KNOWLEDGE BASE ARTICLE

Using SharePoint Column Formatting JSON in List Filter Online

2026 © Vitextra SIA. All rights reserved.

Disclaimer

For the purposes of this Disclaimer:

- **Company** (referred to as either "the Company", "We", "Us" or "Our" in this Disclaimer) refers to Vitextra SIA, LV-1009 Riga, Matise iela 61 - 25.
- **Document** refers to the Knowledgebase Article.
- **You** means the individual accessing the Document, or the company, or other legal entity on behalf of which such individual is accessing or using the Document, as applicable.
- **Website** refers to Vitextra, accessible from https://vitextra.com

Disclaimer

The information contained in the Document is for general information purposes only.

The Company assumes no responsibility for errors or omissions in the contents of the Document.

In no event shall the Company be liable for any special, direct, indirect, consequential, or incidental damages or any damages whatsoever, whether in an action of contract, negligence, or other torts, arising out of or in connection with the use of the contents of the Document. The Company reserves the right to make additions, deletions, or modifications to the contents in the Document at any time without prior notice.

Errors and Omissions Disclaimer

The information given by the Document is for general guidance on matters of interest only. Therefore, errors can occur even if the Company takes every precaution to ensure that the Document's content is current and accurate. Plus, given the changing nature of laws, rules, and regulations, there may be delays, omissions, or inaccuracies in the information contained in the Document.

The Company is not responsible for any errors or omissions, or for the results obtained from the use of this information.

Fair Use Disclaimer

The Company may use copyrighted material that the copyright owner has not explicitly authorized. The Company is making such material available for criticism, comment, news reporting, teaching, scholarship, or research.

The Company believes this constitutes a "fair use" of any such copyrighted material as provided for in section 107 of the United States Copyright law.

If You wish to use copyrighted material from the Document for your own purposes that go beyond fair use, You must obtain permission from the copyright owner.

"Use at Your Own Risk" Disclaimer

All information in the Document is provided "as is", with no guarantee of completeness, accuracy, timeliness or of the results obtained from the use of this information, and without warranty of any kind, express or implied, including, but not limited to warranties of performance, merchantability and fitness for a particular purpose.

The Company will not be liable to You or anyone else for any decision made or action taken in reliance on the information given by the Service or for any consequential, special or similar damages, even if advised of the possibility of such damages.

Contact Us

If you have any questions about this Disclaimer, You can contact Us by email: support@vitextra.com.

Table of contents

Introduction 4

Enabling column formatting 4

What is supported 5

- **Elements** 5
- **Styles** 5
- **Attributes** 6
- **Field tokens** 7
- **Expressions** 8

What is not supported 8

Examples 9

Performance 12

Terms 13

Introduction

Column formatting is a **feature** of **SharePoint Online** that changes how a column is displayed with a **JSON** description: a status shown as a colored pill, a number colored by its value, a person rendered as a link that starts an e-mail. The description is stored on the column itself, and **SharePoint** applies it in every **view** of the **list**.

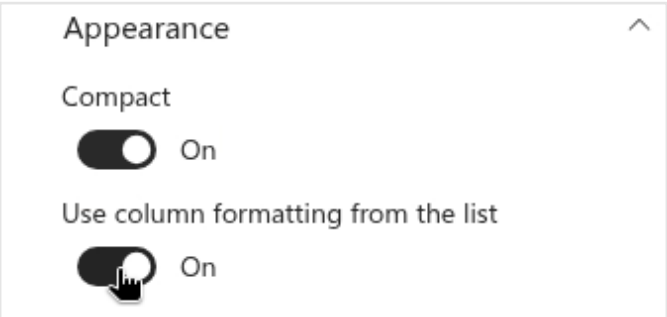
List Filter Online can honor these descriptions. When the option is enabled, the **web part** reads the column formatting JSON of each column that has one and renders the cells of the table from it, so the table on your page looks like the list it comes from. The web part implements the commonly used subset of the column formatting language; this article lists exactly what is supported, what is not, and what happens when a format goes beyond the subset.

Note

The web part applies column formatting only. View formatting - the JSON that formats rows or the whole view - is not read, because the web part does not use list views.

