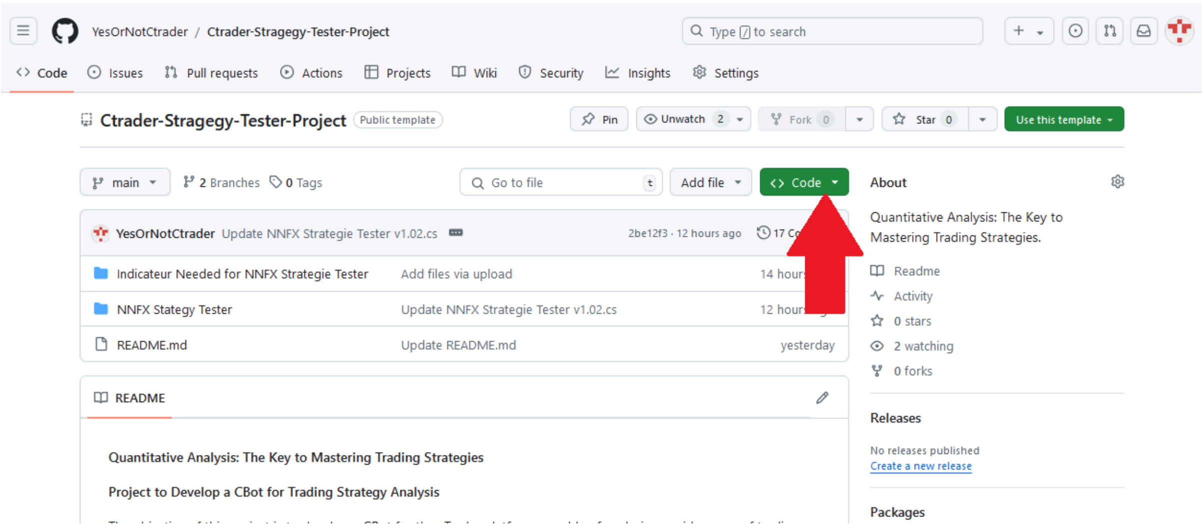
How to Install the NNFX Strategy Tester

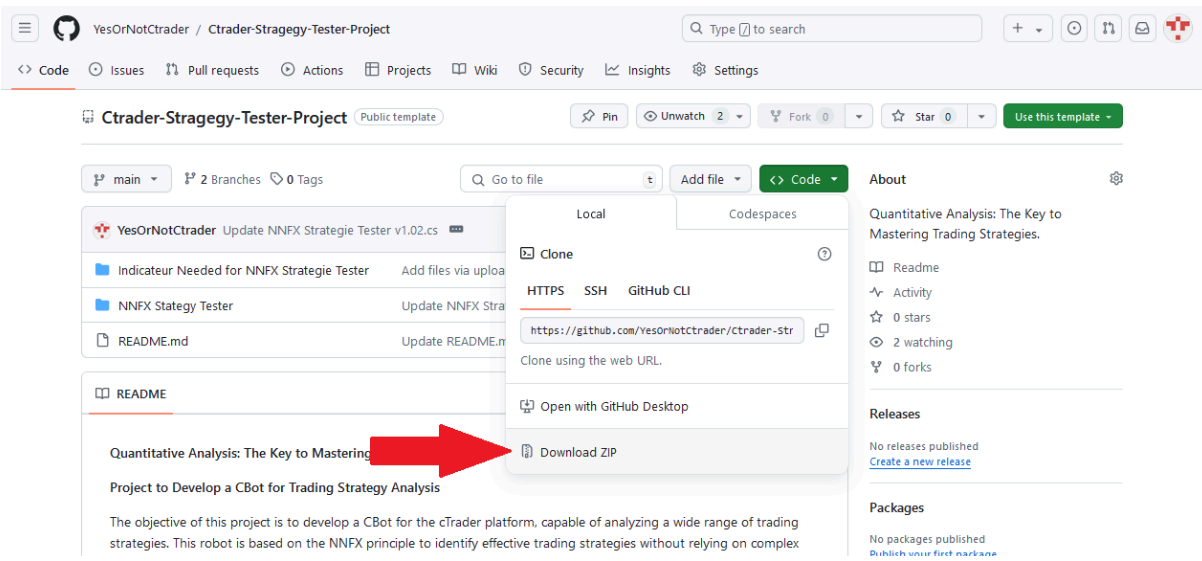
This guide provides two detailed step-by-step methods to install the cBot in the cTrader interface.

I.Download/Install with the ZIP file:

1.
2. Go to the designated page https://github.com/YesOrNotCtrader/Ctrader-Stragegy-Tester-Project
3. Click on the green **Code** button.

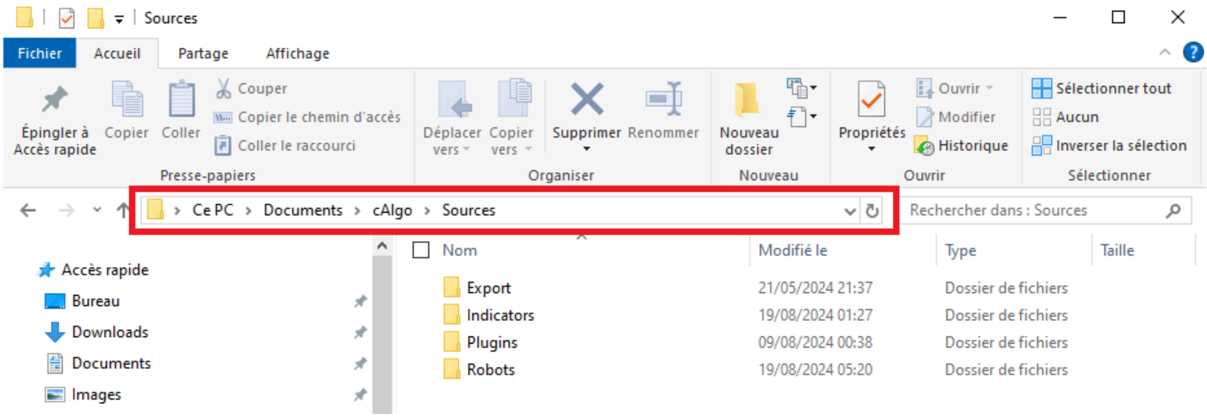


4. Select the **Download ZIP** option to download the file.

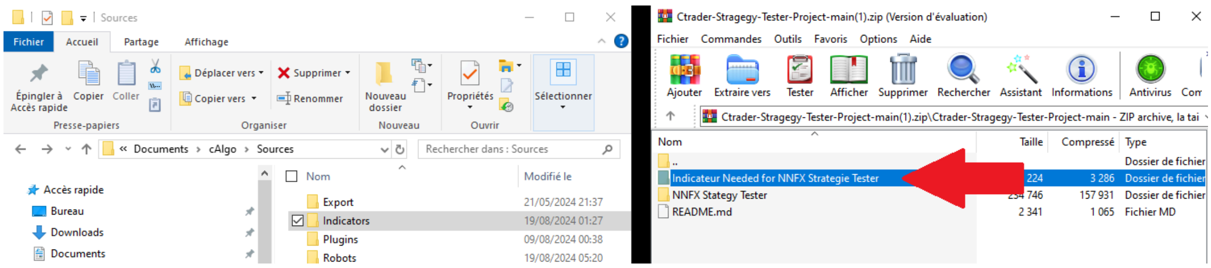


Add the Downloaded Files to the cAlgo Folder:

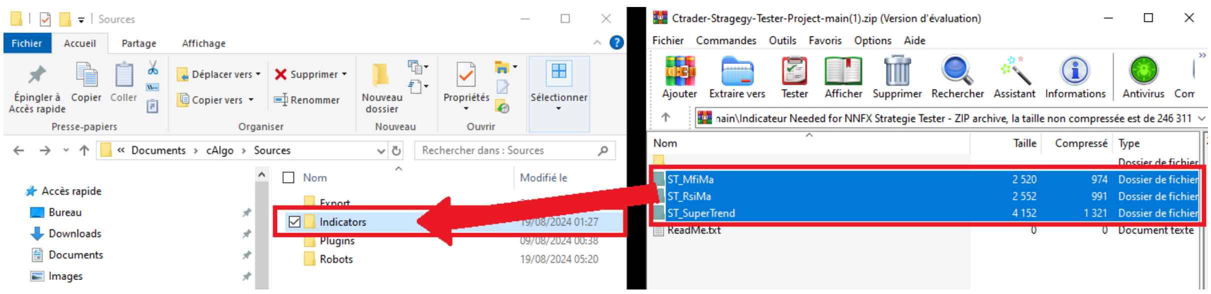
1. Open the **Source** folder for cTrader. This folder is usually located in **Documents -> cAlgo -> Source**.



