Data Store

- Cart Store (wc/store/cart)
- Checkout Store (wc/store/checkout)
- Collections Store (wc/store/collections)
- Payment Store (wc/store/payment)
- Query-State Store (wc/store/query-state)
- Schema Store (wc/store/schema)
- Store Notices Store (wc/store/store-notices)
- Validation Store (wc/store/validation)

This document provides an overview of the data stores registered with `wp.data` for use by various blocks. Store descriptions are exported as constants on the `wc.wcBlocksData` export (externally registered as `@woocommerce/block-data` and enqueued via handle `wc-blocks-data-store`). For any block using the store, it's recommended to import the store description rather than using the reference directly to ensure dependencies are extracted correctly. It is assumed readers have some familiarity with the `wp.data` API. You can read more about that in the @wordpress/data reference guide.

Cart Store (wc/store/cart)

The **Cart Store (wc/store/cart)** is responsible for handling cart-related data and operations. To access this store using a constant, you can use:

```
const { cartStore } = window.wc.wcBlocksData;
```

Checkout Store (wc/store/checkout)

The **Checkout Store (wc/store/checkout)** manages the checkout process, ensuring a seamless experience for users. To access this store using a constant, you can use:

```
const { checkoutStore } = window.wc.wcBlocksData;
```

Collections Store (wc/store/collections)

The **Collections Store (wc/store/collections)** holds data indexed by namespace, model name, and query string. To access this store using a constant, you can use:

```
const { COLLECTIONS_STORE_KEY } = window.wc.wcBlocksData;
```

Payment Store (wc/store/payment)

The **Payment Store (wc/store/payment)** deals with all payment-related data and transactions. To access this store using a constant, you can use:

```
const { paymentStore } = window.wc.wcBlocksData;
```

Query-State Store (wc/store/query-state)

The **Query-State Store (wc/store/query-state)** holds arbitrary values indexed by context and key. It's often used for tracking the state of query objects for a given context. To access this store using a constant, you can use:

```
const { QUERY_STATE_STORE_KEY } = window.wc.wcBlocksData;
```

Schema Store (wc/store/schema)

The **Schema Store (wc/store/schema)** is primarily used for accessing routes and has more of an internal usage. To access this store using a constant, you can use:

```
const { SCHEMA_STORE_KEY } = window.wc.wcBlocksData;
```

Store Notices Store (wc/store/store-notices)

The **Store Notices Store (wc/store/store-notices)** is dedicated to handling various store notices and alerts. To access this store using a constant, you can use:

```
const { storeNoticesStore } = window.wc.wcBlocksData;
```

This `StoreDescriptor` can be passed in the `mapSelect` function of `useSelect` and in `useDispatch`.

Validation Store (wc/store/validation)

The **Validation Store (wc/store/validation)** holds data relating to validation errors, it is primarily used in the Cart and Checkout flows to ensure the Checkout doesn't continue while invalid data is present. To access this store using a constant, you can use:

```
const { validationStore } = window.wc.wcBlocksData;
```

□□□ #2
□ □□□ □□□ 2026-10-31 13:12:04 UTC
□ □□□ □□□ 2026-10-31 13:12:05 UTC