

# PowerBuilder User Group Meeting 2019

May 20 + 21, 2019 | Frankfurt am Main

The PowerBuilder User Group Germany e.V. organizes for its members and all PowerBuilder friends a **PowerBuilder meeting** from 20 to 21 May 2019 in Frankfurt a.M.

Marco  
MEONI

## C# for PowerBuilder Developers

©2019 Appeon Limited and its subsidiaries. All rights reserved.



APPEON®

# DISCLAIMER

**This presentation was authored by volunteer(s) in the Appeon community. This is not a work for hire by Appeon. The views and opinions expressed in this presentation are those of the author(s).**

**Its contents are protected by US copyright law and may not be reproduced, distributed, transmitted, displayed, published or broadcast without the prior written permission of Appeon. All rights belong to their respective owners.**

**Any reference to third-party materials, including but not limited to Websites, content, services, or software, has not been reviewed or endorsed by Appeon. YOUR USE OF THIRD-PARTY MATERIALS SHALL BE AT YOUR OWN RISK.**

**Appeon makes no warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. Appeon assumes no responsibility for errors or omissions.**

# Presentation

## Agenda

- Introduction to C#
  - Building Blocks
  - Flow Control
  - Data Structures
  - OOP Concepts
- Real-life Use Cases
- Discussion
- From PowerShell to C# (Web API)


# Author Profile



Marco MEONI

 [linkedin.com/in/meonimarco](https://www.linkedin.com/in/meonimarco)

 [twitter.com/marcomeoni](https://twitter.com/marcomeoni)

 [marco.meoni@gmail.com](mailto:marco.meoni@gmail.com)

## Key Skills

- PowerServer
- Appeon MVP
- Big Data
- PowerBuilder
- Machine Learning

## Recent Projects

- **2014 - 2019** : Some dozens of migration projects from PowerBuilder to PowerServer Web & Mobile
- **2015 - 2018** : Ph.D in Big Data and Machine Learning for popularity prediction of ~50 PBytes/year of CERN Physics data

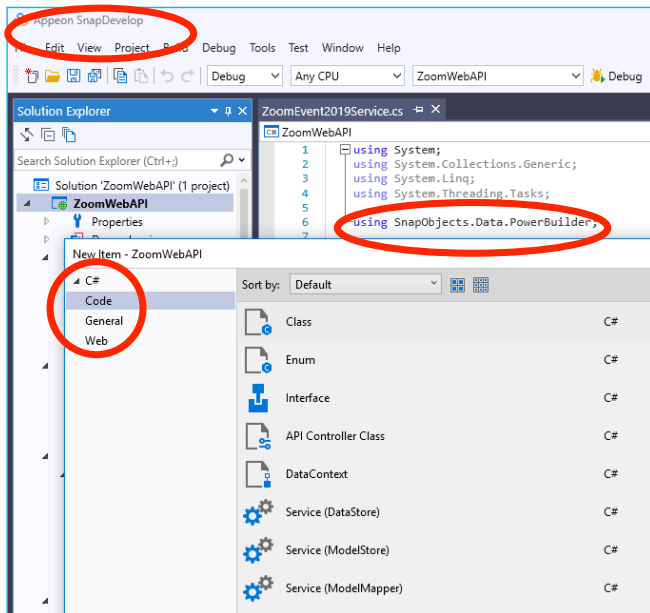
# C# Syntax

Building Blocks

## Background

- C# is the main language of the .NET framework
- .NET includes a large class library named FCL
- Programs written for .NET execute in a sw env called CLR
- However we target .NET Core, MS cross-platform open-source framework
- .NET Core is a combination of ASP.NET MVC and ASP.NET Web API
  - ASP.NET MVC: MS Web framework that implements MVC
  - ASP.NET Web API: MS framework for building RESTful services
  - You can develop using CLI tools, VS 2017, VS Code or SnapDevelop

# SnapDevelop



- SnapDevelop is the new C# IDE included in PB 2019
  - Includes NuGet, the Package Manager for .NET
  - Features IntelliSense from Visual Studio
- SnapObjects is the new C# framework
  - Data components and PowerBuilder DataStore

