FlowNetMaster®

Digital Twins Made Easy

Application Programming Interface (API)

FlowNetMaster® API provides user with the option to build and solve their models by coding them in Pyhton or C# without touching the graphical user interface (GUI). It is convenient for users who would write code rather than drag and drop items into design canvas. API is also quite functional when automation and scripting is preferred.

```csharp
using FlowNetMaster.Core.Medium.CoolProp;
using FlowNetMaster.Core.Node.Fluid;
using FlowNetMaster.Core.Utilities;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using UnitsNet;

namespace FlowNetMaster.Core.Tests
{

    public class ExampleClass
    {

        public void ExampleMethod()
        {
            var model = new Model();
            var fluidName = CoolProp.FluidsV.WATER.ToString();
            var reservoir1 = new InfiniteReservoir(model,
                fluidName.CreateMedium(Temperature.FromKelvins(275), Pressure.FromMegapascals(0.1015)));

            var pipe = new Pipe(Length.FromCentimeters(3), Length.FromMeters(16), Length.FromInches(5.9e-02));

            var reservoir2 = new InfiniteReservoir(model,
                fluidName.CreateMedium(Temperature.FromKelvins(275), Pressure.FromMegapascals(0.101335)));

            model.Add(reservoir1);
            model.Add(reservoir2);
            model.Add(pipe);

            model.Connect(reservoir1.Port1, pipe.Port1);
            model.Connect(reservoir2.Port1, pipe.Port2);
```

```python
import clr
import sys

clr.AddReference("FlowNetMaster.Core")
clr.AddReference("Units")

import FlowNetMaster.Core
import UnitsNet

model = FlowNetMaster.Core.Model()
fluidName = "WATER"

res_1_Temp=UnitsNet.Temperature(275.0,UnitsNet.Units.TemperatureUnit.Kelvin)
res_1_Pres=UnitsNet.Pressure(0.101,UnitsNet.Units.PressureUnit.Megapascal)
res_1_Medium= FlowNetMaster.Core.Utilities.Extentions.CreateMedium(res_1_Temp,res_1_Pres,fluidName)
reservoir1=FlowNetMaster.Core.Node.Fluid.InfiniteReservoir(model,res_1_Medium)

res_2_Temp=UnitsNet.Temperature(275.0,UnitsNet.Units.TemperatureUnit.Kelvin)
res_2_Pres=UnitsNet.Pressure(0.105,UnitsNet.Units.PressureUnit.Megapascal)
res_2_Medium=FlowNetMaster.Core.Utilities.Extentions.CreateMedium(res_2_Temp,res_2_Pres,fluidName)
reservoir2=FlowNetMaster.Core.Node.Fluid.InfiniteReservoir(model,res_2_Medium)


pipe_Diameter=UnitsNet.Length(3.0,UnitsNet.Units.LengthUnit.Centimeter)
pipe_Length=UnitsNet.Length(16.0,UnitsNet.Units.LengthUnit.Meter)
pipe_Roughness=UnitsNet.Length(5.9e-02,UnitsNet.Units.LengthUnit.Inch)

pipe = FlowNetMaster.Core.Node.Fluid.Pipe(pipe_Diameter,pipe_Length,pipe_Roughness)

model.Add(reservoir1)
model.Add(reservoir2)
model.Add(pipe)

model.Connect(reservoir1.Port1, pipe.Port1)
model.Connect(reservoir2.Port1, pipe.Port2)

solver = FlowNetMaster.Core.Solver.SteadyState.NewtonRaphson()
```

*FlowNetMaster® API is not sold seperately. It comes bundled with Professional and Enterprise Editions.

/FlowNetMaster

QUALITY STANDARDS

FlowNetMaster® is developed under ISO 9001 quality assurance system.

ISO 9001:2015