

Custom Magento2 Module for Order processing status via API

1. Installation

1. Unzip the Module inside app/code
2. Run php bin/magento setup: upgrade, php bin/magento s:d:c
3. Make sure the Module is enabled

2. Token Generation

1. Generate Auth Key from Admin

Goto System >> Integration >> Create new Integration >> select "Smartworking - Order Status update via API"

Integrations 🔍 🔔 2 👤 local ▼

[Add New Integration](#)

[Reset Filter](#) 1 records found 20 ▼ per page < 1 of 1 >

Name	Status	Activate	Edit	Delete
<input type="text"/>	<input type="text"/>			
OrderProcessing ▲ Integration not secure	Active	Reauthorize	✎	🗑

Copyright © 2025 Magento Commerce Inc. All rights reserved. Magento ver. 2.4.6
[Privacy Policy](#) | [Report an Issue](#)

Edit "OrderProcessing" Integration ← Back Save

- Permissions
 - All Users
 - Locked Users
 - User Roles
- Action Log
 - Report
 - Bulk Actions
- Other Settings
 - Notifications
 - Show Toolbar
 - Show List
 - Mark as Read
 - Remove
 - Custom Variables
 - Manage Encryption Key
- Global Search
- Smartworking - Order Status update via API**

5. Use Access Token as Bearer Auth in Postman

Edit "OrderProcessing" Integration

← BackSave

Current User Identity Verification

Your Password *

Integration Details

Consumer Key

dfsY555quf0Jecadj5q8j13bfhlkq2r

Consumer Secret

2ogqcm1d6o82iv4kb9g3280aog234w7i

Access Token

5fwq4t3sygpvecjzlm78wae03y3qf7w

Access Token Secret

3hx5y9y2bwo5t07vyxb2n14chwwsjji

3. Optimization

- API is designed and developed considering that different systems will be accessing the order status update functionality for single and multiple order updates
- The order will be processed based on the parameter in the API request (synchronous: true/false) , For bulk updates, synchronous should be passed as False.
- When synchronous is false, orders are processed in the background through the Queue process

4. Caching

Magento's default file based caching is implemented

5. Update Order status with synchronous = false

URL : rest/V1/orderprocessing/changestatus/
TYPE : PUT

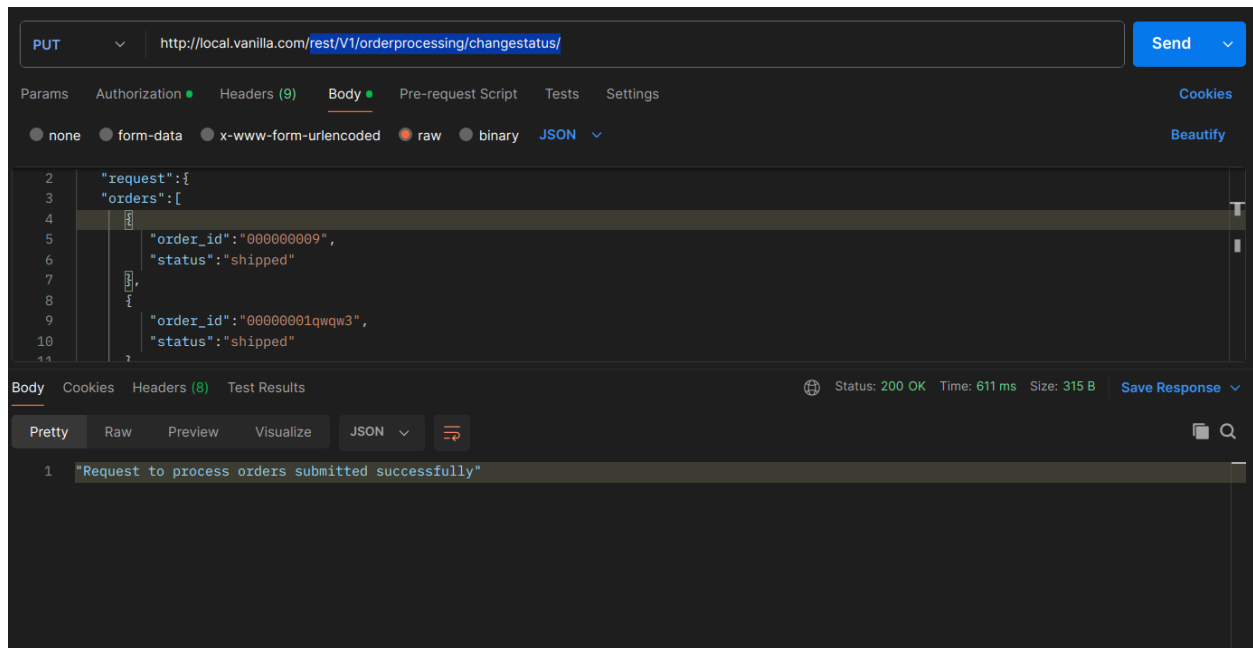
REQUEST :

```

{
  "request":{
    "orders":[
      {
        "order_id":"000000009",
        "status":"shipped"
      },
      {
        "order_id":"00000001qwqw3",
        "status":"shipped"
      }
    ],
    "synchronous" : false
  }
}

