

Υπηρεσίες και Συναλλαγές Ιστού – Χ. Γεωργιάδης

Lab2, VS_2022

Ασκήσεις αξιοποίησης Υπηρεσιών Ιστού σε .NET (Α' μέρος)

1. Υπηρεσία αριθμομηχανής

Στη παρακάτω διεύθυνση:

<http://www.dneonline.com/calculator.asmx>

The screenshot shows a web page titled "Calculator". Below the title, it says "The following operations are supported. For a formal definition, please review the [Service Description](#)". A list of four operations is provided: [Add](#), [Divide](#), [Multiply](#), and [Subtract](#). At the bottom, there is a note: "This web service is using <http://tempuri.org/> as its default namespace." and a recommendation: "Recommendation: Change the default namespace before the XML Web service is made public."

α) “Δοκιμάζουμε” τις 4 μεθόδους στον browser. Βλέπουμε τι χρειάζεται η κάθε μια ως είσοδο (πλήθος και τύπος/μορφή παραμέτρων input) και τι επιστρέφει ως έξοδο (πλήθος και τύπος/μορφή παραμέτρων output)

β) Δημιουργούμε ένα νέο project: ζητάμε File -> New Project και ακολουθούμε τα βήματα της προηγούμενης άσκησης (Lab1, σελ.6, «**Χρήση-κλήση της MyWSservice**»), βάζοντας βέβαια ένα διαφορετικό όνομα στο project (π.χ. CalculatorLab2) και προσέχοντας στην επιλογή των κατάλληλων στοιχείων διάδρασης (controls):

στην web φόρμα, προσθέστε κάτω από την επικεφαλίδα “**Αριθμομηχανή**” τα απαιτούμενα πεδία εισαγωγής/αποτελέσματος καθώς και κουμπιά (4 για τις πράξεις) που θα προκαλούν κλήσεις στις λειτουργίες/μεθόδους της υπηρεσίας και εμφάνιση του αποτελέσματος. Προσθέστε και ένα κουμπί για το καθάρισμα των στοιχείων εισόδου.

Αριθμομηχανή

Αριθμός Α:

+ * - / Καθαρισμός

Αριθμός Β:

ΑΠΟΤΕΛΕΣΜΑ:

γ) Σύνδεση της υπηρεσίας αριθμομηχανής: ακολουθούμε τα βήματα της προηγούμενης άσκησης (Lab1, σελ.7, «Εύρεση της MyWebService»), βάζοντας βέβαια το URL της υπηρεσίας αριθμομηχανής:

<http://www.dneonline.com/calculator.asmx>

δ) Ενδεικτικός κώδικας (για την πρόσθεση-μέθοδος Add και τη διαίρεση-μέθοδος Divide):

```
// δήλωση της μεταβλητής της υπηρεσίας
com.dneonline.www.Calculator myWS = new com.dneonline.www.Calculator();
```

```
protected void btnAdd_Click(object sender, EventArgs e) {
    int a, b, apot;
    a = Convert.ToInt32(textBox1.Text);
    b = Convert.ToInt32(textBox2.Text);
    apot = myWS.Add(a, b);
    Label1.Text = Convert.ToString(apot);
}
```

```
protected void btnDivide_Click(object sender, EventArgs e) {
    int a, b, apot;
    a = Convert.ToInt32(textBox1.Text);
    b = Convert.ToInt32(textBox2.Text);

    if (b != 0)
    {
        apot = myWS.Divide(a, b);
        Label1.Text = Convert.ToString(apot);
    }
    else if (a!=0)
        Label1.Text = "Αδύνατη πράξη";
    else
        Label1.Text = "Αοριστία";
}
```

2. Υπηρεσία Μετατροπής συναλλάγματος:

<https://www.lb.lt/webservices/ExchangeRates/ExchangeRates.asmx>

https://www.lb.lt/webservices/ExchangeRates/ExchangeRates_En.htm

ExchangeRates

WebService provides official (established by Bank of Lithuania) exchange rates of the Litas against Foreign Currencies.