Enabling column formatting

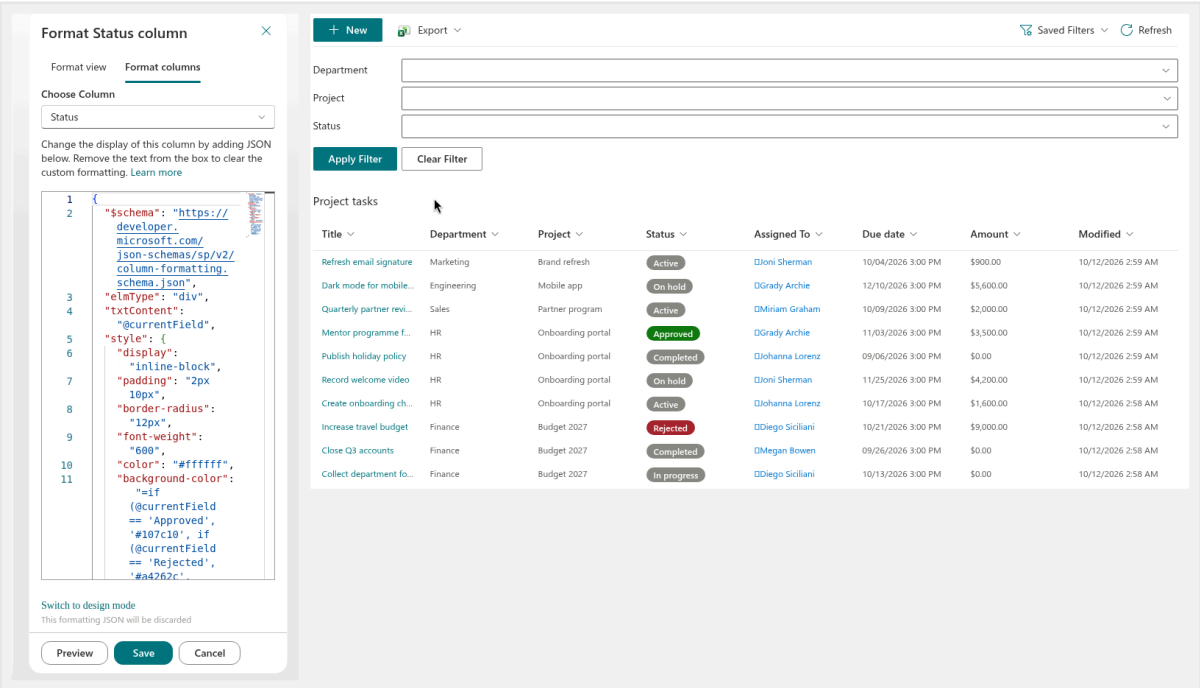
Edit the page, open the property pane of the web part and expand the **Appearance** group. Switch on **Use column formatting from the list**. The option is off by default.



The Use column formatting from the list option

The JSON itself is not entered in the web part. It comes from the definition of the column in the source list - the same definition that SharePoint edits when you open **Column settings** → **Format this column** in the list and save a format. The web part loads it together with the other properties of the columns, so a format set on a column of a list on another **site** is applied as well. Formats are per column: every column that carries a format is rendered from it, and the columns without a format

keep the display formats chosen in the column configurator (see KB10258. Configuring Columns in List Filter Online: Captions, Formats and Display Options).



A column format defined in the list and applied in the web part

When a format is applied to a column, it replaces the cell entirely, including the **Linked to item** option of that column. Put a link into the format itself if the value should stay clickable.

What is supported

Elements

A format is a tree of elements. Each element is an object with the following keys:

- `elmType` - one of `div`, `span`, `a`, `img`, `button` or `p`. Any other value, except the unsupported ones listed below, is rendered as a `div`.
- `txtContent` - the text of the element: a literal string, a field token or an expression.
- `style` - CSS properties of the element (see below).
- `attributes` - HTML attributes of the element (see below).
- `children` - an array of child elements, rendered in order.

Keys the web part does not know, such as `$schema` or `debugMode`, are ignored.

