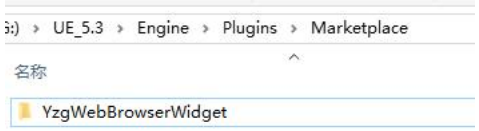
YzgWebBrowser

!NOTE: The YzgBrowser plugin support Windows.

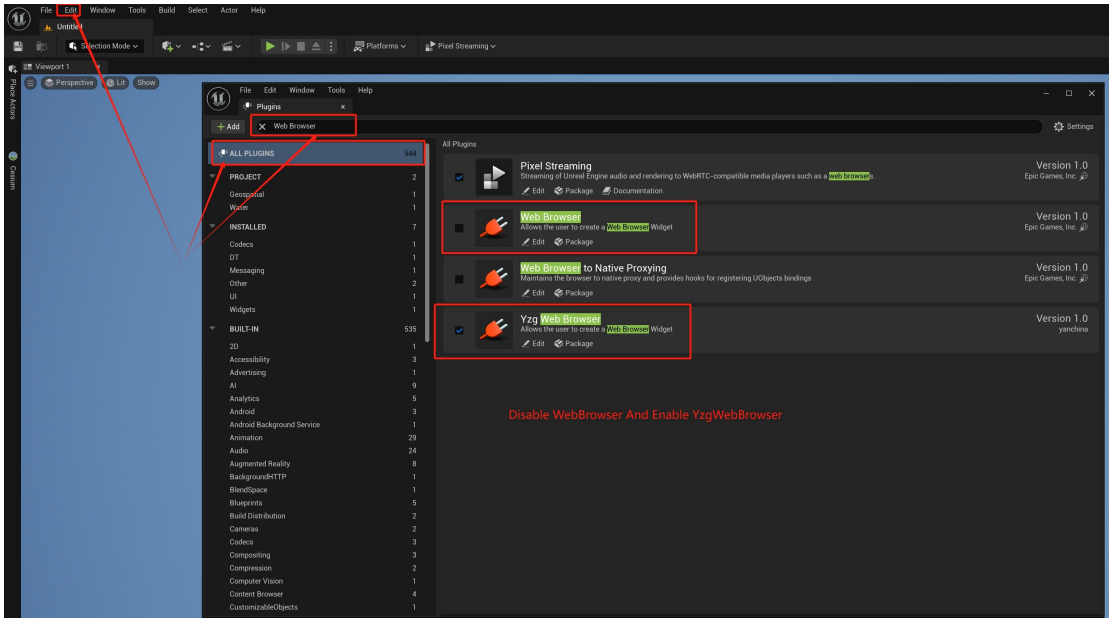
INSTALL

1、Installed YzgWebBrowser From Epic Games Launcher for your engine version.

2、Plugin Path Need In Engine Plugin Path.

