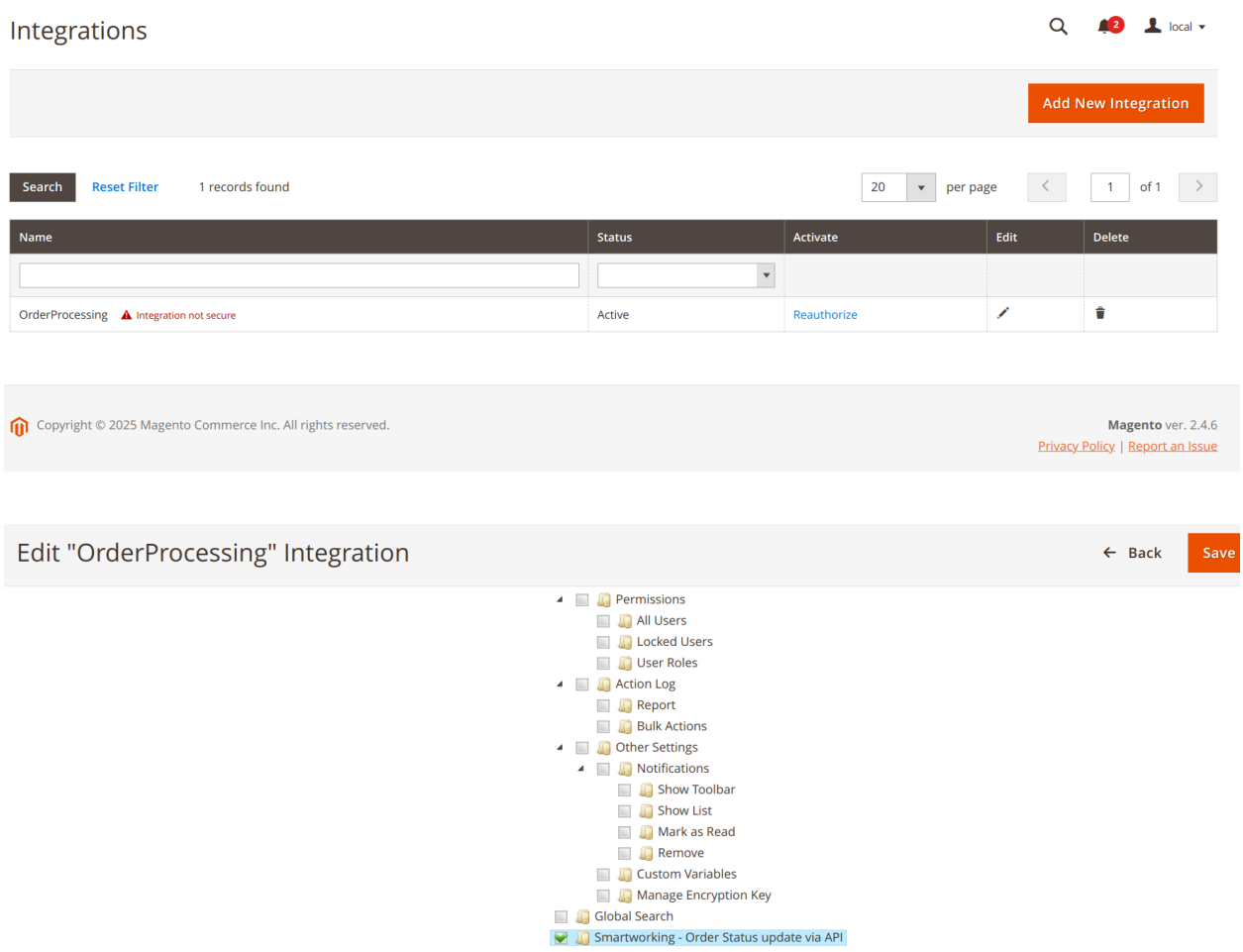
Custom Magento2 Module for Order processing status via API

