



Test Orders

Channable Connect for Magento 2

Before going live with marketplace order imports, you'll want to verify your [Channable Connect](#) configuration works correctly. This guide shows you how to create test orders to validate your setup—shipping methods, invoicing, customer handling, and more—without needing actual marketplace orders.

Where to Find Test Orders

Navigate to **Channable → Orders** in your Magento admin panel and scroll down to the **Webhooks** section.

You'll see a table with three columns:

Column	Description
Store	The store view name
Webhook	The webhook URL to configure in Channable (copy this to your Channable account)
Test	Click "Import Test Order" to create a test order for that store

Each store view has its own row with a unique webhook URL and test link.

Creating a Test Order

There are two ways to create a test order: via the admin panel or via the CLI.

Method 1: Admin Panel (Import Test Order Button)

The easiest way to create a test order is directly from the Webhooks table:

1. Go to **Channable → Orders**
2. Scroll down to the **Webhooks** section
3. Find the store view you want to test in the table
4. Click **Import Test Order** in the Test column

The extension will create a test order using a random product from your catalog and redirect you to the newly created order.

With custom parameters:

If you want more control (specific product, quantity, country), right-click the **Import Test Order** button, copy the URL, and append parameters:

```
{copied-url}/product_id/42/qty/2/country/DE/lvb/1
```

Parameter	Description	Example
<code>product_id</code>	Use a specific product instead of random	<code>product_id/42</code>
<code>qty</code>	Set the order quantity	<code>qty/2</code>
<code>country</code>	Set the customer's country code	<code>country/BE</code>
<code>lvb</code>	Simulate an FBB/LVB (Fulfillment by Bol) order	<code>lvb/1</code>

Important: Magento protects all admin URLs with a security key. The full URL looks like this:

```
https://yourstore.com/admin/channable/order/simulate/key/a1b2c3d4e5f6.../store_id/1
```

The `/key/a1b2c3d4e5f6...` part is a session-specific token that Magento generates to prevent unauthorized access. Without a valid key, you'll be redirected to the dashboard. This is why you should always copy the URL from the **Import Test Order** button — it already contains your current security key.

Method 2: CLI Command

If you prefer the command line or need to run test orders without admin access (e.g. on staging/production via SSH):

```
bin/magento channable:order:simulate --store-id=1 --product-id=42 --qty=2 --country-code=DE
```

Parameter	Description	Example
<code>--store-id</code>	Store view ID (required)	<code>--store-id=1</code>
<code>--product-id</code>	Use a specific product instead of random	<code>--product-id=42</code>
<code>--qty</code>	Set the order quantity	<code>--qty=2</code>
<code>--country-code</code>	Set the customer's country code	<code>--country-code=BE</code>
<code>--lvb</code>	Simulate an FBB/LVB (Fulfillment by Bol) order	<code>--lvb=1</code>

The CLI command outputs the created order increment ID on success.

Parameter Details

product_id

By default, the simulator picks a random enabled simple product from your store. Use `product_id` to test with a specific product.

When to use:

- Testing a specific product that had import issues
- Verifying configurable/grouped/bundle product imports work
- Testing products with special pricing or attributes

Example: `product_id=123`

| qty

Sets the quantity for the test order. Defaults to 1.

When to use:

- Testing stock deduction with multiple items
- Verifying quantity-based shipping rules
- Testing tier pricing behavior

Example: `qty=5`

| country

Sets the country code for the billing and shipping address. Defaults to "NL" (Netherlands).

When to use:

- Testing orders from different countries
- Verifying tax calculation for international orders
- Testing country-specific shipping methods
- US orders will automatically include state information (Texas/TX)

Example: `country=US` , `country=BE` , `country=DE`

| lvb

When set to `1` , simulates a Fulfillment by Bol (FBB/LVB) order. These orders arrive with status "shipped" since bol.com handles fulfillment.