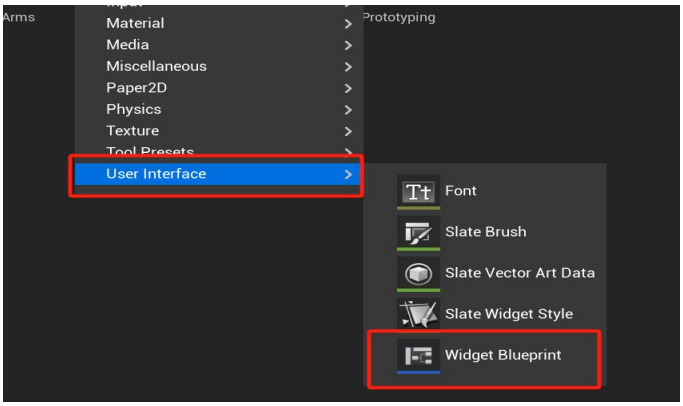


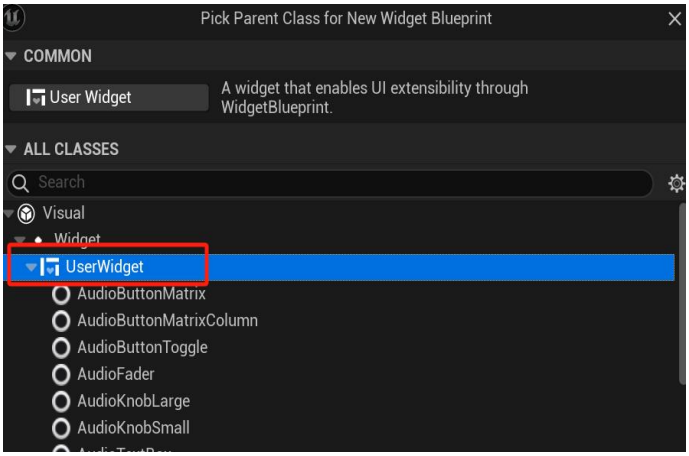
3、Open your project and go to the Plugins option drop-down. Then enable the YzgWebBrowser plugin and disable WebBrowser,then restart editor.



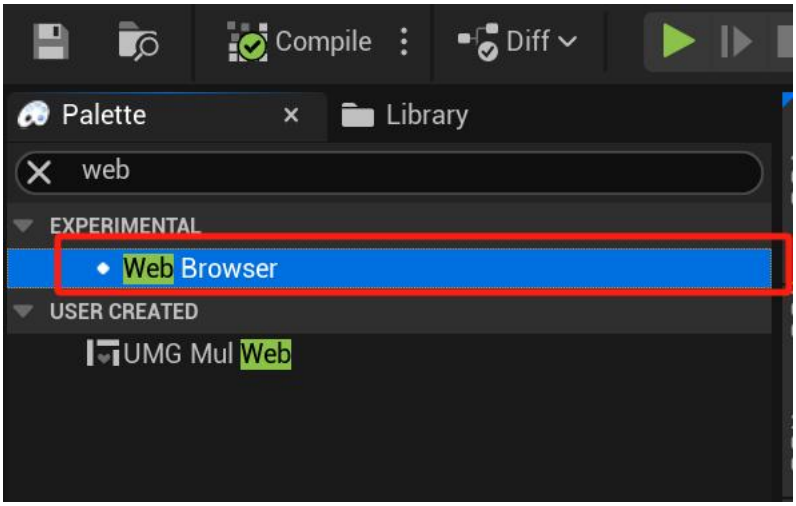
How To Use

1、Create a custom user widget.

