

Στην ενότητα αυτή θα περιγράψουμε δύο βασικές μεθοδολογίες που χρησιμοποιούνται για την ανάπτυξη υπηρεσιών Ιστού. Η μία μεθοδολογία βασίζεται στην χρήση του εργαλείου Apache Axis Java Web Service (JWS) και η άλλη βασίζεται στην χρήση του Apache Axis Web Service Deployment Descriptor (WSDD).

Χρήση Apache Axis Java Web Service (JWS)

Για την καλύτερη κατανόηση της διαδικασίας ανάπτυξης υπηρεσίας Ιστού θα αναπτύξουμε ένα απλό παράδειγμα. Έτσι, σε αυτό το παράδειγμα θα αναπτύξουμε μια απλή μαθηματική υπηρεσία που θα εκτελεί μια αριθμητική πράξη όπως π.χ. την πρόσθεση δύο ακέραιων αριθμών για ένα πελάτη. Η μαθηματική αυτή υπηρεσία δεν είναι καταστατική που σημαίνει ότι η υπηρεσία δεν κρατά το αποτέλεσμα της προηγούμενης αριθμητικής πράξης. Παρακάτω περιγράφονται τα βήματα για την ανάπτυξη, μεταγλώττιση και εκτέλεση προγραμμάτων υπηρεσιών Ιστού:

1. **Εκκίνηση κελύφους Linux:** Πρώτα πρέπει να έχουμε ανοιχτό ένα παράθυρο κελύφους Linux ώστε να εισάγουμε εντολές.
2. **Δημιουργία ενός καταλόγου:** Δημιουργούμε ένα κατάλογο ώστε να αποθηκεύσουμε τα παραδείγματα των προγραμμάτων μας. Έτσι, δημιουργούμε ένα κατάλογο με όνομα `webservices_lab` και μπαίνουμε σε αυτό τον κατάλογο. Οι εντολές είναι οι εξής:

```
$ mkdir webservices_lab  
$ cd webservices_lab
```

3. **Ανάπτυξη της υπηρεσίας σε Java:** Σε αυτό το βήμα γράφουμε το πρόγραμμα Java που θα υλοποιεί την υπηρεσία πρόσθεσης δύο αριθμών. Έτσι, εισάγουμε το παρακάτω πρόγραμμα Java σε ένα απλό επεξεργαστή κειμένου (όπως το `kedit`):

```
public class MyMath {  
    public int add(int x, int y) {  
        return x + y;  
    }  
}
```

Στην συνέχεια αποθηκεύουμε το παραπάνω πρόγραμμα με όνομα `MyMath` και επέκταση `jws`, δηλαδή ως `MyMath.jws` κάτω από το κατάλογο `webservices_lab`. Εναλλακτικά, μπορούμε να μεταφορτώσουμε το πρόγραμμα [MyMath.jws](#) στο κατάλογο `webservices_lab`. Το εργαλείο Axis αναμένει το αρχείο `jws` να βρίσκεται σε μια από τις διάφορες πρότυπες θέσεις. Για να ακολουθήσουμε την απαίτηση αυτή πρέπει να δημιουργήσουμε ένα κατάλογο κάτω από το κατάλογο `axis` της Jakarta Tomcat. Δηλαδή, πρέπει να δημιουργήσουμε ένα κατάλογο με όποιο όνομα θελουμε (π.χ. το `username` μας) κάτω από το κατάλογο `$CATALINA_HOME/webapps/axis/yourusername/`. Η `CATALINA_HOME` είναι μια μεταβλητή περιβάλλοντος που προσδιορίζει την διαδρομή του αρχικού καταλόγου της μηχανής Jakarta Tomcat Java servlet. Στην συνέχεια αντιγράφουμε το αρχείο `jws` κάτω από το κατάλογο που δημιουργήσαμε προηγουμένως έτσι ώστε το Apache Axis να είναι σε θέση να βρίσκει το αρχείο αυτό. Συγκεκριμένα, στο παράθυρο κελύφους εισάγουμε την παρακάτω εντολή:

```
$ cp MyMath.jws $CATALINA_HOME/webapps/axis/yourusername
```

