

Ενδιάμεσο Λογισμικό Διαλειτουργικότητας
(Interoperability Middleware)

Τεκμηρίωση

(Για την έκδοση 1.0)

Ιανουάριος 2004

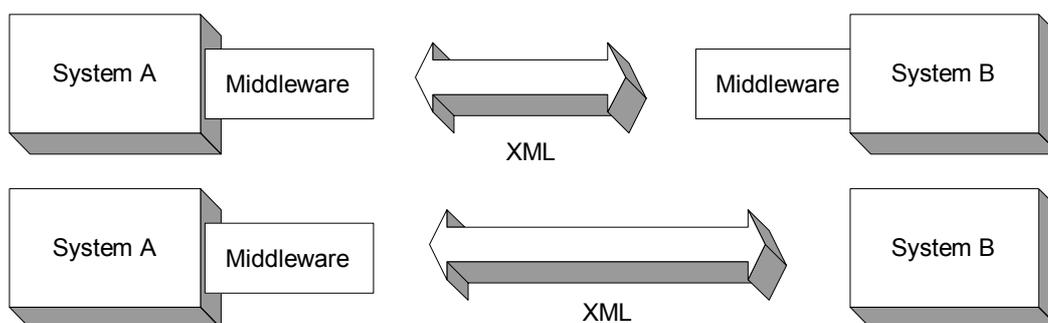
Το Ενδιάμεσο Λογισμικό Διαλειτουργικότητας αναπτύχθηκε από την εταιρία EBS (<http://www.ebs.gr>)

Περιεχόμενα

1	Εισαγωγή	4
2	Οδηγίες εγκατάστασης	6
2.1	Απαιτήσεις – Περιορισμοί	6
2.2	Εγκατάσταση του ΕΛΔ	7
2.3	Εγκατάσταση JDBC driver.....	7
2.4	Ρύθμιση Firewall	8
3	Οδηγίες χρήσης.....	9
3.1	Διαχείριση του ΕΛΔ	9
3.1.1	Πρόσβαση στο ΕΛΔ	9
3.1.2	Διαχείριση των Συνδέσεων σε ΒΔ (Datasources).....	11
3.1.3	Ανάπτυξη Επεκτάσεων	15
3.1.4	Διαχείριση των Επεκτάσεων (Extensions).....	17
3.1.5	Διαχείριση Σχημάτων XSD (Schemas).....	20
3.1.6	Διαχείριση των Υπηρεσιών (Services)	23
3.2	Χρήση των υπηρεσιών του ΕΛΔ	30
3.2.1	Γενική περιγραφή χρήσης	30
3.2.2	Παράδειγμα εφαρμογής πελάτη σε Java	31
	Παράρτημα Α: Δεσμευμένες λέξεις στη Java	33

1 Εισαγωγή

Το Ενδιάμεσο Λογισμικό Διαλειτουργικότητας (ΕΛΔ) σχεδιάστηκε και αναπτύχθηκε με στόχο τη διευκόλυνση ανάπτυξης εφαρμογών διαλειτουργικότητας. Ο ρόλος του φαίνεται στο Σχήμα 1: ουσιαστικά το ΕΛΔ αναλαμβάνει να παρέχει πρόσβαση σε υπηρεσίες και δεδομένα ενός πληροφοριακού συστήματος μέσω web services (XML over HTTP).



Σχήμα 1: Ρόλος του Ενδιάμεσου Λογισμικού Διαλειτουργικότητας (ΕΛΔ)

Ο σχεδιασμός και η υλοποίηση του ΕΛΔ έχει τις εξής βασικές κατευθύνσεις:

- Ανεξαρτησία από την πλατφόρμα υλικού και λογισμικού των πληροφοριακών συστημάτων που υποστηρίζει.

Το ΕΛΔ αναπτύχθηκε σε Java ώστε να μπορεί να λειτουργήσει σε όλες τις διαδεδομένες πλατφόρμες υλικού / λογισμικού συστήματος (MS Windows, Linux, Mac OS, διάφορα Unix, κλπ).

- Ευκολία και απλότητα στη χρήση ώστε να δημιουργούνται web services χωρίς να απαιτείται ή συγγραφή κώδικα, αλλά και δυνατότητα δημιουργίας πολύπλοκων custom connectors μέσω κώδικα.

Το ΕΛΔ έχει υποστηρίξει δύο τρόπους ορισμού υπηρεσιών:

(α) υλοποίηση μέσω απευθείας πρόσβαση στη βάση δεδομένων του πληροφοριακού συστήματος που υποστηρίζει

(β) υλοποίηση μέσω επεκτάσεων που υλοποιούν τη διασύνδεση με το πληροφοριακό σύστημα (custom connectors)

Στην περίπτωση (α) ο χρήστης δηλώνει μέσω του user interface του ΕΛΔ τις λεπτομέρειες του web service και δεν ασχολείται καθόλου με συγγραφή κώδικα λογισμικού για την διασύνδεση με το πληροφοριακό σύστημα. Ο χρήστης απλά δηλώνει τα στοιχεία της βάσης δεδομένων και ένα σύνολο από SQL queries και το ΕΛΔ αναλαμβάνει να δημιουργήσει και να δημοσιοποιήσει τις κατάλληλες υπηρεσίες που προσφέρονται στα απομακρυσμένα πληροφοριακά συστήματα. Στην περίπτωση (β) ο χρήστης χρησιμοποιεί το API¹ του ΕΛΔ (MBeans API) και δημιουργεί την κατάλληλη επέκταση του ΕΛΔ που αναλαμβάνει τη διασύνδεση με το πληροφοριακό σύστημα. Μέσω του user interface του ΕΛΔ, η επέκταση αυτή εγκαθίσταται και δημιουργούνται αυτόματα τα κατάλληλα web

¹ Application Programming Interface

services που παρέχουν πρόσβαση στα απομακρυσμένα πληροφοριακά συστήματα. Αναλυτική περιγραφή και για τις δύο διαδικασίες δίνεται παρακάτω στην παράγραφο 3.1.

Η περίπτωση (α) είναι ιδιαίτερα απλή στη χρήση της και δεν απαιτεί γνώσεις XML και web services, περιορίζεται όμως σε πρόσβαση των δεδομένων του πληροφοριακού συστήματος που υποστηρίζει το ΕΛΔ μέσω SQL queries απευθείας στη βάση δεδομένων του πληροφοριακού συστήματος. Η περίπτωση (β), απαιτεί γνώσεις προγραμματισμού αλλά επιτρέπει την ανάπτυξη επεκτάσεων (custom connectors) που μπορεί εκμεταλλεύεται το συγκεκριμένο API και τις δυνατότητες του εκάστοτε συστήματος – και σε αυτή την περίπτωση όμως δεν απαιτούνται γνώσεις XML/web services καθώς το ΕΛΔ αναλαμβάνει να δημιουργήσει την υπηρεσία αυτόματα όταν δοθεί και δηλωθεί η επέκταση μέσω του user interface.

- Δυνατότητα χρήσης έτοιμων σχημάτων XML (XSD) για τα δεδομένα που ανταλλάσσονται.

Το ΕΛΔ υποστηρίζει τη δήλωση αρχείων XSD που περιγράφουν τους τύπους των δεδομένων που θα ανταλλάσσονται μέσω των web services. Ο χρήστης επιλέγει όταν δηλώνει μια υπηρεσία αν θα χρησιμοποιήσει κάποια αρχεία XSD ή αν θα αφήσει το ΕΛΔ να δημιουργήσει αυτόματα τους τύπους στο αρχείο WSDL που περιγράφει τη νέα υπηρεσία, βασισμένο στα queries που την υλοποιούν (βλέπε 3.1.5 και 3.1.6). Η δυνατότητα έχει εφαρμογή στη περίπτωση (α) καθώς όταν ο χρήστης υλοποιεί επέκταση (περίπτωση β) τα απαιτούμενα αρχεία XSD ενσωματώνονται κατά την υλοποίηση (βλέπε 3.1.3).

2 Οδηγίες εγκατάστασης

2.1 Απαιτήσεις – Περιορισμοί

Ως λογισμικό γραμμένο σε Java το ΕΛΔ για να εκτελεστεί απαιτεί java runtime environment. Παρότι μπορεί να εκτελεστεί με όλα τα Sun-complaint JVM, έχει δοκιμαστεί με Sun JRE 1.4.2. Έτσι, προϋπόθεση να λειτουργήσει το ΕΛΔ είναι να υπάρχει εγκατεστημένο το Sun JRE 1.4.2 (ή το Sun J2SDK 1.4.2) το οποίο δίνεται ελεύθερα για download στο WWW:

- <http://www.java.com/en/index.jsp> (JRE)
- <http://java.sun.com/j2se/1.4.2/download.html> (JDK)

Πριν την εκτέλεση του ΕΛΔ θα πρέπει να οριστεί το environment variable JAVA_HOME με το directory path όπου έχει εγκατασταθεί το JRE ή το JDK.

Το ΕΛΔ είναι μια 3-tier εφαρμογή J2EE που βασίζεται στον open-source J2EE Application Server JBoss (version 3.2.2 για το ΕΛΔ 1.0). Το σύστημα βασίζεται στις υπηρεσίες του JBoss και δεν απαιτεί άλλο λογισμικό, εκτός από την ύπαρξη JVM. Ένα σημείο που θέλει προσοχή είναι η διαθεσιμότητα ορισμένων δικτυακών θυρών (network ports) που απαιτούνται για την λειτουργία του JBoss Application Server – έτσι, στον εξυπηρετητή που θα εγκατασταθεί το ΕΛΔ *δεν πρέπει να δεσμεύονται από άλλο λογισμικό οι εξής θύρες:*

- 8080, 8083, 8090-8093 TCP
- 1099, 1100 TCP
- 4444, 4445 TCP
- 1701 TCP
- 19001 TCP

Προφανώς για να λειτουργήσει σωστά το ΕΛΔ πρέπει να εγκατασταθεί σε υπολογιστή με δικτυακή σύνδεση (ώστε να μπορεί να δεχθεί κλήσεις προς τις web services που προσφέρει και να επικοινωνεί με το πληροφοριακό σύστημα που εξυπηρετεί).

Η επικοινωνία του χρήστη-διαχειριστή με το ΕΛΔ γίνεται μέσω ενός απλού web browser (π.χ. MS IE ver. 5.5+, Mozilla 1.1+, Netscape 6.0+).

Τα web services που δημιουργεί και προσφέρει το ΕΛΔ βασίζονται σε WSDL 1.0 και SOAP 1.2.

Ο χρήστης-διαχειριστής του ΕΛΔ, πρέπει να έχει τις παρακάτω γνώσεις για να μπορέσει να το χρησιμοποιήσει:

- αν σκοπεύει να ορίσει web services που βασίζονται σε απευθείας πρόσβαση στη βάση δεδομένων του πληροφοριακού συστήματος, απαιτείται γνώση SQL, της συγκεκριμένης βάσης δεδομένων (και φυσικά του πληροφοριακού συστήματος) καθώς και τη δυνατότητα να εντοπίσει τον κατάλληλο JDBC driver και τις παραμέτρους του.
- για τον ορισμό web services που βασίζονται σε επεκτάσεις, απαιτείται η εμπλοκή developer που γνωρίζει Java και το API του πληροφοριακού

συστήματος ώστε να γράψει τον κατάλληλο κώδικα σύμφωνα με όσα περιγράφονται στην παράγραφο 3.1.3.

2.2 Εγκατάσταση του ΕΛΔ

Η εγκατάσταση πραγματοποιείται σε δύο απλά βήματα:

1. Αποσυμπίεση του zip archive “InteropMiddleware_ver_M.m.zip” (όπου M.m η έκδοση του ΕΛΔ) σε κάποιο directory του εξυπηρετητή
2. Ορισμός των παρακάτω παραμέτρων λειτουργίας στο αρχείο [ROOT]/server/middleware/conf/general.properties (όπου [ROOT] το path όπου εγκαταστάθηκε το ΕΛΔ):
 - services.temp.dir= full path για αποθήκευση προσωρινών αρχείων (π.χ. /tmp/middleware ή c:/temp/middleware ή c:\\temp\\middleware)
 - services.datasources.package=package name για τις κλάσεις που παράγονται αυτόματα (π.χ. gr.gif.middleware.datasource) – η παράμετρος αυτή δεν υπάρχει λόγος να αλλαχθεί καθώς επηρεάζει κλάσεις που παράγονται εσωτερικά.
 - services.namespace=XML namespace για τα web services που παράγονται αυτόματα από το ΕΛΔ (π.χ. middleware.gif.gr)

Η εκτέλεση του ΕΛΔ γίνεται με την εντολή [ROOT]/bin/run

2.3 Εγκατάσταση JDBC driver

Για να μπορεί το ΕΛΔ να συνδεθεί με τη βάση δεδομένων του εκάστοτε πληροφοριακού συστήματος (για να χρησιμοποιηθεί η περίπτωση λειτουργίας (α)) πρέπει να εγκατασταθεί ο κατάλληλος JDBC driver στο ΕΛΔ. Η εγκατάσταση αυτή γίνεται ως εξής:

- Αντιγραφή του JDBC driver (π.χ. αρχείο classes12.zip για Oracle DBMS, ή αρχείο pg73jdbc3.jar για Postgres 7.3) το στο [ROOT]/lib, όπου [ROOT] το path όπου εγκαταστάθηκε το ΕΛΔ)
- Επανεκκίνηση του ΕΛΔ:
 - Αν το ΕΛΔ εκτελείται, ctrl-C στο παράθυρο εκτέλεσης ή διακοπή του service ή “kill” του unix process, κλπ.
 - Εκτέλεση του [ROOT]/bin/run

Εκτός από την εγκατάσταση του αρχείου του JDBC driver, ο χρήστης θα πρέπει να γνωρίζει και τα υπόλοιπα στοιχεία του (jdbc url) για να τα συμπληρώσει, όπως περιγράφεται παρακάτω, μέσω του user interface όταν ορίζονται τα αντίστοιχα data sources στο ΕΛΔ.

2.4 Ρύθμιση Firewall

Για λόγους απόδοσης και ασφαλείας προτείνεται η εγκατάσταση του ΕΛΔ να γίνεται στο ίδιο δίκτυο με το πληροφοριακό σύστημα που υποστηρίζει, ενώ η πρόσβαση των απομακρυσμένων πληροφοριακών συστημάτων στο ΕΛΔ να γίνεται μέσω firewall που επιτρέπει πρόσβαση στο ΕΛΔ μόνο μέσω της θύρας 8080. Προτείνεται η ρύθμιση του firewall ώστε μόνο τα συγκεκριμένα IPs των απομακρυσμένων πληροφοριακών συστημάτων που προβλέπεται να χρησιμοποιούν τις υπηρεσίες του ΕΛΔ να επιτρέπεται να έχουν πρόσβαση στην θύρα 8080 TCP.

Γενικά, αν κανείς θέλει να ασφαλίσει σε επίπεδο δικτυακής θύρας το ΕΛΔ, πρέπει να γνωρίζει τα εξής:

- Η επικοινωνία του ΕΛΔ με το πληροφοριακό σύστημα που υποστηρίζει γίνεται μέσω των θυρών που υποστηρίζει ο εκάστοτε JDBC driver (π.χ. για Oracle thin JDBC driver η θύρα 1521 TCP), ή των θυρών που απαιτεί η συγκεκριμένη επέκταση του ΕΛΔ.
- Η επικοινωνία με τον χρήστη-διαχειριστή (user interface) γίνεται μέσω απλού web browser στην θύρα 8080 TCP.
- Η επικοινωνία των απομακρυσμένων πληροφοριακών συστημάτων μέσω web services γίνεται μέσω της θύρας 8080 TCP.

Τέλος, ως βασικό μέτρο ασφάλειας, ειδικά όταν τα δεδομένα που παρέχονται από τις υπηρεσίες είναι ευαίσθητα, προτείνεται η χρήση firewall στον εξυπηρετητή που φιλοξενεί το ΕΛΔ, ώστε να επιτρέπεται η εξωτερική πρόσβαση στη δικτυακή θύρα 8080 μόνο στα συγκεκριμένα IPs των εφαρμογών πελάτη που προβλέπεται να χρησιμοποιούν τις web services που παρέχει το ΕΛΔ. Με τον ίδιο τρόπο και η εσωτερική πρόσβαση στο user interface του ΕΛΔ είναι καλό να περιορίζεται στα IPs από όπου εργάζεται ο διαχειριστής του ΕΛΔ.

