

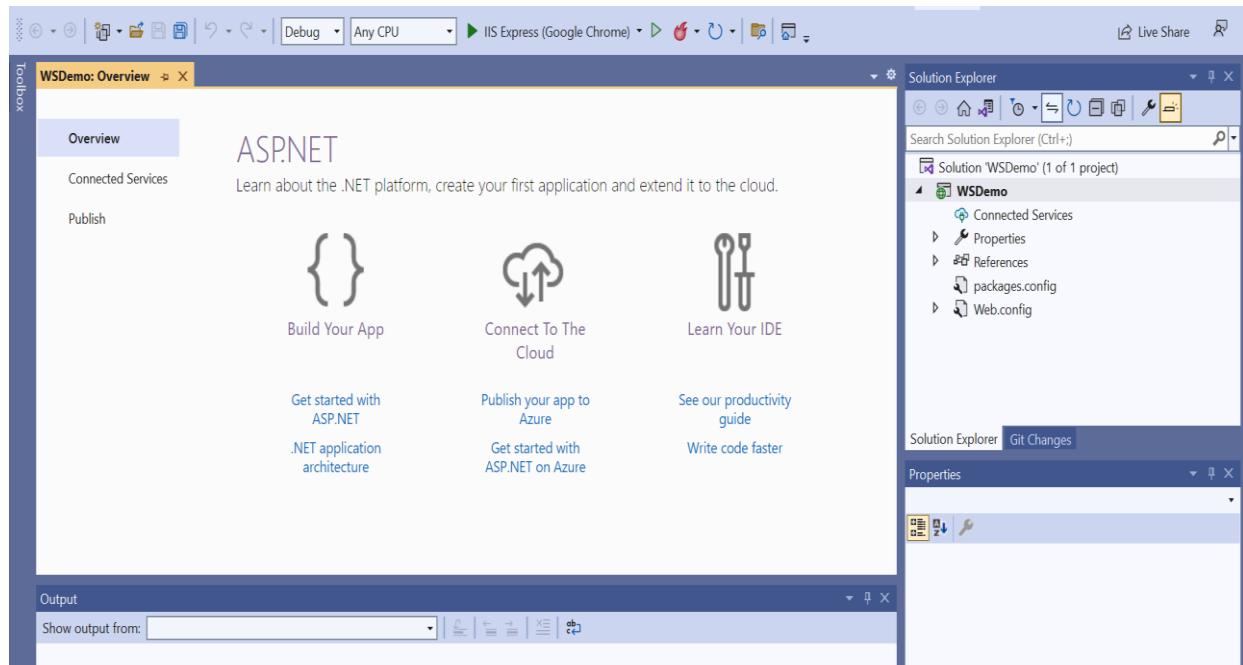
Υπηρεσίες και Συναλλαγές Ιστού – Χ. Γεωργιάδης

Lab1, VS_2022

(Α) Δημιουργία απλής υπηρεσίας ASP.NET

[Στο ίδιο project βρίσκονται και η υπηρεσία (πάροχος) και η ιστοσελίδα κλήσης της υπηρεσίας (καταναλωτής)]

- 1) File -> New Project
- 2) Επιλέγουμε **ASP.NET Web Application(.NET Framework) C#** και το ονομάζουμε **WSDemo**, αποεπιλέγουμε το check box "Configure for HTTPS", και μετά επιλέγουμε **Empty**



(η καρτέλα "WSDemo: Overview" που ανοίγει, σε αυτή τη φάση δεν μας χρειάζεται και μπορούμε να την κλείσουμε)

- 3) Επιλέγουμε πάνω του (από το Solution Explorer) με δεξιά κλικ, Add -> New Item -> Web Service (ASMX) και ονομάζουμε την υπηρεσία **HelloWS.asmx**. Αυτό δημιουργεί αυτόματα το αρχείο HelloWS.asmx.cs που περιέχει τον κώδικα της υπηρεσίας
- 4) Ανοίγουμε το αρχείο HelloWS.asmx.cs και αλλάζουμε το όνομα της συνάρτησης (Web method) από HelloWorld() σε Hello()

Έλεγχος-δοκιμή της υπηρεσίας

- 1) Στο παράθυρο Solution Explorer κάνουμε δεξιά κλικ στο HelloWS.asmx και επιλέγουμε Set As Start Page
- 2) Επιλέγουμε πάνω στο πράσινο βελάκι τον Web Browser που επιθυμούμε και κάνουμε enable το Script Debugging
- 3) Κάνουμε κλικ στο πράσινο βελάκι στην γραμμή εργαλείων standard ή απλά πατάμε F5

- 4) Θα ανοίξει ο browser και θα δούμε την υπηρεσία μας και διάφορες πληροφορίες

Οι παρακάτω λεπτουργίες υποστηρίζονται. Για έναν επίσημο ορισμό, δείτε την [Περιγραφή υπηρεσιών](#).