```

RESPONSE : "Request to process orders submitted successfully"



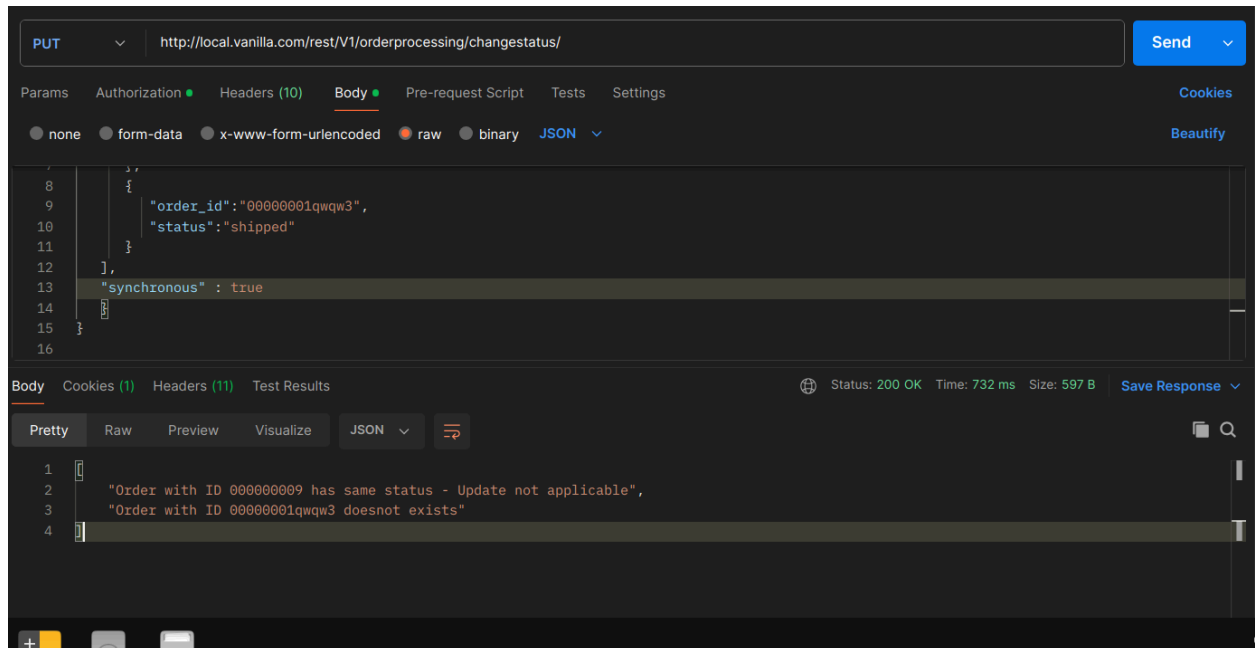
6. Update Order status with synchronous = true

RESPONSE :

```

[
  "Order with ID 000000009 has same status - Update not applicable",
  "Order with ID 00000001qwqw3 doesnot exists"
]

