιέχει πληροφορίες για το πώς και πού μπορεί να εντοπιστεί μια υπηρεσία διαδικτύου.

**BTP ( Business Transaction Protocol )** – Πρωτόκολλο που χρησιμοποιείται από την αρχιτεκτονική B2B για την ανταλλαγή μηνυμάτων XML στα πλαίσια ενός συνόλου συναλλαγών μεταξύ επιχειρήσεων διαμέσου του διαδικτύου.

**businessEntity** – Ένα UDDI XML στοιχείο, που περιέχει πληροφορίες μιας καταχωρημένης στο διαδίκτυο επιχείρησης, πληροφορίες όπως, το όνομά της, πειγραφή της, διεύθυνση και στοιχεία επικοινωνίας.

**businessService** – Ένα UDDI XML στοιχείο, που περιέχει πληροφορίες μιας καταχωρημένης υπηρεσίας διαδικτύου ή μιας ομάδας συσχετιζόμενων υπηρεσιών. Οι πληροφορίες αυτές περιλαμβάνουν, το όνομα, την περιγραφή και μια προαιρετική λίστα στοιχείων *binding Template*

### C

**CGI ( Common Gate Interface )** – Ορίζεται έτσι ο ενδιάμεσος μεταξύ ενός web server και ενός client που αναλαμβάνει την λήψη και επεξεργασία των δεδομένων που ανταλλάσσονται μεταξύ τους.

**Client** – Ένας ρόλος που υπάρχει σε κάθε σύστημα client / server και συνιστά το λογισμικό που αιτείται υπηρεσιών ή πληροφοριών από έναν εξυπηρετητή ( server ).

**COM ( Component Object Model )** – Μια αρχιτεκτονική Λογισμικού σχεδιασμένη από την Microsoft για την ανάπτυξη εφαρμογών που επεξεργάζονται δυαδικά δεδομένα λογισμικού.

**CORBA ( Common Object Request Broker Architecture )** – Μια διεπαφή λογισμικού που αναπτύχθηκε από την OMG ( Object Management Group ) και επιτρέπει την επικοινωνία αντικειμένων στο διαδίκτυο, όντας ανεξάρτητη από την γλώσσα προγραμματισμού και το λειτουργικό τους σύστημα.

**DCOM ( Distributed Component Object Model )** – Μια αρχιτεκτονική που αποτελεί επέκταση της αρχιτεκτονικής COM, η οποία αναπτύχθηκε επίσης, από την Microsoft, ώστε να καταστήσει δυνατή την επικοινωνία μεταξύ αντικειμένων τα οποία είναι τοποθετημένα σε διαφορετικά δίκτυα ή στο Internet.

**DOM ( Document Object Model )** – Μια γενική διεπαφή που επιτρέπει σε εξωτερικά

προγράμματα να επεξεργάζονται τα περιεχόμενα, την δομή και την μορφή εγγράφων.

**DTD ( Document Type Definition )** – Έγγραφο που ορίζει το είδος της κωδικοποίησης των περιεχομένων στις σημάνσεις σε ένα XML ή HTML έγγραφο και τον τρόπο χειρισμού τους από την εφαρμογή που διαβάζει το XML ή HTML έγγραφο.

## E

**ebXML ( Electronic Business using eXtensible Markup Language )** – Πρότυπο ανταλλαγής επιχειρησιακών εγγράφων που ανήκει στην ομάδα προτύπων XML.

## G

**green pages** – Μια ομάδα δεδομένων που χρησιμοποιείται σε ένα UDDI μοντέλο για να προσδιορίσει τεχνικές πληροφορίες για μια υπηρεσία διαδικτύου. Γενικότερα, περιλαμβάνει έναν δείκτη προς ένα εξωτερικό χαρακτηριστικό και μια διεύθυνση εντοπισμού της υπηρεσίας.

## H

**HTTP ( Hyper Text Transfer Protocol )** – Πρωτόκολλο που ορίζει την μορφή των μηνυμάτων που ανταλλάσσονται στο διαδίκτυο και το σύνολο των ενεργειών που πραγματοποιούν οι εξυπηρετητές ( servers ) πάνω σε αυτά.