[Hello](#)

Αυτή η υπηρεσία Web χρησιμοποιεί το <http://tempuri.org/> ως προεπιλεγμένο χώρο ονομάτων.

Σύσταση: Αλλάζετε τον προεπιλεγμένο χώρο ονομάτων πριν γίνεται δημόσια η υπηρεσία XML Web.

Κάθε υπηρεσία XML Web χρειάζεται έναν μοναδικό χώρο ονομάτων, ώστε να μπορούν οι εφαρμογές-πελάτες την ξεχωρίζουν από άλλες υπηρεσίες στο Web. Το <http://tempuri.org/> είναι διορίστημα για υπηρεσίες XML Web που βρίσκονται υπό ανάπτυξη, αλλά οι υπηρεσίες XML Web που έχουν δημοσιευθεί θα πρέπει να χρησιμοποιούν έναν πιο μόνιμο χώρο ονομάτων.

Η παρούσα XML Web χρειάζεται να αναγνωρίζεται από ένα χώρο ονομάτων που ελέγχεται. Για πορέθημα, μπορείτε να χρησιμοποιήσετε το όνομα τομέα Internet της επορίας σας ως μέρος του χώρου ονομάτων. Αν και πολλοί χώροι ονομάτων των υπηρεσιών XML Web μοιάζουν με διαμεμένες URLs, δεν χρειάζεται να δεκχόνται σε πραγματικούς πόρους του Web. (Οι χώροι ονομάτων των υπηρεσιών XML Web είναι URLs.)

Για τη δημιουργία υπηρεσιών XML Web με χρήση του ASP.NET, ο προεπιλεγμένος χώρος ονομάτων μπορεί να αλλάξει χρησιμοποιώντας την ιδιότητα Namespace του χαρακτηριστικού WebService. Το χαρακτηριστικό WebService είναι ένα χαρακτηριστικό που εφαρμόζεται στην κλάση που περιέχει τις μεθόδους της υπηρεσίας XML Web. Ακολουθεί ένα παράδειγμα κώδικα που ορίζει ως χώρο ονομάτων το "<http://microsoft.com/webservices/>".

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // ...
}
```

Visual Basic

```
<WebService(Namespace="http://microsoft.com/webservices/")>
Public Class MyWebService
    ' ...
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    public:
        // ...
}
```

Για περισσότερες λεπτομέρειες σχετικά με πεδία ονόματος XML, δείτε τις συστάσεις του W3C για [χώρους ονομάτων σε XML](#).

Για περισσότερες λεπτομέρειες σχετικά με το WSDL, δείτε την [περιγραφή WSDL](#).

Για περισσότερες λεπτομέρειες σχετικά με τις URLs, δείτε το [RFC 2396](#).

- 5) Για να δείτε την wsdl (xml) περιγραφή της υπηρεσίας που δημιουργήθηκε, πατήστε στον σύνδεσμο «Περιγραφή Υπηρεσιών»

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://tempuri.org/">
    <wsdl:types>
        ...
    </wsdl:types>
    <wsdl:message name="HelloSoapIn">
        ...
    </wsdl:message>
    <wsdl:message name="HelloSoapOut">
        ...
    </wsdl:message>
    <wsdl:portType name="HelloWSSoap">
        <wsdl:operation name="Hello">
            <wsdl:input message="tns:HelloSoapIn"/>
            <wsdl:output message="tns:HelloSoapOut"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="HelloWSSoap" type="tns:HelloWSSoap">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="Hello">
            <soap:operation soapAction="http://tempuri.org/Hello" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="HelloSSoap12" type="tns:HelloSSoap">
        ...
    </wsdl:binding>
    <wsdl:service name="HelloWS">
        ...
    </wsdl:service>
</wsdl:definitions>
```