# Datatype Mapping

.NET	C#	PowerBuilder
System.Boolean	boolean	Boolean
System.Byte	byte	Byte
System.Sbyte	sbyte	
System.Int16	short	Int
System.UInt16	ushort	UInt
System.Int32	int	Long
System.UInt32	uint	Ulong
System.Int64	long	Longlong
System.UInt64	ulong	
System.Single	float	Real
System.Double	double	Double
System.Decimal	decimal	Decimal
System.Char	char	Char
System.String	string	String
System.DateTime	Datetime	Datetime



# Class

```
public class BMI  
{  
}
```

- Just a public class
- Equivalent to PB nonvisualobject

# Methods

```
public class BMI
{
    public float calcBMI(float weight, float height)
    {
        return weight / (height * height);
    }
}
```

Public class with public method

# Fields

```
public class BMI
{
    private static int calcCount;
    public float calcBMI(float weight, float height)
    {
        calcCount++;
        return weight / (height * height);
    }
}
```

A private field (member)

# Properties

```
public class BMI
{
    private static int calcCount;
    public static int ServiceCount {
        get { return calcCount; }
    }
    public float calcBMI(float weight, float height)
    {
        calcCount++;
        return weight / (height * height);
    }
}
```

A property method

# Structs

A Struct definition

```
public struct address {  
    private string street {  
        get { return street.ToUpper(); }  
        set { street = value; }  
    }  
    private string city;  
    private string state;  
    private void clear() {  
        this.street = "";  
        this.city = "";  
        this.state = "";  
    }  
}
```

# Interface

```
interface_movable
{
    int speed{ get; set; };
    void accelerate(int amount);
    void decelerate(int amount);
    void start();
    void stop();
}
```

The Interface contains  
only declarations

# Enumeration

```
public enum Days
{
    monday = 0,
    tuesday = 1,
    wednesday = 2,
    thursday = 3,
    friday = 4,
    saturday = 5,
    sunday = 6
};
```

Declare your own  
enumerations

# Namespace

Namespaces provide levels of code separation

```
namespace pbugg {  
    public class BMI {  
        private static int calcCount;  
        public static int ServiceCount {  
            get { return calcCount; }  
        }  
        public float calcBMI(float weight, float height) {  
            calcCount++;  
            return weight / (height * height);  
        }  
    }  
}
```



# All Together

Appeon  
SnapDevelop  
IDE

Include the standard namespace **System**

Define a class **BMI**

Define the **Main()** method – the program entry point

Print text on the console by method **WriteLine** from class **Console**

```
1 using System;
2
3 namespace PBUGG
4 {
5
6     public interface movable...
14
15     public struct address ...
28
29     public enum Days...
39
40     namespace tutorial
41     {
42         public class BMI
43         {
44             private static int calcCount;
45
46             public static int ServiceCount
47             {
48                 get { return calcCount; }
49             }
50
51             public float calcBMI(float weight, float height)
52             {
53                 calcCount++;
54                 return weight/ (height * height);
55             }
56
57         }
58     }
59
60     public class BMITest
61     {
62         static void Main(string[] args)
63         {
64             BMI bmi = new BMI();
65             Console.WriteLine("BMI Day " + (int)Days.tuesday + ": " + bmi.calcBMI(82.4f, 1.85f));
66             Console.WriteLine("BMI Day " + (int)Days.wednesday + ": " + bmi.calcBMI(82.1f, 1.85f));
67             Console.WriteLine("Nr. Measures: " + BMI.ServiceCount);
68             Console.ReadLine();
69         }
70     }
71 }
```

C:\Program Files\dotnet\dotnet.exe  
BMI Day 1: 24.07597  
BMI Day 2: 23.98831  
Nr. Measures: 2



# C# Syntax

Flow Control

# Iteration Structures

```
for (int i = 0; i < 10; i++)
{
    doSomething();
}

while (true)
{
    doSomething();
}

foreach (var item in myStringArray)
{
    System.Console.WriteLine(item);
}
```

# Conditional Structures

```
if (user == "nobody")
{
    doSomething();
}
else
{
    doSomethingBetter();
}
```

```
switch (caseSwitch)
{
    case 1:
        Console.WriteLine("Case 1");
        break;
    case 2:
        Console.WriteLine("Case 2");
        break;
    default:
        Console.WriteLine("Case Else");
        break;
}
```

# Exception

```
try
{
    // something happens here
}
catch (System.Exception ex)
{
    System.Console.WriteLine("Error: " + ex.Message);
}
```