3 Οδηγίες χρήσης

Στη συνέχεια περιγράφεται η χρήση του ΕΛΔ από την πλευρά του:

- (α) χρήστη-διαχειριστή και του ΕΛΔ, δηλαδή του χρήστη που ορίζει τις υπηρεσίες που προσφέρονται από το ΕΛΔ (web services που έχουν πρόσβαση στο πληροφοριακό σύστημα που υποστηρίζεται),
- (β) του χρήστη των υπηρεσιών του ΕΛΔ, δηλαδή του developer που αναπτύσσει εφαρμογές που αντλούν δεδομένα από το πληροφοριακό σύστημα μέσω των web services του ΕΛΔ.

3.1 Διαχείριση του ΕΛΔ

Ο διαχειριστής του ΕΛΔ χρησιμοποιεί το user interface που περιγράφεται στη συνέχεια για να διαχειριστεί τις υπηρεσίες (web services) που παρέχει το ΕΛΔ σε απομακρυσμένα πληροφοριακά συστήματα και εφαρμογές ολοκλήρωσης (integrating applications). Η υλοποίηση των υπηρεσιών, η επικοινωνία με το πληροφοριακό σύστημα που εξυπηρετεί και όλες οι σχετικές λεπτομέρειες διαχειρίζονται μέσω αυτού του Web-based user interface του ΕΛΔ.

Για την επικοινωνία με το ΕΛΔ ο διαχειριστής χρειάζεται μόνο ένα σύγχρονο web browser (π.χ. MS IE ver. 5.5+, Mozilla 1.1+, Netscape 6.0+).

3.1.1 Πρόσβαση στο ΕΛΔ

Για να συνδεθεί με το ΕΛΔ ο διαχειριστής πρέπει να δώσει στο browser το εξής URL: **http://(server hostname):8080/mw/main**, όπου (server hostname) το όνομα του εξυπηρετητή όπου τρέχει το ΕΛΔ.

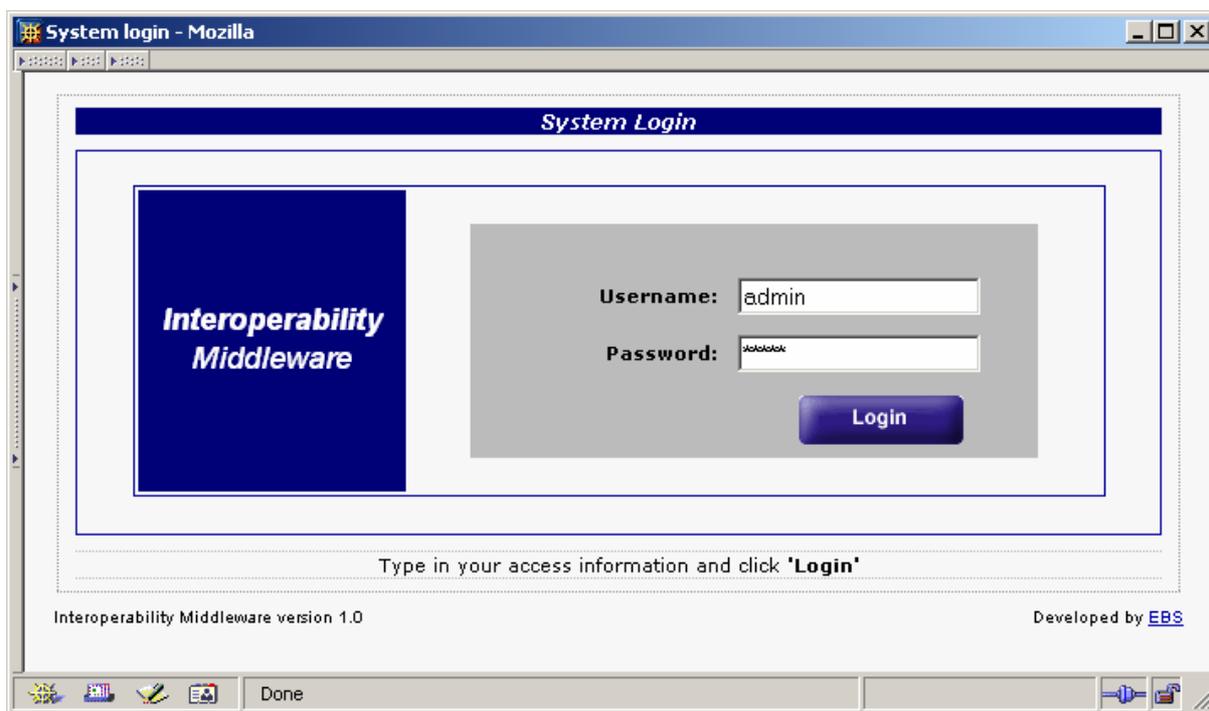
Η αρχική οθόνη παρουσιάζεται στο Σχήμα 2. Η οθόνη αυτή απαιτεί έλεγχο πρόσβασης με username και password, και εμφανίζεται κάθε φορά που ο χρήστης-διαχειριστής συνδέεται στο ΕΛΔ, ή αν ενώ είναι συνδεδεμένος παραμείνει ανενεργός και αρκετή ώρα, οπότε απαιτείται ξανά έλεγχος πρόσβασης.

Ο κωδικός πρόσβασης (username) είναι admin, ενώ στην αρχική του εγκατάσταση το ΕΛΔ (μετά την εγκατάσταση σύμφωνα με την παράγραφο 2.2) το σύνθημα (password) είναι επίσης admin. Για λόγους ασφαλείας, ο διαχειριστής πρέπει αμέσως μετά την εγκατάσταση να αλλάξει το σύνθημα, όπως αυτό περιγράφεται παρακάτω. Παρότι τα συνθήματα αποθηκεύονται κρυπτογραφημένα στην εσωτερική βάση δεδομένων του ΕΛΔ, ο διαχειριστής πρέπει να φροντίσει συνολικά για την ασφάλιση του εξυπηρετητή του ΕΛΔ (βλέπε παράγραφο 2.4).

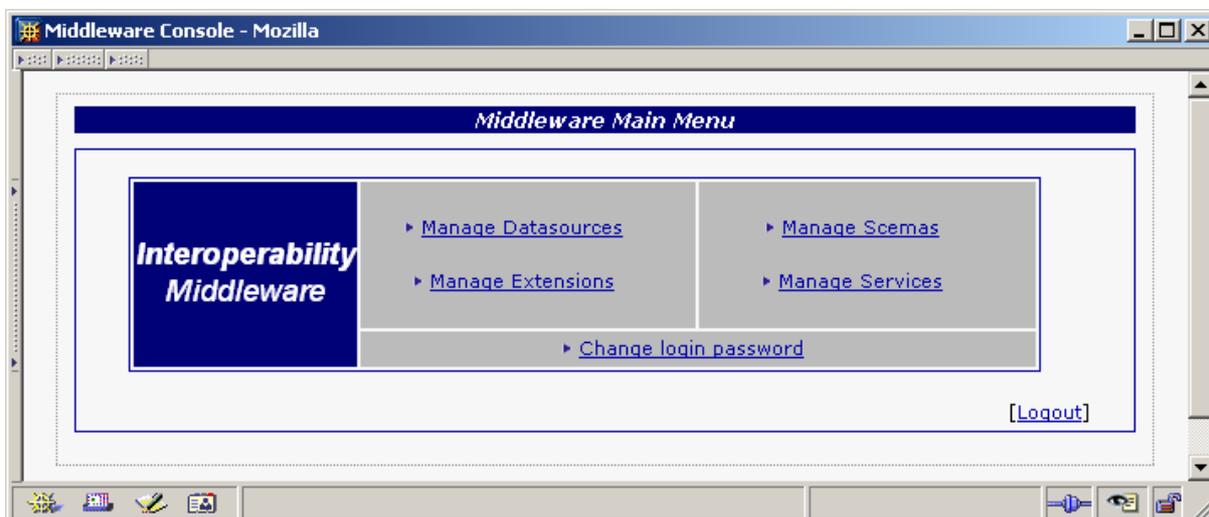
Η επιτυχής είσοδος του κωδικού πρόσβασης και του συνθήματος οδηγεί στην αρχική οθόνη επιλογών (βλέπε Σχήμα 3) και ο χρήστης-διαχειριστής μπορεί να χρησιμοποιήσει κανονικά το user interface του ΕΛΔ.

Γενικά όλες οι οθόνες του ΕΛΔ χρησιμοποιούν τη δομή αυτής της αρχικής οθόνης: ο τίτλος της λειτουργίας στο πάνω μέρος, οδηγίες και βοήθεια στο κάτω μέρος και οι επιλογές στο κέντρο της οθόνης.

Ως γενικός κανόνας: η πλοήγηση μέσα στην εφαρμογή κατά προτίμηση πρέπει να γίνεται μέσω των κατάλληλων υπερ-συνδέσμων (hyperlinks), π.χ. “Back”, “Cancel” και τα διάφορα buttons (π.χ. “< Return”, “Submit Data”).



Σχήμα 2: Αρχική οθόνη εισόδου στο ΕΛΔ (login)

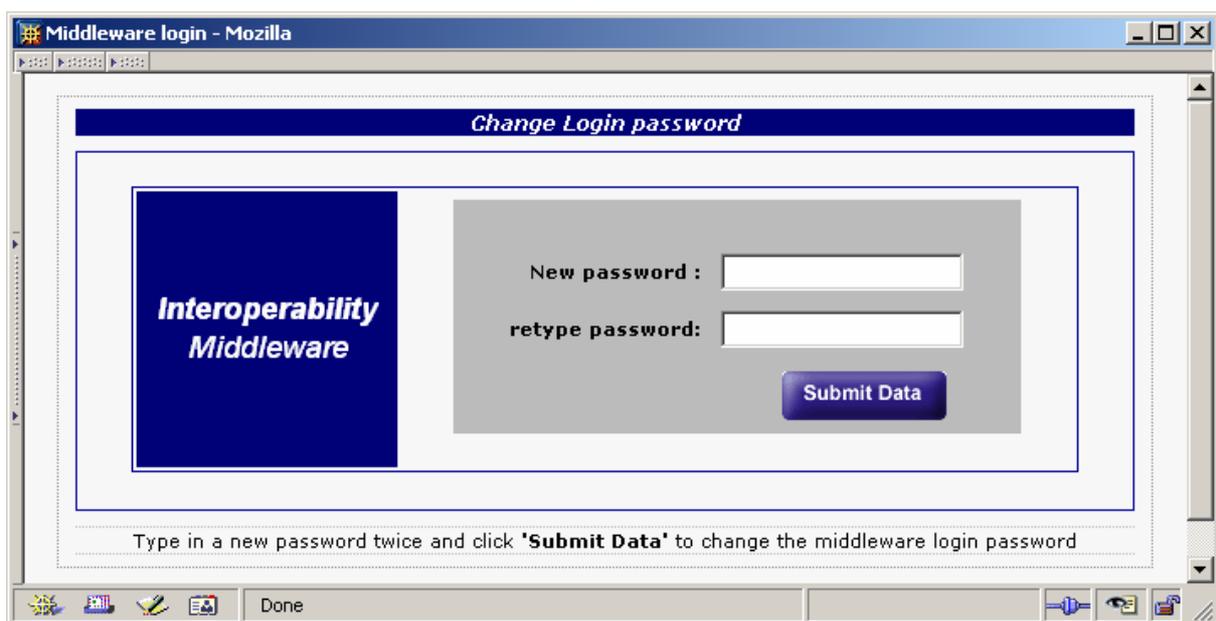


Σχήμα 3: Κεντρική οθόνη του ΕΛΔ

Οι επιλογές της κεντρικής οθόνης είναι οι εξής:

- **Manage Datasources:** Οδηγεί σε οθόνη με λειτουργίες για τη δήλωση απευθείας συνδέσεων του ΕΛΔ με το πληροφοριακό σύστημα (ή τα πληροφοριακά συστήματα) που υποστηρίζει, μέσω της βάσης δεδομένων του. Δηλώνονται τα στοιχεία σύνδεσης (jdbc driver parameters) και τα κατάλληλα SQL queries που θα δημοσιοποιηθούν ως web services του ΕΛΔ. Αναλυτική περιγραφή δίνεται στην παράγραφο 3.1.2.

- **Manage Extensions:** Οδηγεί σε οθόνη με λειτουργίες για την εγκατάσταση και διαχείριση των επεκτάσεων του ΕΛΔ που συνδέονται με το πληροφοριακό σύστημα που υποστηρίζεται και υλοποιούν web services που παρέχουν περισσότερο σύνθετες web services. Περιγραφή για τη δημιουργία των επεκτάσεων δίνεται στην παράγραφο 3.1.3 και για τις λειτουργίες της επιλογής “Manage Extensions” στην παράγραφο 3.1.4.
- **Manage Schemas:** Οδηγεί σε οθόνη με λειτουργίες που επιτρέπουν τη δήλωση και αποθήκευση στο ΕΛΔ αρχείων XSD. Τα αρχεία αυτά δηλώνονται ώστε να μπορούν να χρησιμοποιηθούν στον ορισμό υπηρεσιών από datasources με χρήση προκαθορισμένων τύπων δεδομένων. Αναλυτική περιγραφή δίνεται στην παράγραφο 3.1.5.
- **Manage Services:** Οδηγεί σε οθόνη με λειτουργίες διαχείρισης των υπηρεσιών (web services) που προσφέρει το ΕΛΔ. Από αυτές ορίζονται οι υπηρεσίες καθώς και το τρόπο υλοποίησής τους (datasource, extension), και άλλα στοιχεία τους. Αναλυτική περιγραφή δίνεται στην παράγραφο 3.1.6.
- **Logout:** Επιλογή που είναι διαθέσιμη σε όλες τις οθόνες και αποσυνδέει τον χρήστη (επιστρέφει στην οθόνη ελέγχου πρόσβασης).
- **Change login password:** Επιλογή που οδηγεί στην οθόνη που φαίνεται στο Σχήμα 4. Μέσω αυτής της οθόνης αλλάζει το σύνθημα πρόσβασης στον ΕΛΔ.