```



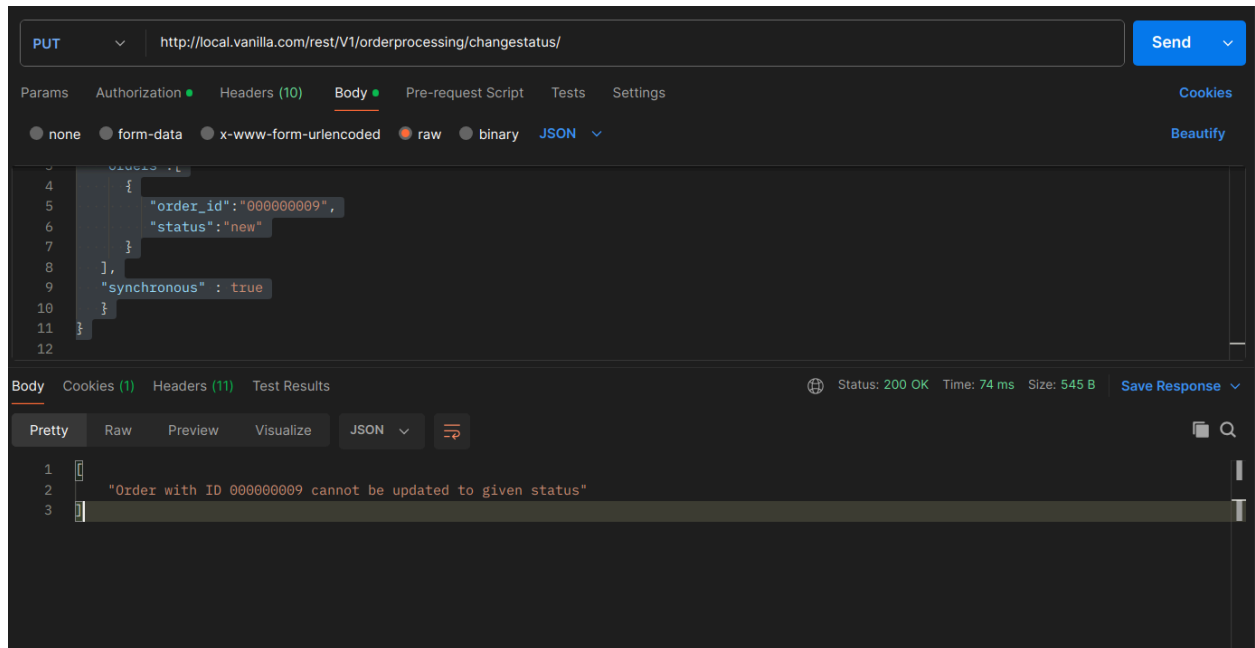
7. Order update when status Transition is not possible

REQUEST

```
{
  "request":{
    "orders":[
      {
        "order_id":"000000009",
        "status":"new"
      }
    ],
    "synchronous" : true
  }
}
```

Response :

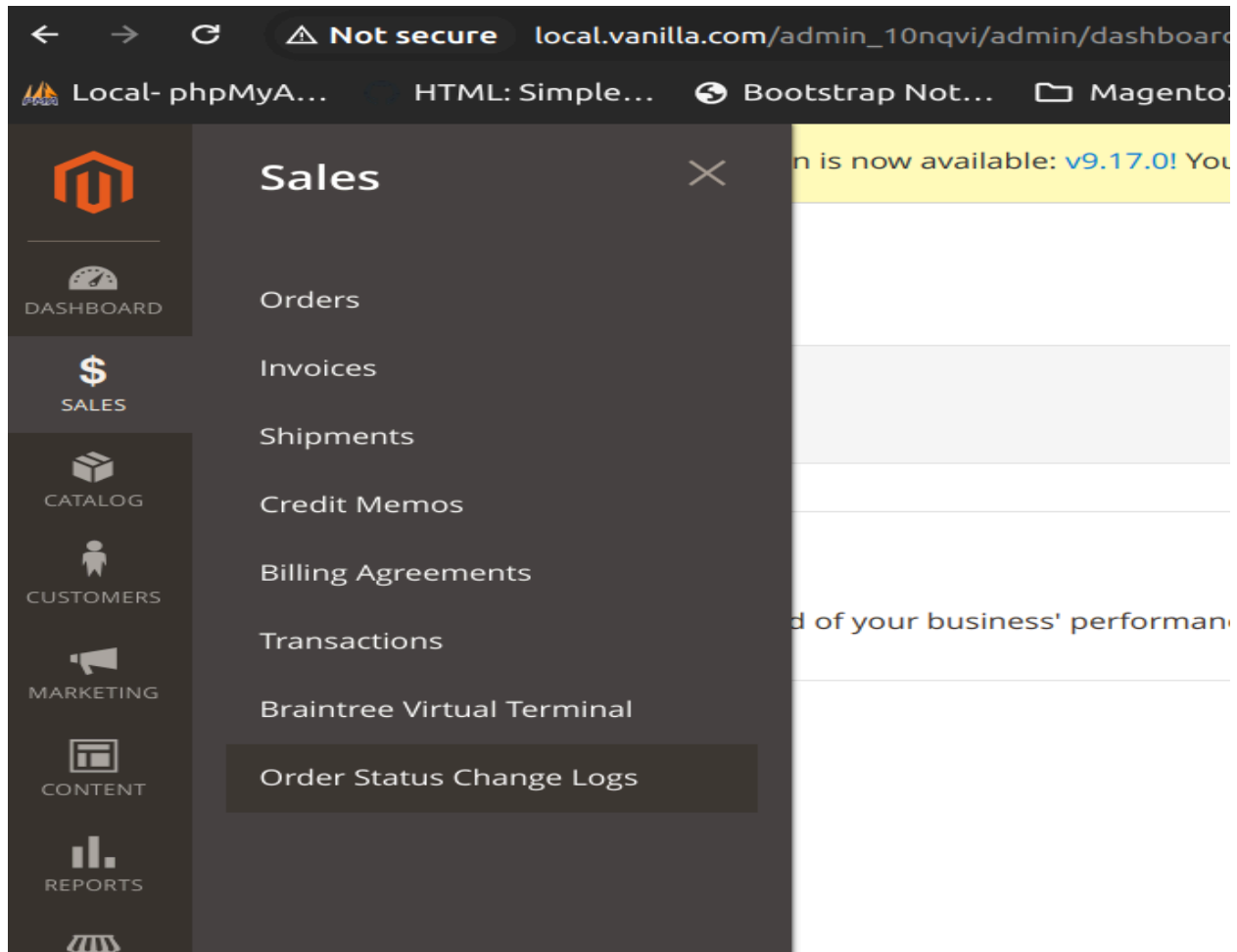
```
[
  "Order with ID 000000009 cannot be updated to given status"
]
```



