



JSON Formatting guide

Quick-reference information
to aid you with column and
view JSON formatting in
SharePoint

JSON Formatting Guide

1. Basics of JSON

JSON (JavaScript Object Notation) is a lightweight data interchange format. In SharePoint, JSON formatting is used to customize the appearance and behavior of lists and libraries.

2. Basic Structure

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/column-formatting.schema.json",
  "elmType": "div",
  "txtContent": "Hello, SharePoint!"
}
```

\$schema: Specifies the JSON schema for validation.

elmType: Defines the HTML element type.

txtContent: Sets the text content of the element.

3. Applying Formatting

Apply JSON formatting to a column by navigating to the column settings, selecting "Format this column," and pasting your JSON code.

4. Simple Conditional Formatting

Apply JSON formatting to a column by navigating to the column settings, selecting "Format this column," and pasting your JSON code.

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/column-formatting.schema.json",
  "elmType": "div",
  "txtContent": "=if(@currentField == 'High', 'Urgent', 'Normal')"
}
```

@currentField: Represents the value of the current field.

if(): Conditional statement to set content based on field value.

5. Icons and Images

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/column-formatting.schema.json",
  "elmType": "div",
  "children": [
    {
      "elmType": "img",
      "attributes": {
        "src": "=/_layouts/15/images/office365.png",
        "alt": "Office 365 Icon"
      }
    }
  ]
}
```

elmType: img: Embeds an image.

src: Specifies the image source.

Common JSON attributes used in SharePoint column formatting

attributes

Specifies additional attributes for the HTML element, such as class, data-*, or other HTML attributes

```
"attributes": {  
  "class": "custom-class",  
  "data-id": "= @currentField.id"  
}
```

children

Defines child elements within the parent HTML element. Used for creating complex structures.

```
"children": [  
  {  
    "elmType": "div",  
    "txtContent": "Child Element"  
  }  
]
```

class

Adds one or more CSS classes to the HTML element, allowing for more detailed styling.

```
"class": "custom-class"
```

customRowAction

Specifies a custom action to be performed when a user clicks on a row. Used for linking to custom actions or workflows.

```
"customRowAction": {  
  "action": "navigate",  
  "actionParams": "= @currentField.url"  
}
```

debugMode

Enables or disables debug mode, providing additional information in the browser's console for troubleshooting.

```
"debugMode": true
```

elmType

Defines the HTML element type to be created in the column. Specifies whether it's a div, span, a (link), or other HTML element.

```
"elmType": "div"
```

iconName

Specifies the Office UI Fabric icon to be displayed alongside the field value.

```
"iconName": "OfficeLogo"
```

if()

Conditional statement that allows you to apply formatting based on a specified condition.

```
"=if(@currentField == 'High', 'Urgent', 'Normal')"
```

operators

Allows you to use operators like ==, !=, >, < in conditional expressions.

```
"=if(@currentField > 100, 'High', 'Low')"
```

\$schema

Specifies the JSON schema for validation. It defines the structure and elements that are allowed in the JSON code.

```
"$schema":  
"https://developer.microsoft.com/json-schemas/sp/column-formatting.schema.json"
```

style

Defines the style (CSS) for the HTML element, allowing you to customize its appearance.

```
"style": {  
  "color": "red",  
  "font-size": "16px"  
}
```

txtContent

Sets the text content of the HTML element. It can be a static string or a dynamic expression.

```
"txtContent": "Hello, SharePoint!"
```

Common JSON attributes used in SharePoint column formatting

columnFormatterReference

will be replaced with the referenced column's formatter JSON

```
"columnFormatterReference": "[$FieldName]"
```

displayName

Specifies a custom display name for the column. Useful for providing a user-friendly name.