# C# Syntax

Data Structures

# Array

```
string[] animals = new string[3];
animals[0] = "deer";
animals[1] = "moose";
animals[2] = "boars";

string[] animals2 = new string[] { "deer", "moose", "boars" };

string[] animals3 = { "deer", "moose", "boars" };

Console.WriteLine("Array size: " + animals.Length);
```

# List

```
List<Car> myCollection = new List<Car>();

myCollection.Add(new Car("Fiat 500 D"));
myCollection.Add(new Car("Lancia Fulvia HF"));
myCollection.Add(new Car("Alfa Romeo Giulietta Spider"));
myCollection.Add(new Car("Fiat Topolino A"));

foreach (var car in myCollection)
{
    System.Console.WriteLine(car.ToString());
}
```



# Collections

- Specialized classes for data storage and retrieval
- Enhancement to the arrays
- Array
  - useful for working with fixed number of strongly-typed objects
- Collections
  - group of objects can grow and shrink dynamically
  - you can assign a key to objects for quick retrieve
  - type safe because can contain only one type of data

## Collection (cont)

- Standard (System.Collections)
  - Store elements of type Object
  - ArrayList, Hashtable, Queue, Stack
- Generic (System.Collections.Generic)
  - Enhance code reuse, type safety, and performance
  - Dictionary<T, T>, List<T>, Queue<T>, SortedList<T>, and Stack<T>
- Concurrent
  - BlockingCollection<T>, ConcurrentDictionary<T, T>, ConcurrentQueue<T>, and ConcurrentStack<T>

# C# Syntax

OOP Concepts

# Class Inheritance

```
public class X {
    public void GetInfo() {
        System.Console.WriteLine("Base Class");
    }
}

public class Y : X {
    public void GetInfo () {
        System.Console.WriteLine("Derived Class");
    }
}

public class TestUnit {
    static void Main(string[] args) {
        Y y = new Y();
        y.GetInfo();
    }
}
```

Use **base.Method()** to call a base class method that has been overridden

# Multiple Inheritance

```
interface IName { void GetName(string x); }
interface ICity { void GetCity(string x); }
interface IAge { void GetAge(int x); }

class User : IName, ICity, IAge
{
    public void GetName(string a) { Console.WriteLine("Name: {0}", a); }
    public void GetCity(string a) { Console.WriteLine("City: {0}", a); }
    public void GetAge(int a)      { Console.WriteLine("Age: {0}", a); }
}

class Program
{
    static void Main(string[] args)
    {
        User u = new User();
        u.GetName("Marco Meoni"); u.GetCity("Florence"); u.GetAge(45);
    }
}
```

# Polymorphism

```
public class Shape {
    public int Height { get; set; }
    public int Width { get; set; }
    public virtual void Draw() { Console.WriteLine("Perform drawing tasks"); }
}
class Circle : Shape {
    public override void Draw() { } // Code to draw a circle...
}
class Rectangle : Shape {
    public override void Draw() { } // Code to draw a rectangle...
}
class Triangle : Shape {
    public override void Draw() { } // Code to draw a triangle...
}
class Program {
    static void Main(string[] args) {
        var shapes = new List<Shape> { new Rectangle(), new Triangle(), new Circle() };
        foreach (var shape in shapes) { shape.Draw(); }
    }
}
```

# C# Code Examples

Real-life Use Cases

# Create XML docs

No need for PBDOM

```
58 public class XML
59 {
60     public void createXML(string fileName)
61     {
62         System.Xml.XmlDocument xdoc = new System.Xml.XmlDocument();
63         System.Xml.XmlDeclaration dec = xdoc.CreateXmlDeclaration("1.0", "UTF-8", null);
64         System.Xml.XmlElement root = xdoc.CreateElement("myroot");
65         xdoc.InsertBefore(dec, xdoc.DocumentElement);
66         xdoc.AppendChild(root);
67         xdoc.Save(fileName);
68     }
69 }
70
71 public class TutorialTest
72 {
73     static void Main(string[] args)
74     {
75         XML xml = new XML();
76         xml.createXML("C:/Users/mmeoni/Desktop/test.xml");
77
78         Console.ReadLine();
79     }
80 }
```



# Read System Props

```
58 public class Utility
59 {
60
61     public string readEnv()
62     {
63         // Get Windows login name
64         string user = Environment.UserName;
65         return user;
66     }
67 }
68
69 public class TutorialTest
70 {
71     static void Main(string[] args)
72     {
73         Utility util = new Utility();
74
75         Console.WriteLine("Windows username: " + util.readEnv());
76     }
77 }
```