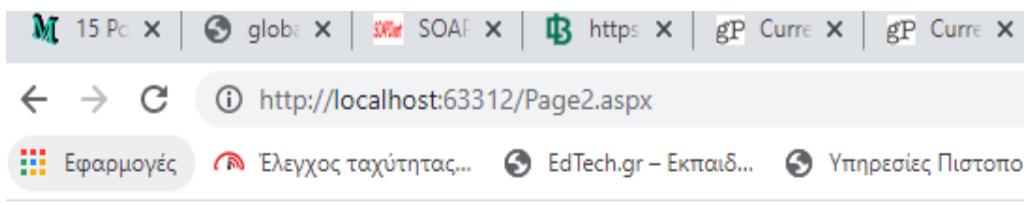
The following operations are supported. For a formal definition, please review the [Service Description](#).

- [**getCurrentExchangeRate**](#)
Returns the current exchange rate (expressed in Litas per 1 currency unit) for the specified currency.
- [**getExchangeAvgRate**](#)
Returns a average exchange rate for the specified currency, between specified dates.
- [**getExchangeRate**](#)
Returns an exchange rate (expressed in Litas per 1 currency unit) for the specified currency and date.
- [**getExchangeRatesByCurrency**](#)
Returns a list containing exchange rates for the specified currency that are between specified dates.
- [**getExchangeRatesByCurrency_XmlString**](#)
Returns a XML formatted string containing exchange rates for the specified currency that are between specified dates.
- [**getExchangeRatesByDate**](#)
Returns a list containing exchange rates for the specified date.
- [**getExchangeRatesByDate_XmlString**](#)
Returns a XML formatted string containing exchange rates for the specified date.
- [**getExchangeRatesXmlSchema**](#)
Returns a Xml Schema of provided Exchange Rates.
- [**getListCurrencies**](#)
Returns a list of currencies.

α) Θα χρησιμοποιήσουμε 2 μεθόδους από αυτήν την υπηρεσία: (i) την μέθοδο `getListOfCurrencies()` για να πάρουμε την λίστα με όλα τα διαθέσιμα νομίσματα, ώστε να μπορεί ο χρήστης να επιλέγει από ποιο νόμισμα προς ποιο νόμισμα θα ζητήσει την μετατροπή. (ii) Την μέθοδο `getExchangeRatesByDate()` για να κάνουμε την μετατροπή. «Δοκιμάζουμε» τις 2 αυτές μεθόδους στον browser. Βλέπουμε τι χρειάζεται η κάθε μια ως είσοδο (πλήθος και τύπος/μορφή παραμέτρων input) και τι επιστρέφει ως έξοδο (πλήθος και τύπος/μορφή παραμέτρων output). Παρατηρούμε ότι η πρώτη μέθοδος δεν χρειάζεται καμία παράμετρο εισόδου, ενώ η δεύτερη χρειάζεται ως είσοδο μια ημερομηνία. Και οι 2 μέθοδοι επιστρέφουν αρχείο XML, με πολλούς περιεχόμενους κόμβους, το οποίο και πρέπει να υποδεχτούμε κατάλληλα.

β) Δημιουργούμε ένα νέο project: ζητάμε File -> New Project και ακολουθούμε τα βήματα της προηγούμενης άσκησης (Lab2, σελ.1), βάζοντας βέβαια ένα διαφορετικό όνομα στο project (π.χ. ConverterLab2) και προσέχοντας στην επιλογή των κατάλληλων στοιχείων διάδρασης (controls):

στην web φόρμα, προσθέστε κάτω από την επικεφαλίδα «**Μετατροπή Συναλλάγματος**» τα απαιτούμενα πεδία εισαγωγής/αποτελέσματος (χρησιμοποιείστε δύο πτυσσόμενες λίστες για την δυνατότητα επιλογής των νομισμάτων «από» και «προς» την μετατροπή) καθώς και κουμπί που θα προκαλεί την κλήση στη λειτουργία/μέθοδο της μετατροπής και την εμφάνιση του αποτελέσματος. Σημειώστε ότι για τη κλήση της πρώτης μεθόδου με την λίστα των νομισμάτων θα αξιοποιήσουμε το συμβάν **Page_Load**, ώστε οι πτυσσόμενες λίστες με την φόρτωση της σελίδας/φόρμας να δέχονται τις τιμές των διαθέσιμων νομισμάτων.



