

# IFunction

Introduced in build 9.1

## Description

The IFunction interface is used to implement custom Excel functions in .NET to be passed in to ExcelWriter Application. You can register your custom logic to a specific Excel Function by using the [Workbook.RegisterCustomFunction](#) method.

### C#

```
public interface IFunction
{
    FunctionValue Calculate(IList<FunctionValue> args, Cell currentCell);
}
```

### vb.net

```
Public Interface IFunction
    Function Calculate(args As IList(Of FunctionValue), currentCell As Cell) As
FunctionValue
End Interface
```

## Examples

## C#

```
class Concat : IFunction
{
    //Your custom function should be called Calculate and have the same signature as the
    sample function below
    public FunctionValue Calculate(ICollection<FunctionValue> args, Cell currentCell)
    {
        //Implement the custom logic you want to calculate your custom/overridden Excel
        function
        StringBuilder result = new StringBuilder("");
        foreach (FunctionValue arg in args)
        {
            result.Append(',');
            switch (arg.Type)
            {
                case FunctionValueType.BOOLEAN:
                    result.Append((bool)arg.Value ? "True" : "False");
                    break;
                case FunctionValueType.STRING:
                    result.Append(arg.Value);
                    break;
                case FunctionValueType.NUMBER:
                    result.Append(((double)arg.Value).ToString("0.#####"));
                    break;
                case FunctionValueType.NULL:
                    result.Append(' ');
                    break;
                case FunctionValueType.RANGE:
                    result.Append(arg.Value.ToString());
                    break;
            }
        }

        //Must return a FunctionValue object
        return new FunctionValue(result.ToString());
    }
}
```

## vb.net

```
Class Concat
    Implements IFunction

    //Your custom function should be called Calculate and have the same signature as the
    sample function below
    Public Function Calculate(args As IList(Of FunctionValue), currentCell As Cell) As
    FunctionValue
        //Implement the custom logic you want to calculate your custom/overridden
    Excel function
        Dim result As New StringBuilder("")
    For Each arg As FunctionValue In args
        result.Append(",")

        Select Case arg.Type
            Case FunctionValueType.[BOOLEAN]
                result.Append(If(CBool(arg.Value), "True", "False"))
            Exit Select
            Case FunctionValueType.[STRING]
                result.Append(arg.Value)
            Exit Select
            Case FunctionValueType.NUMBER
                result.Append(CDbl(arg.Value).ToString("0.#####"))
            Exit Select
            Case FunctionValueType.NULL
                result.Append(" ")
            Exit Select
            Case FunctionValueType.RANGE
                result.Append(arg.Value.ToString())
            Exit Select
        End Select
    Next

    //Must return a FunctionValue object
    Return New FunctionValue(result.ToString())
End Function
End Class
```

## Methods

Name	Description
<a href="#">Calculate(IList&lt;FunctionValue&gt;,Cell)</a>	Calculates a custom function returning a <a href="#">FunctionValue</a> given a list of <a href="#">FunctionValue</a> arguments and the function's current <a href="#">Cell</a> for context.