# Send an Email

```
78 public void sendEmail()  
79 {  
80     System.Net.Mail.SmtpClient smtp = new System.Net.Mail.SmtpClient();  
81  
82     smtp.Host = "smtp.gmail.com";  
83     smtp.Port = 587;  
84     smtp.EnableSsl = true;  
85     smtp.Credentials = new System.Net.NetworkCredential("marco.meoni@gmail.com", "mypwd");  
86     smtp.Timeout = 20000;  
87  
88     smtp.Send("marco.meoni@gmail.com", "info@appeon.com", "Hello", "This is a C# tutorial");  
89 }
```

```
113 var smtp = new SmtpClient  
114 {  
115     Host = "smtp.gmail.com",  
116     Port = 587,  
117     EnableSsl = true,  
118     Credentials = new NetworkCredential("marco.meoni@gmail.com", "mypassword"),  
119     Timeout = 20000  
120 };
```

## Web Resource

```
107 public void webDownload()  
108 {  
109     WebClient wc = new WebClient();  
110     string html = wc.DownloadString("http://www.appeon.com");  
111  
112     Console.WriteLine(html);  
113 }
```

# Write Encoded files

```
115 public void textEncode()  
116 {  
117     System.IO.StreamWriter stw =  
118         new System.IO.StreamWriter("c:/Users/mmeoni/Desktop/test1.out", false, System.Text.Encoding.Unicode);  
119     stw.Write("This text will e encoded");  
120     stw.Close();  
121 }  
122  
123 }  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136
```

- Y ASCII
- Y BigEndianUnicode
- ⊕ Convert
- Y Default
- ⊕ Equals
- ⊕ GetEncoding
- ⊕ GetEncodings
- ⊕ ReferenceEquals
- ⊕ RegisterProvider
- Y Unicode
- Y UTF32
- Y UTF7
- Y UTF8

# Discussion

## I'm a PowerBuilder developer!

- Should I care of C#?
  - Do not underestimate yourself
- PB 2019 includes breakthrough C# Web API feature
- Long history of solutions/hacks to run B.L. on the server
  - Application Server plugin (JEE)
  - SOAP WS (.NET)
  - Assembly via WS (.NET)
  - Stored Procedure
- With C# (Datastore) you can reuse most of the PB DW handling

## I'm a PowerServer developer!

- PowerServer doesn't really separate out B.L. from client app
  - Built-in data connectivity, transaction mgt, DW support, load balancing...
  - ...but PB NVOs remain client code
    - Workarounds: Get/SetFullState, SPs, .NET assembly, WS
    - Non-optimized code suffers of heavy script and multiple server calls
- B.L. separation is the purpose of C# Web API
  - No unsupported PB features
  - C# has more extensive feature set than PB itself
  - Expose app B.L. to the world for integration purposes
- PSW/PSM migration effort can leverage C# Web API too
  - It is not PowerServer vs C# Web API: combine them and get the best

# From PowerScript to C# Web API

My kickoff experience



## Difference

- Case sensitive!
  - Return => return
- String
  - in double quotes, backslash-escaped
  - String(li\_num) => li\_num.ToString()
- Flow Control
  - if ll\_row = 1 then else end if => if(ll\_row == 1) { } else { }

## Difference (cont)

- DataStore operations
  - `li_rc = Ids.SetItem(1, val) => lbool = Ids.SetItem(1, val)`
  - `=> li_row = Ids_cal.InsertRow(0); // row index starts from 0!!!`
  - `ll_val = Ids.GetItemnumber(1, 'col') => ll_val = (long) Ids.GetItemNumber(0, "col")`
  - `=> Ids.SetItem(1, "col", (Int32) Ids_1.ll_val); // cast`

This is how you learn the language!