Styles

The `style` object may set the following CSS properties; every value can be a literal

or an expression:

`background`, `background-color`, `border`, `border-top`, `border-right`, `border-bottom`, `border-left`, `border-color`, `border-style`, `border-width`, `border-radius`, `box-shadow`, `color`, `cursor`, `direction`, `display`, `fill`, `flex-direction`, `flex-wrap`, `align-items`, `justify-content`, `font-family`, `font-size`, `font-style`, `font-weight`, `letter-spacing`, `line-height`, `text-align`, `text-decoration`, `text-overflow`, `text-transform`, `white-space`, `word-break`, `overflow`, `opacity`, `visibility`, `padding`, `padding-top`, `padding-right`, `padding-bottom`, `padding-left`, `margin`, `margin-top`, `margin-right`, `margin-bottom`, `margin-left`, `width`, `min-width`, `max-width`, `height`, `min-height`, `max-height`.

Properties outside this list - for example `position` or `z-index`, which could break the layout of the table - are skipped silently. A value that contains `url(` is skipped as well, so background images cannot be loaded through styles.

Attributes

Attribute	Elements	Behaviour
`class`	all	Sets the CSS class of the element. Class names of SharePoint, such as `sp-field-severity--good`, are accepted but have no styling on the page; use `style` for colors.
`title`	all	Sets the tooltip of the element.
`href`	`a`	The address of the link. Allowed are `http:`, `https:`, `mailto:`, `tel:` and relative addresses; any other scheme is dropped and the element is rendered without a link. Links get `rel="noreferrer"`.

Attribute	Elements	Behaviour
`target`	`a`	The target window of the link; `_blank` (a new tab) when not specified.
`src`, `alt`	`img`	The address and the alternative text of the image. The same address rules as for `href` apply; an image without a valid `src` is not rendered.
`iconName`	all	The name of a Fluent UI icon, such as `Mail` or `CheckMark`, inserted in front of the text of the element.

Other attributes, including `role`, `aria-*`, `d` and `data-interception`, are ignored.

Field tokens

Token	Value
`@currentField`	The value of the formatted column.
`@currentField.prop`	A property of the value: for a person `title`, `email`, `id`, `jobTitle`, `department` or `picture`; for a lookup `lookupValue` or `lookupId`; for a hyperlink `Url` or `Description`.
`[$Field]`	The value of another column of the same item, by its internal name.
`[$Field.prop]`	A property of that value, with the same property names.
`@me`	The e-mail address of the current user.
`@now`	The current date and time.

Tokens may stand alone (`"txtContent": "@currentField"`) or inside a longer string (`"txtContent": "Owner: [$Owner.title]"`). A multi-value column (several people, lookups or terms) is displayed as a comma-separated list; property access such as `[$Owner.email]` on a multi-value column returns the property of the first value. Person and lookup values display their name; dates display as a localized date.

Expressions

Expressions can be written as operator trees - objects with `operator` and `operands` - or as strings that start with `=`, using the Excel-like syntax of SharePoint. Both forms support the same set of operations:

- arithmetic: `+` (also string concatenation), `-`, `*`, `/`, `%`;
- comparison: `==`, `!=`, `>`, `<`, `>=`, `<=` - dates are compared by their time stamp;
- logic: `&&`, `||`, `!`, the conditional `? :` (in a tree, operator `?` or `:` with three operands) and the function `if(condition, whenTrue, whenFalse)`;
- conversion: `toString()`, `Number()`, `Date()`;
- numbers: `abs()`, `floor()`, `ceiling()`;
- strings: `toLowerCase()`, `toUpperCase()`, `indexOf()`, `substring()`, `length()`.

In `=` strings, string literals are written in single or double quotes, numbers as digits, booleans as `true` and `false`; parentheses group sub-expressions, and field tokens are used directly: `=if([$Priority] == 'High', '#a4262c', '#323130')`.