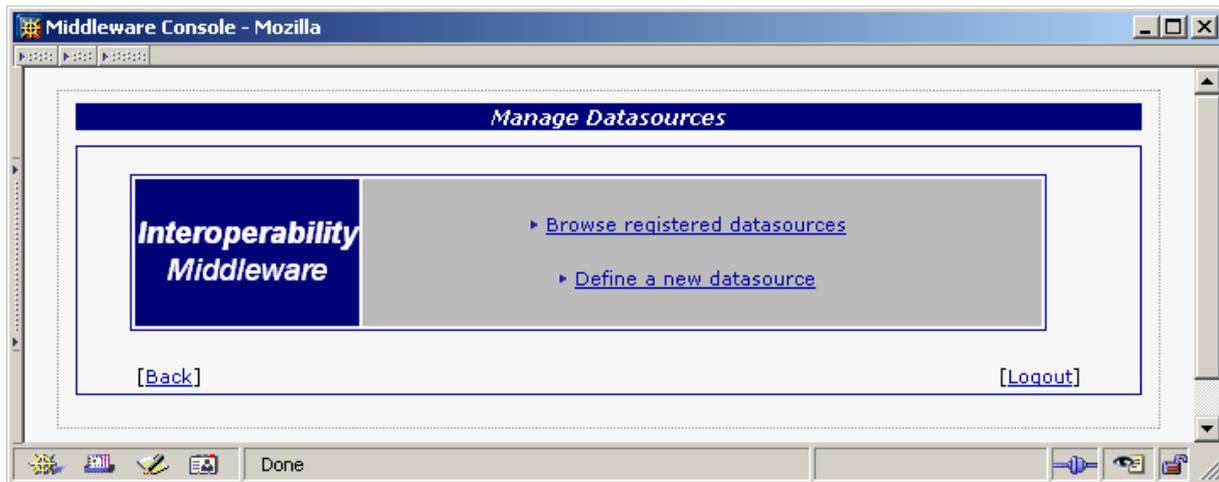


Σχήμα 4: Αλλαγή συνθήματος για την είσοδο στο ΕΛΔ

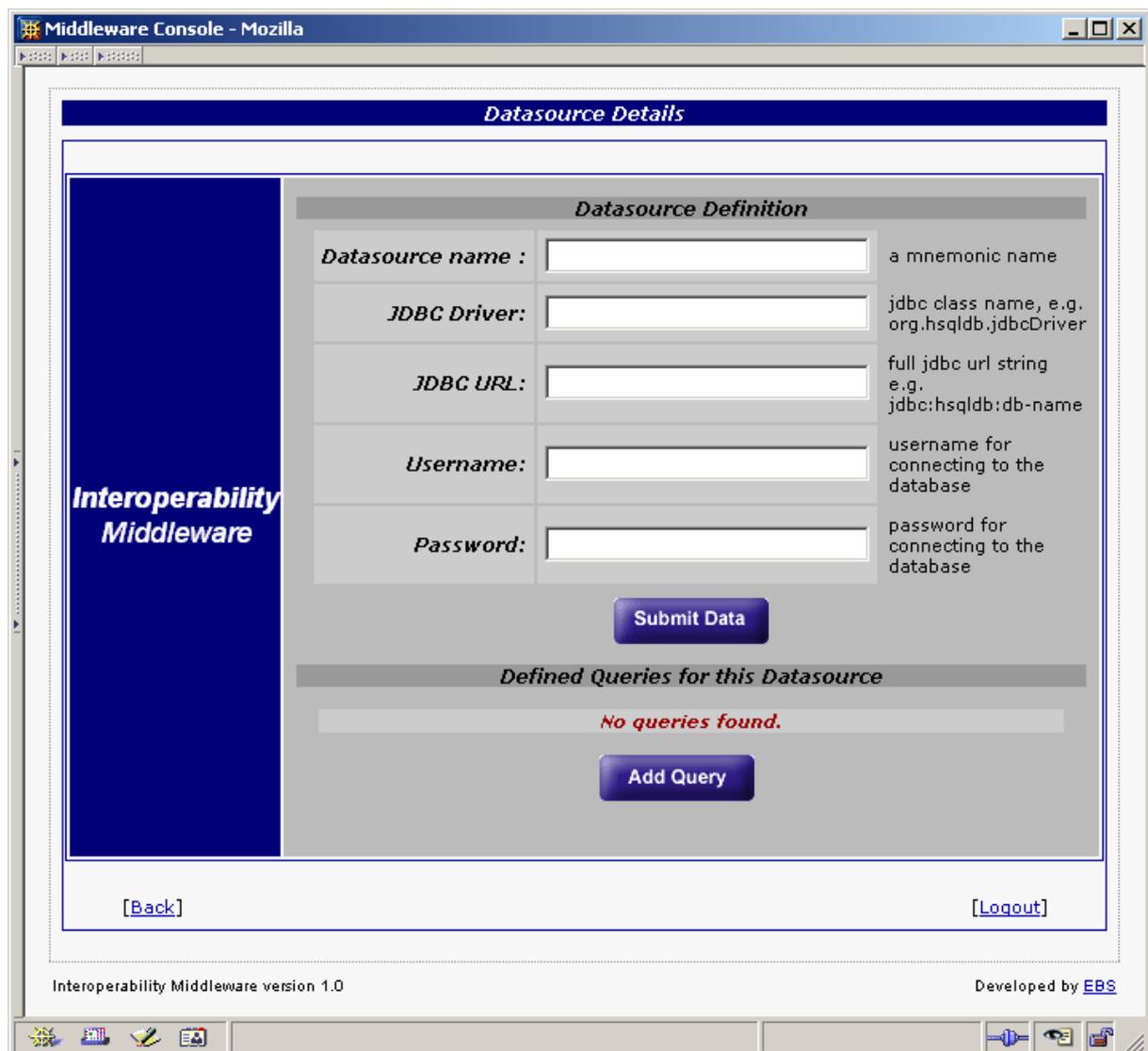
3.1.2 Διαχείριση των Συνδέσεων σε ΒΔ (Datasources)

Η οθόνη με τις επιλογές διαχείρισης συνδέσεων σε ΒΔ παρουσιάζεται στο Σχήμα 5. Οι διαθέσιμες επιλογές είναι: (α) επισκόπηση εγκατεστημένων συνδέσεων και (β) δήλωση νέας σύνδεσης (datasource).

Επιλέγοντας “Define a new datasource” εμφανίζεται η οθόνη ορισμού σύνδεσης με ΒΔ και ορισμού SQL queries, όπως αυτή φαίνεται στο Σχήμα 6.



Σχήμα 5: Κεντρική οθόνη διαχείρισης συνδέσεων ΒΔ

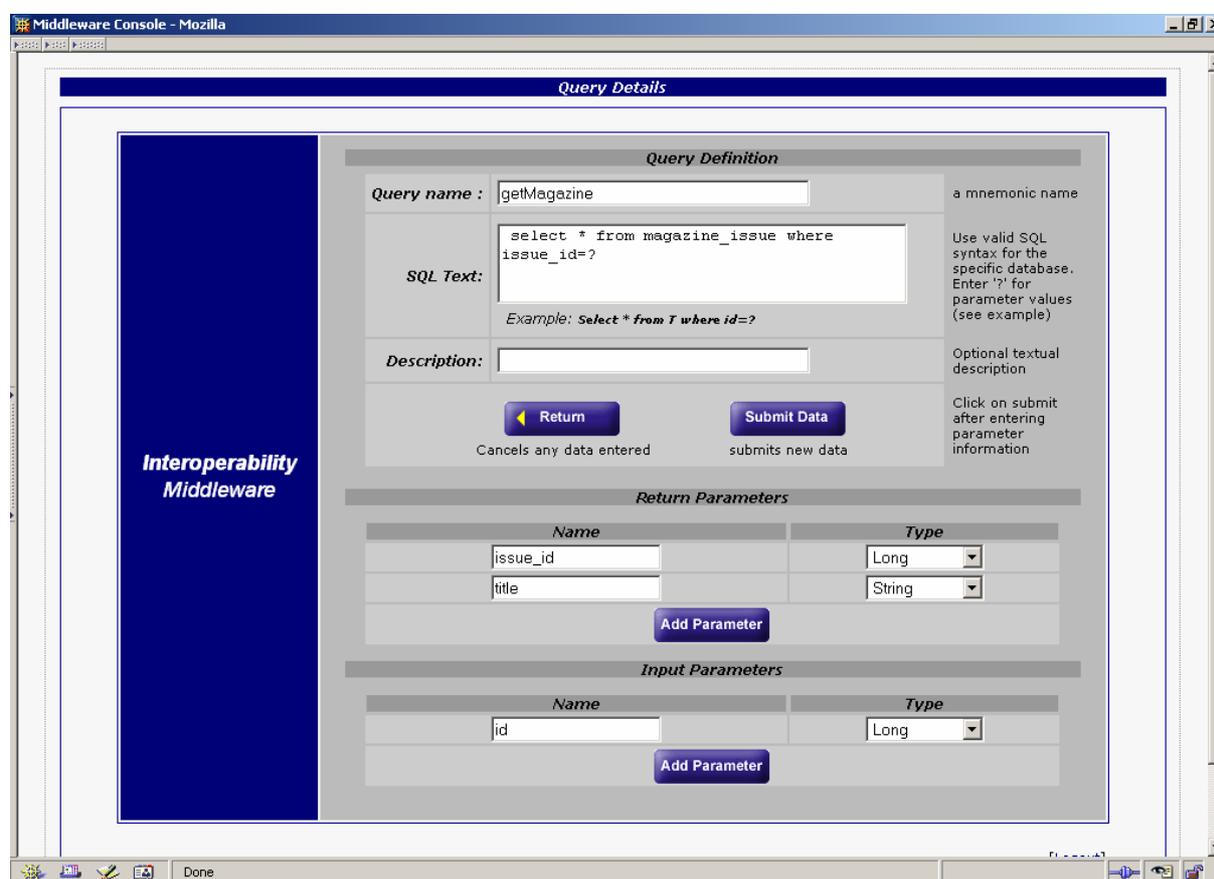


Σχήμα 6: Οθόνη δήλωσης νέας σύνδεσης

Στην οθόνη αυτή συμπληρώνουμε ένα μνημονικό όνομα για τη νέα σύνδεση και τα στοιχεία της σύνδεσης μέσω JDBC (συνήθως μπορεί να τα βρει κανείς στο documentation του JDBC driver ή σε σχετικά παραδείγματα κώδικα που τον συνοδεύουν):

- **JDBC driver:** η κλάση του driver, π.χ. για την PostgreSQL είναι org.postgresql.Driver, για την Oracle oracle.jdbc.driver.OracleDriver.
- **JDBC URL:** περιγραφή για τη σύνδεση με τη βάση, π.χ. για την PostgreSQL είναι jdbc:postgresql://[db server name]/[database name], ενώ για την Oracle είναι jdbc:oracle:thin:@[db server name]:[server port]:[database name].
- **Username:** όνομα χρήστη για τη σύνδεση στη βάση.
- **Password:** σύνθημα χρήστη για τη σύνδεση στη βάση.

Μόλις δοθούν τα στοιχεία αυτά ο χρήστης μπορεί να επιλέξει “Submit Data” για να αποθηκευτεί η δήλωση της σύνδεσης (datasource) που δήλωσε ή να επιλέξει “Add Query” για να προσθέσει και SQL queries. Στη δεύτερη περίπτωση εμφανίζεται η οθόνη ορισμού SQL query (Σχήμα 7).



Σχήμα 7: Οθόνη δήλωσης query για συγκεκριμένη σύνδεση

Στο παράδειγμα εμφανίζονται συμπληρωμένα τα στοιχεία ενός νέου query. Όπως φαίνεται στο σχήμα, συμπληρώνονται τα εξής:

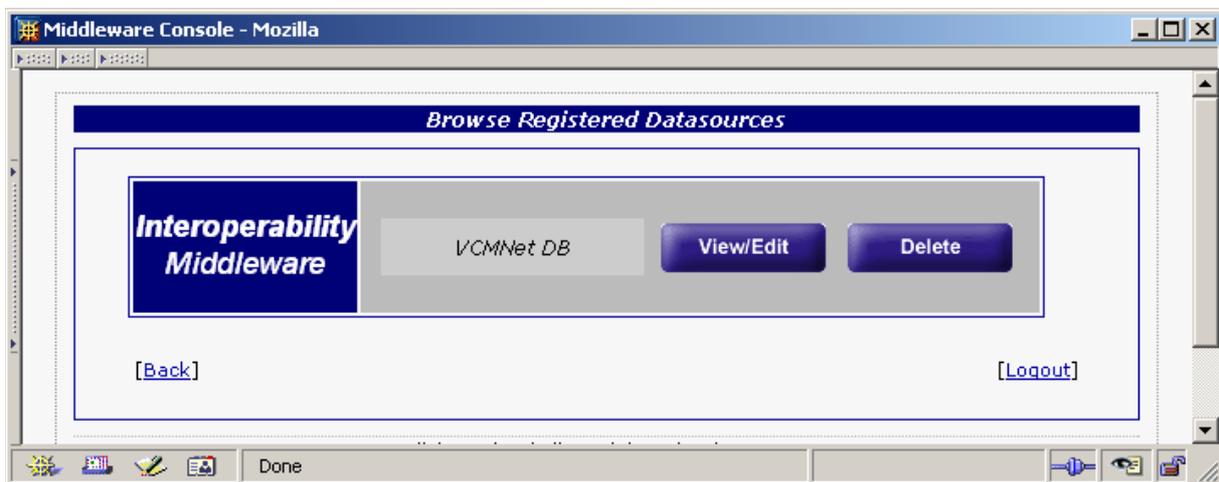
- **Query name:** ένα μνημονικό όνομα που θα χρησιμοποιηθεί και ως όνομα του operation του web service
- **SQL Text:** η περιγραφή σε SQL – σημειώστε ότι οι παράμετροι εισόδου σημειώνονται με ‘?’

- **Description:** μια λεκτική περιγραφή που μπορεί να συμπληρωθεί ως υπενθύμιση για το τι κάνει το συγκεκριμένο query
- Καθορισμός των παραμέτρων εισόδου – εξόδου του query:
 - Επιλέγοντας “Add Parameter” στις δύο περιοχές εισόδου – εξόδου προστίθεται μια ακόμη γραμμή για τον ορισμό μιας παραμέτρου
 - Σε κάθε γραμμή παραμέτρου συμπληρώνεται ένα όνομα και επιλέγεται από τη λίστα ο τύπος που ταιριάζει στην παράμετρο. **Τα ονόματα των παραμέτρων εξόδου πρέπει να ταιριάζουν με τα ονόματα των στηλών του πίνακα που επιστρέφει το query, ενώ τα ονόματα παραμέτρων εισόδου είναι αδιάφορα, παίζει ρόλο όμως η σειρά με την οποία δηλώνονται.**

Σημείωση: Τα ονόματα των queries και των παραμέτρων τους δεν μπορούν να είναι δεσμευμένα ονόματα στη γλώσσα Java (βλέπε Παράρτημα Α).

Όταν συμπληρωθούν όλα τα στοιχεία των παραμέτρων, επιλέγουμε “Submit Data” (η επιλογή “Return” ακυρώνει ότι στοιχεία έχουμε δώσει) και επιστρέφουμε στην οθόνη δήλωσης της σύνδεσης (Σχήμα 6). Εδώ μπορούμε να συμπληρώσουμε και άλλα queries πριν ολοκληρώσουμε τη δήλωση της σύνδεσης με τα queries της επιλέγοντας “Submit Data”. **Προσοχή, τα στοιχεία που έχουμε εισάγει δεν αποθηκεύονται μέχρι να επιλέξουμε “Submit Data”.**

Με την ολοκλήρωση της διαδικασίας επιστρέφουμε στην οθόνη διαχείρισης συνδέσεων, όπου αν επιλέξουμε “Browse registered datasource” εμφανίζεται η οθόνη με τη λίστα των συνδέσεων που παρουσιάζεται στο Σχήμα 8 όπου βλέπουμε τη σύνδεση που μόλις δημιουργήσαμε.

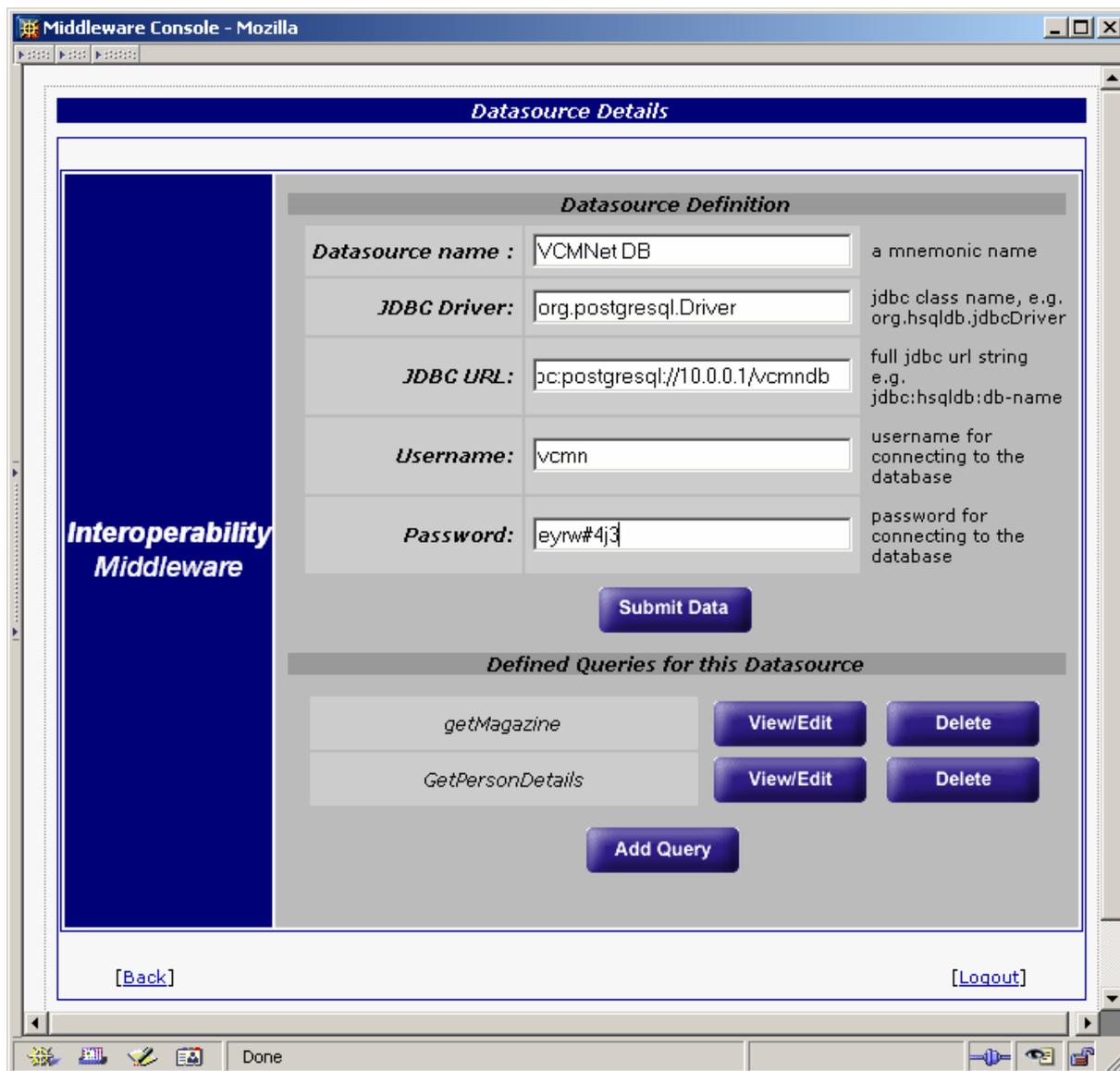


Σχήμα 8: Λίστα εγκατεστημένων συνδέσεων με ΒΔ

Στη γενική περίπτωση εμφανίζονται τα μνημονικά ονόματα όλων των συνδέσεων με την επιλογή να δούμε και πιθανόν να τροποποιήσουμε τα στοιχεία τους ή να τις διαγράψουμε. Η επιλογή της διαγραφής (“Delete”) οδηγεί σε οθόνη επιβεβαίωσης, ενώ η επιλογή “View/Edit” εμφανίζει την οθόνη που παρουσιάζεται στο Σχήμα 9.