- 6) Για να δοκιμάσετε την υπηρεσία που δημιουργήθηκε, πατήστε στον σύνδεσμο «Hello» και στη συνέχεια το πλήκτρο «Κλήση» που καλεί την ομώνυμη μέθοδο/λειτουργία (μοναδική σε αυτήν την υπηρεσία)

The screenshot shows a browser window titled "HelloWS Υπηρεσία Web". The address bar contains "localhost:61087/HelloWS.asmx?op=Hello". The page title is "HelloWS". A message at the top says "Κάντε κλικ εδώ για μια πλήρη λίστα λειτουργιών." Below it, a section titled "SOAP 1.1" contains a "Δοκιμή" button. A note says "Για να δοκιμάσετε τη λειτουργία χρησιμοποιώντας το πρωτόκολλο HTTP POST, κάντε κλικ στο κουμπί 'Κλήση'." A "Κλήση" button is shown. The "SOAP 1.1" section contains a code block:

```

POST /HelloWS.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Hello"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>

```

- 7) Σε xml μορφή βλέπουμε το string «Hello World» να είναι το αποτέλεσμα της κλήσης της υπηρεσίας, οπότε βεβαιωνόμαστε ότι η μέθοδος αυτή λειτουργεί έτσι όπως θέλουμε (αν αυτό θέλουμε...)

The screenshot shows a browser window with the same title and URL as the previous one. The page content says "This XML file does not appear to have any style information associated with it. The document tree is shown below." Below it is the XML response:

```

<string xmlns="http://tempuri.org/">Hello World</string>

```

Χρήση-κλήση της υπηρεσίας

- 1) Επιλέγουμε πάνω στο project (από το Solution Explorer) με δεξιά κλικ, Add -> New Item -> Web Form και έτσι δημιουργίται αυτόματα ένα αρχείο WebForm1.aspx
- 2) Κάνουμε πάνω του διπλό κλικ και επιλέγουμε το Design view
- 3) Από την εργαλειοθήκη Toolbox -> standard σύρουμε ένα εργαλείο τύπου Button μέσα στο προεπιλεγμένο στοιχείο div
- 4) Πατήστε 2 φορές ENTER ώστε να προστεθεί χώρος 2 νέων σειρών μέσα στο div και στην συνέχεια από την εργαλειοθήκη Toolbox -> standard σύρετε ένα εργαλείο τύπου Label
- 5) Στο παράθυρο properties του εικονιδίου Label διαγράψτε το προεπιλεγμένο κείμενο "Label" από την ιδιότητα text
- 6) Κάντε διπλό κλικ στο εικονίδιο button ώστε να ανοίξει ο κώδικας-σκελετός για το function Button1_Click()
- 7) Προσθέτουμε τον παρακάτω κώδικα

```

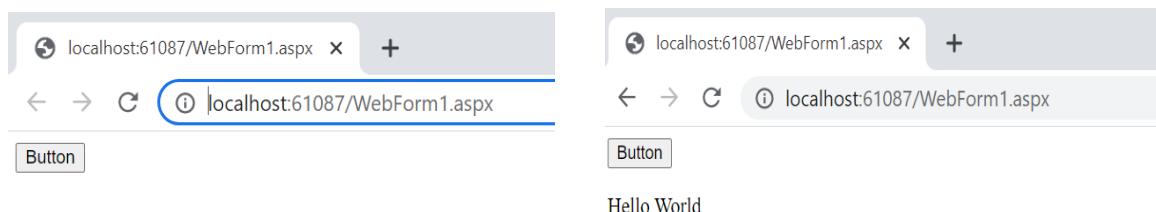
protected void Button1_Click(object sender, EventArgs e)
{
    // HelloWS είναι ο "τύπος" που δημιούργησε η υπηρεσία HelloWS
    // η myHello είναι η μεταβλητή που δηλώνουμε, τύπου HelloWS
    HelloWS myHello = new HelloWS();//

    //κλήση της μεθόδου Hello() της υπηρεσίας HelloWS
    Label1.Text = myHello.Hello();
}

```

- 8) Στο παράθυρο Solution Explorer κάνουμε δεξιά κλικ στο WebForm1.aspx και επιλέγουμε «Set As Start Page».
- 9) Κάνουμε κλικ στο πράσινο βελάκι στην γραμμή εργαλείων ή απλά πατάμε F5

- 10) Θα ανοίξει ο browser και θα μπορέσουμε να ελέγξουμε την συνάρτηση (μέσω του button) και την υπηρεσία μας