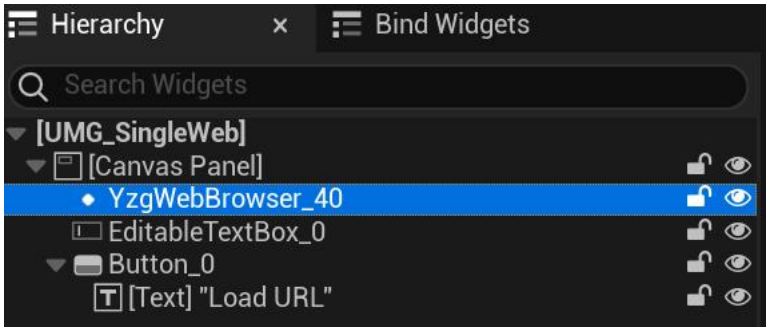


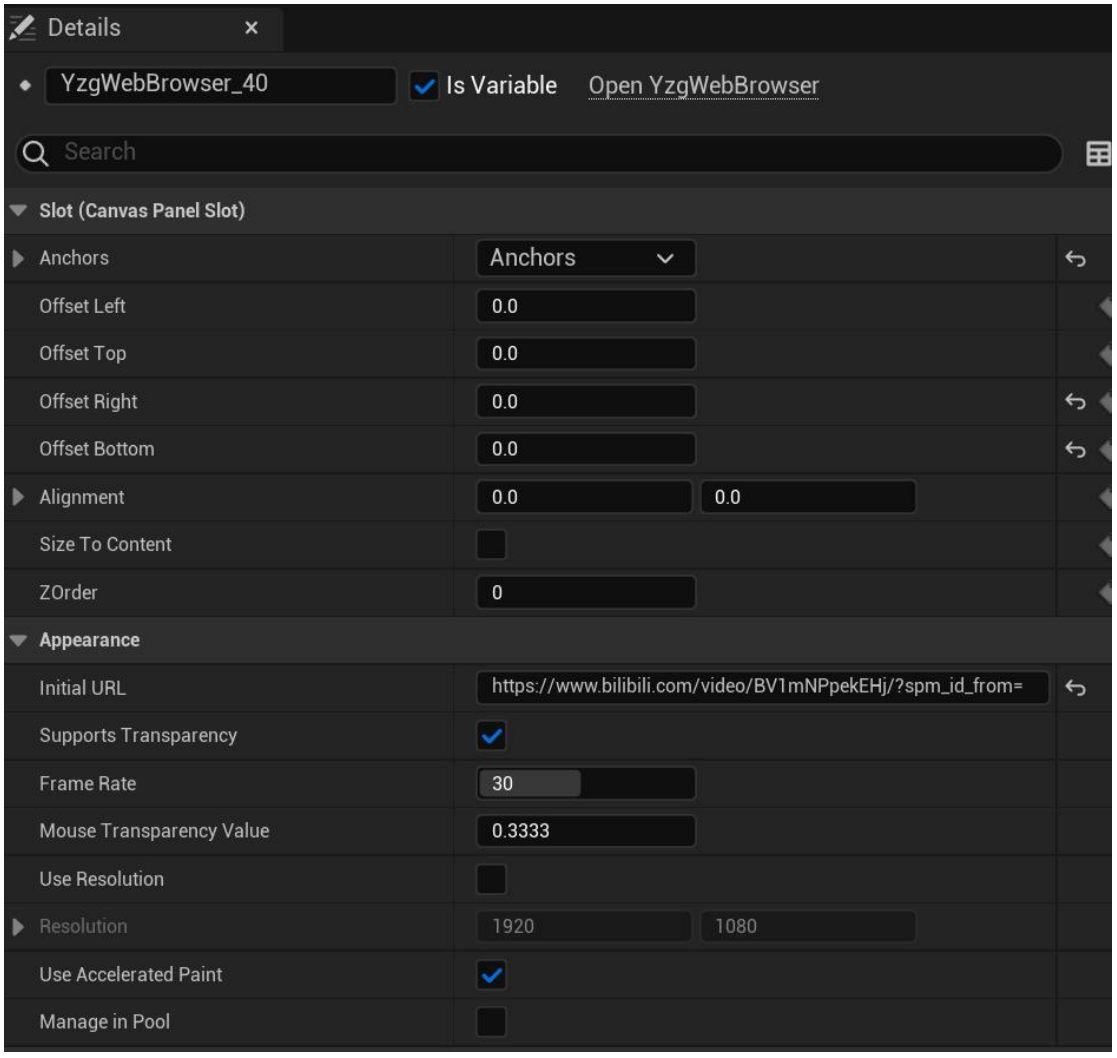


2、Open the widget Blueprint,Drag Web Browser Into widget panel.