Η οθόνη αυτή είναι παρόμοια με εκείνη που χρησιμοποιήθηκε για τη δήλωση της σύνδεσης και επιτρέπει την απλή επισκόπηση των στοιχείων της σύνδεσης και των

queries της ή την τροποποίησή τους με τον ίδιο τρόπο που περιγράφηκε παραπάνω για τον ορισμό τους.



Σχήμα 9: Πληροφορίες / διόρθωση στοιχείων σύνδεσης

3.1.3 Ανάπτυξη Επεκτάσεων

Όπως αναφέρθηκε στην εισαγωγή, οι επεκτάσεις αποτελούν ένα εναλλακτικό τρόπο υλοποίησης / ορισμού υπηρεσιών του ΕΛΔ που προσφέρει πολύ περισσότερες δυνατότητες από της συνδέσεις μέσω ΒΔ, αλλά και περισσότερο πολύπλοκες στη δημιουργία τους (απαιτείται προγραμματισμός σε Java). Η παράγραφος αυτή περιγράφει πως υλοποιούνται οι επεκτάσεις και η επόμενη πως αυτές δηλώνονται / εγκαθίστανται μέσω του user interface του ΕΛΔ.

Οι επεκτάσεις του ΕΛΔ υλοποιούνται ως Java Managed Beans (MBeans), όπως αυτά περιγράφονται στο Java Management Extensions (JMX) Specification. Η έκδοση του JMX Specification που υποστηρίζεται από την έκδοση 1.0 του ΕΛΔ είναι

η 1.0, όπως αυτή υλοποιείται στον JBoss 3.2.2. Σχετικές πληροφορίες μπορεί κανείς να δει στα εξής URL:

- <http://www.jboss.org/developers/projects/jboss/jbossmx>
- <http://java.sun.com/products/JavaManagement/>

Για την ανάπτυξη μιας επέκτασης στο ΕΛΔ, ακολουθούμε τα εξής βήματα:

1. Γράφουμε τον κώδικα του MBean σύμφωνα με τους κανόνες ανάπτυξης του JMX Specification.
2. Πακετάρουμε το MBean σε ένα απλό JAR αρχείο. (Δεν υπάρχει καμία ειδική απαίτηση όσον αφορά το περιεχόμενό του.) Στην παρούσα έκδοση του ΕΛΔ κάθε MBean πρέπει να βρίσκεται σε ξεχωριστό JAR αρχείο.
3. Επιλέγουμε από την κύρια οθόνη Manage Extensions > Define a new extension και συμπληρώνουμε τα στοιχεία του MBean (η αναλυτική περιγραφή δίνεται στην επόμενη παράγραφο - 3.1.4).

Για κάθε επέκταση γράφουμε τουλάχιστον δύο αρχεία: την Java class που υλοποιεί το MBean, συνεπώς και τη διασύνδεση με το πληροφοριακό σύστημα, και το αντίστοιχο Java interface που ορίζει τις μεθόδους που θα είναι προσβάσιμοι στους clients (βλέπε JMX Specification).

Για παράδειγμα το αρχείο MyClass.java (Java Class *MyClass*) που δίνεται παρακάτω υλοποιεί μια επέκταση του ΕΛΔ που συνδέεται με το υποθετικό πληροφοριακό σύστημα και καλεί κάποια συγκεκριμένη business function:

```
package gr.egif.testmbean;
public class MyClass implements MyClassMBean
{
    public String findNameByAFM(String AFM)
    {
        String name;

        // επικοινωνία με το πληροφοριακό σύστημα
        // και κλήση της κατάλληλης business function

        return name;
    }
}
```

Αντίστοιχα το αρχείο MyClassMbean.java (Java Interface *MyClassMbean*) ορίζει τις μεθόδους της επέκτασης που επιτρέπεται να δημοσιοποιηθούν μέσω web services από το ΕΛΔ:

```
package gr.egif.testmbean;
public interface MyClassMBean
{
    public String findNameByAFM(String AFM);
}
```

Φτιάχνουμε ένα αρχείο JAR σύμφωνα με τους γνωστούς κανόνες (τα αρχεία .class πρέπει να βρίσκονται σε directories σύμφωνα με το package name τους, δηλαδή στο παραπάνω παράδειγμα στο gr/egif/testmbean). Το αρχείο αυτό είναι έτοιμο για εγκατάσταση στο ΕΛΔ μέσω του user interface – η διαδικασία αυτή περιγράφεται στην επόμενη παράγραφο (3.1.4).

Στην περίπτωση που η υπηρεσία που σκοπεύουμε να δημιουργήσουμε με τη συγκεκριμένη επέκταση, τότε η υλοποίηση του MBean θα πρέπει να λάβει υπόψη τους συγκεκριμένους τύπους δεδομένων που απαιτούνται. Ο προγραμματιστής θα πρέπει να λάβει υπόψη τη δομή και τα ονόματα των τύπων και των πεδίων ώστε οι μέθοδοι της επέκτασης / MBean που θα υλοποιηθούν και θα δηλωθούν στα παραπάνω αρχεία (class και interface) να δέχονται και να επιστρέφουν τους τύπους αυτούς. Στη γενική περίπτωση οι τύποι αυτοί δίνονται σε αρχεία XSD από τα οποία εξάγονται οι κατάλληλες κλάσεις είτε μέσω αυτόματων εργαλείων (π.χ. WSDL2Java) είτε με χειροκίνητη ανάλυση του αρχείου. Έτσι, στην περίπτωση όπου οι τύποι δεδομένων είναι συγκεκριμένοι, το JAR αρχείο θα πρέπει να περιέχει και τις αντίστοιχες κλάσεις του.

Για παράδειγμα, αν έχει δοθεί το παρακάτω XSD

```
<xsd:complexType name="PersonName">
  <xsd:sequence>
    <xsd:element name="first_name" type="xsd:string" minOccurs="1"
maxOccurs="1" nillable="true"/>
    <xsd:element name="last_name" type="xsd:string" minOccurs="1"
maxOccurs="1" nillable="true"/>
  </xsd:sequence>
</xsd:complexType>
```

θα πρέπει να δημιουργηθεί η κατάλληλη κλάση PersonName

```
public class PersonName
{
    public String first_name;
    public String last_name;
}
```

και να τροποποιηθεί η αντίστοιχη μέθοδος της επέκτασης:

```
public PersonName findNameByAFM(String AFM);
```

Τέλος, το PersonName.class θα πρέπει να συμπεριληφθεί στο JAR που θα δοθεί για εγκατάσταση στο ΕΛΔ.

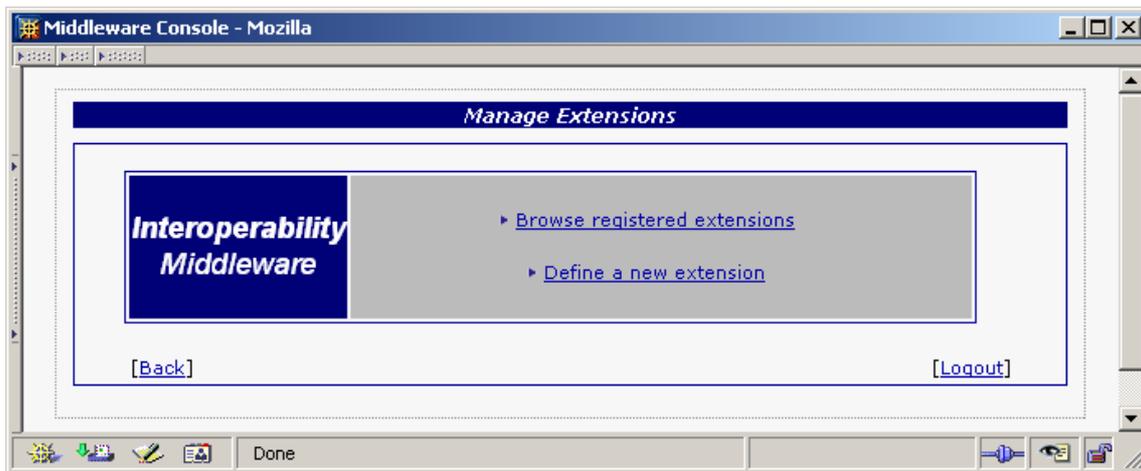
3.1.4 Διαχείριση των Επεκτάσεων (Extensions)

Η επιλογή αυτή επιτρέπει τη δήλωση / εγκατάσταση νέων επεκτάσεων (που δημιουργήθηκαν με τον τρόπο που αναφέρεται στην προηγούμενη παράγραφο) καθώς και την επισκόπηση ή διαγραφή των ήδη εγκατεστημένων επεκτάσεων.

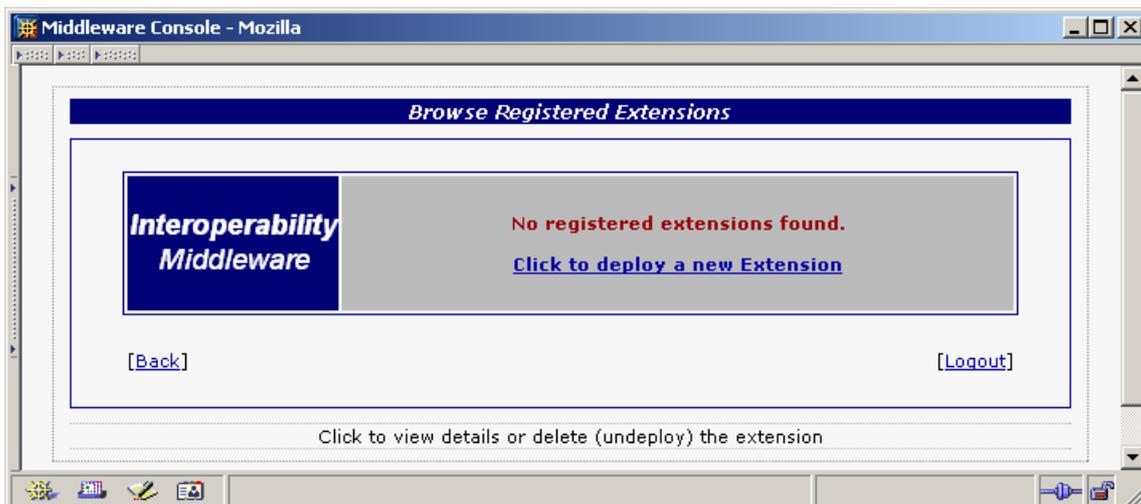
Επιλέγοντας "Manage Extensions" από την αρχική οθόνη, το ΕΛΔ παρουσιάζει την οθόνη που φαίνεται στο Σχήμα 10.

Η πρώτη επιλογή οδηγεί στο Σχήμα 11 ή στο Σχήμα 13, ανάλογα με το αν έχουν δηλωθεί επεκτάσεις ή όχι.

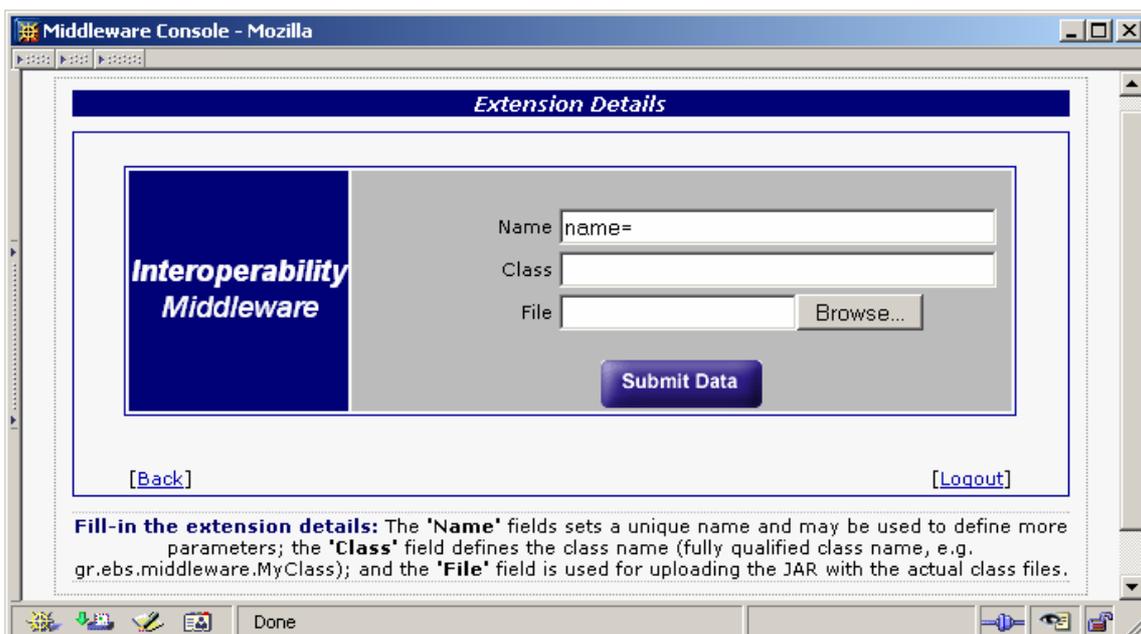
Επιλέγοντας "Click to deploy a new Extension" στο Σχήμα 11, ή "Define a new Extension" στην αρχική οθόνη διαχείρισης, εμφανίζεται η οθόνη δήλωσης / εγκατάστασης επεκτάσεων (Σχήμα 12).



Σχήμα 10: Κεντρική οθόνη διαχείρισης επεκτάσεων



Σχήμα 11: Άδεια λίστα εγκατεστημένων επεκτάσεων



Σχήμα 12: Δήλωση και εγκατάσταση επέκτασης

Στην οθόνη αυτή δίνουμε τα εξής στοιχεία:

- **Name:** Συμπληρώνουμε π.χ. *name=MyMBeanName*. Με την επιτυχημένη εγκατάσταση η επέκταση θα είναι διαθέσιμη με το όνομα που δώσαμε σε αυτό το πεδίο. Το όνομα αυτό μπορεί να είναι όποιο θέλουμε, αρκεί αυτό να ακολουθεί τους κανόνες για JMX object naming *χωρίς όμως το domain name*. Ένα JMX object name έχει τη γενική μορφή

```
domain-name:key1=value1[,key2=value2,...,keyN=valueN]
```

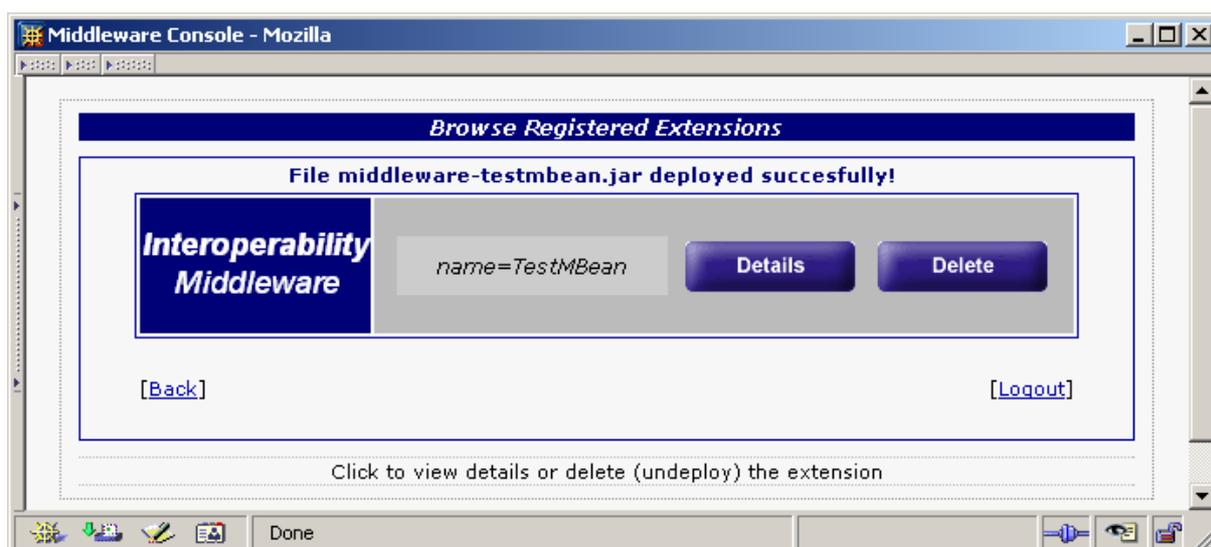
Το domain-name συμπληρώνεται αυτόματα από το ΕΛΔ γι αυτό και δεν πρέπει να το συμπληρώσουμε. Γενικά δίνουμε ένα όνομα μετά το ήδη συμπληρωμένο "name=", αλλά ο έμπειρος προγραμματιστής μπορεί να τροποποιήσει το πεδίο κατάλληλα ώστε να έχει τη μορφή "key1=value1[,key2=value2,...,keyN=valueN]" και να περάσει τις κατάλληλες παραμέτρους στη συγκεκριμένη δήλωση του MBean.

