## **TermSetWebPart Class Documentation**

### **Overview**

### The TermSetWebPart class represents a SharePoint Framework (SPFx) web part that integrates with the SharePoint term store. It allows users to configure and display term sets from a selected term store group. This web part fetches term store groups and term sets using SharePoint REST APIs and allows configuration through the property pane.

### **Imports**

### React: Core React library for building the UI components.

### ReactDom: For rendering React components into the DOM.

### Version: Class for handling version information.

### IPropertyPaneConfiguration, IPropertyPaneDropdownOption, PropertyPaneDropdown: SPFx property pane configuration and controls.

### BaseClientSideWebPart: Base class for creating SPFx web parts.

### IReadonlyTheme: Interface for theme information.

### SPHttpClient: Provides methods for making HTTP requests to SharePoint REST APIs.

### strings: Localized strings for the web part.

### ITermSetProps: Interface for the properties used by the TermSetList component.

### TermSetList: React component that displays term sets in a tree view.

### PropertyFieldMultiSelect: A property pane control for selecting multiple values.

### **Interface Definitions**

#### **ITermSetWebPartProps**

### description (string): Description of the web part.

### selectedTermSets (string[]): Array of selected term set names.

### selectedGroupId (string): ID of the selected term store group.

### **Class: TermSetWebPart**

#### **Methods**

### **render**():

### Renders the TermSetList React component into the DOM.

### Parameters: None

### Returns: Promise<void>

### **getTermSets(groupId: string)**:

### Fetches term sets for the given term store group ID and populates the term set dropdown options.

### Parameters:

### groupId (string): The ID of the term store group.

### Returns: Promise<void>

### **onInit()**:

### Initializes the web part and fetches term store groups.

### Parameters: None

### Returns: Promise<void>

### **getTermStoreGroups()**:

### Fetches term store groups and populates the group dropdown options.

### Parameters: None

### Returns: Promise<void>

### **onThemeChanged(currentTheme: IReadonlyTheme | undefined)**:

### Applies theme changes to the web part.

### Parameters:

### currentTheme (IReadonlyTheme | undefined): The current theme information.

### Returns: void

### **onDispose()**:

### Unmounts the React component from the DOM when the web part is disposed.

### Parameters: None

### Returns: void

### **onPropertyPaneConfigurationStart**():

### Loads term store groups and term sets for the property pane configuration.

### Parameters: None

### Returns: Promise<void>

### **onPropertyPaneFieldChanged(propertyPath: string, oldValue: any, newValue: any)**:

### Handles changes in the property pane fields and updates term sets based on the selected group.

### Parameters:

### propertyPath (string): The path of the changed property.

### oldValue (any): The old value of the property.

### newValue (any): The new value of the property.

### Returns: Promise<void>

### **getPropertyPaneConfiguration()**:

### Configures the property pane with dropdowns for selecting term store groups and term sets.

### Parameters: None

### Returns: IPropertyPaneConfiguration

### Usage

### Rendering: The render method creates and renders the TermSetList component with the current web part context, selected term store group ID, and selected term sets.

### Property Pane Configuration: The getPropertyPaneConfiguration method defines the configuration options available in the property pane, including dropdowns for selecting term store groups and term sets.

### Data Fetching: Methods like getTermSets and getTermStoreGroups are used to fetch data from the SharePoint term store and update dropdown options accordingly.

### **Component: TermSetList**

**Description**: The TermSetList component is designed to fetch and display term sets and their hierarchical terms from the SharePoint Term Store. It uses Fluent UI components to render a tree view of the terms, allowing users to navigate through term sets and select terms. When a term is selected, a custom event is dispatched to communicate the selection to other parts of the application.

**Props**:

* **context (WebPartContext)**:
  + **Description**: Provides the context for the web part, including access to SharePoint resources, such as spHttpClient for making API requests.
  + **Usage**: This context is essential for performing HTTP requests to SharePoint's REST API.
* **groupId (string)**:
  + **Description**: The ID of the term group that contains the term sets to be displayed.
  + **Usage**: Used to filter term sets that belong to a specific term group when fetching data from SharePoint.
* **setNames (string[])**:
  + **Description**: An array of names of term sets to be fetched and displayed.
  + **Usage**: Each name in the array is used to fetch the corresponding term set by its name.

**State**:

* **termSets (TermSet[])**:
  + **Description**: Stores the term sets and their associated terms fetched from SharePoint.
  + **Usage**: Used to render the tree view of term sets and terms.
* **isLoading (boolean)**:
  + **Description**: Indicates whether the term sets are still being fetched.
  + **Usage**: Controls the display of a loading message or the tree view of term sets.
* **selectedTermId (string)**:
  + **Description**: The ID of the currently selected term.
  + **Usage**: Highlights the selected term in the tree view and is used in the handleTermClick method to dispatch the custom event.

**Methods**:

* **fetchTerms**:
  + **Description**: Asynchronously fetches terms for a given term set. If a parent term is provided, it fetches the child terms recursively.
  + **Parameters**:
    - setId (string): The ID of the term set to fetch terms from.
    - parentTermId (string | undefined): The ID of the parent term to fetch child terms. If undefined, fetches top-level terms.
    - parentName (string | undefined): The name of the parent term. Used to set the ParentName of child terms.
    - mainParentId (string | undefined): The ID of the main parent term. Used to set the MainParentId of child terms.
  + **Returns**: A promise that resolves to an array of Term objects.
  + **Usage**: Called recursively to build a hierarchical structure of terms.
* **fetchData**:
  + **Description**: Fetches term sets based on the setNames and groupId props. It then fetches terms for each term set and updates the state.
  + **Returns**: A promise that resolves once term sets and terms are fetched and the state is updated.
  + **Usage**: Initializes the fetching process when the component mounts or when the setNames or groupId props change.
* **handleTermClick**:
  + **Description**: Handles the click event on a term item. Sets the selectedTermId state and dispatches a custom event with the parent category name.
  + **Parameters**:
    - term (Term): The term that was clicked.
  + **Returns**: Nothing (void).
  + **Usage**: Used to update the selection state and notify other parts of the application about the selected term.
* **renderTreeItems**:
  + **Description**: Recursively renders tree items for the terms. Handles rendering of child terms.
  + **Parameters**:
    - terms (Term[]): The terms to render as tree items.
  + **Returns**: JSX elements representing the tree items.
  + **Usage**: Called to build the tree view of terms.
* **renderTermSets**:
  + **Description**: Renders the term sets and their terms as a tree structure.
  + **Parameters**:
    - sets (TermSet[]): The term sets to render.
  + **Returns**: JSX elements representing the term sets as tree structures.
  + **Usage**: Calls renderTreeItems to build the tree view for each term set.

**Rendering**:

* **Loading State**:
  + Displays a loading message ("Loading term sets...") while term sets are being fetched.
* **Loaded State**:
  + Displays the term sets in a hierarchical tree view using Fluent UI’s Tree and TreeItem components.

**Event Handling**:

* **Custom Event**:
  + **Event Name**: catagorySelected
  + **Detail**: Contains the name of the parent category of the selected term.
  + **Usage**: Dispatched when a term is clicked, allowing other components to react to the term selection.

**Error Handling**:

* Logs errors to the console if there are issues fetching term sets or terms.
* Sets termSets to an empty array and updates isLoading to false if data cannot be fetched.

**Styles**:

* Utilizes CSS modules for styling, with styles imported from TermSet.module.scss. The selectedItem and term\_item classes are used to style selected and general term items.