FormTokenField

- Keyboard Accessibility
- Properties
- Usage

A `FormTokenField` is a field similar to the tags and categories fields in the interim editor chrome, or the “to” field in Mail on OS X. Tokens can be entered by typing them or selecting them from a list of suggested tokens.

Up to one hundred suggestions that match what the user has typed so far will be shown from which the user can pick from (auto-complete). Tokens are separated by the “,” character. Suggestions can be selected with the up or down arrows and added with the tab or enter key.

The `value` property is handled in a manner similar to controlled form components. See Forms in the React Documentation for more information.

Keyboard Accessibility

- `left arrow` – if input field is empty, move insertion point before previous token
- `right arrow` – if input field is empty, move insertion point after next token
- `up arrow` – select previous suggestion
- `down arrow` – select next suggestion
- `tab` / `enter` – if suggestion selected, insert suggestion as a new token; otherwise, insert value typed into input as new token
- `comma` – insert value typed into input as new token

Properties

- `value` – An array of strings or objects to display as tokens in the field. If objects are present in the array, they **must** have a property of `value`. Here is an example object that could be passed in as a value:

```
{
    value: '(string) The value of the token.',
    status: "(string) One of 'error', 'validating', or 'success'. Applies styles to token."
    title: '(string) If not falsey, will add a title to the token.',
    onMouserEnter: '(function) Function to call when onMouseEnter event triggered on token.'
    onMouseLeave: '(function) Function to call when onMouseLeave is triggered on token.'
}
```

- `displayTransform` – Function to call to transform tokens for display. (In the editor, this is needed to decode HTML entities embedded in tags – otherwise entities like `&` in tag names are double-encoded like `&`, once by the REST API and once by React).
- `saveTransform` – Function to call to transform tokens for saving. The default is to trim the token value. This function is also applied when matching suggestions against the current value so that matching works correctly with leading or trailing spaces. (In the editor, this is needed to remove leading and trailing spaces from tag names, like wp-admin does. Otherwise the REST API won't save them.)
- `onChange` – Function to call when the tokens have changed. An array of new tokens is passed to the callback.
- `onInputChange` – Function to call when the users types in the input field. It can be used to trigger autocomplete requests.
- `onFocus` – Function to call when the TokenField has been focused on. The event is passed to the callback. Useful for analytics.
- `suggestions` – An array of strings to present to the user as suggested tokens.
- `maxSuggestions` – The maximum number of suggestions to display at a time.
- `tokenizeOnSpace` – If true, will add a token when `TokenField` is focused and `space` is pressed.
- `isBorderless` – When true, renders tokens as without a background.
- `maxLength` – If passed, `TokenField` will disable ability to add new tokens once number of tokens is greater than or equal to `maxLength`.
- `disabled` – When true, tokens are not able to be added or removed.
- `placeholder` – If passed, the `TokenField` input will show a placeholder string if no value tokens are present.
- `help` – Additional description for the control. Only use for meaningful description or instructions for the control. An element containing the description will be programmatically associated to the `FormTokenField` via `aria-describedby`. Defaults to a how-to message (e.g. *Separate with commas or the Enter key.*); pass an empty string to hide it.
- `messages` – Allows customizing the messages presented by screen readers in different occasions:
  - `added` – The user added a new token.
  - `removed` – The user removed an existing token.

    - `remove` – The user focused the button to remove the token.
    - `__experimentalInvalid` – The user tried to add a token that didn't pass the validation.
- `__experimentalRenderItem` – Custom renderer invoked for each option in the suggestion list. The render prop receives as its argument an object containing, under the `item` key, the single option's data (directly from the array of data passed to the `options` prop).
- `__experimentalExpandOnFocus` – If true, the suggestions list will be always expanded when the input field has the focus.
- `__experimentalShowHowTo` – **Deprecated.** Use the `help` prop instead. The `help` prop now defaults to the previous how-to text; if you were passing `__experimentalShowHowTo={ false }` to hide it, pass an empty string to `help` instead.
- `__experimentalValidateInput` – If passed, all introduced values will be validated before being added as tokens.
- `__experimentalAutoSelectFirstMatch` – If true, the select the first matching suggestion when the user presses the Enter key (or space when tokenizeOnSpace is true).
- `tokenizeOnBlur` – If true, add any incompleteTokenValue as a new token when the field loses focus.

Usage

```
import { useState } from 'react';
import { FormTokenField } from '@wordpress/components';

const continents = [
    'Africa',
    'America',
    'Antarctica',
    'Asia',
    'Europe',
    'Oceania',
];

const MyFormTokenField = () => {
    const [ selectedContinents, setSelectedContinents ] = useState( [] );

    return (
        <FormTokenField
            value={ selectedContinents }
            suggestions={ continents }
            onChange={ ( tokens ) => setSelectedContinents( tokens ) }
        />
    );
```

```
};
```

□□□ #2
□ □□□ □□□ 2026-10-31 18:13:17 UTC
□ □□□ □□□ 2026-10-31 18:13:19 UTC