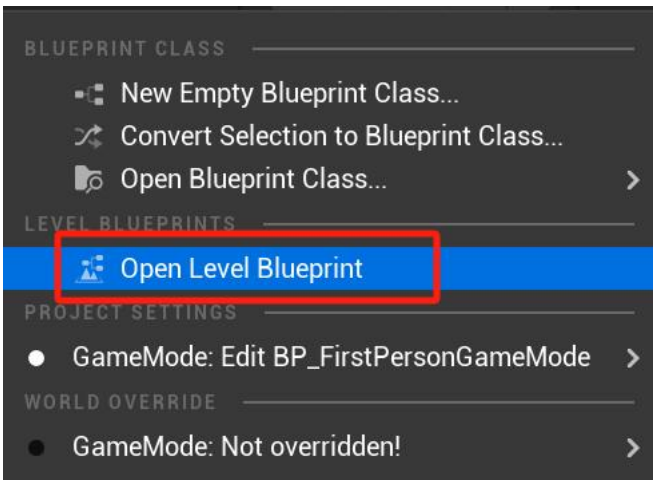


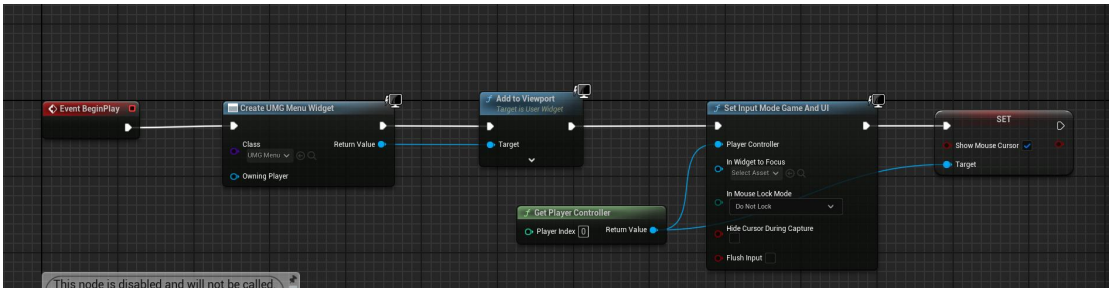
3、select YzgWebBrowser,then set URL in Details Panel.



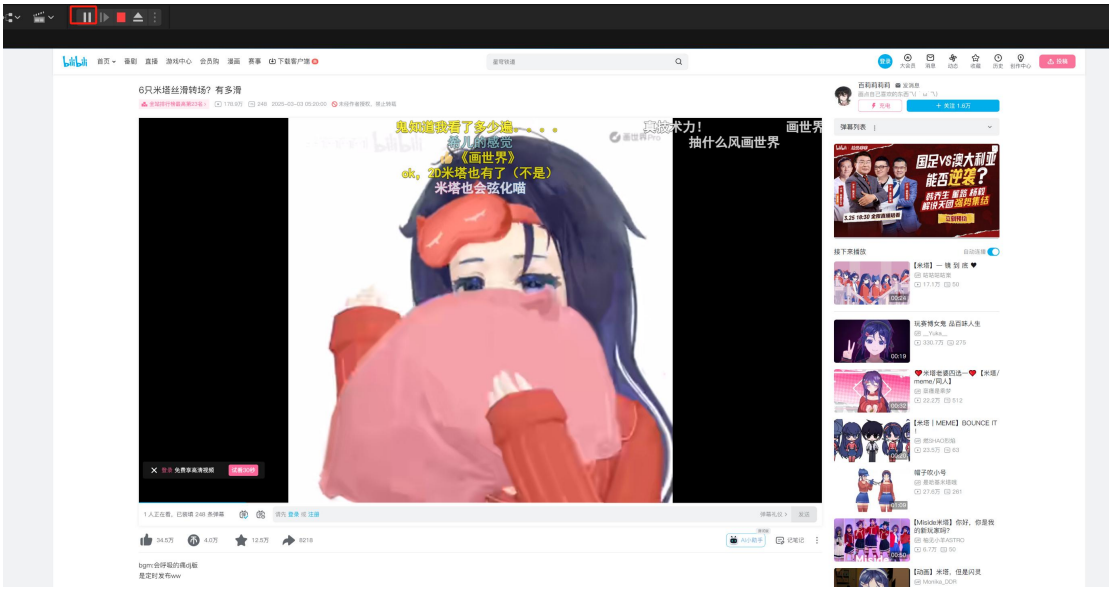


4、Open Level Blueprint, Add the Widget to viewport.





5、Play In Editor.



Details

Initial URL： URL that the browser will initially navigate to. The URL should include the protocol, eg http:// .

Support Transparency ： Should the browser window support transparency.

MouseTransparencyValue: Mouse Click Transparency Value.

Frame rate: browser window Frame Rate,Value Range 1 - 60.

UseAcceleratedPaint: Whether to enable GPU acceleration.