# Not automatic... ...but easy

```
private function integer of_doJob(ref anv_calc anv_calc, ref nv_err anv_err);

Integer li_rc = 1
String ls_temp
Long ll_rows

Datastore lds_req
lds_req = CREATE datastore
lds_req.DataObject = "ds_req_getinfo"
lds_req.SetTransObject(sqlca)
ll_rows = lds_req.Retrieve(anv_calc.il_reqNum)

If ll_rows = 1 Then

    anv_calc.il_calType = lds_req.GetItemNumber(ll_rows, 'type')
    anv_calc.iddt_calDay = lds_req.GetItemDatetime(ll_rows, 'day')
    anv_calc.il_calScore = of_Score(anv_calc.il_calType, anv_calc.iddt_calDay)

Else

    anv_err.is_message = "Retrieve error for req. "+string(anv_calc.il_reqNum)
    li_rc = -1

End If

Destroy lds_req

Return li_rc

end function
```

```
private int DoJob(ref Calc calc, ref Err err)
{
    int rc = 1;

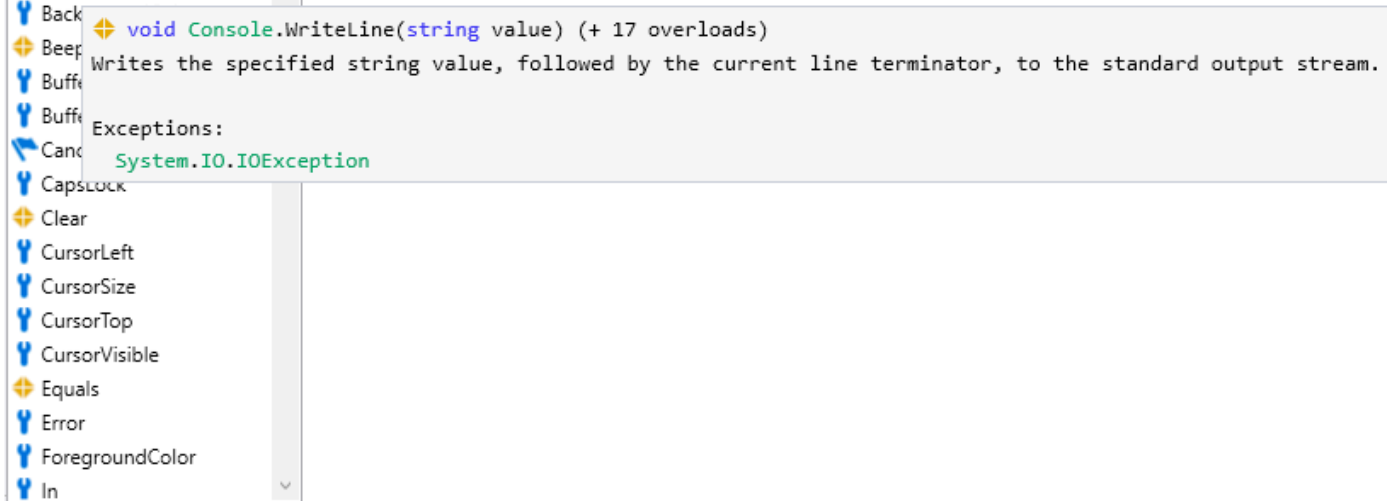
    var dsReq = new DataStore("Ds_Req_Getinfo", _dataContext);
    int rows = dsReq.Retrieve(calc.reqNum);

    if (rows == 1)
    {
        calc.calType = (long)dsReq.GetItemNumber(rows - 1, "type");
        calc.calDay = (DateTime)dsReq.GetItemDateTime(li_rows - 1, "day");
        calc.calScore = Score(calc.calType, calc.calDay);
    }
    else
    {
        err.message = "Retrieve error for req. " + calc.reqNum.ToString();
        rc = -1;
    }

    return rc;
}
```

# IntelliSense

```
107 public void webDownload()  
108 {  
109     WebClient wc = new WebClient();  
110     string html = wc.DownloadString("http://www.appeon.com");  
111  
112     Console.WriteLine(html);  
113 }  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132 public void
```



The image shows a screenshot of a code editor with an IntelliSense dropdown menu open. The code is in C# and shows a method named `webDownload()` that uses `WebClient` to download a string from `http://www.appeon.com` and then calls `Console.WriteLine(html)`. The dropdown menu is positioned over the `Console.WriteLine` call and lists various methods and properties. The top item is `void Console.WriteLine(string value) (+ 17 overloads)`, which is highlighted. Below it, there is a description: "Writes the specified string value, followed by the current line terminator, to the standard output stream." There is also an "Exceptions:" section with `System.IO.IOException` listed. Other items in the list include `Back`, `Beep`, `Buffer`, `Cancel`, `CapsLock`, `Clear`, `CursorLeft`, `CursorSize`, `CursorTop`, `CursorVisible`, `Equals`, `Error`, `ForegroundColor`, and `In`.