```
"displayName": "Status"
```

[\$FieldName]

Reference another field in the view

```
"txtContent": "[$FieldName.displayValue]"
```

inlineEditField

Adds the field editor for the referenced column

```
"inlineEditField": "[$FieldName]",
```

@currentField

Will evaluate to the value of the current field

```
"txtContent": "@currentField"
```

displayValue

Get the default rendered value, based on the column setting

```
"txtContent": "[$FieldName.displayValue]"
```

filePreviewProps

Override the default styles of file type icon and brand type icon in filepreview elmType

```
"elmType": "filepreview",
"attributes": {
  "src": "@thumbnail.medium",
},
"filePreviewProps": {
  "fileTypeIconClass": "sp-css-borderColor-neutralLight",
  "fileTypeIconStyle": {
    "width": "100px"
  },
  "brandTypeIconClass": "sp-css-borderColor-neutralLight",
  "brandTypeIconStyle": {
    "width": "68px"
  }
}
```

customCardProps

Applies custom properties to the card view of a list. Allows for more detailed customization of card views.

```
"customCardProps": {
  "formatter": {
    "elmType": "div",
    "children": [
      {
        "elmType": "p",
        "txtContent": "@currentField"
      }
    ]
  }
}
```

forEach

Used with the "elmType" attribute to iterate over each item in an array and render the specified elements for each item.

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/column-formatting.schema.json",
  "elmType": "div",
  "txtContent": "Skills:",
  "children": [
    {
      "elmType": "div",
      "forEach": "skill in @currentField",
      "txtContent": "[$skill]"
    }
  ]
}
```

@isSelected

Whether this item is selected in the view

```
"additionalRowClass": "=if(@isSelected, 'sp-css-backgroundColor-themeLighter', '')"
```

Common JSON attributes used in SharePoint view formatting

additionalRowClass

Adds additional CSS classes to the entire row, allowing for customized styling based on row data.

```
"additionalRowClass": "=if(@currentField == 'High', 'urgent-row', 'normal-row')"
```

customRowAction

Similar to column formatting, specifies a custom action to be performed when a user clicks on a row.

```
"customRowAction": {  
  "action": "navigate",  
  "actionParams": "= @currentField.url"  
}
```

fillHorizontally

Makes a tile in the tile layout stretch to fill the row.

```
"fillHorizontally": true
```

headerFormatter

Defines a formatter for the entire header row, allowing for custom styling or content.

```
"headerFormatter": {  
  "elmType": "div",  
  "txtContent": "Header Text",  
  "style": {  
    "color": "red"  
  }  
}
```

height

Sets the height of a tile in the tile layout.

```
"height": "200px"
```

hideColumnHeader

Hides the column header for the specified column. Useful for creating a cleaner, more focused view.

```
"hideColumnHeader": true
```

hideFooter

Hides the entire footer section, which typically contains pagination controls.

```
"hideFooter": true
```

rowFormatter

Defines a nested JSON structure to format the entire row. It allows for more complex row-level formatting.

```
"rowFormatter": {  
  "elmType": "div",  
  "style": {  
    "background-color": "=if(@currentField == 'Completed', 'green', 'red')"  
  },  
  "children": [  
    {  
      "elmType": "p",  
      "txtContent": "@currentField"  
    }  
  ]  
}
```

hideSelection

Hides the selection checkbox for list items. Useful for creating more visually streamlined views.

```
"hideSelection": true
```

\$schema

Specifies the JSON schema for validation, similar to column formatting. It defines the structure and elements allowed in the JSON code for view formatting.

```
"$schema":  
"https://developer.microsoft.com/json-schemas/sp/view-formatting.schema.json"
```

width

Sets the width of a tile in the tile layout.

```
"width": "200px"
```

Common Operators used in SharePoint JSON formatting

Operators

+	length
-	abs
/	loopIndex
*	floor
<	ceiling
>	pow
%	substring
==	getDate
!=	getMonth
<=	getYear
>=	toUpperCase
	lastIndexOf
&&	startsWith
toString()	endsWith
Number()	replace
Date()	replaceAll
cos	padStart
sin	padEnd
?	getUserImage
:	addDays
toLocaleString()	addMinutes
toLocaleDateString()	appendTo
toLocaleTimeString()	removeFrom
indexOf	split
toLowerCase	
join	