1. Installation

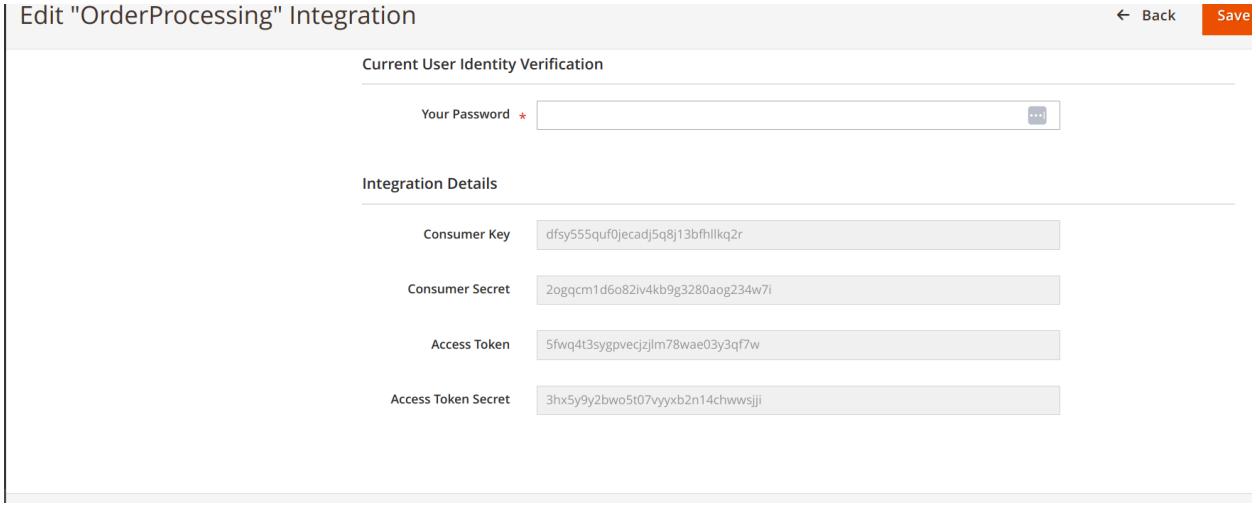
1. Unzip the Module inside app/code
2. Run php bin/magento setup: upgrade, php bin/magento s:d:c
3. Make sure the Module is enabled

2. Token Generation

1. Generate Auth Key from Admin
   Goto System >> Integration >> Create new Integration >> select "Smartworking - Order Status update via API"



5. Use Access Token as Bearer Auth in Postman



3. Optimization
    a. API is designed and developed considering that different systems will be accessing the order status update functionality for single and multiple order updates
    b. 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.
    c. When synchronous is false, orders are processed in the background through the Queue process

4. Caching
    a. No external caching is implemented as the API call is of type PUT. Caching is helpful for GET calls only.

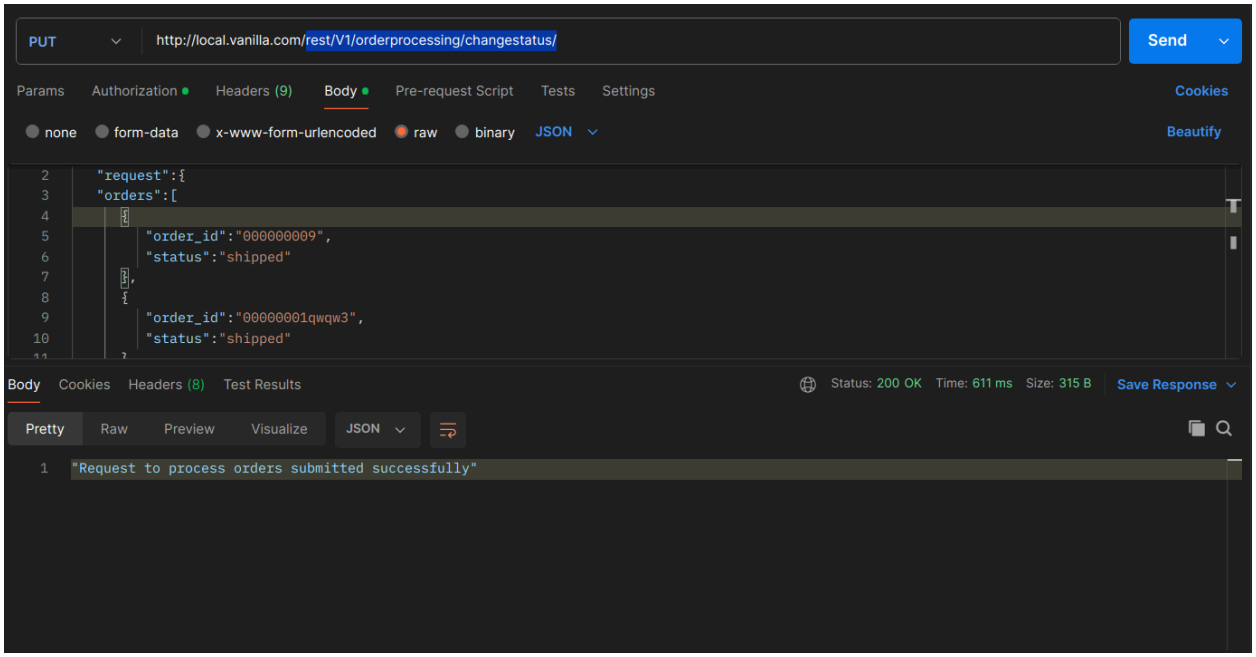
5. Update Order status with synchronous = false

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

REQUEST :