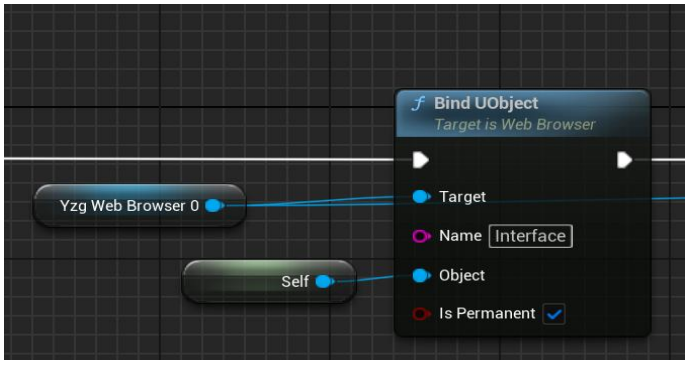
UseResolution: Whether to enable custom render sizes

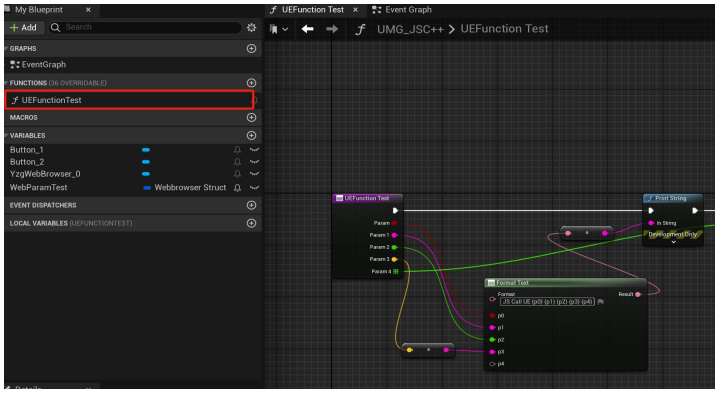
Resolution: Customize the render size

Manage In Pool: Whether to use pool to manage WebBrowser lifetimes

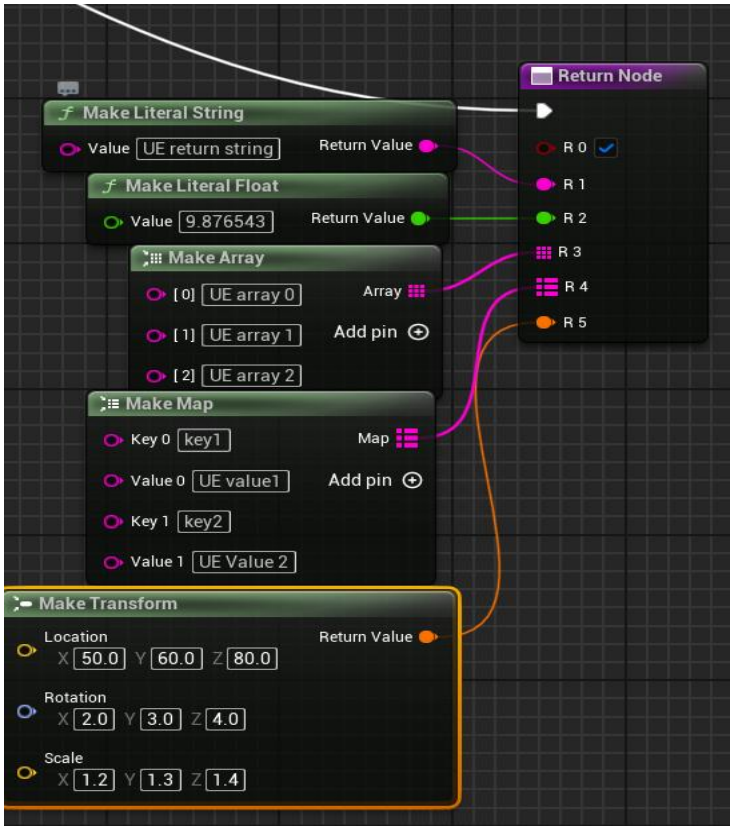
UE And JS Interaction

1、Use “BindUObject” Function To Register UE Object To JS. Then you can Direct Call UE function In JS.





