QueryControls



- Development Guidelines
  - Usage
  - Multiple category selector
  - Props
    - authorList: Author[]
    - categoriesList: Category[]
    - categorySuggestions: Record< Category['name'], Category >
    - maxItems: number
    - minItems: number
    - numberOfItems: number
    - onAuthorChange: (newAuthor: string) => void
    - onCategoryChange: (newCategory: string) => void | FormTokenFieldProps['onChange']
    - onNumberOfItemsChange: (newNumber?: number) => void
    - onOrderChange: (newOrder: 'asc' | 'desc') => void
    - onOrderByChange: (newOrderBy: 'date' | 'title') => void
    - order: 'asc' | 'desc'
    - orderBy: 'date' | 'title' | 'menu_order'
    - orderByOptions: OrderByOption[]
    - selectedAuthorId: number
    - selectedCategories: Category[]
    - selectedCategoryId: number

Development Guidelines

Usage

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

const QUERY_DEFAULTS = {
    category: 1,
    categories: [
        {
            id: 1,
            name: 'Category 1',
            parent: 0,
        },
        {
            id: 2,
            name: 'Category 1b',
            parent: 1,
        },
        {
            id: 3,
            name: 'Category 2',
            parent: 0,
        },
    ],
    maxItems: 20,
    minItems: 1,
    numberOfItems: 10,
    order: 'asc',
    orderBy: 'title',
};

const MyQueryControls = () => {
    const [ query, setQuery ] = useState( QUERY_DEFAULTS );
    const { category, categories, maxItems, minItems, numberOfItems, order, orderBy } = query;

    const updateQuery = ( newQuery ) => {
        setQuery( { ...query, ...newQuery } );
    };
```

```
    return (
        <QueryControls
            { ...{ maxItems, minItems, numberOfItems, order, orderBy } }
            onOrderByChange={ ( newOrderBy ) => updateQuery( { orderBy: newOrderBy } ) }
            onOrderChange={ ( newOrder ) => updateQuery( { order: newOrder } ) }
            categoriesList={ categories }
            selectedCategoryId={ category }
            onCategoryChange={ ( newCategory ) => updateQuery( { category: newCategory } ) }
            onNumberOfItemsChange={ ( newNumberOfItems ) =>
                updateQuery( { numberOfItems: newNumberOfItems } )
            }
        />
    );
};
```

Multiple category selector

The `QueryControls` component now supports multiple category selection, to replace the single category selection available so far. To enable it use the component with the new props instead: `categorySuggestions` in place of `categoriesList` and the `selectedCategories` array instead of `selectedCategoryId` like so:

```
const QUERY_DEFAULTS = {
    orderBy: 'title',
    order: 'asc',
    selectedCategories: [
        {
            id: 1,
            value: 'Category 1',
            parent: 0,
        },
        {
            id: 2,
            value: 'Category 1b',
            parent: 1,
        },
    ],
    categories: {
        'Category 1': {
```

```
            id: 1,
            name: 'Category 1',
            parent: 0,
        },
        'Category 1b': {
            id: 2,
            name: 'Category 1b',
            parent: 1,
        },
        'Category 2': {
            id: 3,
            name: 'Category 2',
            parent: 0,
        },
    },
    numberOfItems: 10,
};

const MyQueryControls = () => {
    const [ query, setQuery ] = useState( QUERY_DEFAULTS );
    const { orderBy, order, selectedCategories, categories, numberOfItems } = query;

    const updateQuery = ( newQuery ) => {
        setQuery( { ...query, ...newQuery } );
    };

    return (
        <QueryControls
            { ...{ orderBy, order, numberOfItems } }
            onOrderByChange={ ( newOrderBy ) => updateQuery( { orderBy: newOrderBy } ) }
            onOrderChange={ ( newOrder ) => updateQuery( { order: newOrder } ) }
            categorySuggestions={ categories }
            selectedCategories={ selectedCategories }
            onCategoryChange={ ( category ) => updateQuery( { selectedCategories: category } )
}
            onNumberOfItemsChange={ ( newNumberOfItems ) =>
                updateQuery( { numberOfItems: newNumberOfItems } )
            }
        />
    );
```

```
};
```

The format of the categories list also needs to be updated to match the expected type for the category suggestions.

Props

`authorList`: `Author[]`

An array of the authors to select from.

- Required: No
- Platform: Web

`categoriesList`: `Category[]`

An array of categories. When passed in conjunction with the `onCategoryChange` prop, it causes the component to render UI that allows selecting one category at a time.

- Required: No
- Platform: Web

`categorySuggestions`: `Record< Category[ 'name' ], Category >`

An object of categories with the category name as the key. When passed in conjunction with the `onCategoryChange` prop, it causes the component to render UI that enables multiple selection.

- Required: No
- Platform: Web

`maxItems`: `number`

The maximum number of items.

- Required: No
- Default: 100
- Platform: Web

`minItems`: `number`

The minimum number of items.

- Required: No

- Default: 1
- Platform: Web

`numberOfItems`: `number`

The selected number of items to retrieve via the query.

- Required: No
- Platform: Web

`onAuthorChange`: `( newAuthor: string ) => void`

A function that receives the new author value. If not specified, the author controls are not rendered.

- Required: No
- Platform: Web

`onCategoryChange`: `( newCategory: string ) => void | FormTokenFieldProps[ 'onChange' ]`

A function that receives the new category value. If not specified, the category controls are not rendered.
The function's signature changes depending on whether multiple category selection is enabled or not.

- Required: No
- Platform: Web

`onNumberOfItemsChange`: `( newNumber?: number ) => void`

A function that receives the new number of items. If not specified, then the number of items range control is not rendered.

- Required: No
- Platform: Web

`onOrderChange`: `( newOrder: 'asc' | 'desc' ) => void`

A function that receives the new order value. If this prop or the `onOrderByChange` prop are not specified, then the order controls are not rendered.

- Required: No
- Platform: Web

`onOrderByChange`: `( newOrderBy: 'date' | 'title' ) => void`

A function that receives the new orderby value. If this prop or the `onOrderChange` prop are not specified, then the order controls are not rendered.

- Required: No
- Platform: Web

`order`: `'asc' | 'desc'`

The order in which to retrieve posts.

- Required: No
- Platform: Web

`orderBy`: `'date' | 'title' | 'menu_order'`

The meta key by which to order posts.

- Required: No
- Platform: Web

`orderByOptions`: `OrderByOption[]`

The meta key by which to order posts.

- Required: No
- Platform: Web

`selectedAuthorId`: `number`

The selected author ID.

- Required: No
- Platform: Web

`selectedCategories`: `Category[]`

The selected categories for the `categorySuggestions` prop.

- Required: No
- Platform: Web

`selectedCategoryId`: `number`

The selected category for the `categoriesList` prop.

- Required: No
- Platform: Web

□□□ #2
□ □□□ □□□ 2026-10-31 18:14:19 UTC
□ □□□ □□□ 2026-10-31 18:14:20 UTC