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

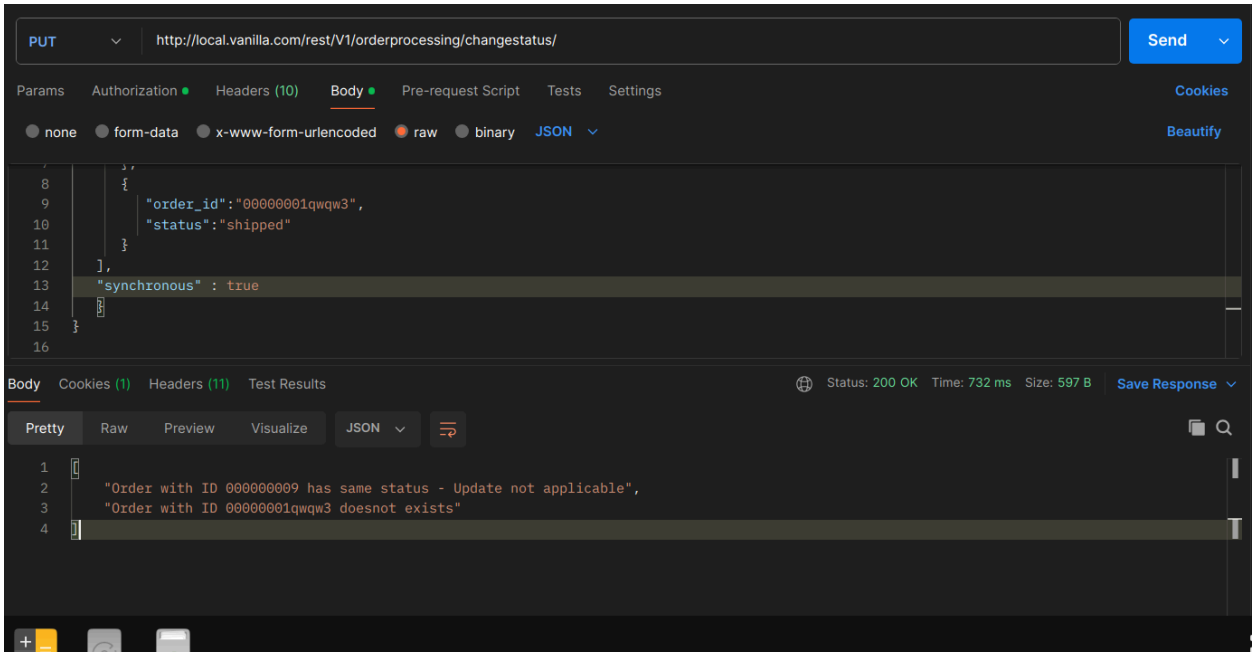
RESPONSE : "Request to process orders submitted successfully"



6. Update Order status with synchronous = true

RESPONSE :

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



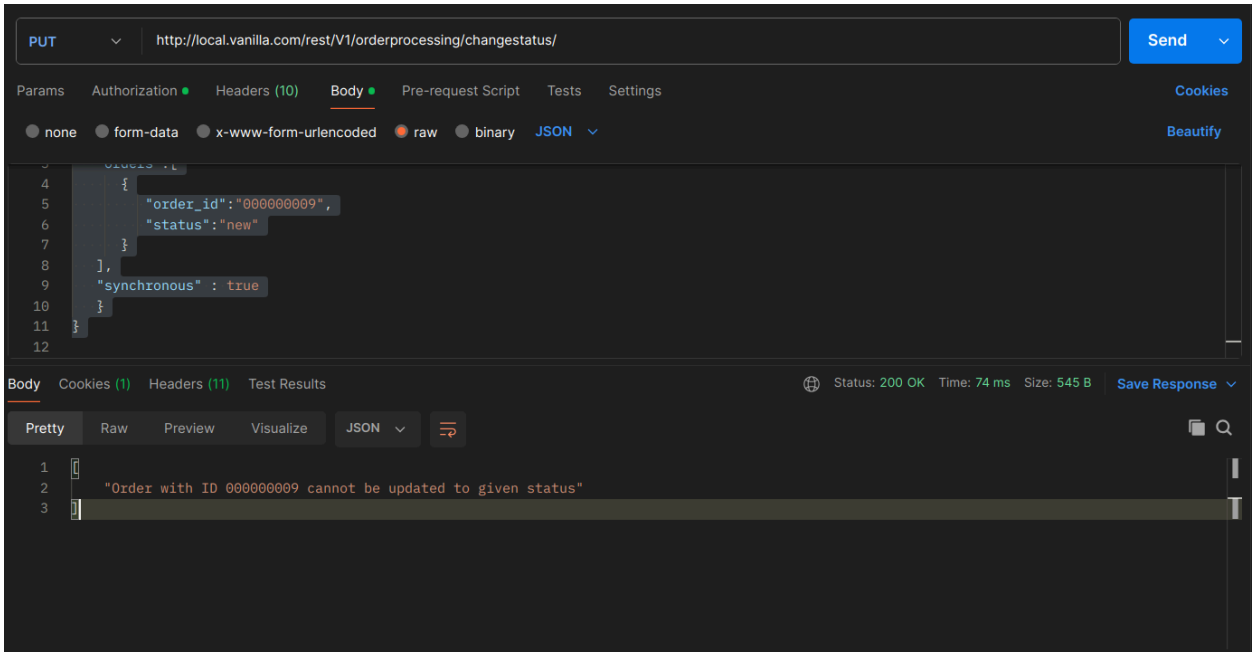
7. Order update when status Transition is not possible