Μετατροπή Συναλλάγματος

Δώστε νόμισμα προέλευσης: EUR Euro

Δώστε νόμισμα προορισμού:

Αποτέλεσμα: 15,67

γ) Σύνδεση της αντίστοιχης υπηρεσίας (όπως και στη προηγούμενη άσκηση Lab2, σελ.3)

δ) Ενδεικτικός απαιτούμενος κώδικας:

```
// δήλωση της μεταβλητής της υπηρεσίας
lt.lb.www.ExchangeRates myWS2 = new lt.lb.www.ExchangeRates();

protected void Page_Load(object sender, EventArgs e)
{
    if (IsPostBack == false) // αν είναι η πρώτη φορά που φορτώνει η σελίδα
    {
        // λαμβάνω σε XML κόμβο-ρίζα <Currencies> τα διαθέσιμα νομίσματα,
        // πχ. «EUR», «GBP» κλπ.
        XmlNode node = myWS2.getListOfCurrencies();

        // οι XML κόμβοι μέσα στη ρίζα είναι της μορφής <item>
        XmlNodeList nodes = node.SelectNodes("//item");

        // Διαπερνάμε τους xml nodes και βάζουμε όποια πεδία τους θέλουμε
        // στις πινακοποιημένες λίστες
        foreach (XmlNode nd in nodes)
        {
            lstCurFrom.Items.Add(nd["currency"].InnerText + " " +
                nd["description"].InnerText);
            lstCurTo.Items.Add(nd["currency"].InnerText + " " +
                nd["description"].InnerText);
        }
        lstCurFrom.SelectedIndex = 26;
        //αρχικοποιούμε στο Ευρώ την πρώτη λίστα
        lstCurTo.SelectedIndex = 27;
        //αρχικοποιούμε στην Αγγλική Λίρα την δεύτερη λίστα
    }
}
```

```

protected void Button1_Click(object sender, EventArgs e)
{
    decimal f, rateFrom=0, rateTo=0;
    string n1, n2;

    n1 = Convert.ToString(lstCurFrom.SelectedItem).Substring(0,3);
    n2 = Convert.ToString(lstCurTo.SelectedItem).Substring(0,3);

    //Δυστυχώς μόνο για ιστορικά στοιχεία λειτουργεί αυτή η free Web
    //Υπηρεσία
    // Λαμβάνω σε XML κόμβο-ρίζα <ExchangeRates> την αντιστοιχία
    XmlNode node = myWS2.getExchangeRatesByDate("2014-12-31");

    // οι XML κόμβοι μέσα στη ρίζα είναι και εδώ της μορφής <item>
    XmlNodeList nodes = node.SelectNodes("//item");

    //Διαπέραση των xml nodes
    foreach (XmlNode nde in nodes)
    {
        if (nde["currency"].InnerText == n1)
        // εντοπίζουμε το νόμισμα της πρώτης λίστας (προέλευση)

        // η ισοτιμία δίνεται στα διάφορα νομίσματα σε διαφορετικές
        // ποσότητες...
        rateFrom = Convert.ToDecimal(nde["rate"].InnerText) /
            Convert.ToInt32(nde["quantity"].InnerText);

        if (nde["currency"].InnerText == n2)
        // εντοπίζουμε το νόμισμα της δεύτερης λίστας (προορισμός)

        // και εδώ χρειάζεται αναγωγή στη μονάδα
        rateTo = Convert.ToDecimal(nde["rate"].InnerText) /
            Convert.ToInt32(nde["quantity"].InnerText);
    }

    // ακρίβεια 2 δεκαδικών ψηφίων
    f = Math.Round(Convert.ToDecimal(textBox1.Text) * rateFrom/rateTo, 2);
    Label1.Text = Convert.ToString(f);
}

```

ΠΡΟΣΟΧΗ:

Στις πτυσσόμενες λίστες προσέξτε η ιδιότητα AutoPostBack να είναι True, π.χ.

```

<asp:DropDownList ID="lstCurFrom" runat="server" AutoPostBack="True">
</asp:DropDownList>

```