## I

**IP** – ( Internet Protocol ) Είναι το βασικό πρωτόκολλο που χρησιμοποιείται για την δομολόγηση πακέτων δεδομένων στο διαδίκτυο.

**IP Address** – Μια 32 – μπιτη διεύθυνση που προσδιορίζει μονοσήμαντα ένα τερματικό στο διαδίκτυο.

**ISO 3166** – Πρότυπο ταξινόμησης κωδικών χωρών σχεδιασμένο από τον Διεθνή Οργανισμό Προτυποποίησης ( International Organization for Standardization ). Για παράδειγμα, οι ΗΠΑ έχουν τον κωδικό US. Χρησιμοποιώντας το πρότυπο ISO 3166 οι επιχειρήσεις οι οποίες είναι καταχωρημένες σε UDDI μπορούν να προσδιορίσουν την γεωγραφική τους τοποθεσία.

## J

**JAXP ( Java API for XML Processing )** – Μια πλατφόρμα που μετατρέπει τα μηνύματα XML σε μορφή συμβατή με οποιαδήποτε υλοποίηση μιας επιχείρησης.

## M

**MathML** – Γλώσσα που υποστηρίζεται από την XML και χρησιμοποιείται για την δημιουργία μαθηματικών τύπων.

**Message** – Η βασική μονάδα επικοινωνίας που στηρίζει την μεταφορά δεδομένων μεταξύ client και server.

## P

**P2P ( Peer to Peer )** – Αρχιτεκτονική επιχειρησιακών υπηρεσιών Ιστού που επιτρέπει σε μια ομάδα εφαρμογών να έχουν κοινά χαρακτηριστικά ώστε να είναι δυνατή η επικοινωνία μεταξύ τους.

## R

**Registry** – Ένα κεντρικό σημείο αναφοράς μέσα σε μία αρχιτεκτονική προσανατολισμένη στις υπηρεσίες, το οποίο τοποθετεί και διαχειρίζεται πληροφορίες για μια υπηρεσία. Οι πληροφορίες αυτές περιγράφουν το είδος των υπηρεσιών, πώς αυτές χρησιμοποιούνται και πώς μπορούν να επικοινωνήσουν με άλλα συστατικά μέσα στο διαδίκτυο.

**rpcrouter** – Ένα Apache SOAP servlet το οποίο λαμβάνει αιτήσεις SOAP και τις μεταφέρει στις κατάλληλες κλάσεις υπηρεσιών για επεξεργασία.

## S

**SAML ( Security Assertion Markup Language )** – Αναπτύχθηκε από την OASIS ( Organization for the Advancement of Structured Information Standards ) και είναι υπεύθυνη για τον χειρισμό των πληροφοριών πιστοποίησης και αυθεντικοποίησης που ανταλλάσσονται μεταξύ των επιχειρησιακών εφαρμογών.

**Search Engine** – Δηλαδή μηχανή αναζήτησης. Διατηρεί ευρετήρια με web sites με την βοήθεια των οποίων εντοπίζει και κατηγοριοποιεί το κάθε web site.

**Service** – Ένα Μοντέλο επαναχρησιμοποιήσιμου λογισμικού το οποίο είναι ανεξάρτητο από εφαρμογές και πλατφόρμες μέσα από τις οποίες τρέχει. Έχουν ένα καλώς ορισμένο γραφικό περιβάλλον και επιτρέπουν την ένα προς ένα χαρτογράφηση μεταξύ των επιχειρησιακών εργασιών και των IT συστατικών που χρειάζονται για την εκτέλεση των εργασιών.

**Service description** – Ένα επίπεδο του πρωτοκόλλου των υπηρεσιών διαδικτύου το οποίο είναι υπεύθυνο για την περιγραφή της δημόσιας διασύνδεσης μιας συγκεκριμένης υπηρεσίας διαδικτύου.