UE Function can return valus, In JS You can use a function param to receive UE Return Values.

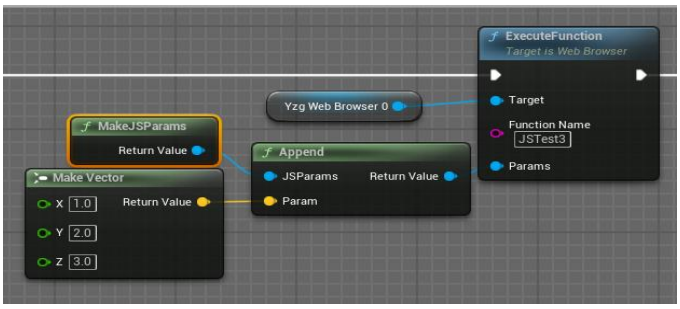


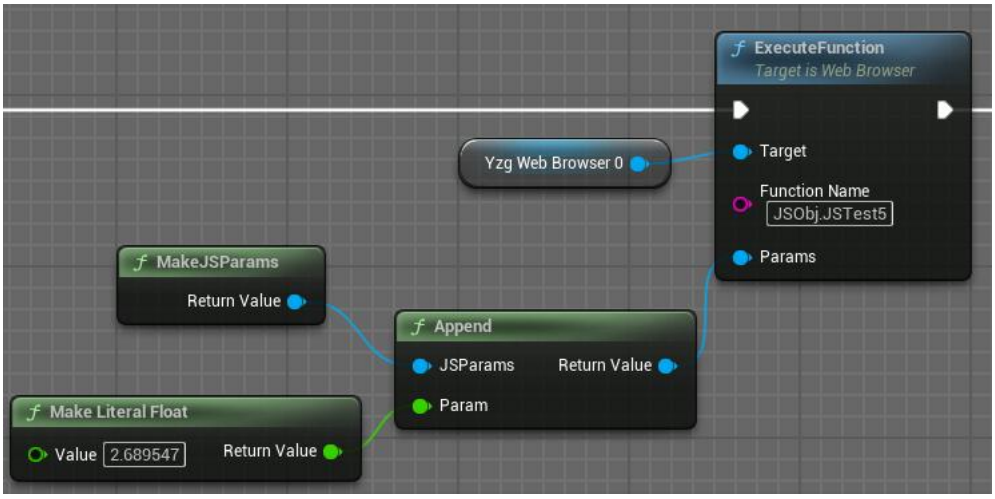

```
function CallUE() {
    var vec = {};
    vec.X = 2.365;
    vec.Y = 3.569;
    vec.Z = 4.13;

    var arr = [7.456,8.889,9.963];

   ue.Interface.UEFunctionTest(true,"js String",5.5689,vec, arr, function(r0,r1,r2,r3,r4,r5) {
        console.log("UE return call back");

        console.log("UE return Param0 " + r0);
        console.log("UE return Param1 " + r1);
        console.log("UE return Param2 " + r2);
        console.log("UE return Param3 " + r3);
        console.log("UE return Param4 " + r4);
        console.log("UE return Param4 " + r4.key1 + " " + r4.key2);
        console.log("UE return Param5 " + r5 +" " + r5.Translation.X + " " +r5.Translation.Y + " "  + " " + r5.Rotation.Y+ " " + r5.Scale3D.Z);

   });
}
```

2、UE Call JS

Use “ExecuteFunction” Call JS Function.

