SRP_Json GetValue

Returns a descendant value within a given JSON Entity.

Syntax

```
Result = SRP_Json(Handle, "GetValue", Path, Default)
```

Returns

A descendant's value if it was found, "" if not.

Parameters

Parameter	Description
Handle	Handle to a JSON Entity. **Required.**
Path	A formatted string indicating the desired descendant. **Required.**
Default	An alternative value to return if the descendant's value is *null*. **Optional.** If this argument is ommitted, then a *null* value will return "<null>".

Remarks

The GetValue service returns the value of a descendant entity somewhere within the current JSON Entity Handle. If the descendant is an object or array, then you'll get the entire JSON string for that entity. Otherwise, you'll get the element's value without extraneous formatting (like quotes around strings).

The term descendant is intentional because you can extract members or elements deep within the JSON tree in a single call. Let's use the following JSON as an example:

```
{"menu": {
    "header": "SVG Viewer",
    "items": [
        {"id": "Open"},
        {"id": "OpenNew", "label": "Open New"},
        null,
        {"id": "ZoomIn", "label": "Zoom In"},
        {"id": "ZoomOut", "label": "Zoom Out"},
        {"id": "OriginalView", "label": "Original View"},
        null,
        {"id": "Quality"},
        {"id": "Pause"},
        {"id": "Mute"},
        null,
        {"id": "Find", "label": "Find..."},
        {"id": "FindAgain", "label": "Find Again"},
        {"id": "Copy"},
        {"id": "CopyAgain", "label": "Copy Again"},
        {"id": "CopySVG", "label": "Copy SVG"},
        {"id": "ViewSVG", "label": "View SVG"},
        {"id": "ViewSource", "label": "View Source"},
        {"id": "SaveAs", "label": "Save As"},
        null,
        {"id": "Help"},
        {"id": "About", "label": "About Adobe CVG Viewer..."}
    ]
}}
```

In this example, we have a root object with a single child object called "menu" as it's only member. That object contains a member called items, which is an array, and each element in the array is either null or a child object. As you can see, it would be cumbersome to extract a single descendant as it would require getting a child handle, followed by another child handle, and so on. That is why the GET service supports a path.

The path is a specially formatted string that instructs the service how to drill down into the JSON. If you pass a single member name or index, then you'll get the JSON string for that object. For example, we can get the "menu" object like so:

```
MenuJSON = SRP_Json(ObjectHandle, "GetValue", "menu")
```

If you want to get a child of menu, you separate each member name with a period, like so:

```
ItemsJSON = SRP_Json(ObjectHandle, "GetValue", "menu.items")
```

If you want to get an element within an array, you use square brackets instead of periods. This is how we drill down directly to the twelfth element within the items array:

```
ItemJSON = SRP_Json(ObjectHandle, "GetValue", "menu.items[12]")
```

And this is how we get the handle to the "label" member of the second item in the items array:

```
Label = SRP_Json(ObjectHandle, "GetValue", "menu.items[2].label")
```

Important. The Path parameter is a convenient way to drill into a JSON document. It reserves five characters: period, left square bracket, right square bracket, left angle bracket, and right angle bracket. If the member you want to access uses one of these characters, you must escape them by doubling up. Here's an example of a JSON document with periods in the member names and how to access them.

```
{
    "first.name" : "John"
    "last.name"  : "Doe"
}


// INCORRECT: This will get you "" because it will try to get "first" and then "name"
Name = SRP_Json(ObjectHandle, "GetValue", "first.name")


// CORRECT: This will get you "first.name"
Name = SRP_Json(ObjectHandle, "GetValue", "first..name")
```


This service always returns JSON Strings or values. If you want to get the JSON Entity Handle, you need to use Get instead.

Example

```
// Test parsing an existing JSON string and then drilling down into specific values
SampleJSON  = '{"menu": {'
SampleJSON := '    "header": "SVG Viewer",'
SampleJSON := '    "items": ['
SampleJSON := '        {"id": "Open"},'
SampleJSON := '        {"id": "OpenNew", "label": "Open New"},'
SampleJSON := '        null,'
SampleJSON := '        {"id": "ZoomIn", "label": "Zoom In"},'
SampleJSON := '        {"id": "ZoomOut", "label": "Zoom Out"},'
SampleJSON := '        {"id": "OriginalView", "label": "Original View"},'
SampleJSON := '        null,'
SampleJSON := '        {"id": "Quality"},'
SampleJSON := '        {"id": "Pause"},'
SampleJSON := '        {"id": "Mute"},'
SampleJSON := '        null,'
SampleJSON := '        {"id": "Find", "label": "Find..."},'
SampleJSON := '        {"id": "FindAgain", "label": "Find Again"},'
SampleJSON := '        {"id": "Copy"},'
SampleJSON := '        {"id": "CopyAgain", "label": "Copy Again"},'
SampleJSON := '        {"id": "CopySVG", "label": "Copy SVG"},'
SampleJSON := '        {"id": "ViewSVG", "label": "View SVG"},'
SampleJSON := '        {"id": "ViewSource", "label": "View Source"},'
SampleJSON := '        {"id": "SaveAs", "label": "Save As"},'
SampleJSON := '        null,'
SampleJSON := '        {"id": "Help"},'
SampleJSON := '        {"id": "About", "label": "About Adobe CVG Viewer..."}'
SampleJSON := '    ],'
SampleJSON := '    "testbool": "true",'
SampleJSON := '    "testint": "1234567890",'
SampleJSON := '    "testreal": "1234567890.987654321"'
SampleJSON := '}}'

 // Parse it, then grab some values
If SRP_Json(ObjectHandle, "Parse", SampleJSON) EQ "" then
    TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.items[1].id")
    TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.items[12].label")
    TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.items[3]")
    TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.testbool")
    TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.testint")
    TestValue = SRP_Json(ObjectHandle, "GetValue", "menu.testreal")
    SRP_Json(ObjectHandle, "Release")
end
```

See Also

Get