(B) Ανάπτυξη Ανεξάρτητης Υπηρεσίας Web

[Σε διαφορετικά projects/μηχανές βρίσκονται η υπηρεσία (πάροχος) και η ιστοσελίδα κλήσης της υπηρεσίας (καταναλωτής)]

Δημιουργία της MyWSservice

(Κάνουμε, όπως και στη προηγούμενη άσκηση δημιουργίας υπηρεσίας, τα πρώτα 3 βήματα, ονομάζοντας απλώς διαφορετικά την υπηρεσία)

- 1) File -> New Project
- 2) Επιλέγουμε ASP.NET Web Application (.NET Framework), το ονομάζουμε TestWS, αποεπιλέγουμε το check box "Configure for HTTPS", και επιλεγούμε Empty
- 3) Επιλέγουμε Add->New Item->Web Service (ASMX) και ονομάζουμε την υπηρεσία MyWSservice.asmx. Αυτό δημιουργεί αυτόματα το αρχείο MyWSservice.asmx.cs που περιέχει τον κώδικα της υπηρεσίας.
- 4) Άλλαζουμε το namespace από <http://tempuri.org> σε π.χ. <http://mywebservice.org>
(Το βήμα αυτό αλλαγής namespace χρειάζεται όταν θέλουμε η υπηρεσία που δημιουργούμε να μπορεί να είναι προσβάσιμη «εξωτερικά», δηλ. να κληθεί από διαφορετικά projects/μηχανήματα)
- 5) Αντικαθιστούμε την προεπιλεγμένη συνάρτηση/μέθοδο HelloWorld() με την ακόλουθη:

```
[WebMethod]
public string ServerTime() {
    string st;
    st = DateTime.Now.ToString();
    return st;
}
```

- 6) Κάτω από την συνάρτηση αυτή προσθέτουμε την ακόλουθη:

```
[WebMethod]
public int CalculatePercentage(int myTotal, int myValue) {
    int cp;
    cp = (int)myValue * 100 / myTotal;
    return cp;
}
```

- 7) Στο παράθυρο Solution Explorer κάνουμε δεξιά κλικ στο MyWService.aspx και επιλέγουμε «Set As Start Page»
- 8) Κάνουμε κλικ στο πράσινο βελάκι στην γραμμή εργαλείων ή απλά πατάμε F5
- 9) Θα ανοίξει ο browser και θα μπορέσουμε να ελέγχουμε τις 2 μεθόδους (operations) της υπηρεσίας μας

MyWService

Οι παρακάτω λειτουργίες υποστηρίζονται. Για έναν επισημο ορισμό, δείτε την [Περιγραφή υπηρεσιών](#).

- [CalculatePercentage](#)
- [ServerTime](#)

localhost:51940/MyWService.asmx/ServerTime

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<string xmlns="http://mywebservice.org/">11/9/2022 12:44:34 μμ</string>
```

Κάντε κλικ εδώ για μια πλήρη λίστα λειτουργιών.

CalculatePercentage

Δοκιμή

Για να δοκιμάσετε τη λειτουργία χρησιμοποιώντας το πρωτόκολλο HTTP POST, κάντε κλικ στο κουμπί 'Κλήση'.

Παράμετρος	Τιμή
myTotal:	<input type="text" value="200"/>
myValue:	<input type="text" value="100"/>

Kλήση

SOAP 1.1

Ακολουθεί ένα δείγμα αίτησης και απόκρισης SOAP 1.1. Οι θέσεις αντικειμένου που εμφανίζονται θα πρέπει να αντικατασταθούν με πραγματικές τιμές.

```
POST /MyWService.asmx HTTP/1.1
Host: localhost
```

localhost:51940/MyWService.asmx/CalculatePercentage

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<int xmlns="http://mywebservice.org/">50</int>
```