8. Order status logs

All the updates in the status for an order are logged in the database and can be viewed in the admin

Goto Sales >> Order Status Change Logs



Order Status Change Logs

Search  2  local ▾

<input type="checkbox"/>	ID	Order ID	Old Status	New Status	Created
<input type="checkbox"/>	8	000000009	processing	shipped	Mar 22, 2025 4:01:13 PM

Order status change logs with mass action and filters

Filters

Default View

Columns

ID

from

to

Created

from

to

Order ID

Old Status

New Status

Cancel

Apply Filters

Actions

1 records found

20 per page

<

1

of 1

>

	ID	Order ID	Old Status	New Status	Created
<input type="checkbox"/>	9	000000009	shipped	closed	Apr 16, 2025 3:24:50 PM

Copyright © 2025 Magento Commerce Inc. All rights reserved.

Magento ver. 2.4.6

Order Status Change Logs

Search

Notifications

local

✓ A total of 1 record(s) have been deleted.

Filters

Default View

Columns

Actions

0 records found

20 per page

<

1

of 1

>

	ID	Order ID	Old Status	New Status	Created
We couldn't find any records.					

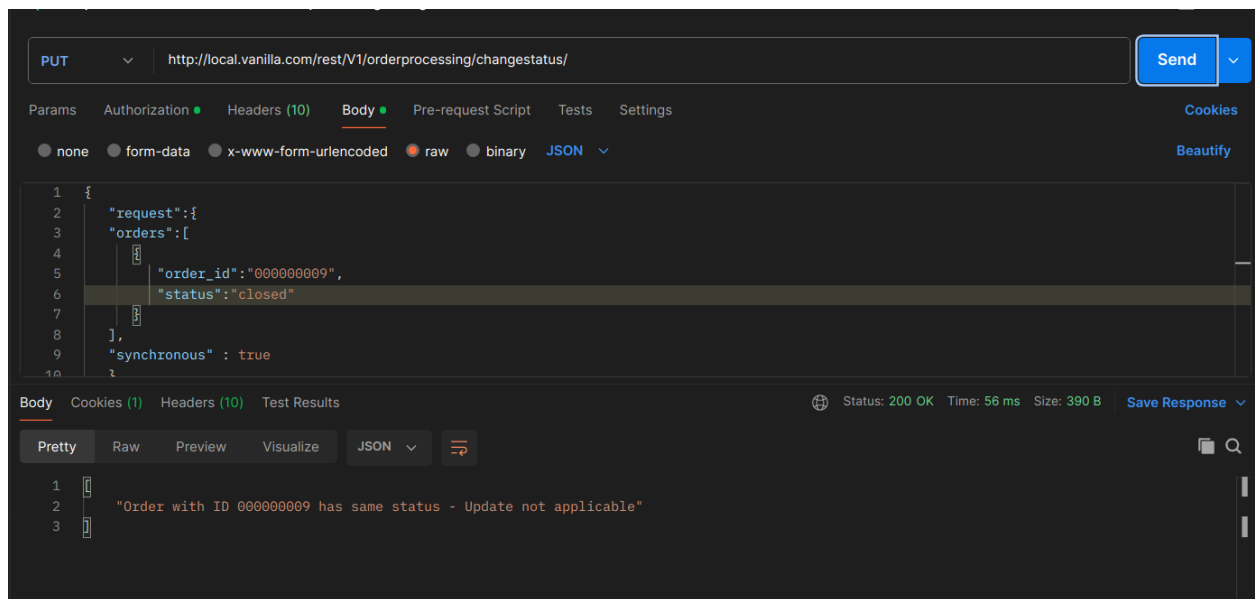
Copyright © 2025 Magento Commerce Inc. All rights reserved.