What is not supported

The web part does not implement the following parts of the column formatting language:

- `forEach` - repeating an element for each value of a multi-value column;
- `customRowAction` - buttons that start a flow, open a panel or set a field value;
- `elmType: "filepreview"` - document previews;
- `elmType: "svg"` and `path` - inline vector graphics;
- the tokens `@rowIndex`, `@window`, `@thumbnail`, `@currentWeb` and other tokens not listed above;
- the operators and functions not listed above, such as `toLocaleString()`, `toLocaleDateString()`, `startsWith()`, `padStart()`, `replace()`, `split()`, `join()`, `getDate()`, `addDays()`, `getUserImage()`, `cos()` and `sin()`.

A format that uses `forEach`, `customRowAction`, `filepreview`, `svg`, an unknown operator or function, an unknown `@` token in an `=` expression or a syntax error in an `=`

expression cannot be rendered. In that case the web part gives up on the format and renders the column with its default display - the format chosen in the column configurator - exactly as if the option were off for that column. The other columns are not affected. Unknown attributes and style properties, on the other hand, are simply skipped and the rest of the format is applied.

Warning

The formats produced by the column formatting designer in SharePoint and the samples published by Microsoft often use constructs outside the supported subset, such as `forEach`, `customRowAction`, `getUserImage()` or `toLocaleDateString()`. A format that works in the list may therefore fall back to the default display in the web part. Always check the result on the page after enabling the option, and simplify the format if a column is not rendered as expected.

Note

Values can arrive as text formatted with the list's regional settings. Use `Number()` or `Date()` only when the value is in a format those functions recognize; localized decimal separators, thousands separators and date formats may not convert correctly. Referencing another column does not guarantee a raw numeric or date value. Check the actual values and keep the formatting simple when conversion is unreliable.

`[$Field]` references. Use `@me` and `@now` as standalone tokens or in `=` expressions. An unknown `@` token causes fallback in an `=` expression; in ordinary text it remains literal.

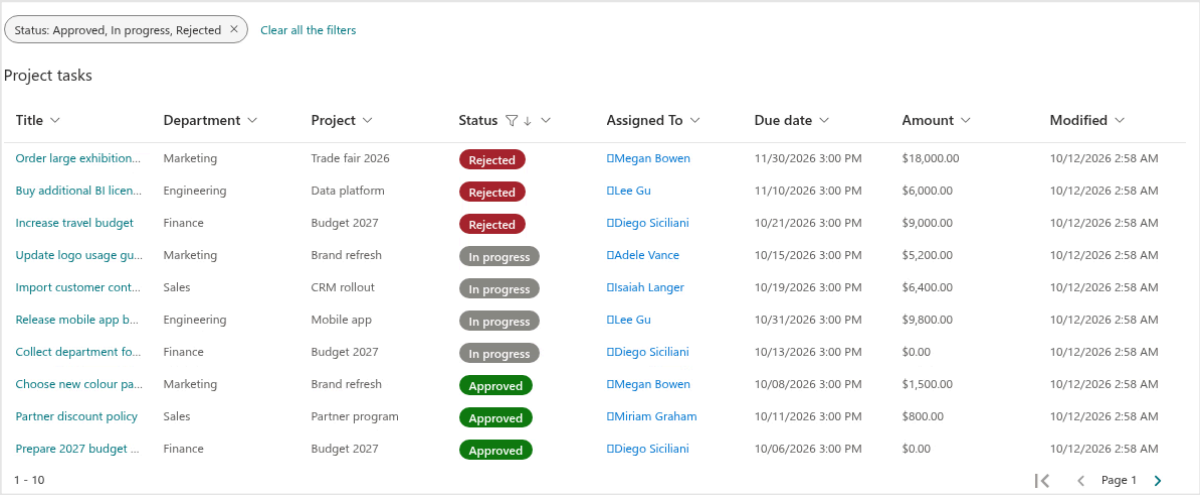
Examples

The following formats stay within the supported subset.