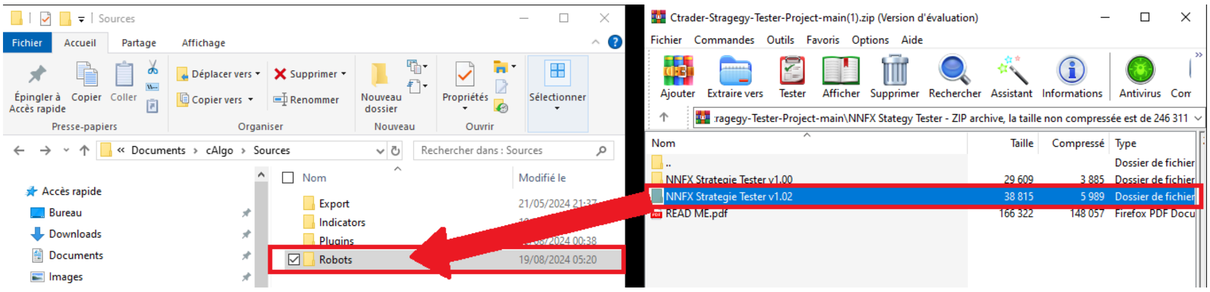
2. Open the downloaded ZIP file and click on “ Indicateur Needed For NNFX Strategie Tester ”.



3. Select all folders named **"ST_..."** (ctrl + click) and Drag and drop these folders into the **Indicators** folder located within the Source folder (from step 1).

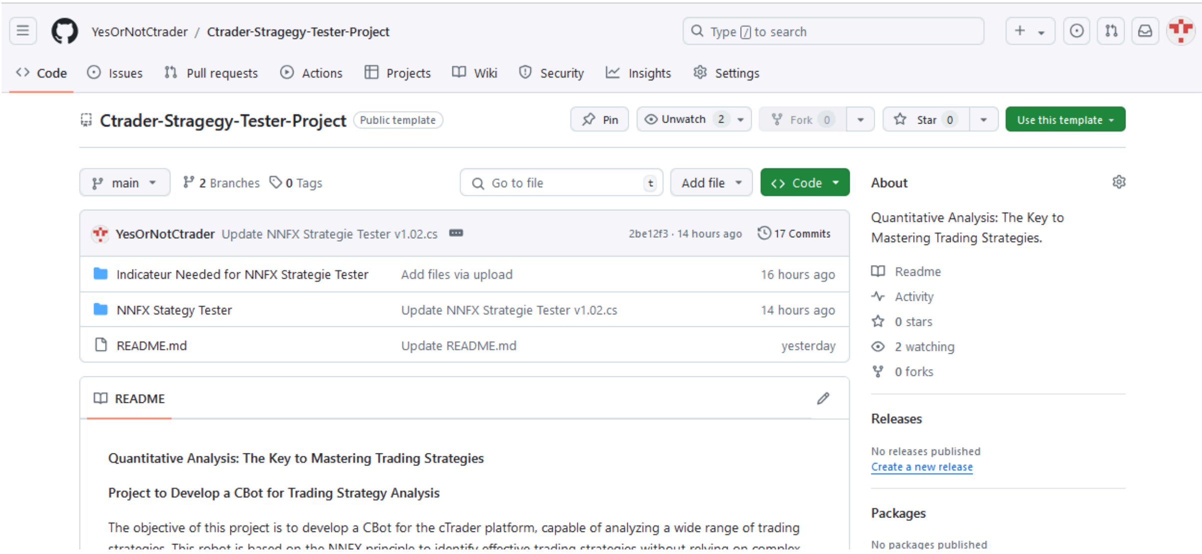


4. Return to the **root** of the downloaded folder (click on "..") and open the folder containing the latest version of cbot located in **"NNFX Strategy Tester"** Folder. For this **demonstration, we are using version 1.02** (the higher the number, the newer the version), and drag the downloaded folder into **Robots**.

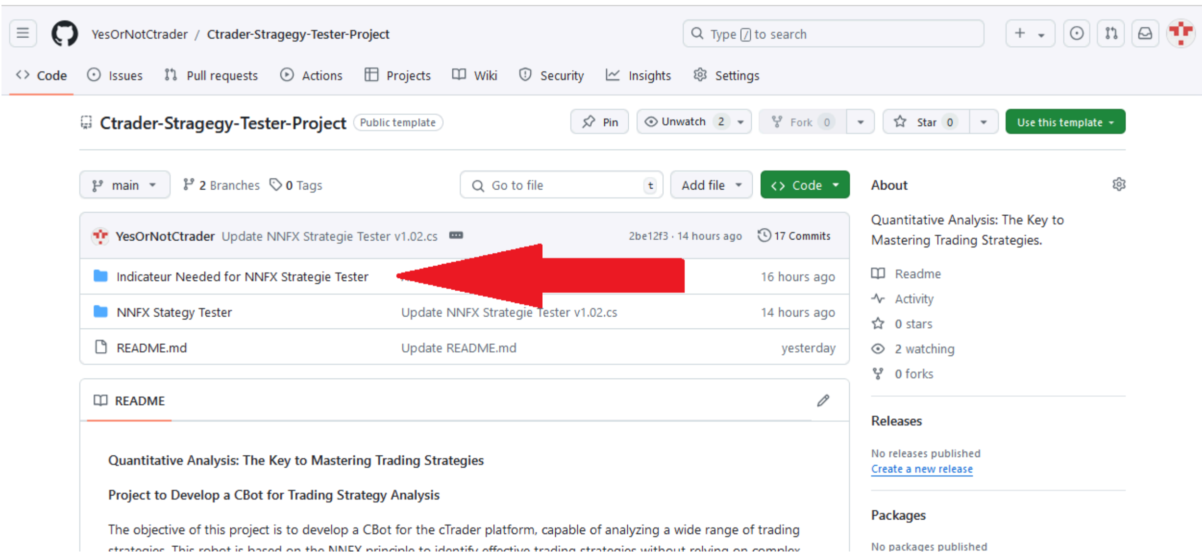


II.Create With “ .Cs “ Folder

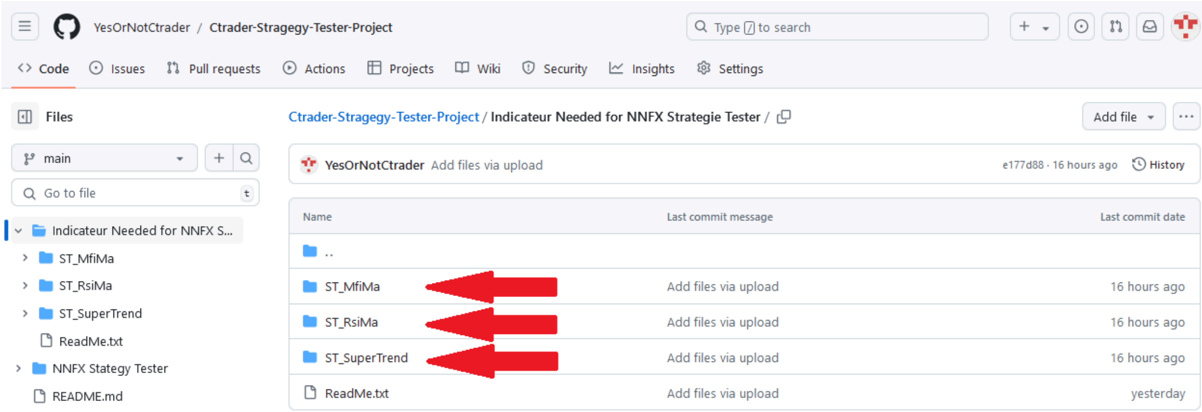
1. Open the GitHub link. : https://github.com/YesOrNotCtrader/Ctrader-Stragegy-Tester-Project

