

Paths

Several SRP JsonX routines rely on paths to reference elements within a document. Paths are specially formatted strings, akin to a mini query language, that tell SRP JsonX how to find an element. Paths are also returned when you call [SRP_JsonX_State](#) or [SRP_JsonX_Trace](#). Those are current paths, paths that point to the current element of each document.

Empty Path

The simplest path is an empty one.

```
Result = SRP_JsonX_Get()
```

Calling SRP_JsonX_Get with no path is like calling it with a path set to "". When a path is empty, SRP JsonX uses the current path.

Simple Path

The next simplest path is a single name or number. If the current element is an object, we can pass a name to reference a member in that object

```
Result = SRP_JsonX_Get("firstname")
```

If the current element is an array, we can pass an index into that array.

```
Result = SRP_JsonX_Get(2)
```

In either case, if the member name is not found or the index is out of bounds, null is returned.

Complex Path

The real power of the path is to drill deep into the document without having to manage handles or pointers. 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..." }
  ]
}
```

When we parse the above json, the current path points to the root of the document. So, all our paths will be relative to the root.

To get the entire menu, we use a simple path

```
Result = SRP_JsonX_Get("menu")
```

To get the member of a sub-object, separate each member name with a period. This example gets the items array of the menu object.

```
Result = SRP_JsonX_Get("menu.items")
```

To get an array element, surround the array index with square brackets or angle brackets. This examples gets item 12 in the items array.

```
Result = SRP_JsonX_Get("menu.items[12]")
```

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

```
Result = SRP_JsonX_Get("menu.items[2].label")
```

Escaping Reserved Characters

The Path syntax 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_JsonX_Get("first.name")

// CORRECT: This will get you "first.name"
Name = SRP_JsonX_Get("first..name")
```