4. **Παραγωγή των απαραίτητων αρχείων Java για την υπηρεσία Ιστού:** Σε αυτό το βήμα γίνεται ο ορισμός της διεπαφής για την μαθηματική υπηρεσία. Όπως, είναι γνωστό η διεπαφή ορίζεται χρησιμοποιώντας την γλώσσα WSDL που προσδιορίζει ποιές πράξεις μέσω της υπηρεσίας Ιστού θα είναι διαθέσιμες στα προγράμματα πελάτες. Σε αυτό το παράδειγμα, πρέπει να έχουμε την διεπαφή σε Java από το έγγραφο WSDL. Για το λόγο αυτό, το Apache Axis διαθέτει το εργαλείο WSDL2Java που παράγει τα απαραίτητα αρχεία Java για την υλοποίηση της υπηρεσίας από το αρχείο MyMath.jws που δημιουργήσαμε στο προηγούμενο βήμα. Έτσι, χρησιμοποιούμε την εντολή:

```
$ java -classpath $AXISCLASSPATH org.apache.axis.wsdl.WSDL2Java \
http://localhost:8080/axis/yourusername/MyMath.jws?wsdl
```

Η δεύτερη παράμετρος της παραπάνω εντολής είναι μια διεύθυνση URL που προσδιορίζει το διακομιστή Ιστού (στο παράδειγμα μας είναι ο localhost) που βρίσκεται το αρχείο MyMath.jws.

Το αποτέλεσμα της εκτέλεσης του προγράμματος WSDL2Java είναι η δημιουργία ενός καταλόγου με όνομα localhost μέσα στον τρέχοντα κατάλογο (π.χ. /home/yourusername/webaservices_lab/). Μέσα στον κατάλογο localhost υπάρχει ένας υποκατάλογος με όνομα axis, ο οποίος έχει έναν άλλο υποκατάλογο με όνομα το yourusername που πάλι έχει ακόμα έναν υποκατάλογο με όνομα MyMath_jws. Παρατηρούμε λοιπόν ότι δημιουργήθηκαν μια σειρά καταλόγων σύμφωνα με τα μέρη της διεύθυνσης, URL <http://localhost:8080/axis/yourusername/MyMath.jws?wsdl>. Μέσα στον κατάλογο MyMath_jws υπάρχουν τέσσερα αρχεία πηγαίου κώδικα Java τα οποία είναι τα εξής:

- MyMath.java: κώδικας διεπαφής Java για την κλάση MyMath.
- MyMathService.java: κώδικας διεπαφής Java που περιλαμβάνει την μέθοδο getMyMath.
- MyMathServiceLocator.java: κώδικας για την κλάση Java MyMathServiceLocator.
- MyMathSoapBindingStub.java: κώδικας για την κλάση Java MyMathSoapBindingStub.

Τα παραπάνω αρχεία είναι απαραίτητα ώστε η MyMath να μπορεί να χρησιμοποιηθεί ως υπηρεσία Ιστού.

5. **Μεταγλώττιση των παραγόμενων αρχείων Java:** Στο βήμα αυτό μεταγλωττίζουμε τα τέσσερα αρχεία Java που δημιουργήθηκαν από το προηγούμενο βήμα με την παρακάτω εντολή:

```
$ javac -classpath $AXISCLASSPATH \
localhost/axis/yourusername/MyMath_jws/*.java
```

6. **Ανάπτυξη προγράμματος πελάτη μιας υπηρεσίας Ιστού:** Έτσι, στο βήμα αυτό γράφουμε ένα πρόγραμμα πελάτη σε Java που θα υλοποιεί την κλήση μιας υπηρεσίας Ιστού. Εισάγουμε το παρακάτω πρόγραμμα σε έναν απλό επεξεργαστή κειμένου (όπως το kedit):

```
import localhost.axis.yourusername.MyMath_jws.MyMathServiceLocator;
```

```

import localhost.axis.yourusername.MyMath_jws.MyMathService;
import localhost.axis.yourusername.MyMath_jws.MyMath;
public class MyMathClient {
    public static void main(String args[]) throws Exception {
        MyMathService service = new MyMathServiceLocator();
        MyMath myMath = service.getMyMath();
        int x = (new Integer(args[0])).intValue();
        int y = (new Integer(args[1])).intValue();
        System.out.println("The add of " + args[0] + " and " + args[1]
+ " is " + myMath.add(x,y));
    }
}