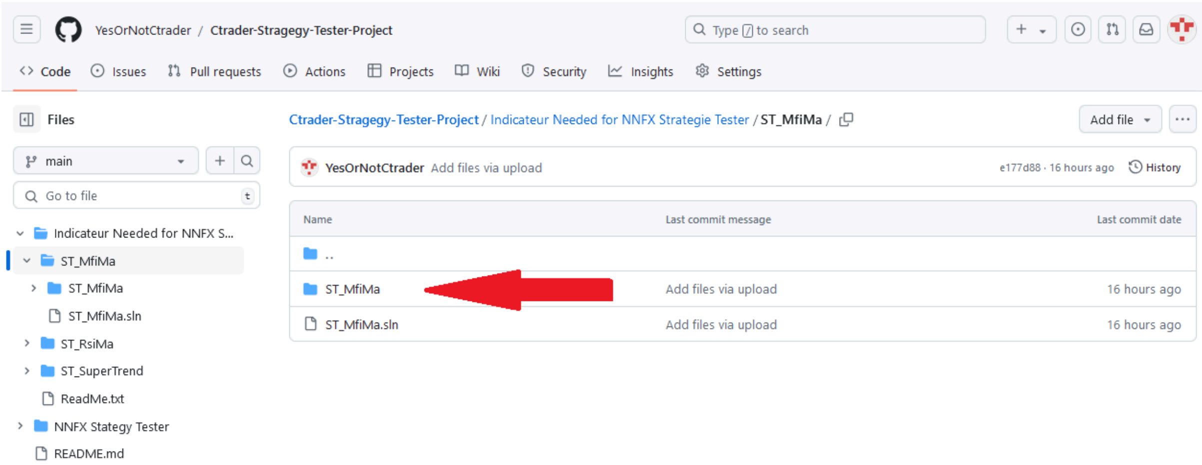


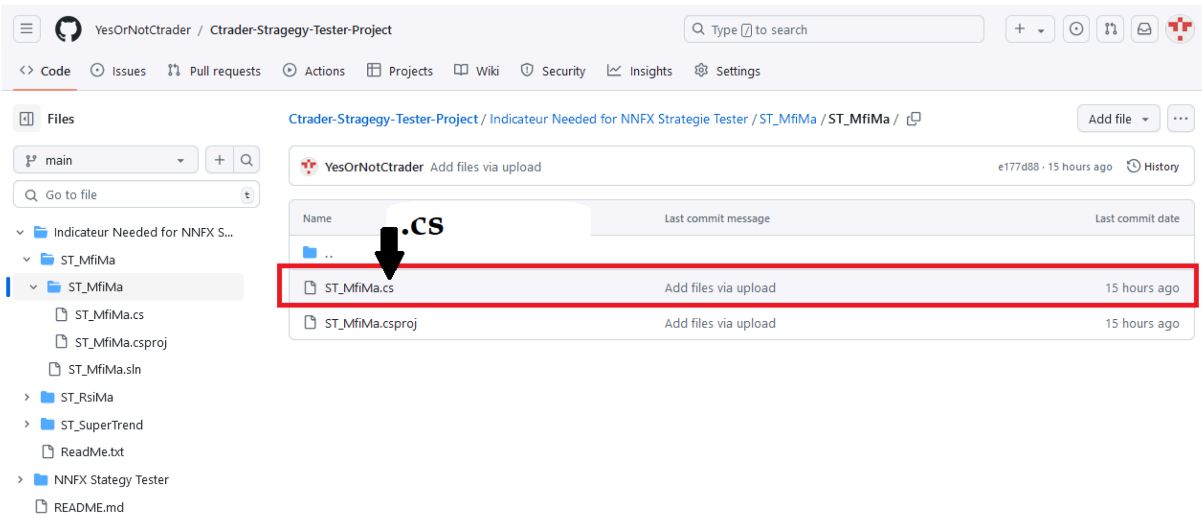
2. Click on "Indicators Needed for NNFX Strategy Tester."



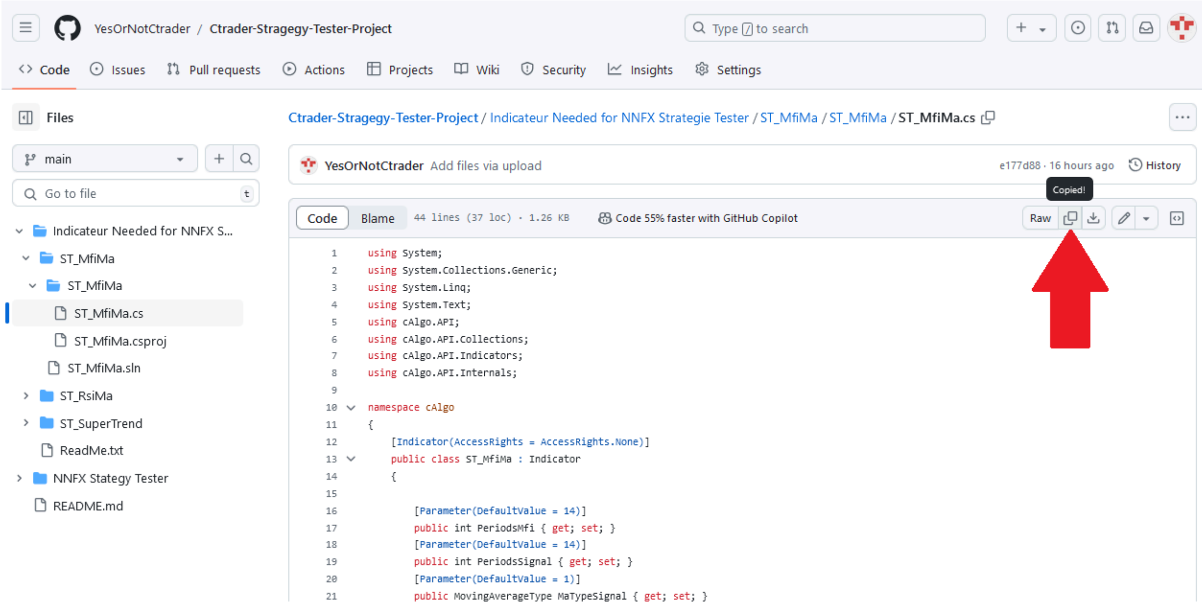
3. Now, for each folder named "ST_...", click until you reach a file with the `.cs` extension.