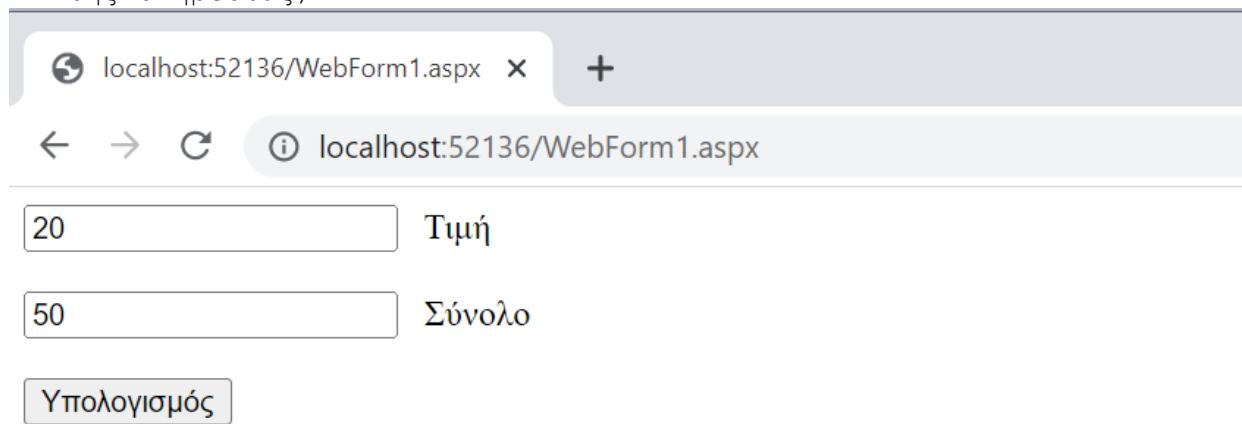
Χρήση-κλήση της MyWSservice (από διαφορετικό project/μηχάνημα)

- 1) Ανοίγουμε (κατά προτίμηση ζητάμε με δεξί κλικ Run As Administrator) ένα δεύτερο στιγμιότυπο του Visual Studio 2022 και ζητάμε File -> New Project

(τα επόμενα 3 βήματα είναι όπως στη προηγούμενη άσκηση χρήσης-κλήσης της υπηρεσίας, δίνοντας βέβαια άλλο όνομα στο project)

- 2) Επιλέγουμε ASP.NET Web Application(.NET Framework), το ονομάζουμε MyConsumer, όχι το "Configure for HTTPS" και επιλέγουμε Empty
- 3) Επιλέγουμε πάνω στο site (από το Solution Explorer) με δεξί κλικ, Add -> New Item -> Web Form και έτσι δημιουργείται αυτόματα ένα αρχείο WebForm1.aspx
- 4) Κάνουμε κλικ κάτω στο Design και ανοίγει το design view

(η επιλογή των εργαλείων διάδρασης που τοποθετούμε στην φόρμα, εξαρτάται από τη μορφή των στοιχείων εισόδου που απαιτεί μια μέθοδο υπηρεσίας αλλά και από τη μορφή των αποτελεσμάτων (έξοδος) της υπηρεσίας)



40% = ποσοστό που υπολογίστηκε

11/9/2022 3:00:43 μμ = η ώρα στον server/πάροχο

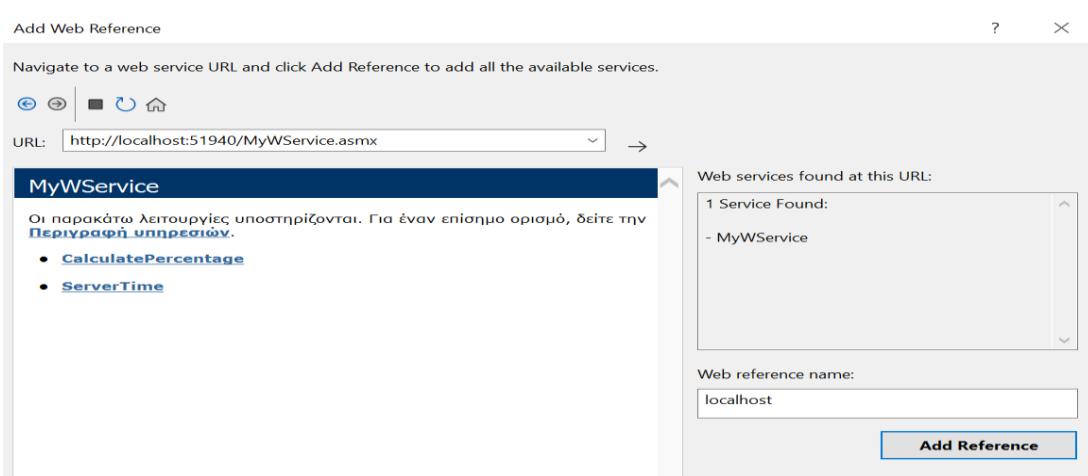
- 5) Από την εργαλειοθήκη Toolbox -> standard σύρουμε ένα εικονίδιο τύπου TextBox μέσα στο προεπιλεγμένο στοιχείο div
- 6) Στο παράθυρο properties του textbox ορίζουμε την ιδιότητα ID σε tbMyValue (μια είσοδος για την μέθοδο **CalculatePercentage()**)
- 7) Από την εργαλειοθήκη Toolbox -> standard σύρουμε ένα εικονίδιο τύπου Label μέσα δίπλα από το textbox. Ορίζουμε την ιδιότητα Text σε 'Τιμή'. Στη συνέχεια κάνουμε κλικ δεξιά του label μας και πατάμε ENTER για να αλλάξει η γραμμή
- 8) Από την εργαλειοθήκη Toolbox -> standard σύρουμε ένα ακόμα εικονίδιο τύπου TextBox μέσα στο προεπιλεγμένο στοιχείο div (κάτω από το προηγούμενο)
- 9) Στο παράθυρο properties του textbox ορίζουμε την ιδιότητα ID σε tbMyTotal (άλλη μια είσοδος για την μέθοδο **CalculatePercentage()**)
- 10) Από την εργαλειοθήκη Toolbox->standard σύρουμε ένα εικονίδιο τύπου Label μέσα δίπλα από το textbox. Ορίζουμε την ιδιότητα Text σε