**Service provider** – Στοιχείο της αρχιτεκτονικής των υπηρεσιών διαδικτύου που αναλαμβάνει τον ρόλο του host που υλοποιεί μια υπηρεσία και την καθιστά διαθέσιμη στο διαδίκτυο. Είναι ανάλογο του server στην αρχιτεκτονική client / server.

**Service registry** – Στοιχείο της αρχιτεκτονικής των υπηρεσιών διαδικτύου, που αποτελεί έναν κατάλογο από υπηρεσίες. Συγκεκριμένα, οι προγραμματιστές μπορούν να συνδεθούν σε έναν κατάλογο, να δημοσιεύσουν νέες υπηρεσίες, ή να βρουν ήδη υπάρχουσες υπηρεσίες.

**Service requestor** – Στοιχείο της αρχιτεκτονικής των υπηρεσιών διαδικτύου, που αποτελεί τον χρήστη μιας υπηρεσίας διαδικτύου, ο οποίος χρησιμοποιεί μια υπηρεσία συνδεδεμένος στο διαδίκτυο και στένοντας αιτήματα XML. Είναι ανάλογο του client στην αρχιτεκτονική client / server.

**Service Orientation** – Μια λογική που εφαρμόζεται στις επιχειρησιακές διεργασίες κατά την οποία οι διεργασίες αυτές αποτελούν συνδεδεμένες, χαμηλής σύζευξης διεργασίες που υποστηρίζονται από υπηρεσίες.

**SMIL** – Γλώσσα που υποστηρίζεται από την XML και χρησιμοποιείται για την παρουσίαση δεδομένων υπό μορφή πολυμέσων.

**SOAP ( Simple Object Access Protocol )** – Πρωτόκολλο ανταλλαγής XML μηνυμάτων το οποίο περιλαμβάνει τρεις κύρια συστατικά: envelope, header, body και είναι σχεδιασμένο για την ανταλλαγή πληροφοριών μέσα σε ένα διαμοιραζόμενο περιβάλλον.

**SOAP Body** – Το στοιχείο SOAP Body αποτελεί το “σώμα” του SOAP μηνύματος. Περιέχει λεπτομερείς πληροφορίες που είναι απαραίτητες για την απομακρυσμένη επίκληση διαδικασιών, συμπεριλαμβανομένων, του ονόματος της μεθόδου, των παραμέτρων ή επιστρεφόμενων τιμών. Το στοιχείο Body, προαιρετικά, μπορεί να περιέχει ένα στοιχείο Fault για τον χειρισμό σφαλμάτων.

**SOAP Envelope** – Το στοιχείο SOAP Envelope αποτελεί ενθυλάκωση ενός XML μηνύματος που αποστέλλεται μέσω SOAP. Προσδιορίζει την έκδοση του SOAP πρωτοκόλλου και αποτελείται, προαιρετικά από ένα στοιχείο SOAP Header και απαραίτητα, από ένα στοιχείο SOAP Body.

**SOAP Header** – Το στοιχείο SOAP Header αποτελεί ένα προαιρετικό στοιχείο ενός XML εγγράφου και ενθυλακώνει επίπεδα ιδιοτήτων που προσδιορίζουν το μήνυμα SOAP.

## T

**tModel** – Τεχνικό Μοντέλο. Αποτελεί ένα στοιχείο UDDI XML που χρησιμοποιείται για να ορίσει δείκτες προς εξωτερικά τεχνικά χαρακτηριστικά.

## U

**UDDI ( Universal Description Discovery and Integration )** – Λογική καταχώρησης και περιγραφής των Web Services στο διαδίκτυο, η οποία βασίζεται στην XML.

## W

**Web Service** – Ένα σύνολο εφαρμογών που τοποθετείται στο Internet και στηρίζεται σε πρωτόκολλα Internet και επικοινωνίας επιτρέποντας την επικοινωνία και ανταλλαγή μηνυμάτων μεταξύ διαφορετικών εφαρμογών.

**white pages** – Μία ομάδα δεδομένων που χρησιμοποιείται σε μία καταχώριση UDDI για να προσδιορίσει πληροφορίες μίας επιχείρησης, όπως το όνομα, η περιγραφή και η διεύθυνση.