YesOrNotCtrader / Ctrader-Stragegy-Tester-Project
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Files
main
Go to file
Indicateur Needed for NNFX S...
ST_MfiMa
ST_RsiMa
ST_SuperTrend
ReadMe.txt
NNFX Stategy Tester
README.md
Ctrader-Stragegy-Tester-Project / Indicateur Needed for NNFX Strategie Tester /
Add file
YesOrNotCtrader Add files via upload
e177d88 · 16 hours ago
History
Name
Last commit message
Last commit date
..
ST_MfiMa
Add files via upload
16 hours ago
ST_RsiMa
Add files via upload
16 hours ago
ST_SuperTrend
Add files via upload
16 hours ago
ReadMe.txt
Add files via upload
yesterday

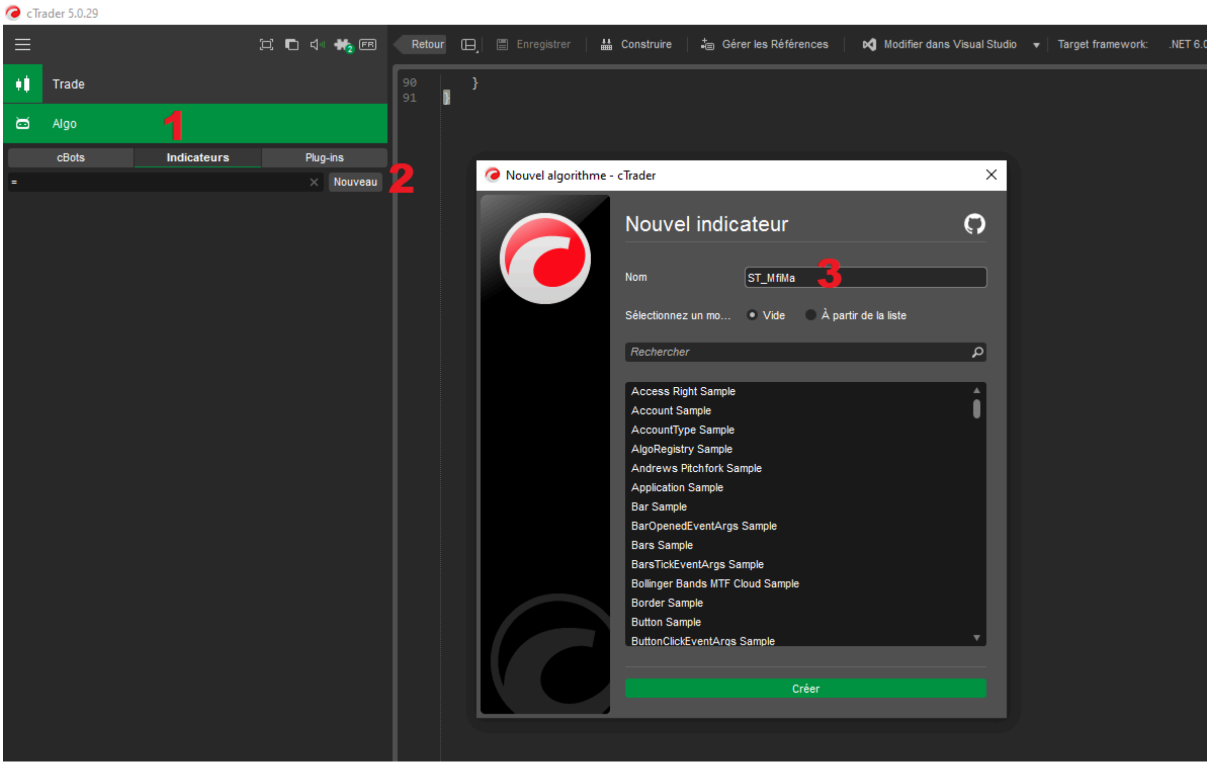

YesOrNotCtrader / Ctrader-Stragegy-Tester-Project
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Files
main
Go to file
Indicateur Needed for NNFX S...
ST_MfiMa
ST_MfiMa
ST_MfiMa.sln
ST_RsiMa
ST_SuperTrend
ReadMe.txt
NNFX Stategy Tester
README.md
Ctrader-Stragegy-Tester-Project / Indicateur Needed for NNFX Strategie Tester / ST_MfiMa /
Add file
YesOrNotCtrader Add files via upload
e177d88 · 16 hours ago
History
Name
Last commit message
Last commit date
..
ST_MfiMa
Add files via upload
16 hours ago
ST_MfiMa.sln
Add files via upload
16 hours ago


YesOrNotCtrader / Ctrader-Stragegy-Tester-Project
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Files
main
Go to file
Indicateur Needed for NNFX S...
ST_MfiMa
ST_MfiMa
ST_MfiMa.cs
ST_MfiMa.csproj
ST_MfiMa.sln
ST_RsiMa
ST_SuperTrend
ReadMe.txt
NNFX Stategy Tester
README.md
Ctrader-Stragegy-Tester-Project / Indicateur Needed for NNFX Strategie Tester / ST_MfiMa / ST_MfiMa /
Add file
YesOrNotCtrader Add files via upload
e177d88 · 15 hours ago
History
Name
.cs
Last commit message
Last commit date
..
ST_MfiMa.cs
Add files via upload
15 hours ago
ST_MfiMa.csproj
Add files via upload
15 hours ago

4. Copy the code obtained



5. Go to cTrader, navigate to the Algo section, click on Indicators, and then on New. Name the indicator (we recommend using the name listed in the folder), then click Create.



6. Once the indicator is created, do the following:

- Click on the code

```
using System;
using cAlgo.API;
using cAlgo.API.Collections;
using cAlgo.API.Indicators;
using cAlgo.API.Internals;

namespace cAlgo
{
    [Indicator(AccessRights = AccessRights.None)]
    public class ST_MfiMa1 : Indicator
    {
        [Parameter(DefaultValue = "Hello world!")]
        public string Message { get; set; }

        [Output("Main")]
        public IndicatorDataSeries Result { get; set; }

        protected override void Initialize()
        {
            // To learn more about cTrader Automate visit our Help Center:
            // https://help.ctrader.com/ctrader-automate

            Print(Message);
        }

        public override void Calculate(int index)
        {
            // Calculate value at specified index
            // Result[index] =
        }
    }
}
```