A status pill. The choice value is shown on a rounded background whose color depends on the value:

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/
sp/v2/column-formatting.schema.json",
  "elmType": "div",
```

```
  "txtContent": "@currentField",
  "style": {
    "display": "inline-block",
    "padding": "2px 10px",
    "border-radius": "12px",
    "font-weight": "600",
    "color": "#ffffff",
    "background-color": "=if(@currentField == 'Approved',
'#107c10', if(@currentField == 'Rejected', '#a4262c',
'#8a8886'))"
  }
}
```



A Status column rendered as colored pills

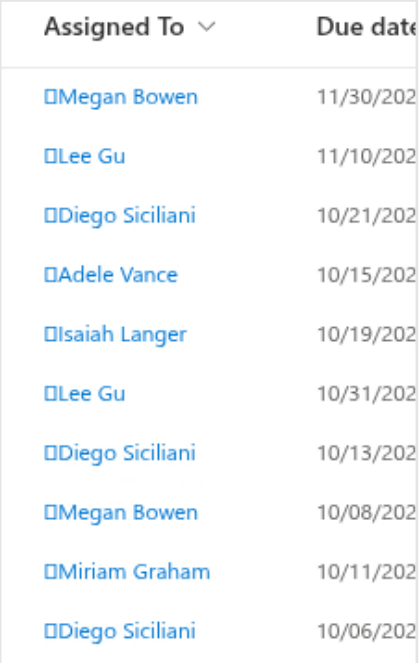
A number colored by its value, written as an operator tree. Amounts above 1000 are shown in red and bold:

```
{
  "elmType": "span",
  "txtContent": "@currentField",
  "style": {
    "color": {
      "operator": "?",
      "operands": [
        { "operator": ">", "operands": [ { "operator":
"Number()", "operands": [ "@currentField" ] }, 1000 ] },
        "#a4262c",
        "#107c10"
      ]
```

```
    },
    "font-weight": "=if(Number(@currentField) > 1000, '700',
'400')"
  }
}
```

A link with an icon. The person in the Assigned To column is rendered as a mail link with an envelope icon; the format is set on the Assigned To column itself and uses the properties of the person value:

```
{
  "elmType": "a",
  "attributes": {
    "href": "='mailto:' + @currentField.email",
    "title": "=@currentField.title",
    "iconName": "Mail",
    "target": "_blank"
  },
  "txtContent": "@currentField.title",
  "style": {
    "color": "#0078d4",
    "text-decoration": "none"
  }
}
```



A person column rendered as a mail link with an icon

Performance

The JSON of a column is parsed once, the first time a cell of that column is rendered, and the parsed tree is reused for every other cell of the column and for every later page of items. Only the tokens and expressions are evaluated per cell. Enabling the option therefore adds very little to the time the table needs to appear, even on long pages.

Terms

Column Formatting
Column formatting is a way to customize how fields in lists and libraries are displayed by using a JSON definition, for example to show icons, colored indicators, or links, without writing code.

Feature
A feature is a set of SharePoint elements (artifacts), such as site columns, content types, web parts, application pages, custom actions, list templates, list instances, and others.

JSON
JSON (JavaScript Object Notation) is a lightweight, text-based data-interchange format that is easy for people to read and write and easy for software to parse and generate. It is widely used by REST APIs, configuration files, and web applications to exchange structured data.

List
A list is a storage unit in SharePoint. A list can contain columns of different types, content types, and views. A document library is a special type of list and therefore inherits the functionality of a list.

SharePoint
SharePoint is a web-based collaboration platform that provides content management, document management, workflow applications, search, and other tools that empower employees and teams to work together. SharePoint is also a secure place to store, organize, and access shared information.

SharePoint Online
The cloud version of SharePoint, a web-based collaboration platform within Microsoft 365 that provides an extensive range of capabilities to store, share, and manage the digital information, content, and knowledge of an organization.

Site
A SharePoint site is a container for information within the SharePoint platform. Sites typically offer features such as document libraries, lists, calendars, discussion boards,

and workflows, facilitating collaboration, communication, and document management within teams or across departments. Sites can be customized and configured to suit the specific needs and workflows of an organization.

View
A view is a saved configuration that determines which items of a list or library are displayed and how, including the columns, sorting, filtering, grouping, and layout. Views can be personal or public.

Web Part
A web part is a primary building block of functionality or content placed on a SharePoint page. Some web parts support customization, such as configuring the web part or changing its appearance.