```

Στο παραπάνω πρόγραμμα εισάγουμε τρεις δηλώσεις import όπου καλέσαμε το πρόγραμμα WSDL2Java χρησιμοποιώντας το υπολογιστή localhost. Τέλος, το παραπάνω πρόγραμμα το αποθηκεύουμε ως MyMathClient.java κάτω από το κατάλογο webservices_lab. Εναλλακτικά μπορούμε να μεταφορτώσουμε το πρόγραμμα [MyMathClient.java](#) στο κατάλογο webservices_lab.

- Μεταγλώττιση του προγράμματος πελάτη:** Στην συνέχεια, μεταγλωττίζουμε το πρόγραμμα πελάτη με την παρακάτω εντολή:

```
$ javac -classpath \$AXISCLASSPATH:. MyMathClient.java
```

- Εκτέλεση προγράμματος υπηρεσίας Ιστού:** Εκτελούμε το πρόγραμμα πελάτη με την εντολή:

```
$ java -classpath \$AXISCLASSPATH:. MyMathClient 6 5
```

Μετά την εκτέλεση του προγράμματος MyMathClient, παίρνουμε την παρακάτω έξοδο στην οθόνη:

```
The add of 6 and 5 is 11
```

Χρήση Apache Axis Web Service Deployment Descriptor (WSDD)

Η ανάπτυξη υπηρεσίας Ιστού με την προσέγγιση JWS είναι απλή διότι το Apache Axis αυτόματα εντοπίζει το αρχείο jws, μεταγλωττίζει την κλάση και μετατρέπει σωστά τις κλήσεις SOAP σε κλήσεις Java της υπηρεσίας. Όμως, η προσέγγιση αυτή έχει περιορισμούς όπως μπορεί να αναπτύξει μόνο απλές υπηρεσίες Ιστού και δεν υποστηρίζει σύνθετους τύπους δεδομένων. Για το λόγο αυτό υπάρχει η προσέγγιση WSDD που μπορεί να αναπτύξει υπηρεσίες Ιστού που υποστηρίζουν σύνθετους τύπους δεδομένων. Τα βήματα για την ανάπτυξη της υπηρεσίας Ιστού με την προσέγγιση WSDD είναι παρόμοια με εκείνη την JWS. Για την επίδειξη της προσέγγισης WSDD θα χρησιμοποιήσουμε το παράδειγμα της μαθηματικής υπηρεσίας που είχαμε αναπτύξει για την προηγούμενη προσέγγιση JWS. Επομένως, παρακάτω περιγράφονται τα βήματα για την ανάπτυξη, μεταγλώττιση και εκτέλεση προγραμμάτων υπηρεσιών Ιστού σε Java:

- Εκκίνηση κελύφους Linux:** Πρώτα πρέπει να έχουμε ανοιχτό ένα παράθυρο κελύφους Linux ώστε να εισάγουμε εντολές.
- Δημιουργία ενός καταλόγου:** Δημιουργούμε ένα κατάλογο ώστε να αποθηκεύσουμε τα παραδείγματα των προγραμμάτων μας. Έτσι, δημιουργούμε ένα κατάλογο με όνομα webservices_lab και μπαίνουμε σε

αυτό του κατάλογο. Οι εντολές είναι οι εξής:

```
$ mkdir webservices_lab  
$ cd webservices_lab
```