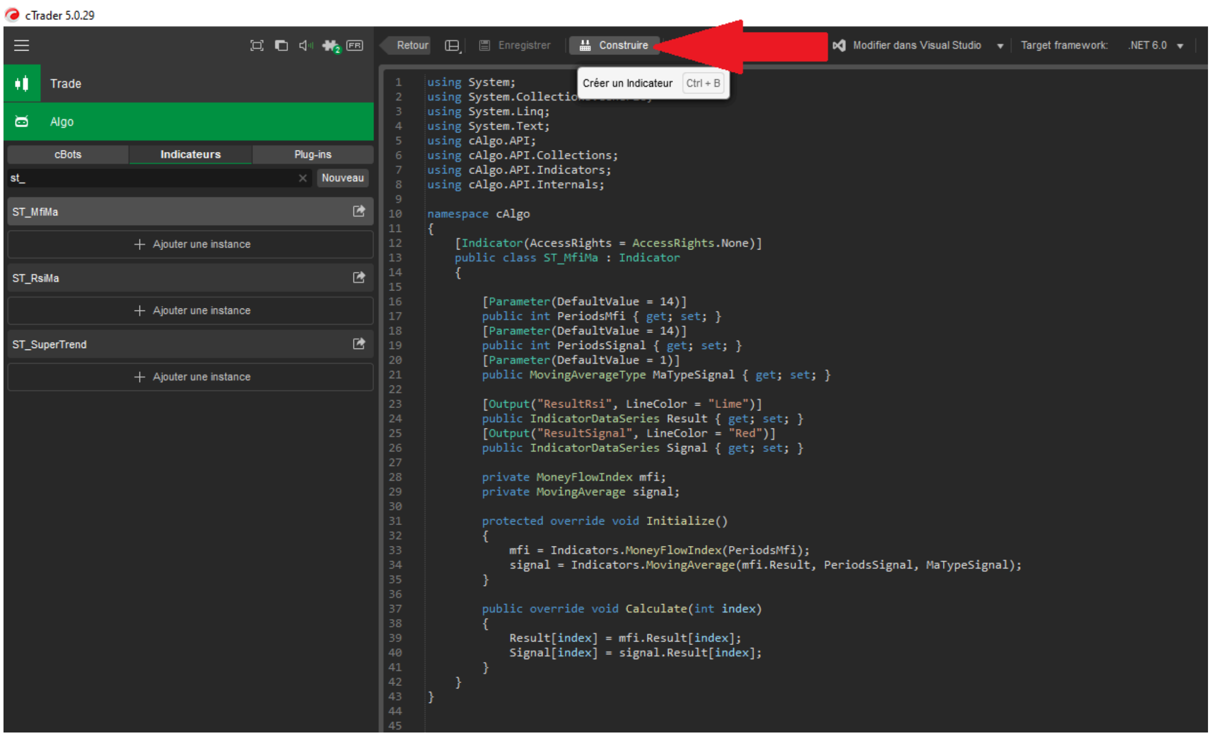
- Press `Ctrl + A` (to select all)

```
using System;
using cAlgo.API;
using cAlgo.API.Collections;
using cAlgo.API.Indicators;
using cAlgo.API.Internals;

namespace cAlgo
{
    [Indicator(AccessRights = AccessRights.None)]
    public class ST_MfiMa1 : Indicator
    {
        [Parameter(DefaultValue = "Hello world!")]
        public string Message { get; set; }

        [Output("Main")]
        public IndicatorDataSeries Result { get; set; }

        protected override void Initialize()
        {
            // To learn more about cTrader Automate visit our Help Center:
            // https://help.ctrader.com/ctrader-automate

            Print(Message);
        }

        public override void Calculate(int index)
        {
            // Calculate value at specified index
            // Result[index] =
        }
    }
}
```

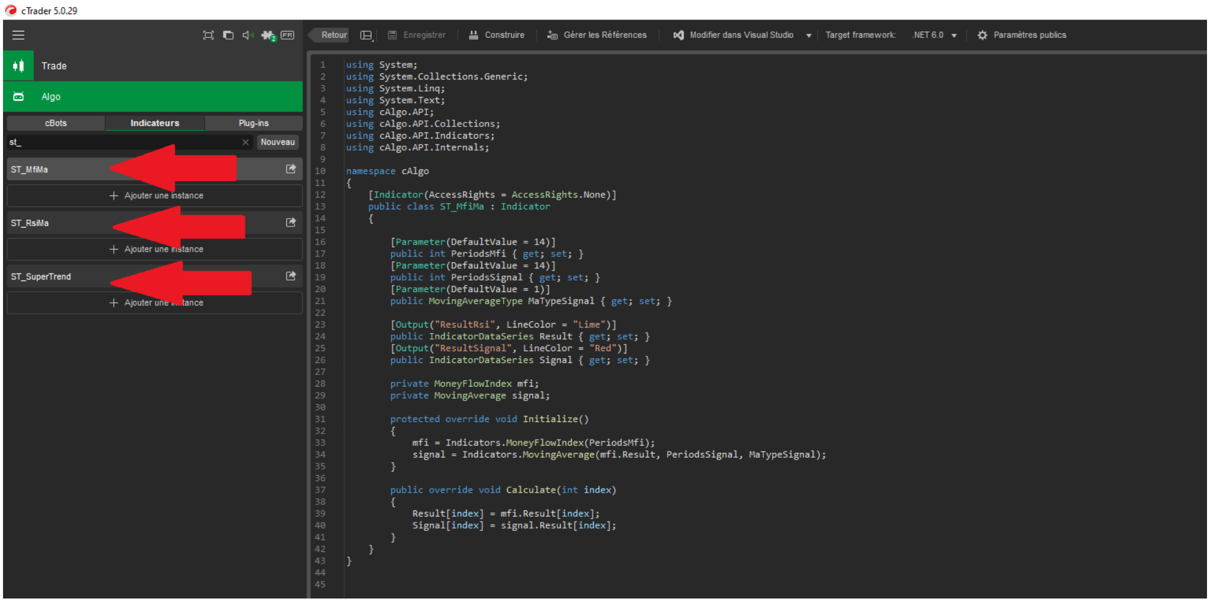
- Press `Ctrl + V` (to paste)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using cAlgo.API;
using cAlgo.API.Collections;
using cAlgo.API.Indicators;
using cAlgo.API.Internals;

namespace cAlgo
{
    [Indicator(AccessRights = AccessRights.None)]
    public class ST_MfiMa : Indicator
    {

        [Parameter(DefaultValue = 14)]
        public int PeriodsMfi { get; set; }
        [Parameter(DefaultValue = 14)]
        public int PeriodsSignal { get; set; }
        [Parameter(DefaultValue = 1)]
        public MovingAverageType MaTypeSignal { get; set; }

        [Output("ResultRsi", LineColor = "Lime")]
        public IndicatorDataSeries Result { get; set; }
        [Output("ResultSignal", LineColor = "Red")]
        public IndicatorDataSeries Signal { get; set; }

        private MoneyFlowIndex mfi;
        private MovingAverage signal;

        protected override void Initialize()
        {
            mfi = Indicators.MoneyFlowIndex(PeriodsMfi);
            signal = Indicators.MovingAverage(mfi.Result, PeriodsSignal, MaTypeSignal);
        }

