

@yuntijs/dom-helpers

npm invalid downloads package not found or too new

A TypeScript library for advanced DOM element height calculations, with support for complex layouts including CSS Grid, Flexbox, and nested containers.

Features

- 🎯 **Accurate height calculation** for complex CSS layouts (Grid, Flexbox, Block)
- ⚙️ **Recursive depth control** to balance performance and precision
- ⚡ **Multiple calculation modes** (fast, standard, precise, grid-optimized)
- 🛠️ **Configurable options** for margins, padding, and scroll height
- 🌐 **Universal compatibility** works with all modern browsers
- 🔒 **Full TypeScript support** with comprehensive type definitions

Install

```
npm install @yuntijs/dom-helpers
# or
pnpm install @yuntijs/dom-helpers
# or
yarn add @yuntijs/dom-helpers
```

Quick Start

```
import { getContentHeight, getContentHeightFast } from '@yuntijs/dom-helpers';

const container = document.getElementById('my-container');

// Basic usage
const height = getContentHeight(container);

// Fast calculation (performance-optimized)
const fastHeight = getContentHeightFast(container);

// Custom configuration
const preciseHeight = getContentHeight(container, {
  maxDepth: 5,
  includeMargins: true,
  includePadding: true,
  layoutTypes: ['grid', 'flex', 'block']
});
```

API Reference

Main Functions

getContentHeight(container, options?)

The primary function for calculating element content height.

Parameters:

- `container`: `HTMLElement` - The DOM element to calculate height for
- `options?`: `HeightCalculationOptions` - Configuration options

Returns: `number` - The calculated height in pixels

Preset Functions

```
// Fast mode: 1 layer depth, uses scrollHeight
getContentHeightFast(container: HTMLElement): number

// Standard mode: 3 layers, optimized for grid/flex
getContentHeightPrecise(container: HTMLElement): number

// Grid optimized: specialized for CSS Grid layouts
getContentHeightForGrid(container: HTMLElement): number
```

Configuration Options

```
interface HeightCalculationOptions {
  maxDepth?: number;           // Maximum recursion depth (default: 3)
  includeMargins?: boolean;    // Include element margins (default: true)
  includePadding?: boolean;    // Include container padding (default: true)
  useScrollHeight?: boolean;   // Use scrollHeight as fallback (default:
false)
  layoutTypes?: string[];      // Layout types for deep calculation
                                (default: ['grid', 'flex', 'block'])
}
```

Preset Configurations

Preset	Max Depth	Layout Types	Use Case
fast	1	-	Performance-critical scenarios
standard	3	grid, flex	General purpose, balanced
precise	5	grid, flex, block	High accuracy needed
gridOptimized	2	grid	CSS Grid specific layouts

Usage Examples

Basic Height Calculation

```
import { getContentHeight } from '@yuntijs/dom-helpers';

const container = document.querySelector('.content-wrapper');
const height = getContentHeight(container);
console.log(`Content height: ${height}px`);
```

Grid Layout Optimization

```
import { getContentHeightForGrid } from '@yuntijs/dom-helpers';

const gridContainer = document.querySelector('.grid-container');
const gridHeight = getContentHeightForGrid(gridContainer);

// Apply calculated height
gridContainer.style.height = `${gridHeight}px`;
```

Custom Configuration

```
import { getContentHeight } from '@yuntijs/dom-helpers';

const complexLayout = document.querySelector('.complex-layout');

const height = getContentHeight(complexLayout, {
  maxDepth: 4,
  includeMargins: false,
  includePadding: true,
  useScrollHeight: true,
  layoutTypes: ['grid', 'flex']
});
```

Performance Comparison

```
import {
  getContentHeight,
  getContentHeightFast,
  getContentHeightPrecise
} from '@yuntijs/dom-helpers';

const element = document.getElementById('dynamic-content');

// Fast but less accurate
```

```
const fastHeight = getContentHeightFast(element);

// Balanced approach
const standardHeight = getContentHeight(element);

// Slow but most accurate
const preciseHeight = getContentHeightPrecise(element);

console.log({ fastHeight, standardHeight, preciseHeight });
```

When to Use Each Mode

Fast Mode (`getContentHeightFast`)

- ✓ Real-time calculations during scrolling/resizing
- ✓ Simple layouts without complex nesting
- ✓ Performance is critical

Standard Mode (`getContentHeight`)

- ✓ Most common use cases
- ✓ Grid and Flexbox layouts
- ✓ Good balance of speed and accuracy

Precise Mode (`getContentHeightPrecise`)

- ✓ Complex nested layouts
- ✓ Accuracy is more important than speed
- ✓ Final calculations for layout positioning

Grid Optimized (`getContentHeightForGrid`)

- ✓ CSS Grid specific issues
- ✓ Grid items with dynamic content
- ✓ Grid containers with auto-sizing

Browser Support

- Chrome 60+
- Firefox 55+
- Safari 12+
- Edge 79+

Development

```
# Install dependencies
pnpm install

# Development mode
```

```
npm run dev

# Build library
npm run build

# Run tests
npm test

# Type checking
npm run type-check
```

LICENSE

MIT