@wordpress/viewport

- Installation
- Usage
  - Data Module
  - Higher-Order Components
    - ifViewportMatches
    - store
    - withViewportMatch
- Contributing to this package

Viewport is a module for responding to changes in the browser viewport size. It registers its own data module, updated in response to browser media queries on a standard set of supported breakpoints. This data and the included higher-order components can be used in your own modules and components to implement viewport-dependent behaviors.

Installation

Install the module

```
npm install @wordpress/viewport --save
```

*This package assumes that your code will run in an **ES2015+** environment. If you're using an environment that has limited or no support for such language features and APIs, you should include the polyfill shipped in `@wordpress/babel-preset-default` in your code.*

Usage

The standard set of breakpoint thresholds is as follows:

Name	Pixel Width

`huge`	1440
`wide`	1280
`large`	960
`medium`	782
`small`	600
`mobile`	480

Data Module

The Viewport module registers itself under the `core/viewport` data namespace and is exposed from the package as `store`.

```
import { select } from '@wordpress/data';
import { store } from '@wordpress/viewport';

const isSmall = select( store ).isViewportMatch( '< medium' );
```

The `isViewportMatch` selector accepts a single string argument `query`. It consists of an optional operator and breakpoint name, separated with a space. The operator can be `<` or `>=`, defaulting to `>=`.

```
import { select } from '@wordpress/data';
import { store } from '@wordpress/viewport';

const { isViewportMatch } = select( store );
const isSmall = isViewportMatch( '< medium' );
const isWideOrHuge = isViewportMatch( '>= wide' );
// Equivalent:
//  const isWideOrHuge = isViewportMatch( 'wide' );
```

Higher-Order Components

This package provides a set of HOCs to author components whose behavior should vary depending on the viewport.

ifViewportMatches

Higher-order component creator, creating a new component which renders if the viewport query is satisfied.

Related

- withViewportMatches

Usage

```
function MyMobileComponent() {
    return <div>I'm only rendered on mobile viewports!</div>;
}

MyMobileComponent = ifViewportMatches( '< small' )( MyMobileComponent );
```

Parameters

- *query* `ViewportQuery`: Viewport query.

Returns

- Higher-order component.

store

Store definition for the viewport namespace.

Related

- https://github.com/WordPress/gutenberg/blob/HEAD/packages/data/README.md#createReduxStore

withViewportMatch

Higher-order component creator, creating a new component which renders with the given prop names, where the value passed to the underlying component is the result of the query assigned as the object's value.

Related

- isViewportMatch

Usage

```
function MyComponent( { isMobile } ) {
    return <div>Currently: { isMobile ? 'Mobile' : 'Not Mobile' }</div>;
}
```

```
MyComponent = withViewportMatch( { isMobile: '< small' } )( MyComponent );
```

Parameters

- *queries* `ViewportQueries`: Object of prop name to viewport query.

Returns

- Higher-order component.

Contributing to this package

This is an individual package that's part of the Gutenberg project. The project is organized as a monorepo. It's made up of multiple self-contained software packages, each with a specific purpose. The packages in this monorepo are published to npm and used by WordPress as well as other software projects.

To find out more about contributing to this package or Gutenberg as a whole, please read the project's main contributor guide.

□□□ #2
□ □□□ □□□ 2026-10-31 18:20:37 UTC
□ □□□ □□□ 2026-10-31 18:20:38 UTC