Magento ver. 2.4.6
[Privacy Policy](#) | [Report an Issue](#)

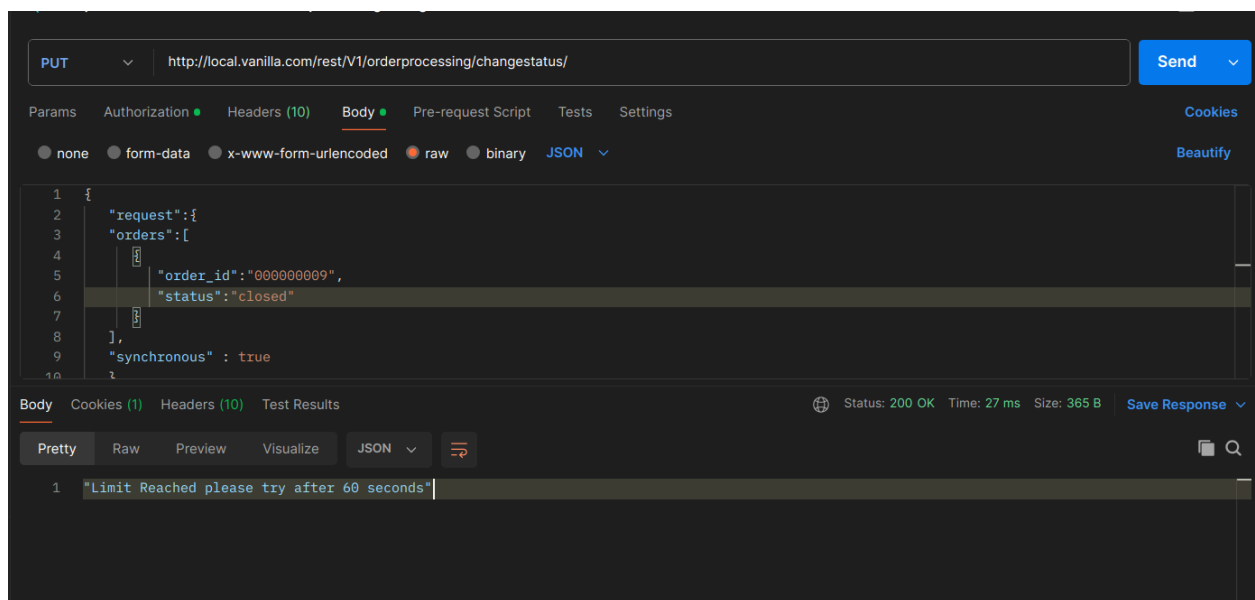
9. Rate Limiter

Added a Class to limit API calls per minute from a particular IP address. The solution is based on sessions and variables related to limit and usage can be modified in the code

1 Normal Request



2. Rate Limit reached



10. Unit Test

Unit test are located at
app/code/Smartworking/CustomOrderProcessing/Test/Unit

To Test from CLI run : `vendor/phpunit/phpunit/phpunit -c dev/tests/unit/phpunit.xml.dist app/code/Smartworking/CustomOrderProcessing/Test/Unit/Model/Api/OrderProcessingTest.php`

```
maneshlalwani@EE-MaheshLalwani: /var/www/html/vanilla$ vendor/phpunit/phpunit/phpunit -c dev/tests/unit/phpunit.xml.dist app/code/Smartworking/CustomOrderProcessing/Test/Unit/Model/Api/OrderProcessingTest.php
PHPUnit 9.6.21 by Sebastian Bergmann and contributors.

.....
Time: 00:00.017, Memory: 12.00 MB

OK (5 tests, 5 assertions)
maneshlalwani@EE-MaheshLalwani: /var/www/html/vanilla$
```

11. Integration Test

1. Create an integration database

```
CREATE DATABASE magento_integration_tests;
```

2. Update Database configurations in file below

```
dev/tests/integration/etc/install-config-mysql.php
```

3. To avoid repeated installation of database, make below configurations

```
Update <const name="TESTS_CLEANUP" value="disabled"/>
```

In file

```
dev/tests/integration/phpunit.xml
```

4. Test from CLI: `vendor/phpunit/phpunit/phpunit -c dev/tests/integration/phpunit.xml app/code/Smartworking/CustomOrderProcessing/Test/Integration/Model/Api/OrderProcessingTest.php`