@wordpress/global-styles-engine

- Overview
- API
  - Reading Data
    - getStyle(globalStyles, path, blockName?, shouldDecodeEncode?)
    - getSetting(globalStyles, path, blockName?)
  - Writing Data
    - setStyle(globalStyles, path, newValue, blockName?)
    - setSetting(globalStyles, path, newValue, blockName?)
  - Utility Functions
    - getPalettes(globalStyles)
    - generateGlobalStyles(globalStyles)
    - mergeGlobalStyles(base, user)
- Data Structure
- Features
- Usage

A generic library for reading and writing global styles data structures (theme.json format).

Overview

This is a framework-agnostic utility library that provides functions for manipulating global styles objects. It handles the complex nested structure of theme.json format, including block-specific styles and settings.

API

Reading Data

`getStyle(globalStyles, path, blockName?, shouldDecodeEncode?)`

Get a style value from the global styles object.

```
// Get global text color
const textColor = getStyle( globalStyles, 'color.text' );

// Get button background color
const buttonBg = getStyle( globalStyles, 'color.background', 'core/button' );

// Get raw value without CSS variable resolution
const rawValue = getStyle( globalStyles, 'color.text', undefined, false );
```

`getSetting(globalStyles, path, blockName?)`

Get a setting value with fallback hierarchy (block-specific → global).

```
// Get global color palette
const colors = getSetting( globalStyles, 'color.palette' );

// Get paragraph-specific font sizes
const fontSizes = getSetting(
    globalStyles,
    'typography.fontSizes',
    'core/paragraph'
);
```

Writing Data

`setStyle(globalStyles, path, newValue, blockName?)`

Immutably set a style value. Returns a new global styles object.

```
// Set global text color
const updated = setStyle( globalStyles, 'color.text', '#333333' );

// Set button background color
const updated = setStyle(
    globalStyles,
    'color.background',
    '#0073aa',
    'core/button'
);
```

`setSetting(globalStyles, path, newValue, blockName?)`

Immutably set a setting value. Returns a new global styles object.

```
// Set global color palette
const updated = setSetting( globalStyles, 'color.palette', newPalette );

// Set block-specific settings
const updated = setSetting(
    globalStyles,
    'typography.fontSizes',
    sizes,
    'core/heading'
);
```

Utility Functions

`getPalettes(globalStyles)`

Extract color palettes organized by origin (theme, custom, default).

`generateGlobalStyles(globalStyles)`

Generate CSS from global styles object.

`mergeGlobalStyles(base, user)`

Merge multiple global styles objects with proper precedence.

Data Structure

The global styles object follows the theme.json schema:

```
{
  styles: {
    color: { text: '#000', background: '#fff' },
    typography: { fontSize: '16px' },
    elements: {
      button: { color: { background: 'blue' } }
    },
    blocks: {
      'core/paragraph': { color: { text: '#333' } }
    }
  },
  settings: {
    color: { palette: [...] },
    typography: { fontSizes: [...] },
    blocks: {
      'core/paragraph': { color: { text: true } }
    }
  }
}
```

Features

- **Immutable Updates**: All setter functions return new objects
- **Type Safety**: Full TypeScript support
- **CSS Variable Resolution**: Automatic resolution of CSS custom properties
- **Block-Specific Styles**: Support for block and element-specific styling
- **Fallback Hierarchy**: Smart fallback from block → global → default settings
- **Framework Agnostic**: No dependencies on React, WordPress, or other frameworks

Usage

This library is designed to be used as the foundation for any global styles editing interface. It provides the core data manipulation functions while remaining completely generic and reusable.

It is a prerequisite for use that blocks and their styles be globally registered, because block styles are fetched from the global blocks store.

□□□ #2
□ □□□ □□□ 2026-10-31 18:21:46 UTC
□ □□□ □□□ 2026-10-31 18:21:48 UTC