        public override void Calculate(int index)
        {
            Result[index] = mfi.Result[index];
            Signal[index] = signal.Result[index];
        }
    }
}
```

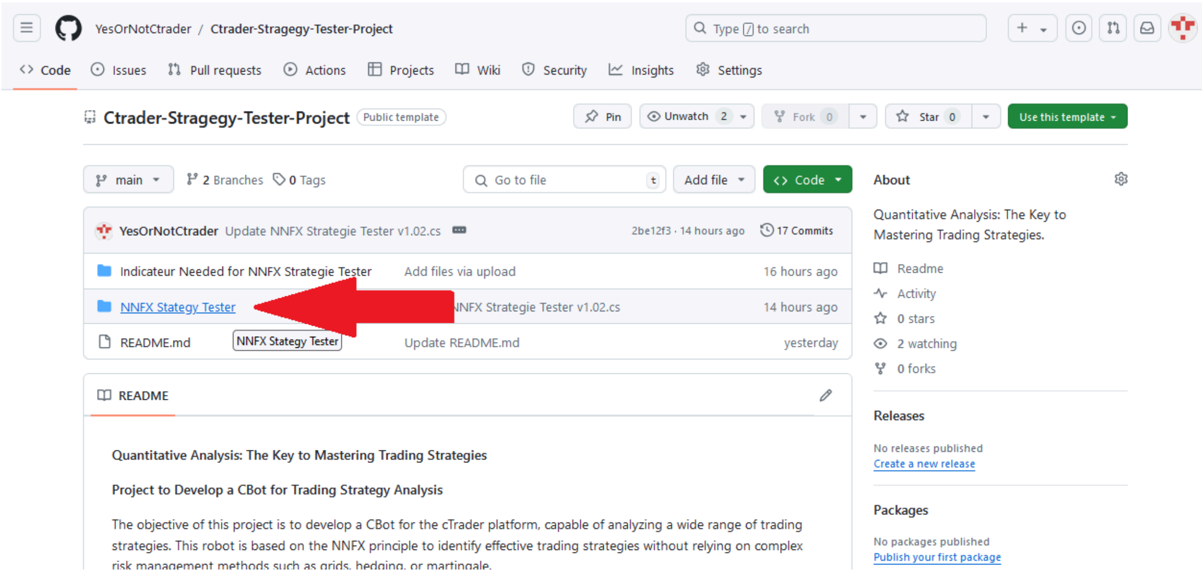
7. Finally, click Create. If you encounter an error, it's possible that the old indicator code was not cleared before you pasted the new code, so restart from step 4 and go directly to step 6.



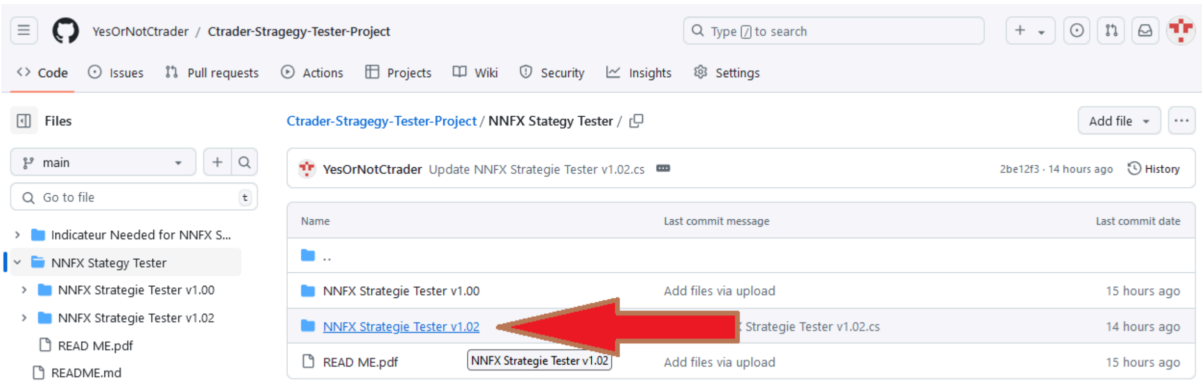
8. Repeat this process for all indicators named "ST_...".

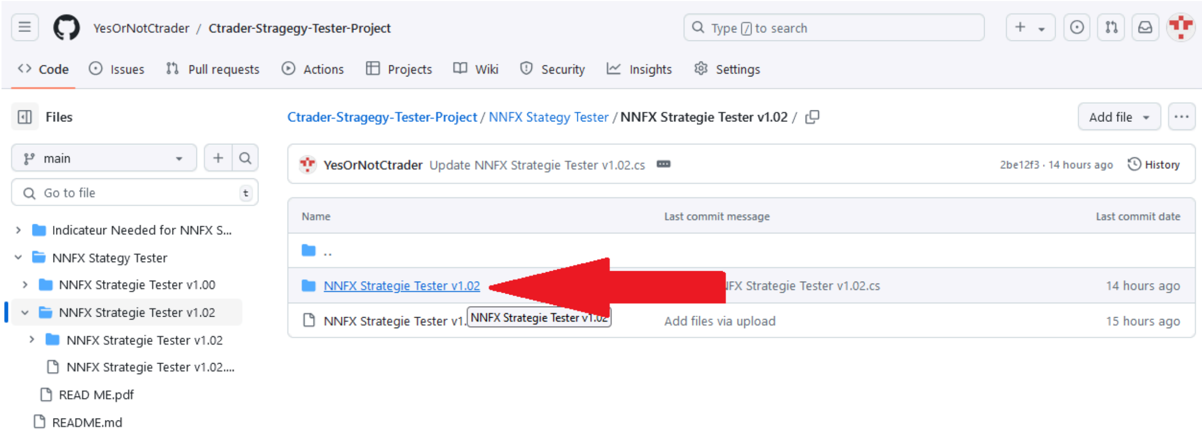


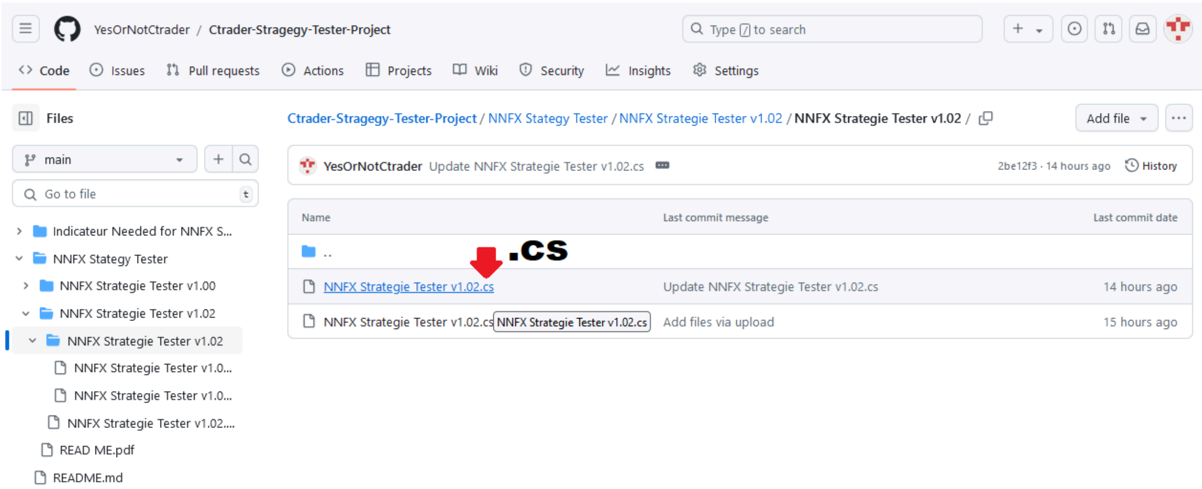