Ας σημειωθεί ότι αυτό που θα συμπληρώσουμε στο πεδίο Name πρέπει να είναι μοναδικό (να μην χρησιμοποιείται ήδη από κάποια άλλη εγκατεστημένη επέκταση του ΕΛΔ).

- **Class:** Το όνομα της κλάσης που υλοποιεί το MBean (όχι του interface). Για το παράδειγμα υλοποίησης επεκτάσεων της προηγούμενης παραγράφου (3.1.3) το όνομα της κλάσης είναι *gr.egif.testmbean.MyClass*.
- **File:** Πατάμε το πλήκτρο browse και στο file browser που εμφανίζεται επιλέγουμε το αρχείο JAR από εκεί που το έχουμε αποθηκεύσει (στο παράδειγμα της προηγούμενης παραγράφου).

Μόλις πατήσουμε Submit Data, το ΕΛΔ θα δημιουργήσει μία νέα επέκταση που θα είναι διαθέσιμη με το όνομα που δώσαμε στο πεδίο name.

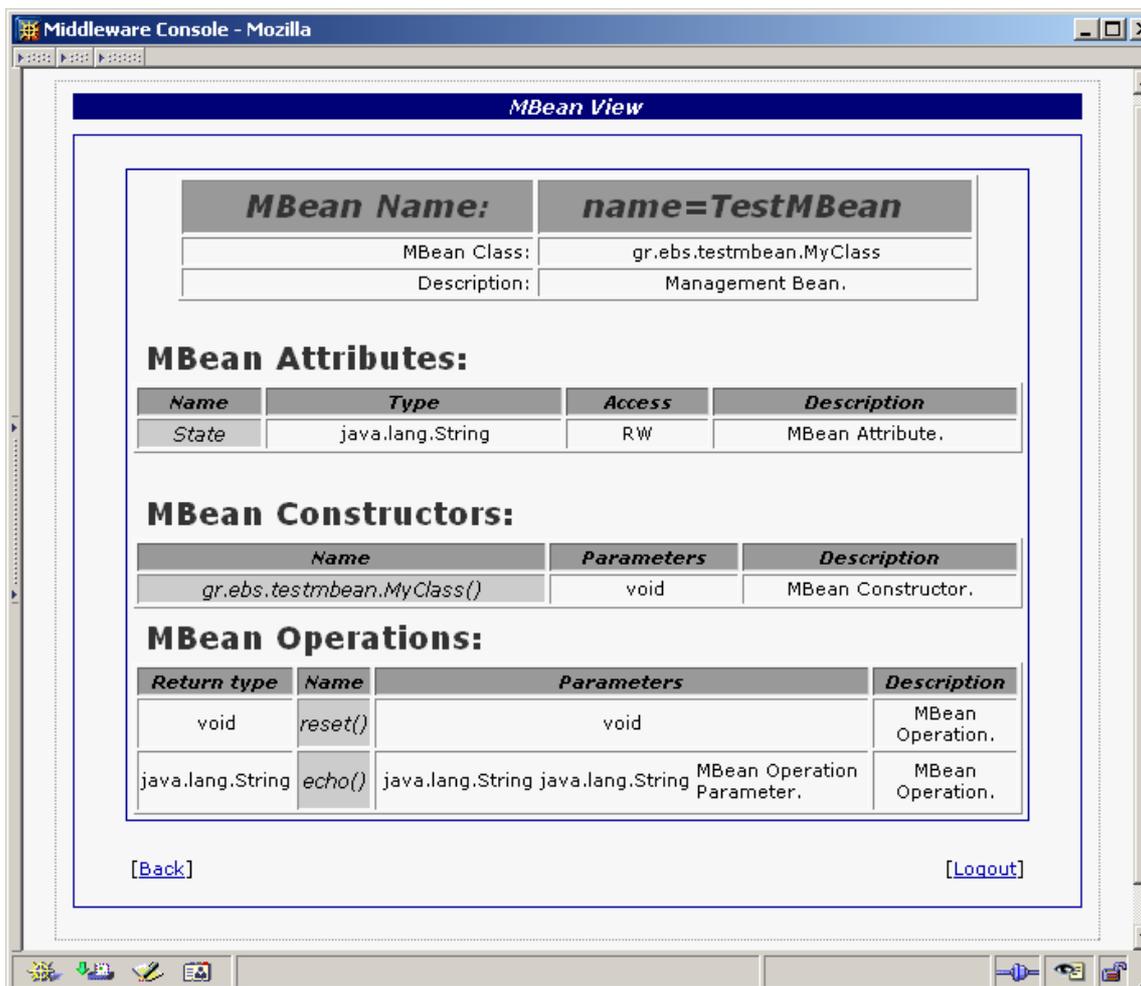
Η επιτυχής δήλωση / εγκατάσταση της επέκτασης επιβεβαιώνεται στην οθόνη με τη λίστα των εγκατεστημένων επεκτάσεων (Σχήμα 8).



Σχήμα 13: Λίστα εγκατεστημένων επεκτάσεων (στην γενική περίπτωση χωρίς το μήνυμα)

Η επιλογή "Delete" επιτρέπει τη διαγραφή της επέκτασης από το ΕΛΔ (μετά από επιβεβαίωση), ενώ η επιλογή "Details" εμφανίζει την οθόνη με τα στοιχεία της

επέκτασης (όπως αυτά αναγνωρίστηκαν από το ΕΛΔ κατά την εγκατάσταση της επέκτασης) όπως παρουσιάζονται στο Σχήμα 14.

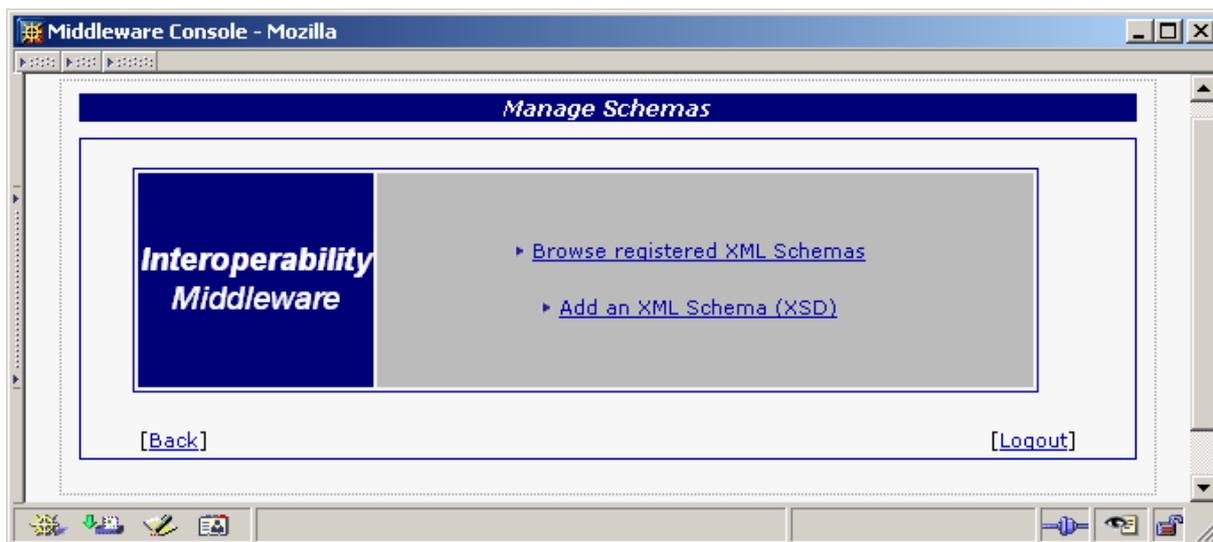


Σχήμα 14: Στοιχεία εγκατεστημένης επέκτασης

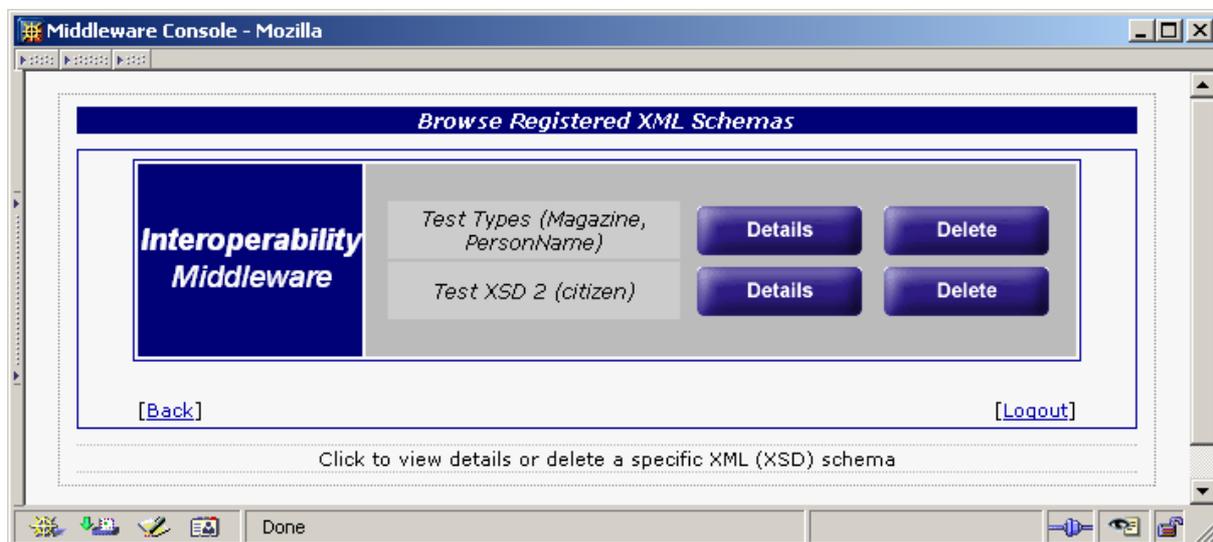
3.1.5 Διαχείριση Σχημάτων XSD (Schemas)

Η επιλογή "Manage Schemas" οδηγεί στην οθόνη που παρουσιάζεται στο Σχήμα 15. σε αναλογία με τις άλλες επιλογές διαχείρισης που παρουσιάστηκαν στην προηγούμενες παραγράφους, η οθόνη αυτή παρέχει δύο επιλογές για: (α) την επισκόπηση των δηλωμένων / εγκατεστημένων αρχείων XSD στο ΕΛΔ, (β) τον ορισμό νέου σχήματος.

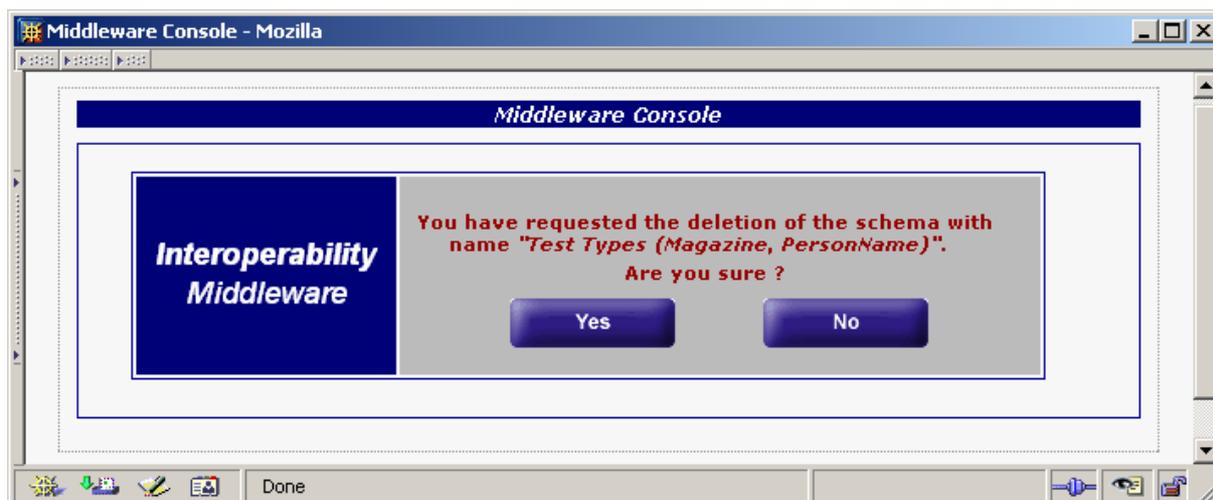
Επιλέγοντας "Browse registered XML schemas" εμφανίζεται λίστα με τα εγκατεστημένα σχήματα (όπως παρουσιάζεται στο Σχήμα 16) και παρέχονται επιλογές για επισκόπηση του σχήματος ή διαγραφή του. Στο συγκεκριμένο παράδειγμα έχουν δηλωθεί / εγκατασταθεί δύο σχήματα. Αν επιλέξουμε τη διαγραφή τους (π.χ. πατώντας το "Delete" δεξιά από το πρώτο) το ΕΛΔ ζητάει επιβεβαίωση όπως φαίνεται στο Σχήμα 17. Αν επιλέξουμε το πλήκτρο "Details" εμφανίζεται παράθυρο με την περιγραφή του σχήματος σε XML, όπως αυτό παρουσιάζεται στο Σχήμα 18.