'Σύνολο'. Στη συνέχεια κάνουμε κλικ δεξιά του label μας και πατάμε ENTER δύο φορές

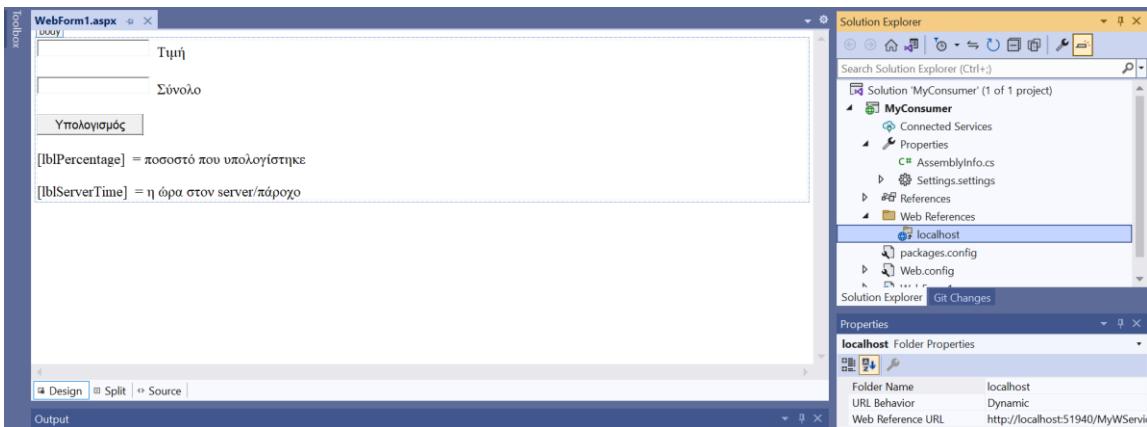
- 11) Από την εργαλειοθήκη Toolbox -> standard σύρουμε ένα εικονίδιο τύπου Button και ορίζουμε την ιδιότητα ID σε btnCalculate. Ορίζουμε και την ιδιότητα text σε 'Υπολογισμός'. Στη συνέχεια κάνουμε κλικ δεξιά του button και πατάμε ENTER
- 12) Από την εργαλειοθήκη Toolbox -> standard σύρουμε ένα εικονίδιο τύπου Label και ορίζουμε την ιδιότητα ID σε lblPercentage, (έξοδος για την μέθοδο **CalculatePercentage()**). Επίσης σβήνουμε το περιεχόμενο της ιδιότητας text.
- 13) Από την εργαλειοθήκη Toolbox -> standard σύρουμε ένα ακόμα εικονίδιο τύπου Label δεξιά από το προηγούμενο και ορίζουμε την ιδιότητα ID σε lblPercentageLabel. Ορίζουμε την ιδιότητα text σε '= ποσοστό που υπολογίστηκε' και κάνουμε την ιδιότητα **Visible -> False**. Στη συνέχεια κάνουμε κλικ δεξιά του label μας και πατάμε ENTER δύο φορές
- 14) Από την εργαλειοθήκη Toolbox -> standard σύρουμε ένα εικονίδιο τύπου Label και ορίζουμε την ιδιότητα ID σε lblServerTime (έξοδος για την μέθοδο **ServerTime()** - η μέθοδος αυτή δεν έχει είσοδο, αλλά μόνο έξοδο). Επίσης σβήνουμε το περιεχόμενο της ιδιότητας text.
- 15) Από την εργαλειοθήκη Toolbox -> standard σύρουμε ένα ακόμα εικονίδιο τύπου Label δεξιά από το προηγούμενο και ορίζουμε την ιδιότητα text σε '= η ώρα στον server'.