9. Once the indicators are created, click on "NNFX Strategy Tester."



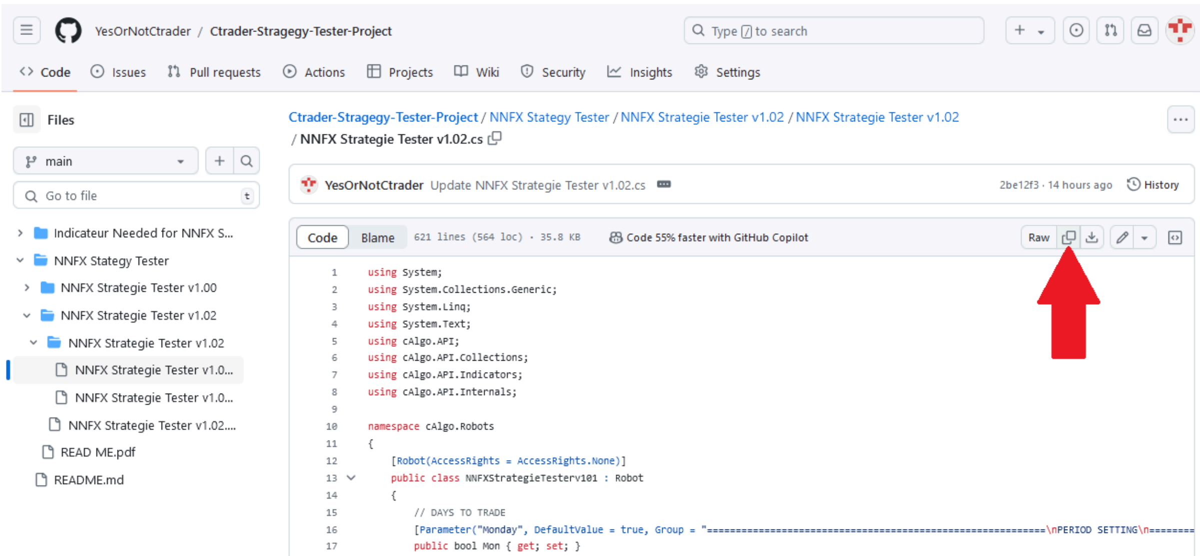
10. Now, select the latest version of "NNFX Strategy Tester v....". Click until you reach a file with the `.cs` extension.





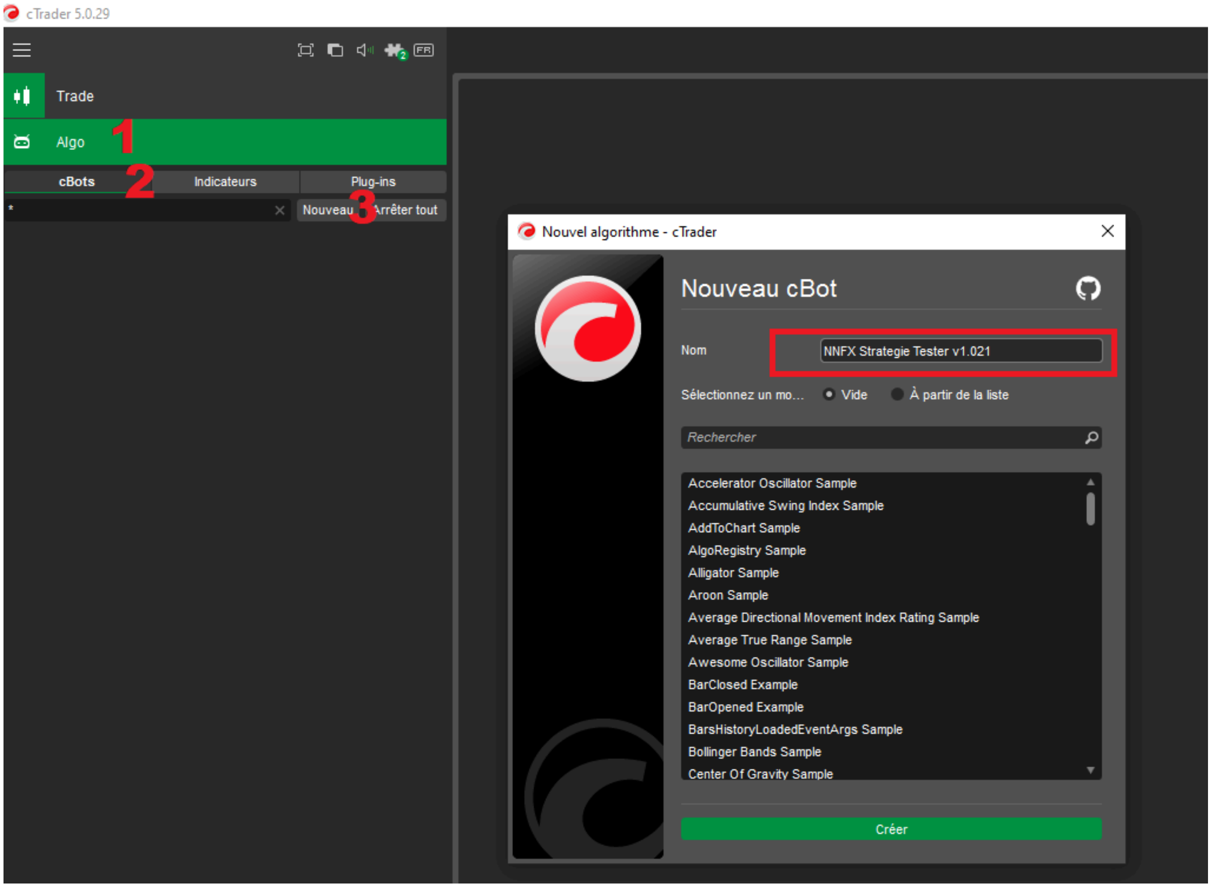


11. Copy the code obtained,



12. then go to cTrader, navigate to the Algo section, click on cBots, and then on New, Name the cBot (we recommend using the name listed in the folder), then click

Create.



13. Once the cBot is created, do the following:

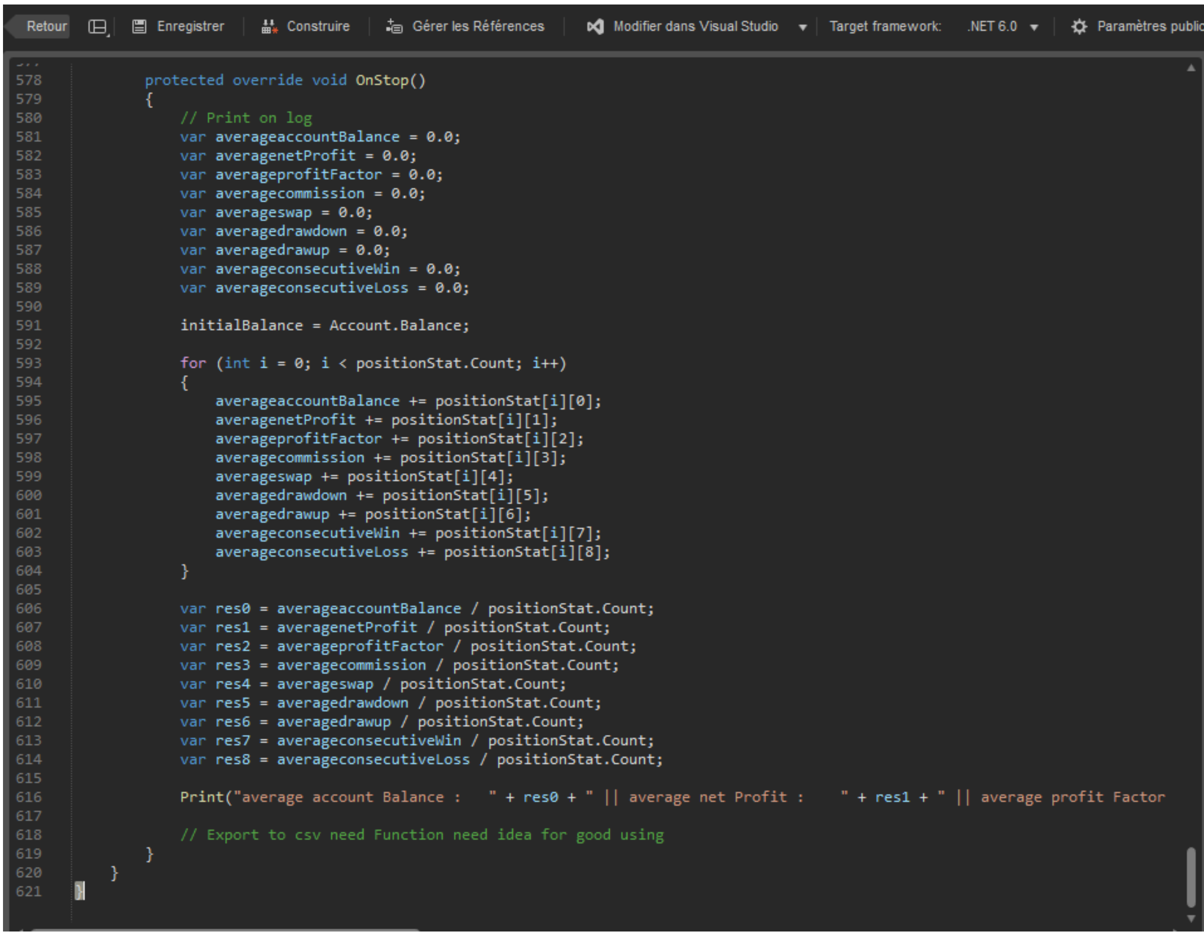
- Click on code

