

📁 WooCommerce CRUD



📁

- [CRUD 📁](#)
- [CRUD 📁📁](#)
 - [📁](#)
 - [Getter 📁 Setter 📁](#)
 - [📁📁](#)
 - [📁📁📁](#)
- [CRUD 📁📁](#)
 - [📁📁📁📁](#)
 - [📁📁📁📁](#)
 - [📁📁📁](#)
 - [📁📁📁](#)
 - [📁📁📁](#)
 - [📁📁📁](#)
 - [📁📁📁](#)
 - [📁📁📁](#)
 - [📁📁 / 📁](#)
 - [📁📁📁](#)
 - [📁📁](#)
 - [📁📁 \(Billing\)](#)
 - [📁📁](#)
 - [📁📁](#)
 - [📁](#)

CRUD 📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁

——📁📁📁📁📁📁📁📁

[WooCommerce 3.0 📁📁](#) [CRUD 📁📁](#) 📁📁📁

WooCommerce 📁📁

📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁

WordPress 📁📁📁📁

```
getter  setter
```

- `api` - `api`
- `api` - `api`
- `api` - `api`
- `api` - `api`
- `api` - `api` REST API `cli` `cli`
- `api` - `api`

The diagram illustrates the data flow from WC_Data to CRUD and then to \$data and ID. WC_Data is represented by a box with 10 slots. An arrow points from WC_Data to a CRUD box with 10 slots. From the CRUD box, an arrow points to a \$data box with 3 slots and an ID box with 2 slots.

\$data

```
/**
 * ██████████
 * @since 3.0.0
 * @var array
 */
protected $data = array(
    'code' => '',
    'amount' => 0,
    'date_created' => '',
    'date_modified' => '',
    'discount_type' => 'fixed_cart',
    'description' => '',
    'date_expires' => '',
    'usage_count' => 0,
```

```

[]'individual_use'           => false,
[]'product_ids'             => array(),
[]'excluded_product_ids'    => array(),
[]'usage_limit'             => 0,
[]'usage_limit_per_user'    => 0,
[]'limit_usage_to_x_items'  => 0,
[]'free_shipping'           => false,
[]'product_categories'      => array(),
[]'excluded_product_categories' => array(),
[]'exclude_sale_items'      => false,
[]'minimum_amount'          => '',
[]'maximum_amount'          => '',
[]'email_restrictions'      => array(),
[]'used_by'                 => array(),
);

```

Getter □ Setter □□

The diagram illustrates a memory layout. At the top, there is a horizontal row of 16 small squares, representing a 16-byte array. Below this array, on the left, is a light blue rounded rectangle containing the text '\$data'. To the right of the '\$data' box is another horizontal row of 16 small squares, representing a pointer variable. A vertical line connects the first square of the '\$data' box to the first square of the second row of squares, indicating that '\$data' points to the start of the 16-byte array.

```
getter □ setter □□□□
getter □ setter □□□□□□
```

set_used_by() get_used_by()

Getter ☐☐☐

```
/**
 * 使用上下文
 * @since 3.0.0
 * @param string $context
 * @return array
 */
public function get_used_by( $context = 'view' ) {
    return $this->get_prop( 'used_by', $context );
}
```

Setter ☐ ☐ ☐

```
/**
 * ████████████████████
 * @since 3.0.0
 * @param array $used_by
 * @throws WC_Data_Exception
```

```

    */
    public function set_used_by( $used_by ) {
        $this->set_prop( 'used_by', array_filter( $used_by ) );
    }

```

set_prop get_prop WC_Data

[illegible]

--	--	--	--	--	--	--

 \$context

[view](#)[illegible]

☐ edit ☐☐☐☐☐☐

--	--	--	--

CRUD []

CRUD

--	--	--

```

/**
 * 优惠券ID
 * @param mixed $data 优惠券ID
 */
public function __construct( $data = '' ) {
    parent::__construct( $data );

    if ( $data instanceof WC_Coupon ) {
        $this->set_id( absint( $data->get_id() ) );
    } elseif ( is_numeric( $data ) && 'shop_coupon' === get_post_type( $data ) ) {
        $this->set_id( $data );
    } elseif ( ! empty( $data ) ) {
        $this->set_id( wc_get_coupon_id_by_code( $data ) );
        $this->set_code( $data );
    } else {
        $this->set_object_read( true );
    }

    $this->data_store = WC_Data_Store::load( 'coupon' );
    if ( $this->get_id() > 0 ) {
        $this->data_store->read( $this );
    }
}

```

--	--	--	--	--

☐ CRUD

Delete

```

/**
 * Save should create or update based on object existence.
 *
 * @since 2.6.0
 * @return int
 */
public function save() {
    if ( $this->data_store ) {
        // TODO: Add logic to update or create based on object existence
        do_action( 'woocommerce_before_' . $this->object_type . '_object_save', $this, $this->data_store );

        if ( $this->get_id() ) {
            $this->data_store->update( $this );
        } else {
            $this->data_store->create( $this );
        }

        return $this->get_id();
    }
}

```

ID / ID
delete

```

/**
 * Delete an object, set the ID to 0, and return result.
 *
 * @since 2.6.0
 * @param bool $force_delete
 * @return bool result
 */
public function delete( $force_delete = false ) {
    if ( $this->data_store ) {
        $this->data_store->delete( $this, array( 'force_delete' => $force_delete ) );
        $this->set_id( 0 );
        return true;
    }
    return false;
}

```

CRUD



```

$product = new WC_Product_Simple();
$product->set_name( 'My Product' );
$product->set_slug( 'myproduct' );
$product->set_description( 'A new simple product' );
$product->set_regular_price( '9.50' );
$product->save();

$product_id = $product->get_id();

```



```

$coupon = new WC_Coupon( $coupon_id );
$coupon->set_discount_type( 'percent' );
$coupon->set_amount( 25.00 );
$coupon->save();

```

--	--	--	--	--	--

```
$customer = new WC_Customer( $user_id );  
$email    = $customer->get_email();  
$address  = $customer->get_billing_address();  
$name     = $customer->get_first_name() . ' ' . $customer->get_last_name();
```

--	--	--	--	--

-   "  " 
 •   "  " 
 •   "  " 

--	--	--	--	--

- **(Name)**
- **(Discount)**
- **(Use)**
- **(Expiration Date)**
- **(Other Settings)**

--	--	--	--	--

[illegible]

- **(Name)**
- **(Discount)**
- **(Count)**
- **(Status)**

--	--	--	--



- [illegible]

- / □□

- ☐ (Import)

•  

[illegible]

• ☐ (Mail) ☐

- ☐ (Amount)

• [] " [] "

- _____