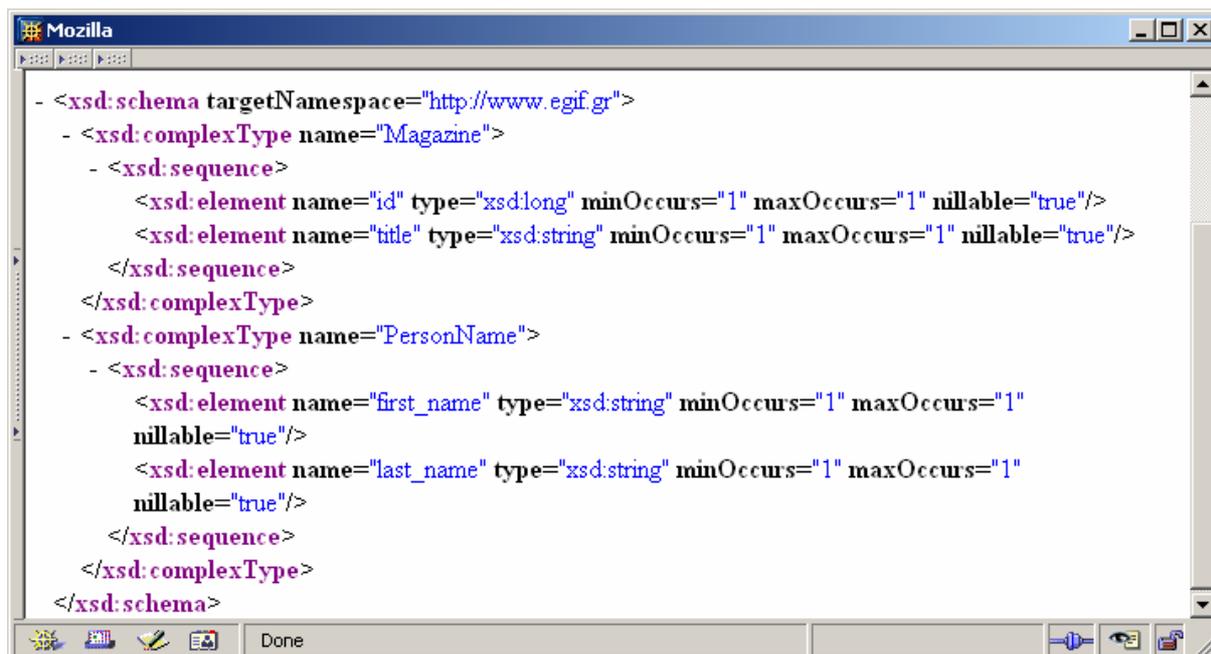
Σχήμα 15: Κεντρική οθόνη διαχείρισης σχημάτων XSD



Σχήμα 16: Λίστα εγκατεστημένων σχημάτων

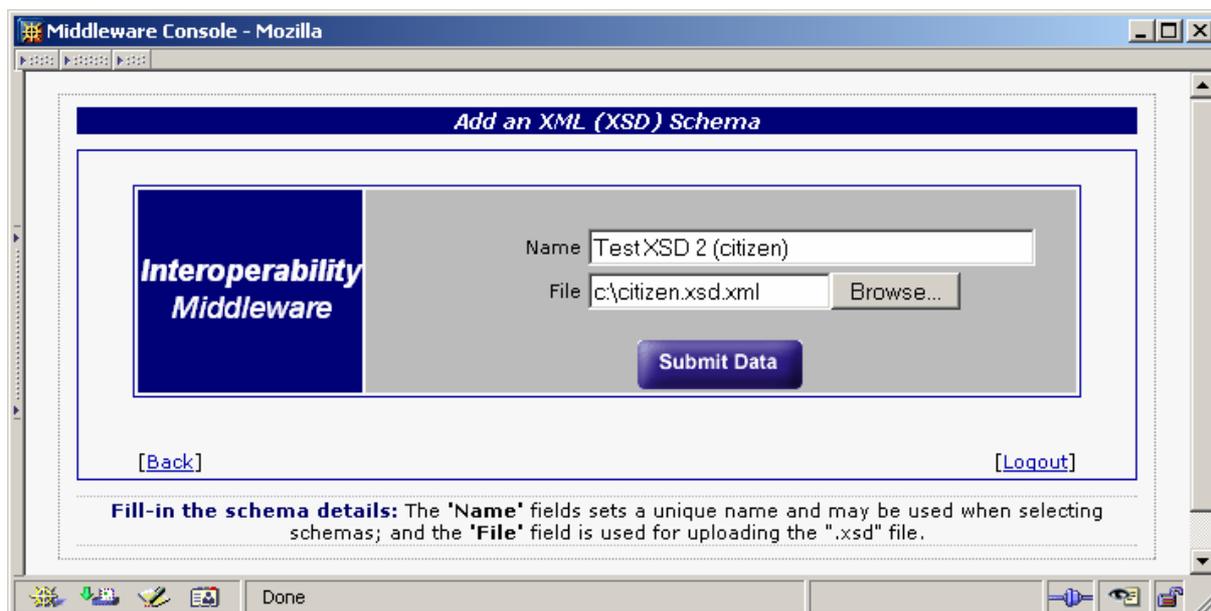


Σχήμα 17: Επιβεβαίωση διαγραφής



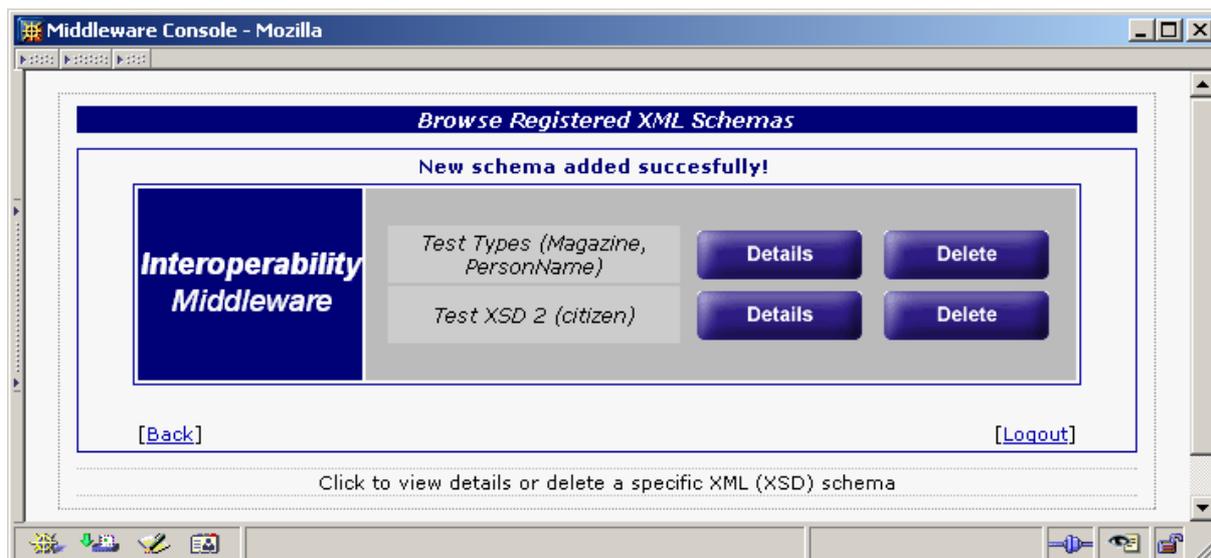
Σχήμα 18: Περιεχόμενα του σχήματος XSD

Η δήλωση νέου σχήματος γίνεται μέσω της οθόνης που παρουσιάζεται στο Σχήμα 19. Ο χρήστης-διαχειριστής δίνει μια λεκτική περιγραφή για το σχήμα και επιλέγει με το file dialog, που εμφανίζεται όταν πατήσει το “Browse”, το κατάλληλο αρχείο XSD.



Σχήμα 19: Οθόνη δήλωσης / εγκατάστασης σχήματος XSD

Τα δεδομένα αποστέλλονται στο ΕΛΔ όταν πατηθεί το “Submit Data”, αποθηκεύονται στο ΕΛΔ και εμφανίζεται η οθόνη επιτυχίας με την πλήρη λίστα των XSD (Σχήμα 20).

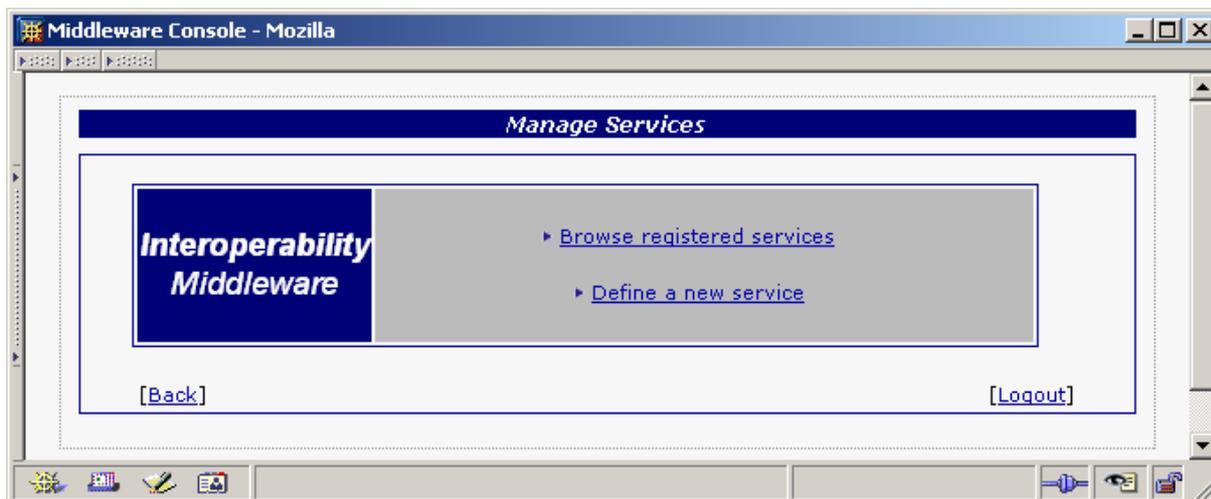


Σχήμα 20: Επιτυχής δήλωση νέου σχήματος

Προσοχή: στα XSD που δηλώνονται στο ΕΛΔ θα πρέπει όλοι οι τύποι δεδομένων (data types) να έχουν την παράμετρο `nillable=true`: για παράδειγμα `<element name="name" nillable="true" type="xsd:string"/>`²

3.1.6 Διαχείριση των Υπηρεσιών (Services)

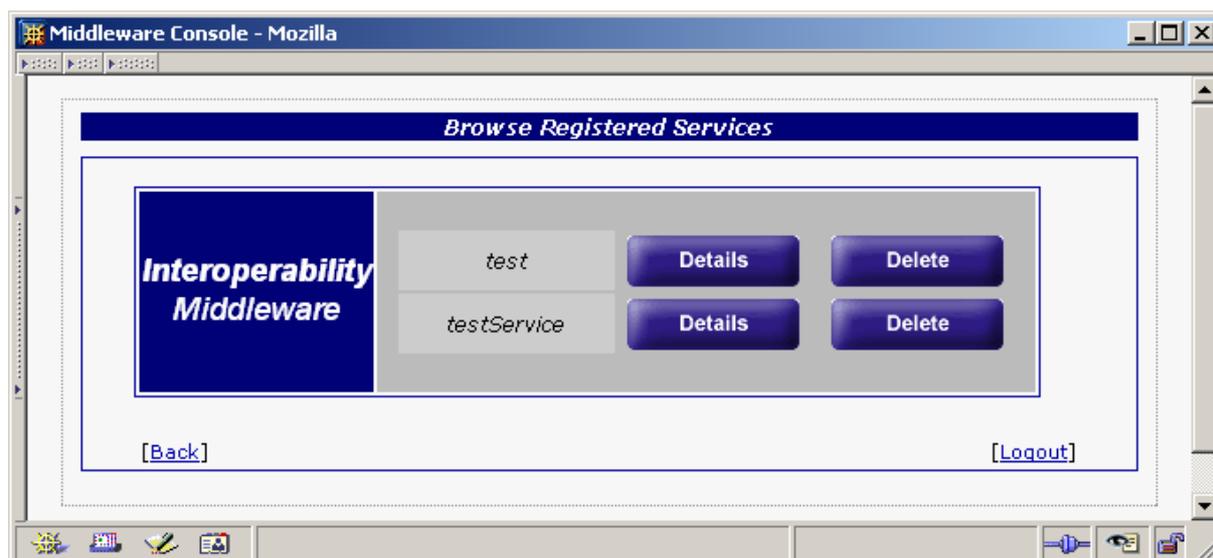
Η επιλογή “Manage Services” της κεντρικής οθόνης επιλογών του ΕΛΔ εμφανίζει την οθόνη που παρουσιάζεται στο Σχήμα 22.



Σχήμα 21: Κεντρική οθόνη διαχείρισης υπηρεσιών

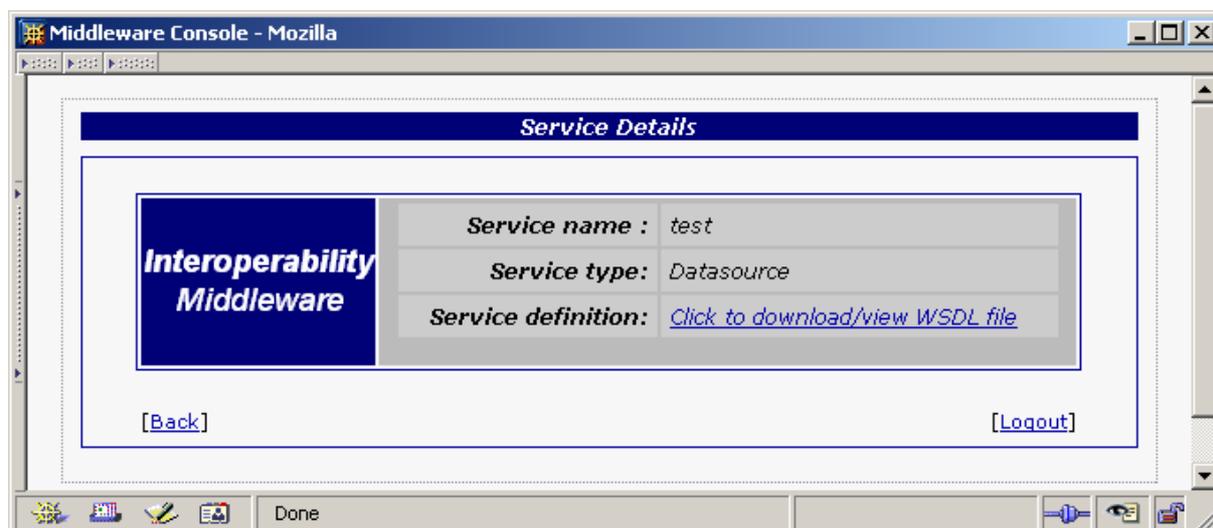
Στο συγκεκριμένο παράδειγμα εμφανίζονται δύο εγκατεστημένες υπηρεσίες. Για κάθε υπηρεσία είναι διαθέσιμες οι επιλογές που είδαμε και σε όλα τις παρόμοιες οθόνες του ΕΛΔ: λεπτομέρειες (“Details”) και διαγραφή (“Delete”) με επιβεβαίωση.

² Η αμέσως επόμενη έκδοση του ΕΛΔ θα προσθέτει αυτόματα την παράμετρο αυτή στους τύπους δεδομένων των αρχείων XSD που δηλώνονται.



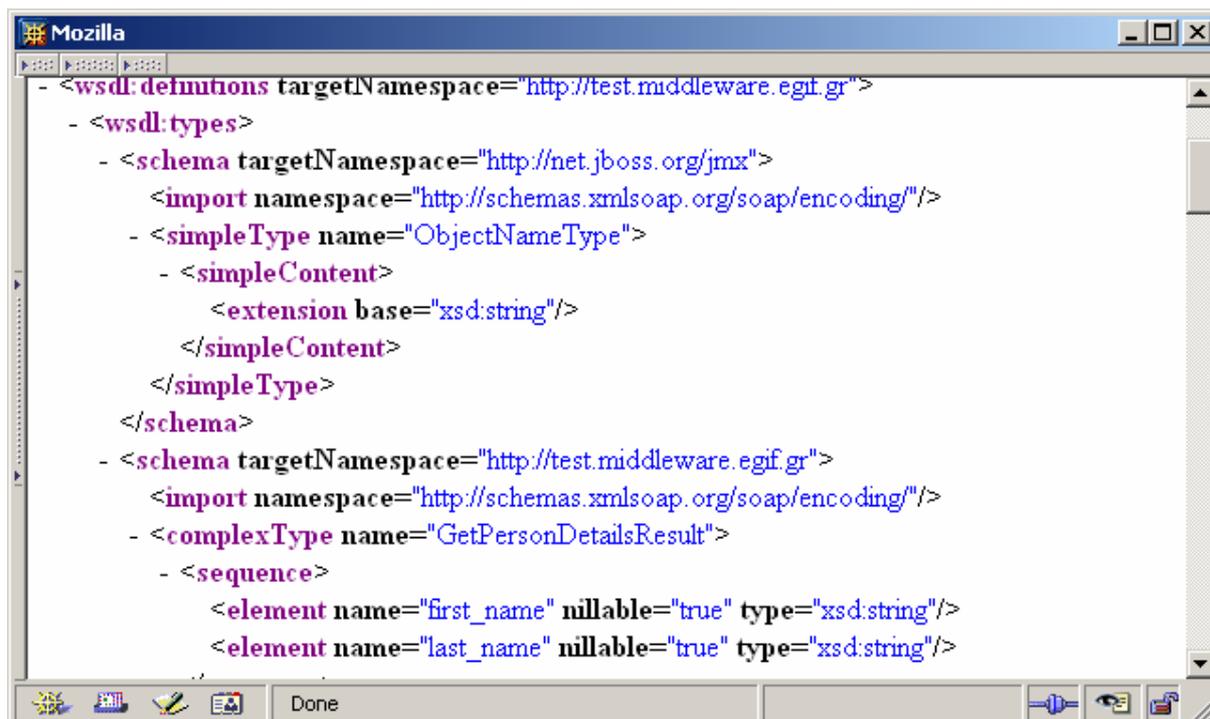
Σχήμα 22: Λίστα εγκατεστημένων υπηρεσιών

Επιλέγοντας να δούμε τις λεπτομέρειες εμφανίζεται η οθόνη που παρουσιάζεται στο Σχήμα 23. Εμφανίζεται το όνομα της υπηρεσίας, ο τύπος της καθώς και ένας υπερ-σύνδεσμος στην περιγραφή της υπηρεσίας σε WSDL. Οι δυνατοί τύποι των υπηρεσιών είναι δύο: (α) υλοποίηση με σύνδεση στη βάση (Datasource) – δηλαδή με χρήση δηλωμένου datasource και SQL queries, ή (β) υλοποίηση με επέκταση (Extension), με χρήση δηλαδή κάποιου από τις εγκατεστημένες επεκτάσεις.