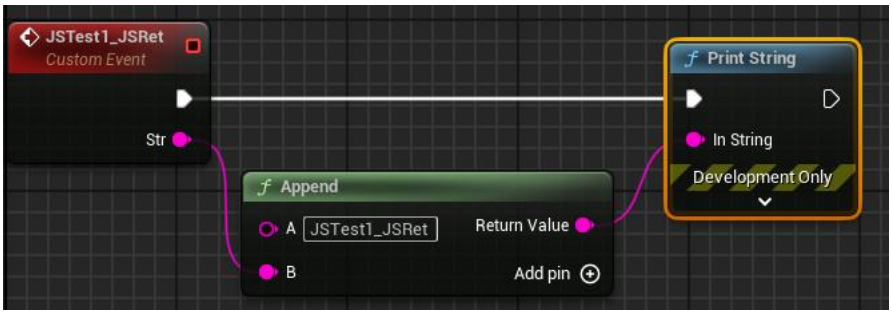
JS Function name can like [JSObj].[FunctionName] Or [FunctionName]





JS Function return value, you can use a UE function to recieve, UE Function Name Must [JSFunctionName]_JSRet like this.

```
function JSTest1(param)
{
    console.log("call JSTest1, return string == " + " Param String " + param);
    return "Call JS return string";
}
```

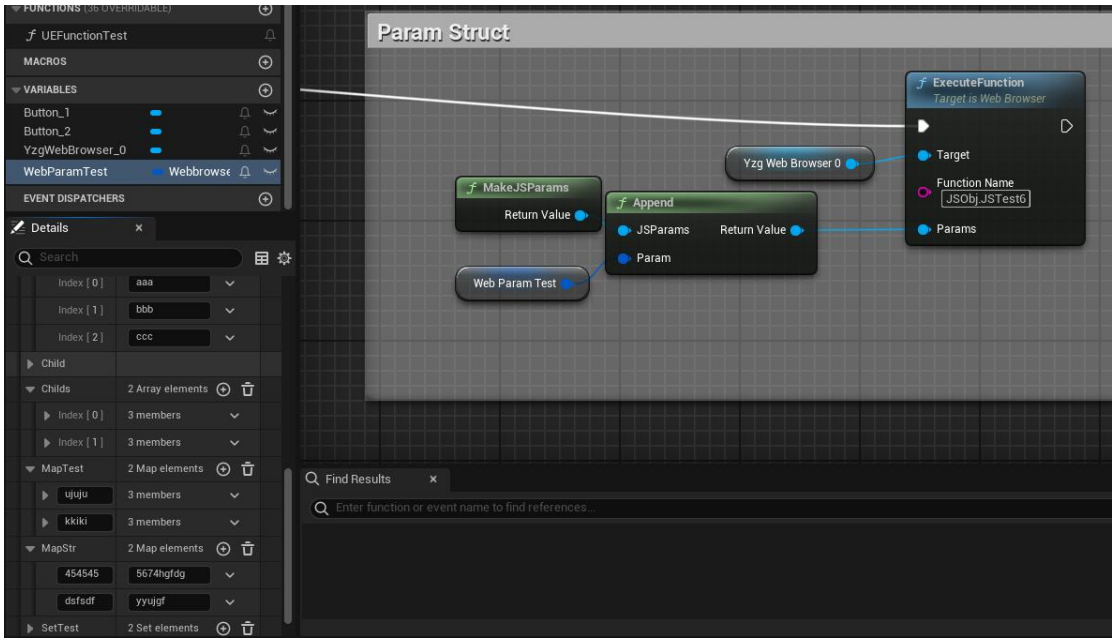


Value Type Convert

You can use “Make JS Params” Node to Build JS Params, UE Value Type Can Convert to JS Value Type Direct.

UE String、int32、int64、float 、Name、Text and Struct can convert to JS Value Type. JS value type can also convert to UE value type Too.