Binary Arithmetic Operators

+
-
/
*
<
>
%
==
!=
<=
>=

Working With Date Values

toLocaleString()
Displays a date with full date and time

toLocaleDateString()
Displays just the date

toLocaleTimeString()
Displays just the time

```
"elmType": "div",  
"txtContent": {  
  "operator": "toLocaleString()",  
  "operands": ["@currentField"]  
}
```

Unary Operators (one parameter)

toString()

Returns a string representing an object.

"txtContent": "=toString(45)"

Date()

Returns a date/time object from the parameter.

"txtContent": "=Date('1/2/2023')"

toDateString()

Returns a date in a friendly format.

"txtContent": "=toDateString(@now)"

toLowerCase()

Converts the a to lower case.

"txtContent": "=toLowerCase('Hello, World!')"

Binary Operators (two parameters)

indexOf()

Searches for an instance of a string within another string. Returns the position.

```
"txtContent": "=indexOf('SharePoint', 'Point')"
```

lastIndexOf

Returns the position of the last instance of a string within another string.

```
"txtContent": "=indexOf('Apples Oranges Oranges', 'Oranges')"
```

getUserImage()

Returns the profile image of a user.

```
"src": "=getUserImage('user@organization.com', 'small')"
```

split()

Splits a string into an array by using another string as a separator

```
"txtContent": "=split('JSON Formatting Rocks', ' ')"
```

Ternary Operators (three parameters)

substring()

Returns part of a string, from a beginning index to the end index (starting at 0)

```
"txtContent": "=substring('SharePoint', 5, 6)" (returns 'P')
```

replace()

Replace part of a string or array with another value, and returns the result. Only the first instance of the substring is replaced.

```
"txtContent": "=replace('Hello world', 'world', 'everyone')"
```

replaceall()

Replace part of a string or array with another value, and returns the result. All instances of the substring are replaced.

```
"txtContent": "=replaceall('S-h-a-r-e-P-o-i-n-t', '-', '')"
```

split()

Splits a string into an array by using another string as a separator

```
"txtContent": "=split('JSON Formatting Rocks', ' ')"
```

Special Strings used in SharePoint JSON formatting

@currentField

Evaluates to the value (or object) of the current column being formatted.

"txtContent": "@currentField.title",

@me

Returns the email address of the currently logged on user

"txtContent": "@me"

@now

Returns the current date/time

"txtContent": "@now"

@window.innerHeight

Returns the height of the browser window (in pixels) at the time the list is displayed

@window.innerWidth

Returns the width of the browser window (in pixels) at the time the list is displayed

@thumbnail.small, @thumbnail.medium, and @thumbnail.large

Displays a thumbnail of an image in a library in a predefined size

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
  "elmType": "img",
  "attributes": {
    "src": "@thumbnail.small",
  }
  "style": {
    "width": "100%",
    "max-width": "100%"
  }
}
```

@thumbnail.<bounding size>

Displays a thumbnail of an image in a library in the specified size

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
  "elmType": "img",
  "attributes": {
    "src": "@thumbnail.200x400",
  }
  "style": {
    "width": "100%",
    "max-width": "100%"
  }
}
```

Using this guide

Print these pages out. Hang them on the wall. Keep them on your desk. Whereever you'd like to keep these, just have them handy. Don't try to memorize these attributes, although some you will just by repeatedly using them. What's more important is knowing what's possible with JSON formatting, and where to find information on the syntax and attributes (hint hint.. this guide!). You can always refer back to this guide.

Keep creating. Keep innovating. Keep being your best self.

Follow Me Online

- [Subscribe and Watch My Free YouTube Videos](#)
- [Follow me on LinkedIn](#)
- [Get exclusive content on Twitter](#)
- [Read my blog posts on my website](#)
- [Find out more about my online courses](#)