3. **Ανάπτυξη της υπηρεσίας σε Java.** Σε αυτό το βήμα γράφουμε το πρόγραμμα Java που θα υλοποιεί την υπηρεσία πρόσθεσης δύο αριθμών. Εδώ, δεν παρουσιάζουμε το πρόγραμμα Java αφού είναι ακριβώς ίδιο με το πρόγραμμα στο βήμα 3 της προηγούμενης προσέγγισης. Όμως, αποθηκεύουμε το πρόγραμμα με όνομα MyMath και επέκταση java, δηλαδή ως MyMath.java κάτω από το κατάλογο webservices_lab. Εναλλακτικά, μπορούμε να μεταφορτώσουμε το πρόγραμμα [MyMath.java](#) στο κατάλογο webservices_lab. Στην συνέχεια, μεταγλωττίζουμε το αρχείο αυτό ώστε να παράγουμε τα αρχεία class. Σύμφωνα με το εργαλείο Axis πρέπει τα αρχεία class να τοποθετηθούν στις κατάλληλες θέσεις. Για το λόγο αυτό πρέπει να δημιουργήσουμε ένα κατάλογο με όποιο όνομα θέλουμε (π.χ. το username μας) κάτω από το κατάλογο \$CATALINA_HOME/webapps/axis/WEB-INF/classes/ της Jakarta Tomcat. Τέλος, αντιγράφουμε το αρχείο class κάτω από το κατάλογο που δημιουργήσαμε προηγουμένως έτσι ώστε το Apache Axis να είναι σε θέση να βρίσκει το αρχείο αυτό. Συγκεκριμένα, στο παράθυρο κελύφους εισάγουμε την παρακάτω εντολή:

```
$ cp MyMath.java $CATALINA_HOME/webapps/axis/WEB-INF/classes/yourusername
```

4. **Δημιουργία αρχείου WSDD:** Για να αναπτύξουμε την υπηρεσία πρέπει να διαμορφώσουμε το αρχείο WSDD για την περιγραφή της υπηρεσίας. Έτσι, παρακάτω φαίνονται τα περιεχόμενα του αρχείου deploy.wsdd για την μαθηματική υπηρεσία που θέλουμε να αναπτύξουμε:

```
<deployment xmlns="http://xml.apache.org/axis/wsdd/"  
             xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">  
  <service name="MyMath" provider="java:RPC">  
    <parameter name="className" value="MyMath"/>  
    <parameter name="allowedMethods" value="*" />  
  </service>  
</deployment>
```

Εναλλακτικά, μπορούμε να μεταφορτώσουμε το αρχείο [deploy.wsdd](#) στο κατάλογο webservices_lab. Η δομή του αρχείου WSDD περιέχει ένα στοιχείο XML το deployment. Το στοιχείο αυτό ορίζει το όνομα περιοχής Java. Μέσα στο στοιχείο αυτό υπάρχει το στοιχείο service που ορίζει την υπηρεσία. Το στοιχείο service μπορεί να είναι ένα ή όλα από τα παρακάτω:

- ένα request flow
- ένα pivot Handler το οποίο είναι το service provider
- ένα response flow

Στην περίπτωση μας, το provider είναι η "Java:RPC", δηλαδή κλήση απομακρυσμένων διαδικασιών. Η ετικέτα parameter προσδιορίζει ποια κλάση (π.χ. MyMath) πρέπει να φορτώσει ο παροχέας και ποιοι μέθοδοι της κλάσης πρέπει να καλούνται. Στο παράδειγμα μας, προσδιορίζουμε ότι θα φορτωθεί η κλάση MyMath και ότι θα καλείται οποιαδήποτε μέθοδο της κλάσης. Πρέπει να σημειώσουμε ότι τα αρχεία WSDD είναι επαναχρησιμοποιήσιμα και ότι απαιτούν ελάχιστες τροποποιήσεις στην

περίπτωση που θέλουμε να χρησιμοποιήσουμε άλλες υπηρεσίες όπως π.χ. το όνομα της υπηρεσίας και το όνομα της κλάσης.