- Support UE Base Value Type And JS Base Value Type Convert.
- Support UE Map and JS Object Convert.
- Support UE Array And JS Array Convert.
- Support UE Set And JS Array Convert.
- Support UE Native Struct And Blueprint Struct.




```
function JSTest6(param)
{
    // UE Transfoirm Serialize Is Translation  Rotation And Scale3D, Rotation Save As Quaternion
console.log("============================ JSTest6 ========================" + param);

    console.log("call JSTest6 blueprint Struct  "
        + "\nParam=\n Str="
        + param.Str + "\n Float="
        + param.Float + "\n Name="
        + param.Name + "\n Text="
        + param.Text + "\n Integer="
        + param.Integer + "\n ArrayStr="
        + param.ArrayStr[0] + " " + param.ArrayStr[1] + " " + param.ArrayStr[2] + " " + param.ArrayStr[3] + "\n Child=\n   "
        + param.Child.Str + " " + param.Child.Float + " " + param.Child.Array[0] + " " + param.Child.Array[1] + "\n Childs=   "
        + param.Childs[0].Str + " " + param.Childs[0].Float + " " + param.Childs[0].Array[0] + " " + param.Childs[0].Array[1]
          + " " + param.Childs[0].Array[2] + "\n   "
        + param.Childs[1].Str + " " + param.Childs[1].Float + " " + param.Childs[1].Array[0] + " " + param.Childs[1].Array[1] + " "
          + param.Childs[1].Array[2] + " " + param.Childs[1].Array[3] + "\n MapTest=\n   Key=ujuju"
        + param.MapTest["ujuju"].Str + " " + param.MapTest["ujuju"].Float + " " + param.MapTest["ujuju"].Array[0] + " "
          + param.MapTest["ujuju"].Array[1] + "\n   Key=kkiki"
        + param.MapTest["kkiki"].Str + " " + param.MapTest["kkiki"].Float + " " + param.MapTest["kkiki"].Array[0] + " "
          + param.MapTest["kkiki"].Array[1] + " " + param.MapTest["kkiki"].Array[2] + "\n MapStr="
        + param.MapStr["454545"] + " " + param.MapStr["dsfsdf"] + "\n SetTest=\n   "
        + param.SetTest[0].Translation.X + " " + param.SetTest[0].Translation.Y + " " + param.SetTest[0].Translation.Z + "\n   "
        + param.SetTest[0].Rotation.X + " " + param.SetTest[0].Rotation.Y + " " + param.SetTest[0].Rotation.Z + "\n   "
        + param.SetTest[0].Scale3D.X + " " + param.SetTest[0].Scale3D.Y + " " + param.SetTest[0].Scale3D.Z + "\n   "
        + param.SetTest[1].Translation.X + " " + param.SetTest[1].Translation.Y + " " + param.SetTest[1].Translation.Z + "\n   "
        + param.SetTest[1].Rotation.X + " " + param.SetTest[1].Rotation.Y + " " + param.SetTest[1].Rotation.Z + "\n   "
        + param.SetTest[1].Scale3D.X + " " + param.SetTest[1].Scale3D.Y + " " + param.SetTest[1].Scale3D.Z + "\n SetStr="
        + param.SetStr[0] + " " + param.SetStr[1]
        );
```

```
    var val = {};
    val.Str = "jsstr";
    val.Float = 1.256;
    val.Name = "jsName";
    val.Text = "jsText";
    val.Integer = 2548;
    val.ArrayStr = ["js1", "js2","js3","js4","js5"];
    val.Child = {};
    val.Child.Str = "Child.Str";
    val.Child.Float = 3.568;
    val.Child.Array = ["Childjs1", "Childjs2","Childjs3","Childjs4","Childjs5"
    val.Childs = [val.Child,val.Child,val.Child];

    val.MapTest = {};
    val.MapTest["eee"] = val.Child;
    val.MapTest["yyy"] = val.Child;
    val.MapTest["iii"] = val.Child;

    val.MapStr = {};
    val.MapStr["xxx"] = "yyy";
    val.MapStr["aaa"] = "bbb";
    val.MapStr["ccc"] = "ddd";
    // UE Transform
    var element = {};
    element.Translation = {};
    element.Translation.X = 1,
    element.Translation.Y = 2;
    element.Translation.Z = 3;
    element.Rotation = {};
    element.Rotation.X = 4,
    element.Rotation.Y = 5;
    element.Rotation.Z = 6;
    element.Scale3D = {};
    element.Scale3D.X = 7,
    element.Scale3D.Y = 8;
    element.Scale3D.Z = 9;
    // UE Set Value
    val.SetTest = [element,element,element];
    val.SetStr = ["xxx","ffff","kkkk"];

    return val;
}
```