When to use:

- Testing FBB/LVB order handling
- Verifying auto-ship functionality

- Testing stock behavior for fulfilled orders

Example: `lvb=1`

What Gets Created

A test order includes:

Customer Information:

- Name: "Test From Channable"
- Email: dontemail@me.net
- Company: "TestCompany" / "Do Not Ship"
- Phone: 01234567890

Address:

- Street: "Street" with house number 1
- City: UTRECHT
- Zip: 1000 AA
- Country: Based on `country` parameter (default: NL)

Order Details:

- Channel: "Channable Test"
- Order ID: TEST-`{random}`
- Channable ID: `{random 5-digit number}`
- Shipping method: Prime
- Payment method: bol
- Commission: 10% of product price
- Memo: "Test Order from Channable"

Testing Scenarios

Basic Order Import Test

1. Create a simple test order without parameters
2. Verify the order appears in Magento
3. Check shipping method is assigned correctly
4. Verify invoice is created (if auto-invoice is enabled)

5. Check customer data (or guest order if customer import is disabled)

International Order Test

```
bin/magento channable:order:simulate --store-id=1 --country-code=DE
```

Verify:

- Correct country appears on addresses
- Tax calculation is correct for the country
- Shipping method handles international orders

FBB/LVB Order Test

```
bin/magento channable:order:simulate --store-id=1 --lvb=1
```

Verify:

- Order imports with "shipped" status
- Auto-ship creates shipment (if enabled)
- Stock is handled correctly (if "Disregard stock" is enabled)
- Order reaches "Complete" status (if auto-invoice + auto-ship enabled)

Specific Product Test

```
bin/magento channable:order:simulate --store-id=1 --product-id=42 --qty=3
```

Verify:

- Correct product is added to order
- Quantity matches parameter
- Stock is deducted correctly
- Pricing is calculated properly

Troubleshooting Test Orders

"Order import not enabled for this store"

Order import must be enabled for the store view you're testing.

Solution: Go to **Channable** → **Orders** → **Enable** and set to Yes for your store view.

"Product not found" or empty order

The random product selector couldn't find an eligible product.

Solution:

- Use `product_id` parameter with a specific enabled product
- Ensure you have enabled simple products in the store
- Check that products have stock (unless backorders are enabled)

Test order created but configuration seems wrong

The test order bypasses some validation that real orders go through.

Solution: After testing basic functionality, also test with a real (or staging) Channable webhook to verify the full flow.

Expert: Customizing Test Data

For developers who need to modify the test order data structure, the simulation logic is in:

```
Magmodules\Channable\Service\Order\ImportSimulator
```

Key Methods

getTestData(array \$params): array

Returns the complete order data array in Channable format. This is where you can modify:

- Customer details (name, email, phone)
- Address information
- Price structure (subtotal, shipping, commission)
- Product data format

getProductData(array \$params): array

Retrieves product information for the test order. Supports:

- Specific product by ID (`product_id` parameter)
- Random product selection
- Custom price override (`price` parameter—not exposed in URL)

getRandomProduct(): DataObject

Selects a random enabled product. Respects:

- Store filter
- Product type settings (includes grouped/bundle if enabled in config)
- Status filter (enabled products only)

Example: Testing with Custom Data

If you need to test specific scenarios not covered by URL parameters, you can temporarily modify `getTestData()`

:

```
// In getTestData(), modify the customer data:
"customer" => [
  "gender" => "female",
  "phone" => "0612345678",
  "email" => "test@example.com",
  "first_name" => "Custom",
  "last_name" => "Test",
  "company" => "My Test Company",
  "business_order" => true, // Test B2B order
],
```

Or test different price scenarios:

```
"price" => [
  "currency" => "EUR",
  "subtotal" => 100.00,
  "shipping" => 5.95,
  "discount" => 10.00,
  "total" => 95.95,
  "transaction_fee" => 2.50, // Test CDiscount fee
  "commission" => 15.00,
],
```

Note: Remember to revert any code changes after testing, or create a separate development branch for testing modifications.

Need More Help?

Documentation:

- [All Help Articles](#) - Complete documentation overview

Support:

- [Contact Support](#) - Get help from our team

For a complete overview of features and configuration options, see the Channable Connect extension on magmodules.eu

All articles for Channable Connect

Installation

1	Installation using Composer (recommended)
2	Installation using the Adobe Marketplace
3	Install through FTP and SSH

Configuration

1	Quick Start Guide
2	Product Feed Configuration
3	Orders Configuration
4	Product Updates Configuration

Troubleshooting

1	The price of at least one product is missing a currency
2	Troubleshooting
3	Test Orders (current)
4	Test Feed

Grids

1	Returns Grid
2	Product Updates Grid
3	Orders Grid

Background

1	How does the Return process work
2	CLI Commands
3	Best Practices