mytest\_ws - Apeon PowerBuilder

File Edit View Insert Design Run Tools Window Help

mail@marcomeoni.com

mytest\_ws (C:\Marco\PB\ZSupport\ZOOM)
 

- dbconnector (C:\Marco\PB\ZSupport\ZOOM)
- zoomevent (C:\Marco\PB\ZSupport\ZOOM)
  - a\_zoomevent.pbl (C:\Marco\PB\ZSupport\ZOOM)
  - p\_zoomevent2019.pbl (C:\Marco\PB\ZSupport\ZOOM)
    - d\_cal\_d1\_main
    - d\_nkw\_l1\_all
    - d\_nkw\_l2\_demodisplay
    - d\_par\_l1\_all
    - d\_req\_l1\_where\_par\_rsn
    - n\_tr
    - nv\_appmanager
    - w\_main
    - p\_calculationbase.pbl (C:\Marco\PB\ZSupport\ZOOM)
      - ds\_nkw\_calcbase
      - ds\_req\_getcalcpai
      - nat\_cal\_calculation
        - nv\_n\_zoomevent\_calc
        - Properties
        - Events
        - Functions
          - of\_preinsert (ref data)
          - of\_preinsert\_bcd (ref data)
          - of\_preinsert\_calcbase (ref data)
          - of\_preinsert\_setvalues (ref data)
          - of\_presave (ref data)
          - of\_presave\_calculation (ref data)
          - of\_presave\_get\_bcd (ref data)
          - of\_presave\_setvalues (ref data)
          - classname () returns : integer
          - getcontextservice (string e) returns : string
          - getparent () returns : string
          - postevent (string e) returns : string
          - postevent (string e, long i) returns : string
          - postevent (string e, long i, long lo) returns : string
          - postevent (trigevent e) returns : string
          - postevent (trigevent e, long i) returns : string
          - postevent (trigevent e, long i, long lo) returns : string
          - triggerevent (string e) returns : string
          - triggerevent (string e, long i) returns : string
          - triggerevent (string e, long i, long lo) returns : string
          - triggerevent (trigeven e) returns : string
          - triggerevent (trigeven e, long i) returns : string
          - triggerevent (trigeven e, long i, long lo) returns : string
          - typeof () returns object
      - Structures
      - Controls
      - uo\_zoomevent2019
      - p\_messages.pbl (C:\Marco\PB\ZSupport\ZOOM)
      - nat\_errorinfo
      - p\_generalfunctions.pbl (C:\Marco\PB\ZSupport\ZOOM)

Script - of\_presave\_calculation (ref nat\_cal\_calculation anat\_cal\_calculation, datastore nat\_cal\_calculation) (ref nat\_cal\_calculation) (ref nat\_cal\_calculation)

Access	Return Type	Function Name
private	integer	of_presave_calculation
Pass By	Argument Type	Argument Name
reference	nat_cal_calculation	anat_cal_calculation
value	datastore	ads_nkw
reference	nat_errorinfo	anat_errorinfo

```

integer li_rc = 1
date idt_Today

// living costs
if anat_cal_calculation.il_calLivingType = 1 then
    // Home
    anat_cal_calculation.il_calLivingCostsCal = ads_nkw.getitemnumber(1,"nkw")
else
    // External
    anat_cal_calculation.il_calLivingCostsCal = ads_nkw.getitemnumber(2,"nkw")
end if

// Total costs
anat_cal_calculation.il_calTotalCosts = anat_cal_calculation.il_calLivingCostsCal

// Revenu
anat_cal_calculation.il_calRevenuFranchise = ads_nkw.getitemnumber(3,"nkw")
if anat_cal_calculation.il_calRevenuEff < anat_cal_calculation.il_calRevenuFranchise
    anat_cal_calculation.il_calRevenuCal = 0
else
    anat_cal_calculation.il_calRevenuCal = anat_cal_calculation.il_calRevenuEff
end if

anat_cal_calculation.il_calRevenuTotal = anat_cal_calculation.il_calRevenuCal

anat_cal_calculation.il_calDeficit = anat_cal_calculation.il_calRevenuCal - anat_cal_calculation.il_calTotalCosts

if anat_cal_calculation.il_calDeficit >= 0 then
    anat_cal_calculation.il_calScholarship = 0
else