**WSDL ( Web Services Definition Language )** – Γλώσσα περιγραφής των Web Services στηριζόμενη σε XML.

**WSFL** – Αρχιτεκτονική επιχειρησιακών υπηρεσιών Ιστού που επιτρέπει σε μια ομάδα εφαρμογών να έχουν κοινά χαρακτηριστικά ώστε να είναι δυνατή η επικοινωνία μεταξύ τους.

## X

**Xlink ( XML Linking Language )** – Παρέχει τον σκελετό για την διασύνδεση όλων των XML πόρων μαζί. Χρησιμοποιείται για την δημιουργία υπερσυνδέσμων.

**XML ( eXtensible Markup Language )** – Γλώσσα περιγραφής των Web εγγράφων, η οποία επιτρέπει την δημιουργία σημάτων προσαρμοσμένες στις ανάγκες και τις απαιτήσεις μιας επιχείρησης. Εκτός από δεδομένα περιλαμβάνει και μεταδεδομένα.

**XML Namespaces** – Οι χώροι ονομάτων χρησιμοποιούνται για την διαφοροποίηση των ονομάτων των στοιχείων στα XML έγγραφα, ώστε να αποφεύγονται οι συγκρούσεις ονομάτων.

**Xpointer ( XML Pointer Language )** – Παρέχει έναν σκελετό για την διευθυνσιοδότηση εσωτερικών δομών μέσα σε ένα XML έγγραφο, όπως για παράδειγμα, των στοιχείων, των χαρακτηριστικών και του περιεχομένου.

**XSL ( eXtensible Stylesheet Language )** - Γλώσσα η οποία δημιουργήθηκε για την περιγραφή των λογιστικών φύλλων για XML έγγραφα.

## Υ

**yellow pages** – Μία ομάδα δεδομένων που χρησιμοποιείται σε καταχωρήσεις UDDI για την κατηγοριοποίηση των επιχειρήσεων και των υπηρεσιών που προσφέρονται μέσω του web.



## ΕΠΙΛΟΓΟΣ

Στην Πτυχιακή παρουσιάστηκε η τεχνολογία των Web Services, εξετάστηκε η αρχιτεκτονική τους και τα δομικά στοιχεία που αποτελούνται, καθώς και οι τεχνολογίες που χρησιμοποιούνται, XML, UDDI, WSDL και εστιάσαμε περισσότερο στα Web Services που υλοποιούνται με SOAP.

Μιλήσαμε για τον τρόπο που γίνεται η κλήση των λειτουργιών των Web Services, τον τρόπο επικοινωνίας πελατών και παρόχων Web Services και τον τρόπο ανταλλαγής μηνυμάτων. Αναλύθηκε, το πώς καταχωρείται μία επιχείρηση ή ένα Web Service στο διαδίκτυο, δηλαδή το κομμάτι που αναφέρεται στην βάση δεδομένων που υπάρχει πίσω από την τεχνολογία των Web Services. Παρουσιάστηκε η τεχνολογία των Java Servlets, ως αντιπαράθεση της τεχνολογίας των Web Services, καθώς και το ποιά κομμάτια της πρώτης τεχνολογίας μπορούν να χρησιμοποιηθούν στην τεχνολογία των Web Services. Μιλήσαμε για τα θέματα ασφάλειας των Web Services, ένας τομέας που εξελίσσεται δυναμικά, όσο δυναμικά εξελίσσεται η τεχνολογία των Web Services.

Τέλος, παρουσιάστηκε η εφαρμογή που αναπτύχθηκε, ώστε παρουσιαστεί πλήρως και με παραδείγματα η τεχνολογία των Web Services και να γίνει επίδειξη της λειτουργίας τους στην πράξη. Επίσης, με αφορμή την χρήση του NetBeans ως περιβάλλον ανάπτυξης της εφαρμογής, κρίθηκε απαραίτητο να παρουσιαστεί με ένα παράδειγμα, ο τρόπος που μπορούμε να δημιουργήσουμε ένα Web Service με αυτό.