5. **Εγκατάσταση του αρχείου WSDD στο Apache Axis:** Στο βήμα αυτό πρέπει να περάσουμε το αρχείο wsdd που δημιουργήσαμε από το προηγούμενο βήμα στο εργαλείο AdminClient του Apache Axis. Το οποίο θα μεταγλωττίζει την υπηρεσία Ιστού με βάση τις παραμέτρους που περιγράφει το αρχείο wsdd και θα εγκατασταθεί η υπηρεσία στην κατάλληλη θέση με την παρακάτω εντολή:

```
$ java -cp $AXISCLASSPATH org.apache.axis.client.AdminClient /  
-lhttp://localhost:8080/axis/services/AdminService deploy.wsdd
```

Για να επιβεβαιώσουμε ότι η υπηρεσία Ιστού εγκαταστάθηκε σωστά, μπορούμε να ελέγξουμε την λίστα των εγκατεστημένων υπηρεσιών του περιβάλλοντος Axis. Για να γίνει αυτό χρησιμοποιούμε ένα περιηγητή Ιστού και κάνουμε κλικ στην επιλογή "View the list of deployed Web services" πάνω στην αρχική σελίδα του Axis. Στην συνέχεια, εμφανίζεται η σελίδα των υπηρεσιών όπου θα πρέπει να φαίνεται και η υπηρεσία μας MyMath αν εγκαταστάθηκε επιτυχώς.

6. **Παραγωγή των απαραίτητων αρχείων Java για την υπηρεσία Ιστού:** Σε αυτό το βήμα είναι παρόμοιο με το βήμα 4 της προσέγγισης JWS. Έτσι, χρησιμοποιούμε την παρακάτω εντολή:

```
$ java -classpath $AXISCLASSPATH org.apache.axis.wsdl.WSDL2Java \  
http://localhost:8080/axis/services/MyMath?wsdl
```

Το αποτέλεσμα της εκτέλεσης του προγράμματος WSDL2Java είναι ότι δημιουργεί τέσσερα αρχεία πηγαίου κώδικα Java στον κατάλογο localhost/axis/services/MyMath τα οποία είναι παρόμοια με εκείνα του βήματος 4 της προσέγγισης JWS.

7. **Μεταγλώττιση των παραγόμενων αρχείων Java:** Στο βήμα αυτό μεταγλωττίζουμε τα τέσσερα αρχεία Java που δημιουργήθηκαν από το προηγούμενο βήμα με την παρακάτω εντολή:

```
$ javac -classpath \ $AXISCLASSPATH localhost/axis/services/MyMath/*.java
```

8. **Ανάπτυξη προγράμματος πελάτη μιας υπηρεσίας Ιστού:** Έτσι, στο βήμα αυτό γράφουμε ένα πρόγραμμα πελάτη σε Java που θα υλοποιεί την κλήση μιας υπηρεσίας Ιστού. Εισάγουμε το παρακάτω πρόγραμμα σε έναν απλό επεξεργαστή κειμένου (όπως το kedit):

```
import localhost.*;  
public class MyMathClient {  
    public static void main(String args[]) throws Exception {  
        MyMathService service = new MyMathServiceLocator();  
        MyMath myMath = service.getMyMath();  
        int x = (new Integer(args[0])).intValue();  
        int y = (new Integer(args[1])).intValue();  
        System.out.println("The add of " + args[0] + " and " + args[1]  
        + " is " + myMath.add(x,y));  
    }  
}
```

Τέλος, το παραπάνω πρόγραμμα το αποθηκεύουμε ως MyMathClient.java

κάτω από το κατάλογο `webservices_lab`. Εναλλακτικά μπορούμε να μεταφορτώσουμε το πρόγραμμα [MyMathClient.java](#) στο κατάλογο `webservices_lab`.

9. **Μεταγλώττιση του προγράμματος πελάτη:** Στην συνέχεια, μεταγλωττίζουμε το πρόγραμμα πελάτη με την παρακάτω εντολή:

```
$ javac -classpath $AXISCLASSPATH:. MyMathClient.java
```

10. **Εκτέλεση προγράμματος υπηρεσίας Ιστού.** Εκτελούμε το πρόγραμμα πελάτη με την εντολή:

```
$ java -classpath $AXISCLASSPATH:. MyMathClient 6 5
```

Μετά την εκτέλεση του προγράμματος `MyMathClient`, παίρνουμε την παρακάτω έξοδο στην οθόνη:

```
The add of 6 and 5 is 11
```