```

Apeon SnapDevelop

File Edit View Project Build Debug Tools Test Window Help

Debug Any CPU ZoomWebAPI Debug

Solution Explorer

Search Solution Explorer (Ctrl+)

Solution 'ZoomWebAPI' (1 project)

- ZoomWebAPI
  - Properties
  - Dependencies
  - BL
    - Impl
      - NatCalCalculation.cs
      - NatErrorInfo.cs
      - ZoomEvent2019Service.cs
      - IZoomEvent2019Service.cs
      - Controllers
        - DemandController.cs
        - NormController.cs
        - PartnerController.cs
        - ZoomEvent2019Controller.cs
      - Services
        - Impl
          - DemandService.cs
          - NormService.cs
          - PartnerService.cs
          - IDemandService.cs
          - INormService.cs
          - IPartnerService.cs
        - zoomevent2019.pbw
          - zoomevent2019.pbl
            - p\_calculationbase.pbl
              - ds\_nkw\_calcbase
              - ds\_req\_getcalcpai
            - p\_zoomevent2019.pbl
              - d\_cal\_d1\_main.cs
              - d\_cal\_d1\_main.src
              - d\_nkw\_l2\_demo
              - d\_par\_l1\_all.cs
              - d\_par\_l1\_all.src
              - d\_req\_l1\_where\_par\_rsn
        - zoomevent2019.pbw
          - appsettings.Development.json
          - appsettings.json
          - DataContext1.cs
          - Program.cs
          - Startup.cs

ZoomEvent2019Service.cs\*

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using SnapObjects.Data.PowerBuilder;

namespace ZoomWebAPI.BL.Impl
{
    public class ZoomEvent2019Service : IZoomEvent2019Service
    {
        private readonly DataContext1 _dataContext;

        public ZoomEvent2019Service(DataContext1 dataContext)
        {
            _dataContext = dataContext;
        }

        public IDataStore PreInsert(int req_rsn)
        {
            var lds_cal = new DataStore("D_Cal_D1_Main", _dataContext);
            int cal_row = lds_cal.InsertRow(0);
            lds_cal.SetItem(cal_row, "calreq_rsn", req_rsn);

            var lnat_cal_calculation = new NatCalCalculation(_dataContext);
            var lnat_errorinfo = new NatErrorInfo(_dataContext);

            var lds_nkw = new DataStore("Ds_Nkw_calcbase", _dataContext);
            int ll_nkw_rowmax = lds_nkw.Retrieve();

            lnat_cal_calculation.il_calRsn = 1000; // Ever
            lnat_cal_calculation.il_calReqRsn = req_rsn;

            int li_rc = this.PreInsertBCD(ref lnat_cal_calculation);
            if (li_rc == 1) li_rc = this.PreInsertCalculation(ref lnat_cal_calculation);
            if (li_rc == 1) li_rc = this.PreInsertSetValues(ref lnat_cal_calculation);

            return lds_cal;
        }

        private int PreInsertBCD(ref NatCalCalculation anat_cal_calculation)
        {
            int li_rc = 1;

            var lds_req = new DataStore("Ds_Reg_Getcalcpai", _dataContext);
            int li_rows = lds_req.Retrieve(anat_cal_calculation.il_calReqRsn);

            if (li_rows == 1)
            {
                anat_cal_calculation.il_calReqLivingType = anat_cal_calculation.iddt_calParBirthday;
            }
            else
            {
                anat_errorinfo.is_message = "Program error"
            }
        }
    }
}

```

MVC

## What's next?

- Ready to create your first C# Web API...
- ...with SnapDevelop

# Connect with the Appeon Community



[community.appeon.com](https://community.appeon.com)

Discussions, tech articles and videos, free online training, and more.



[facebook.com/AppeonPB](https://facebook.com/AppeonPB)

Encourage us with a “like”, see cool pics, and get notified of upcoming events.



[twitter.com/AppeonPB](https://twitter.com/AppeonPB)

Follow Appeon and community members to get the latest tech news.



[linkedin.com](https://linkedin.com)

Build up your career profile, and stay in contact with other professionals.



[youtube.com/c/AppeonHQ](https://youtube.com/c/AppeonHQ)

Share important Appeon videos with others; no account registration required.



[google.appeon.com](https://google.appeon.com)

Follow Appeon and community members to get the latest tech news.

**Thank you!**

**Questions?**