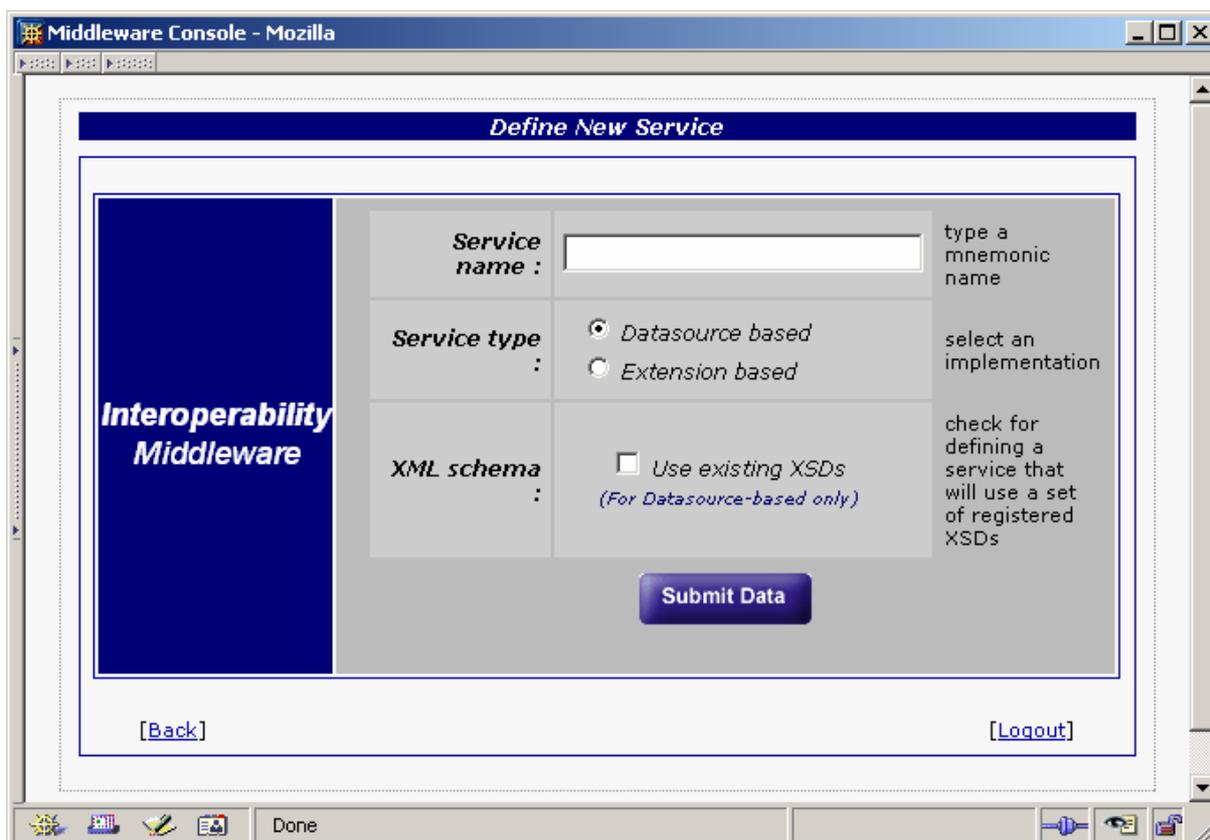
Σχήμα 23: Στοιχεία υπηρεσίας

Επιλέγοντας το σύνδεσμο του αρχείου WSDL της υπηρεσίας ανοίγει παράθυρο που έχει τη μορφή που παρουσιάζεται στο Σχήμα 24. Το παράθυρο επιτρέπει την επισκόπηση της περιγραφής XML / WSDL της συγκεκριμένης υπηρεσίας και μπορεί να αποθηκευτεί (στους περισσότερους web browsers με δεξί click και επιλογή αποθήκευσης) ώστε να αποσταλεί στους developers της εφαρμογής πελάτη.



Σχήμα 24: WSDL description υπηρεσίας

Η δεύτερη επιλογή της αρχικής οθόνης διαχείρισης υπηρεσιών του ΕΛΔ εμφανίζει την οθόνη που παρουσιάζεται στο Σχήμα 25. Αυτή επιτρέπει τον ορισμό μιας νέας υπηρεσίας.



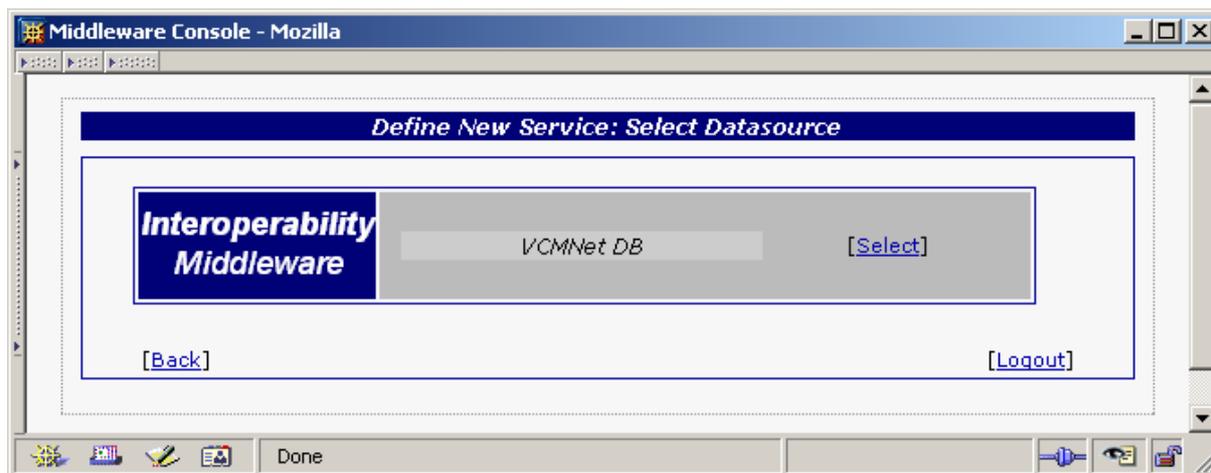
Σχήμα 25: Ορισμός νέας υπηρεσίας του ΕΛΔ

Οι βασικές επιλογές για τον ορισμό μιας υπηρεσίας είναι: (α) ένα όνομα για την υπηρεσία, (β) επιλογή του τύπου, (γ) αν η υπηρεσία θα βασιστεί σε datasource, αν θα βασιστεί σε δεδομένο σχήμα XSD

Προσοχή: Τα ονόματα των υπηρεσιών δεν μπορούν είναι δεσμευμένα ονόματα στη γλώσσα Java (βλέπε Παράρτημα Α). Επίσης, καθώς εσωτερικά θα δημιουργηθούν κλάσεις και αντικείμενα με αυτό το όνομα, δεν θα πρέπει να περιέχονται κενά και γενικά μη αλφαριθμητικοί χαρακτήρες.

Όπως αναφέρθηκε και παραπάνω, η υλοποίηση μπορεί να γίνει μέσω datasource ή extension του ΕΛΔ. Στην πρώτη περίπτωση υπάρχει η δυνατότητα χρήσης σχήματος XSD που έχει δηλωθεί στο ΕΛΔ ώστε οι τύποι δεδομένων που χρησιμοποιούν τα SOAP operations να ορίζονται από αυτά τα σχήματα. Στην περίπτωση υλοποίησης με επέκταση κάτι τέτοιο δεν χρειάζεται, καθώς τα όποια XSD μπορούν να ληφθούν υπόψη κατά τη συγγραφή της επέκτασης (όπως αναφέρεται παραπάνω στην παράγραφο 3.1.3).

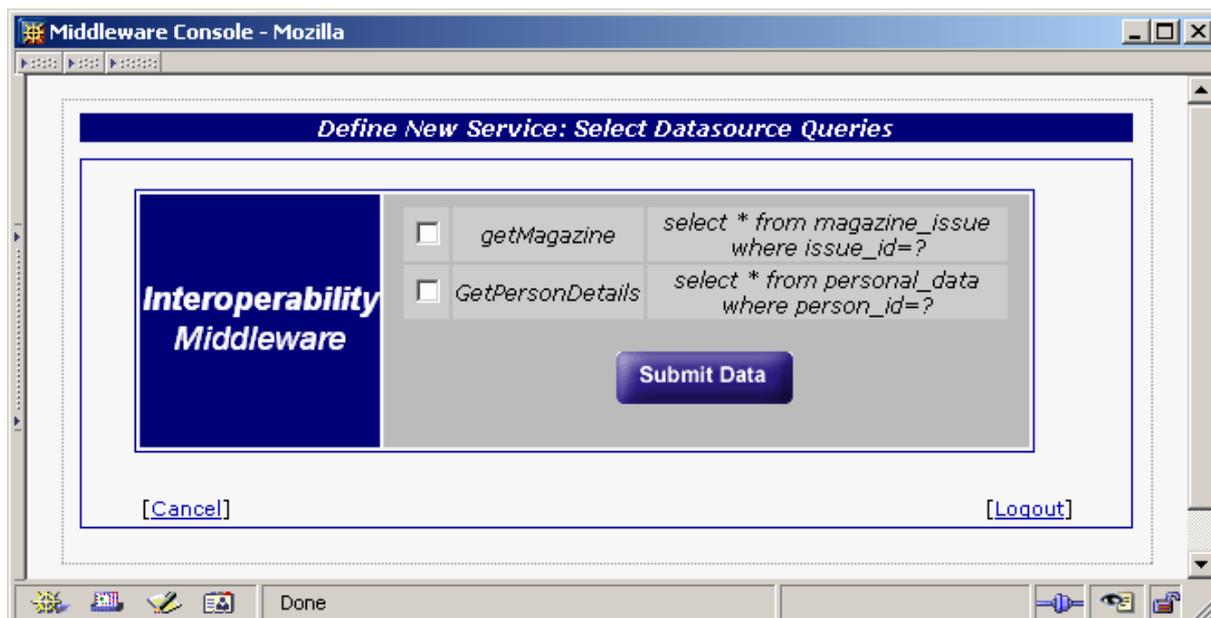
Στην περίπτωση λοιπόν που επιλεγεί υλοποίηση με datasource ή επόμενη οθόνη εμφανίζει τα διαθέσιμα datasources για να επιλέξει ο χρήστης-διαχειριστής, όπως φαίνεται στο Σχήμα 26 (στη συγκεκριμένη περίπτωση έχει δηλωθεί μόνο μια σύνδεση).



Σχήμα 26: Επιλογή Datasource για τη νέα υπηρεσία

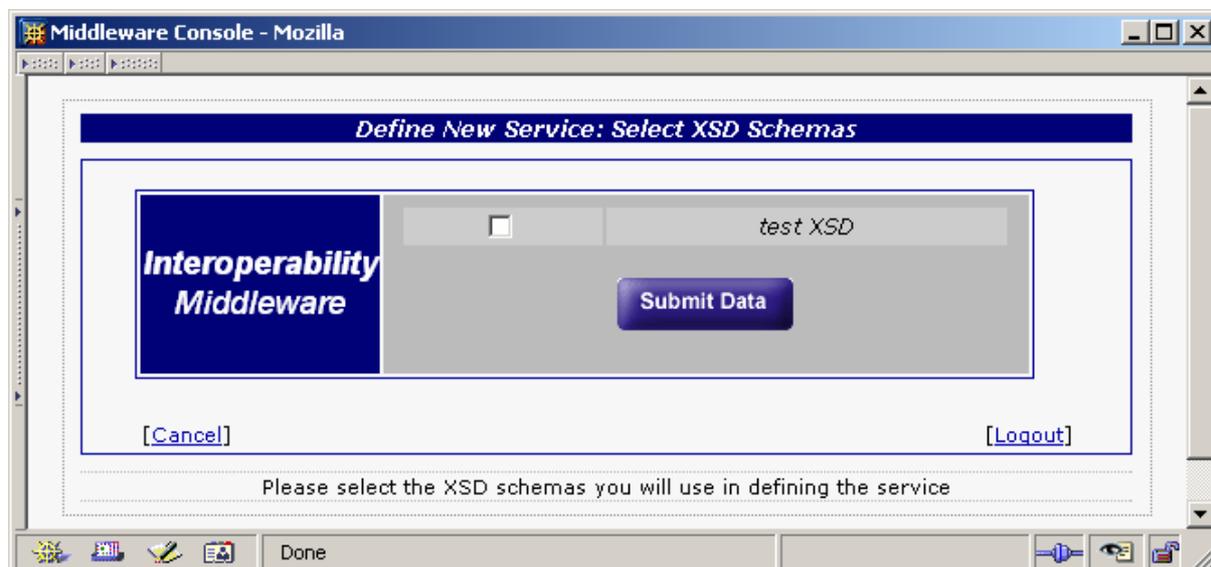
Η επιλογή συγκεκριμένου datasource οδηγεί στην οθόνη που παρουσιάζεται στο Σχήμα 27, όπου πρέπει να επιλεγεί ένα ή περισσότερα SQL queries του datasource. Για κάθε query που επιλέγεται θα δημιουργηθεί και ένα αντίστοιχο SOAP operation στο συγκεκριμένο web service.

Αν δεν είχε επιλεγεί η χρήση σχήματος XSD, μετά από αυτή την οθόνη το ΕΛΔ δημιουργεί την κατάλληλη υπηρεσία και την ενεργοποιεί (η διαδικασία απαιτεί μερικά δευτερόλεπτα ανάλογα με τη χρησιμοποίηση του εξυπηρετητή του ΕΛΔ). Μόλις η δημιουργία της υπηρεσίας ολοκληρωθεί, εμφανίζεται η οθόνη λεπτομερειών της υπηρεσίας (Σχήμα 23) και ο χρήστης-διαχειριστής μπορεί να δει την περιγραφή WSDL που δημιουργήθηκε για τη νέα υπηρεσία. Το ΕΛΔ μπορεί να δεχθεί και να εξυπηρετήσει αμέσως αιτήσεις στη νέα αυτή υπηρεσία, ενώ σε κάθε επανεκκίνησή του ενεργοποιεί όλες τις δηλωμένες υπηρεσίες.



Σχήμα 27: Επιλογή queries για τη νέα υπηρεσία

Σε περίπτωση που επιλέχθηκε η χρήση σχήματος XSD, τότε μετά την επιλογή SQL queries εμφανίζεται η οθόνη που παρουσιάζεται στο Σχήμα 28. Εδώ δίνεται η λίστα με τα διαθέσιμα (δηλωμένα στο ΕΛΔ) σχήματα και ο χρήστης-διαχειριστής καλείται να επιλέξει ένα ή περισσότερα σχήματα.