Εύρεση της MyWService

- 1) Ανοίγουμε το προηγούμενο στιγμιότυπο του Visual Studio (TestWS), που είναι ο πάροχος της υπηρεσίας
- 2) Εκτελούμε (αν δεν εκτελείται ήδη) την υπηρεσία myWService πατώντας το πράσινο βελάκι
- 3) Στον browser που ανοίγει αντιγράφουμε το URL της υπηρεσίας (π.χ. <http://localhost:51940/MyWService.asmx>)
- 4) Επιστρέφουμε στο δεύτερο στιγμιότυπο του Visual Studio (MyConsumer), που είναι ο καταναλωτής της υπηρεσίας. Στο Solution Explorer κάνουμε δεξιά κλικ στο myConsumer -> Add -> Service Reference. Προσοχή: από το 'Advanced' επιλέξτε στο τμήμα Compatibilty τον επιλογή Add Web Reference
- 5) Κάνουμε επικόλληση το URL, πατάμε το βελάκι (Go) και όταν εμφανιστεί το μήνυμα «1 service found» επιλέγουμε το «Add Reference»



Βλέπουμε στο Solution Explorer να εμφανίζεται τμήμα "Web References"



Προσθήκη κώδικα

- 1) Ζητάμε View -> Code
- 2) Βάζουμε τον ακόλουθο κώδικα μέσα στην κλάση WebForm1:

```
...
public partial class WebForm1 : System.Web.UI.Page
{
    // localhost.MyWSService είναι ο "τύπος" που δημιούργησε η υπηρεσία
    // ή myWS είναι η μεταβλητή που δηλώνουμε, τύπου localhost.MyWSService
    localhost.MyWSService myWS = new localhost.MyWSService();

    ...
protected void Page_Load(object sender, EventArgs e){

    // η κλήση της μεθόδου ServerTime() δεν χρειάζεται παράμετρους εισόδους
    // επιστρέφεται string, το οποίο εμφανίζεται μέσω του label lblServerTime
    lblServerTime.Text = myWS.ServerTime();
}
```

- 3) Με διπλό κλικ στο κουμπί «Υπολογισμός» της φόρμας, δημιουργείται ο «σκελετός» του κώδικα εξυπηρέτησης του κουμπιού, τον οποίο και συμπληρώνουμε:

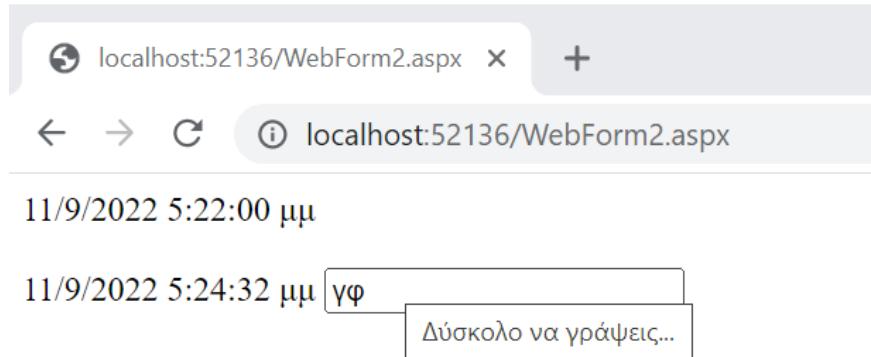
```
protected void btnCalculate_Click (object sender, EventArgs e){

    // σε 2 ακέραιες μεταβλητές, υποδεχόμαστε ότι έδωσε ο χρήστης
    // ως απαραίτητη είσοδο για τη κλήση της μεθόδου CalculatePercentage()
    int myVal = Convert.ToInt32(tbMyValue.Text);
    int myTot = Convert.ToInt32(tbMyTotal.Text);

    // το αποτέλεσμα (ακέραιος) της μεθόδου της υπηρεσίας
    // μετατρέπεται σε string, για να εμφανιστεί μέσω του label lblPercentage
    lblPercentage.Text = Convert.ToString(myWS.CalculatePercentage(myTot,
        myVal)) + "%";
    lblPercentageLabel.Visible = true;
}
```