```
using System;
using cAlgo.API;
using cAlgo.API.Collections;
using cAlgo.API.Indicators;
using cAlgo.API.Internals;

namespace cAlgo.Robots
{
    [Robot(AccessRights = AccessRights.None, AddIndicators = true)]
    public class NNFXStrategieTesterv1021 : Robot
    {
        [Parameter(DefaultValue = "Hello world!")]
        public string Message { get; set; }

        protected override void OnStart()
        {
            // To learn more about cTrader Automate visit our Help Center:
            // https://help.ctrader.com/ctrader-automate

            Print(Message);
        }

        protected override void OnTick()
        {
            // Handle price updates here
        }

        protected override void OnStop()
        {
            // Handle cBot stop here
        }
    }
}
```

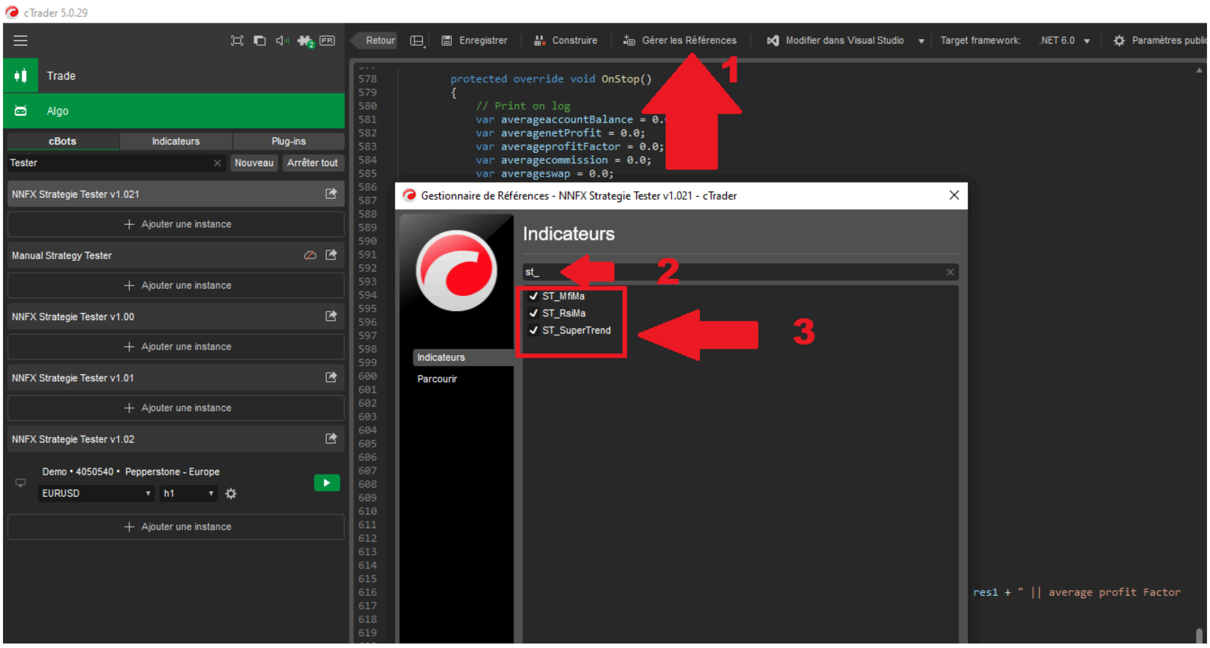
- Press `Ctrl + A` (to select all)

```
using System;
using cAlgo.API;
using cAlgo.API.Collections;
using cAlgo.API.Indicators;
using cAlgo.API.Internals;

namespace cAlgo.Robots
{
    [Robot(AccessRights = AccessRights.None, AddIndicators = true)]
    public class NNFXStrategieTesterv1021 : Robot
    {
        [Parameter(DefaultValue = "Hello world!")]
        public string Message { get; set; }

        protected override void OnStart()
        {
            // To learn more about cTrader Automate visit our Help Center:
            // https://help.ctrader.com/ctrader-automate

            Print(Message);
        }

        protected override void OnTick()
        {
            // Handle price updates here
        }

        protected override void OnStop()
        {
            // Handle cBot stop here
        }
    }
}
```

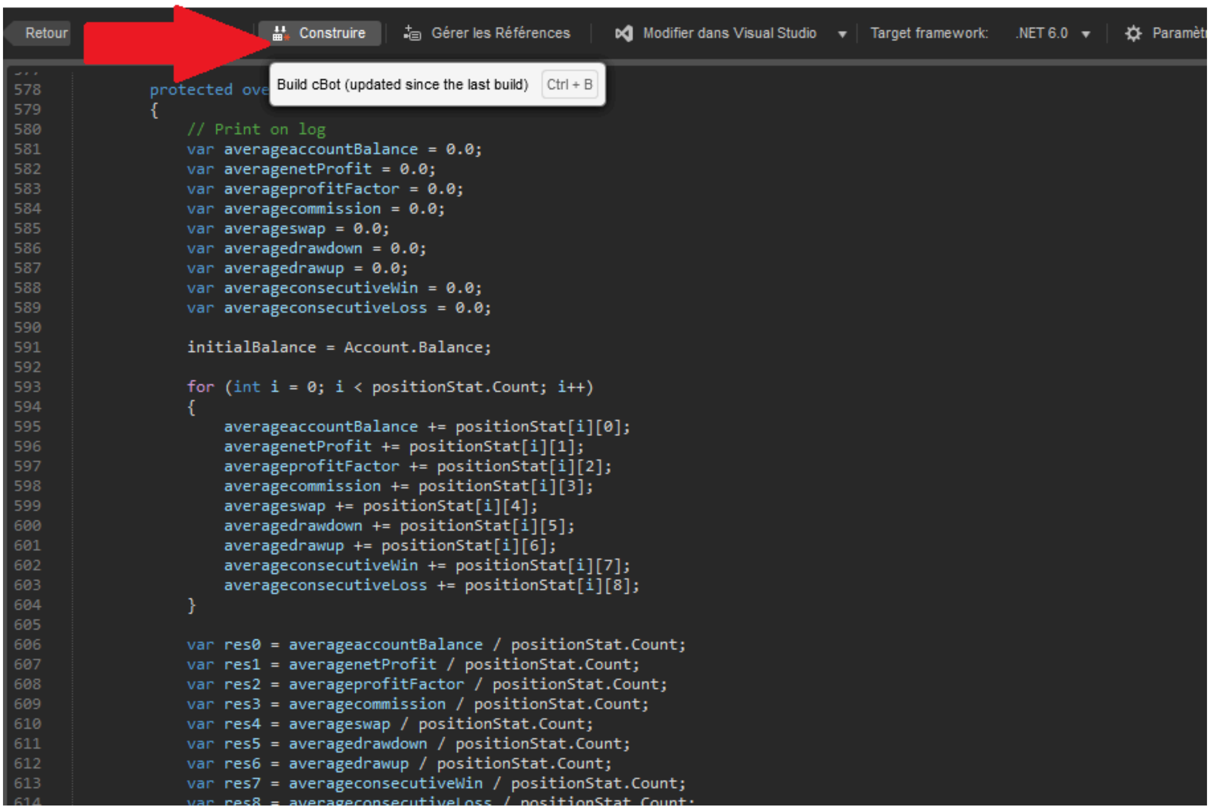
○ Press `Ctrl + V` (to paste)



14. Now that you have the code, click on Manage References.Select all the indicators necessary for this robot. Since we named them "ST_", enter "ST_" in the search bar and select all the listed indicators.



15. Finally, click Create. If you encounter an error, it’s possible that the old indicator code was not cleared before you pasted the new code, so restart from step 4 and go directly to step 6.



<u>Congratulations, you did it!</u>