Σχήμα 28: Επιλογή XSD για τη νέα υπηρεσία

Με την επιλογή των σχημάτων, το ΕΛΔ αναλύει τους τύπους δεδομένων και εμφανίζει την οθόνη που δίνεται στο Σχήμα 29. Σε αυτή την οθόνη πρέπει να επιλεγούν οι τύποι για τις παραμέτρους εισόδου και το αντικείμενο εξόδου³. Ο χρήστης δίνει τα ονόματα παραμέτρων που θα εμφανιστούν στο WSDL και επιλέγει τους τύπους (η λίστα διαθέσιμων τύπων έχει προκύψει από τα αρχεία XSD που

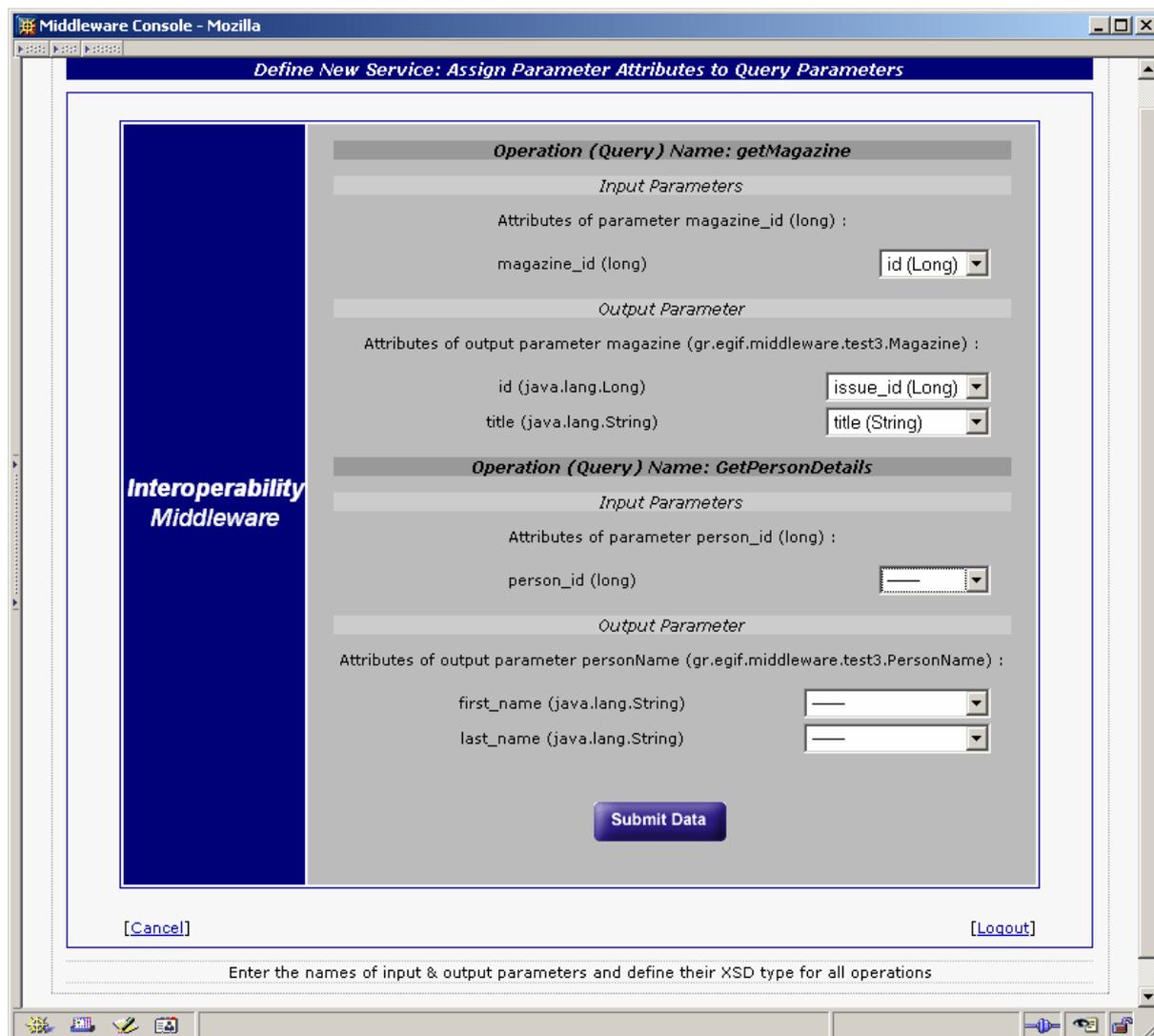
³ Στην περίπτωση που χρησιμοποιείται XSD το ΕΛΔ θεωρεί ότι όλες οι παράμετροι εξόδου ομαδοποιούνται σε ένα αντικείμενο

επιλέχθηκαν στο προηγούμενο βήμα). Η αντιστοίχιση αυτή γίνεται για κάθε SQL query που έχει επιλεγεί.



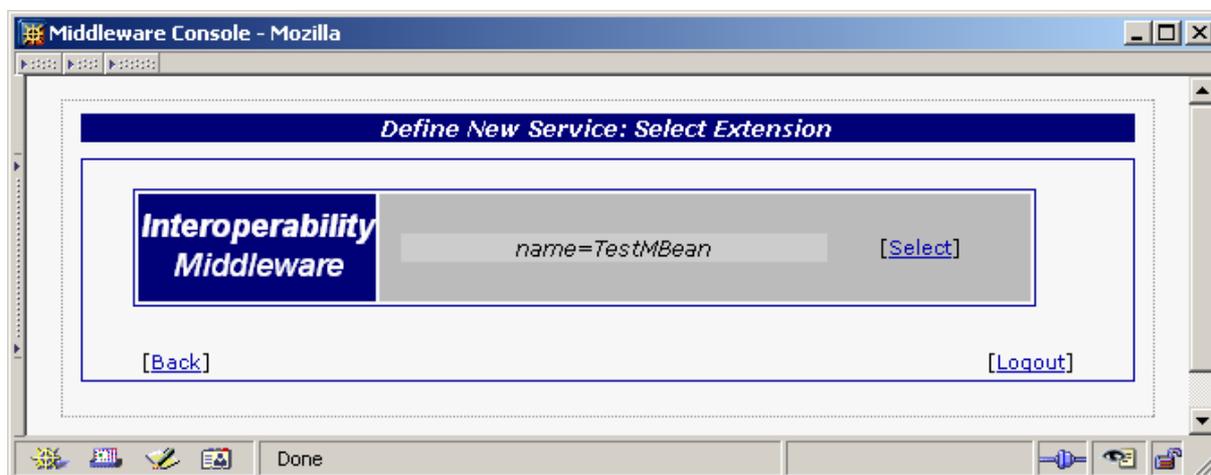
Σχήμα 29: Ανάθεση τύπων δεδομένων XSD στις παραμέτρους των queries

Η ολοκλήρωση της διαδικασίας οδηγεί στην οθόνη που περιγράφει το Σχήμα 30. Εδώ ο χρήστης καλείται να αντιστοιχίσει τις παραμέτρους εισόδου – εξόδου για κάθε SQL query με τις παραμέτρους και τα επιμέρους πεδία τους που δήλωσε στην προηγούμενη οθόνη Σχήμα 29. Αν και αυτή η διαδικασία ολοκληρωθεί ορθά, τότε η υπηρεσία δημιουργείται και ενεργοποιείται από το ΕΛΔ, το οποίο εμφανίζει την οθόνη λεπτομερειών της υπηρεσίας με τον σύνδεσμο στην περιγραφή WSDL.

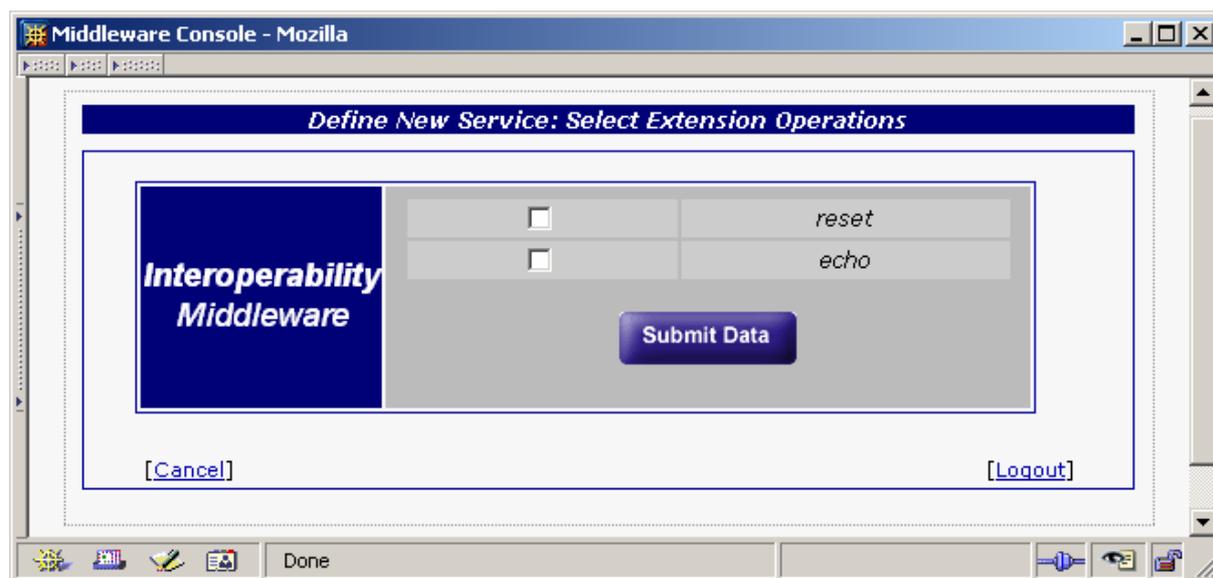


Σχήμα 30: Αντιστοίχιση πεδίων παραμέτρων XSD στις παραμέτρους του SQL query

Τελευταία περίπτωση είναι η υλοποίηση με επέκταση (επιλογή τύπου Extension). Τότε το ΕΛΔ εμφανίζει την οθόνη επιλογής επέκτασης, αντί datasource, όπως φαίνεται στο Σχήμα 31.



Σχήμα 31: Επιλογή επέκτασης για τη νέα υπηρεσία



Σχήμα 32: Επιλογή μεθόδων της επέκτασης για τη νέα υπηρεσία

Στη συνέχεια, ο χρήστης επιλέγει ποια από τα methods της επέκτασης θα χρησιμοποιηθούν από τη νέα υπηρεσία (αντί των SQL queries της προηγούμενης περίπτωσης) και η υπηρεσία δημιουργείται – το ΕΛΔ εμφανίζει την οθόνη λεπτομερειών της υπηρεσίας με τον σύνδεσμο στην αντίστοιχη περιγραφή WSDL.

3.2 Χρήση των υπηρεσιών του ΕΛΔ

Για να χρησιμοποιηθούν οι υπηρεσίες που προσφέρονται από το ΕΛΔ τα απομακρυσμένα πληροφοριακά συστήματα

3.2.1 Γενική περιγραφή χρήσης

Για να έχει ένα απομακρυσμένο πληροφοριακό σύστημα ή εφαρμογή ολοκλήρωσης (integrating application) πρόσβαση στις υπηρεσίες (web services) που προσφέρει το ΕΛΔ θα πρέπει να χρησιμοποιήσει το αρχείο περιγραφής WSDL που παράγεται αυτόματα από το ΕΛΔ όταν δηλωθεί με επιτυχία η υπηρεσία (βλέπε παράγραφο 3.1.6).

Όλα τα σύγχρονα εργαλεία και περιβάλλοντα λογισμικού προσφέρουν τη δυνατότητα με είσοδο το αρχείο WSDL να δημιουργηθεί αυτόματα ο κατάλληλος κώδικας πελάτη (client source code) που θα ενσωματωθεί στην εφαρμογή για την κλήση των web services που περιγράφει η WSDL. Τέτοια εργαλεία και δυνατότητες προσφέρει το περιβάλλον ανάπτυξης Microsoft .NET, τα περισσότερα J2EE IDEs (π.χ. από την Oracle, IBM), το Borland Delphi, αλλά και open source εργαλεία⁴.

Γενικά, ο «χρήστης» των υπηρεσιών, αυτός δηλαδή που θα καλέσει και θα χρησιμοποιήσει τα web services που προσφέρει το ΕΛΔ εκ μέρους του εκάστοτε πληροφοριακού συστήματος, δεν χρειάζεται τίποτα περισσότερο από το αρχείο

⁴ Τα εργαλεία που αναφέρονται είναι ενδεικτικά και δεν υποδηλώνουν προτίμηση – καθώς οι web services αποκτούν όλο και μεγαλύτερη εξάπλωση και αποδοχή από την αγορά ο αριθμός των εργαλείων, IDEs και CASE tools αυξάνει με ραγδαίους ρυθμούς. Όλοι οι μεγάλοι κατασκευαστές εργαλείων ανάπτυξης λογισμικού παρέχουν τέτοιες λύσεις ή έχουν ανακοινώσει ότι σύντομα θα παρέχουν.

WSDL που περιγράφει την συγκεκριμένη υπηρεσία, και το username / password που ελέγχει την πρόσβαση σε αυτή, αν έχει περιοριστεί η πρόσβαση.

3.2.2 Παράδειγμα εφαρμογής πελάτη σε Java

Ακολουθεί περιγραφή παραδείγματος δημιουργίας client (κυρίως για έλεγχο ορθής λειτουργίας) σε Java. Το παράδειγμα προϋποθέτει τη χρήση του open-source Jakarta Axis ή κάποιο από τα γνωστά web service toolkits (π.χ. από τους SUN, IBM, Oracle) ανάλογα με το περιβάλλον και τα εργαλεία ανάπτυξης που θα χρησιμοποιήσει κανείς.

Αφού κατεβάσουμε το αρχείο περιγραφής της υπηρεσίας σε μορφή WSDL (βλέπε παράγραφο 3.1.6) ακολουθούμε τα εξής βήματα:

1. Τρέχουμε την εφαρμογή wsdl2java για να παραχθεί ο κώδικας των κλάσεων που μεταφράζουν τις κλήσεις σε java στο κατάλληλο σχήμα σε XML.
2. Γράφουμε τον κώδικα της εφαρμογής μας χρησιμοποιώντας τις κλάσεις που παραχθήκαν στο πορηγούμενο βήμα.
3. Καλούμε την υπηρεσία.

Για παράδειγμα έστω ότι η περιγραφή σε wsdl ήταν το παρακάτω αρχείο:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://example.targetNamespace"
xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:apacheSOAP="http://xml.apache.org/xml-soap"
xmlns:impl="http://example.targetNamespace" xmlns:intf="http://example.targetNamespace"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns1="http://net.jboss.org/jmx" xmlns:tns2="http://www.egif.gr/address"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <schema targetNamespace="http://www.egif.gr/address"
xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <complexType name="StreetType">
        <sequence>
          <element name="name" nillable="true" type="xsd:string"/>
          <element name="number" nillable="true" type="xsd:int"/>
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>

  <wsdl:message name="memberRequest">
    <wsdl:part name="memberId" type="xsd:long"/>
  </wsdl:message>
  <wsdl:message name="memberResponse">
    <wsdl:part name="street" type="tns2:StreetType"/>
  </wsdl:message>

  <wsdl:portType name="ExamplePortType">
    <wsdl:operation name="member" parameterOrder="memberId">
      <wsdl:input message="impl:memberRequest" name="memberRequest"/>
      <wsdl:output message="impl:memberResponse" name="memberResponse"/>
    </wsdl:operation>
  </wsdl:portType>

  <wsdl:binding name="ExampleSoapBinding" type="impl:ExamplePortType">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="member">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="memberRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://example.targetNamespace" use="encoded"/>
      </wsdl:input>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

```
</wsdl:input>
  <wsdl:output name="memberResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://example.targetNamespace" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>

<wsdl:service name="ExampleService">
  <wsdl:port binding="impl:ExampleSoapBinding" name="Example">
    <wsdlsoap:address location="http://host/jboss-net/services/Example"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Πίνακας 1: Αρχείο Service.wsdl (παράδειγμα)

Εκτελώντας την εντολή:

```
% java org.apache.axis.wsdl.WSDL2Java Service.wsdl
```

Θα παραχθούν τα παρακάτω αρχεία:

```
ExamplePortType.java
ExampleService.java
ExampleServiceLocator.java
ExampleSoapBinding.java
StreetType.java
```

Τα αρχεία αυτά ορίζουν java classes (binding classes) που απαιτούνται από το client entity για να καλέσει και να χρησιμοποιήσει τη web service. Για παράδειγμα ένα απλό client application που καλεί την υπηρεσία που περιγράφει το παραπάνω αρχείο WSDL είναι της μορφής:

```
public class Client
{
    public static void main(String [] args) throws Exception {

        ExampleService service = new ExampleServiceLocator();
        ExamplePortType port = service.getExample();

        long parameter = 1;
        // κλήση της υπηρεσίας
        StreetType result = port.member(parameter);

        // κώδικας που χρησιμοποιεί το αποτέλεσμα (result)
        ...
    }
}
```

Πίνακας 2: Παράδειγμα απλού client application

Παράρτημα Α: Δεσμευμένες λέξεις στη Java

Οι παρακάτω λέξεις δεν μπορούν να χρησιμοποιηθούν ως ονόματα υπηρεσιών, queries ή παραμέτρων των queries στον ορισμό υπηρεσιών και datasources του ΕΛΔ.

abstract	double	int	strictfp **
boolean	else	interface	super
break	extends	long	switch
byte	final	native	synchronized
case	finally	new	this
catch	float	package	throw
char	for	private	throws
class	goto *	protected	transient
const *	if	public	try
continue	implements	return	void
default	import	short	volatile
do	instanceof	static	while