===== προαιρετικό =====

Υλοποίηση AJAX



Σε μια νέα σελίδα WebForm2.aspx:

- 1) Εισαγωγή ενός label (Label1) στο <div> του <body>
- 2) Εισαγωγή κώδικα:

```
localhost.MyWS myWS = new localhost.MyWS();
...
protected void Page_Load(object sender, EventArgs e){
    Label1.Text = myWS.ServerTime();
}
```

- 3) Βάλτε την WebForm2 ως Start Page και κάντε εκτέλεση - δοκιμή
- 5) Πάρτε από το toolbox (κατηγορία AJAX Extensions) ένα ScriptManager και βάλτε το κάτω από το Label1
- 6) Από κάτω του, βάλτε ένα UpdatePanel (είναι και αυτό στα AJAX Extensions)
- 7) Γράψτε μέσα στο Update Panel μια ετικέτα <ContentTemplate>
- 8) Στο χώρο του <ContentTemplate>, βάλτε ένα Timer, ένα label (Label2) και ένα textbox. Ειδικά για το Timer, προσθέστε μια ιδιότητα Interval="1000", που σημαίνει ότι το συμβάν Tick του timer θα γίνεται κάθε 1000 msec, δηλαδή κάθε ένα second. Ο πηγαίος κώδικας του WebForm2.aspx θα έχει γίνει έτσι:

```
WebForm2.aspx
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
            <asp:ScriptManager ID="ScriptManager1" runat="server">
            </asp:ScriptManager>
            <asp:UpdatePanel ID="UpdatePanel1" runat="server">
                <ContentTemplate>
                    <asp:Timer runat="server" Interval="1000" ID="timer1"
                        OnTick="timer1_Tick"></asp:Timer>
                    <asp:Label ID="Label2" runat="server"></asp:Label>
                    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
                </ContentTemplate>
            </asp:UpdatePanel>
        </div>
    </form>
</body>
```

- 9) Κάντε διπλό κλικ πάνω στον timer για να ανοίξει ο κώδικας για το timer1_Tick(). Γράψτε μέσα:

```
Label2.Text = myWS.ServerTime();
```

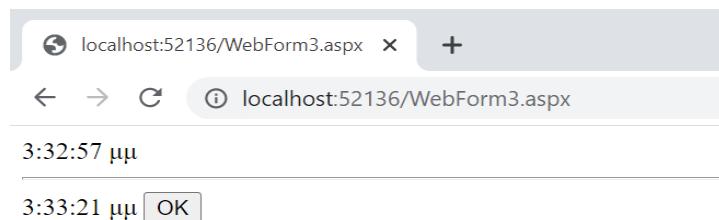
- 10) Η τελική μορφή του WebForm2.aspx.cs είναι:

WebForm2.aspx.cs

```
localhost.MyWSservice myWS = new localhost.MyWSservice();
...
protected void Page_Load(object sender, EventArgs e) {
    Label1.Text = myWS.ServerTime();
}
protected void timer1_Tick(object sender, EventArgs e) {
    Label2.Text = myWS.ServerTime();
}
```

- 11) Προσέξτε πόσο δύσκολο είναι να γράψει κανείς μέσα στο textbox λόγω του timer

Ασύγχρονη ενημέρωση για στοιχείο σε UpdatePanel



- 1) Σε μια νέα σελίδα WebForm3, βάλτε τα ακόλουθα στοιχεία/τον ακόλουθο κώδικα:

WebForm3.aspx

```
<body>
<form id="form1" runat="server">
<div>
    <%= DateTime.Now.ToString("T") %>
    <hr />
    <asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
    <asp:UpdatePanel ID="UpdatePanel1" runat="server">
        <ContentTemplate>
            <%= DateTime.Now.ToString("T") %>
            <asp:Button ID="Button1" runat="server" Text="OK" />
        </ContentTemplate>
    </asp:UpdatePanel>
</div>
</form>
</body>
```

- 2) Βάλτε την WebForm3 ως Start Page

- 3) Εκτελέστε το project: με το πάτημα του κουμπιού OK προκαλείται η ενημέρωση της ώρας.