REQUEST

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

Response :

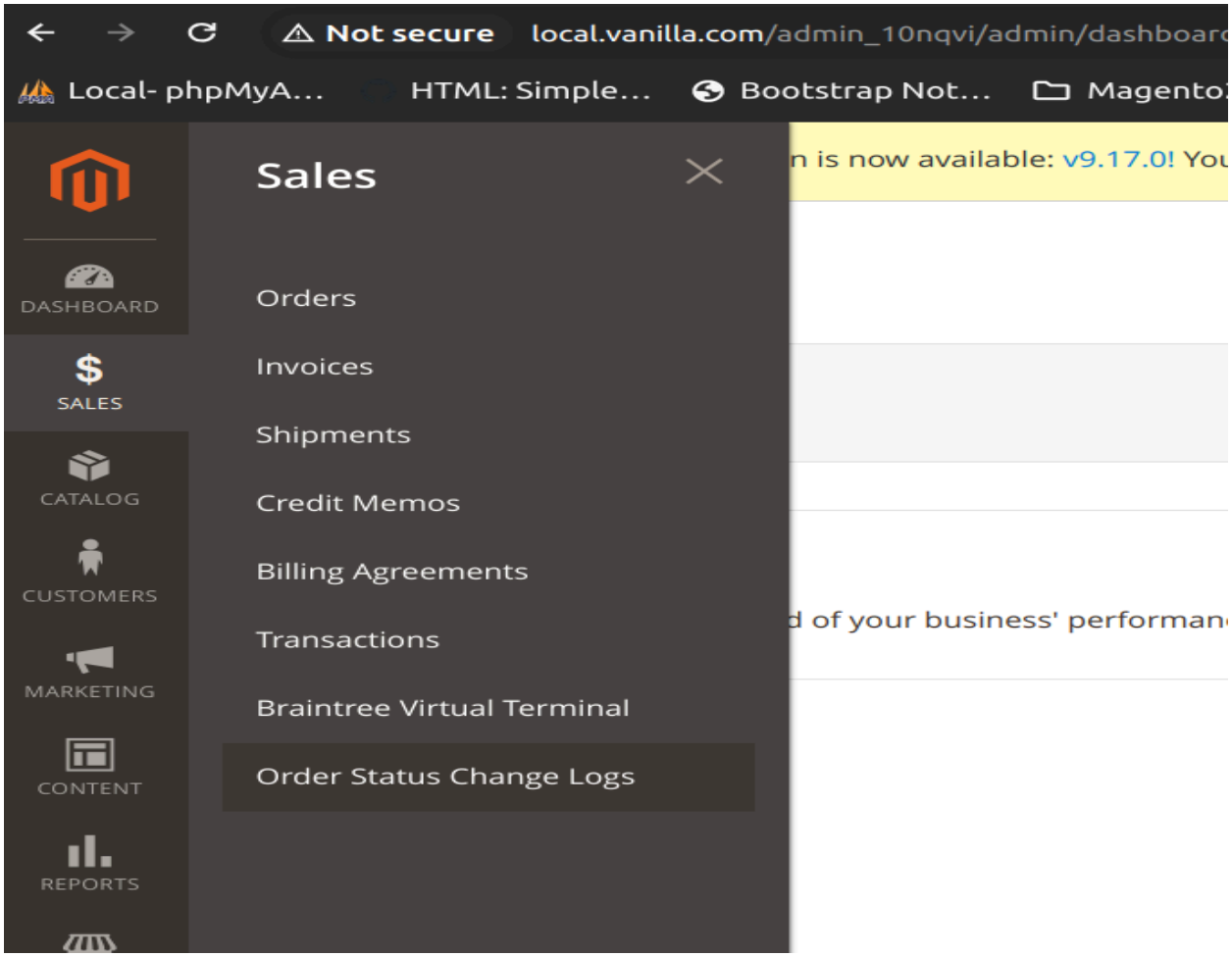
```
[
  "Order with ID 000000009 cannot be updated to given status"
]
```

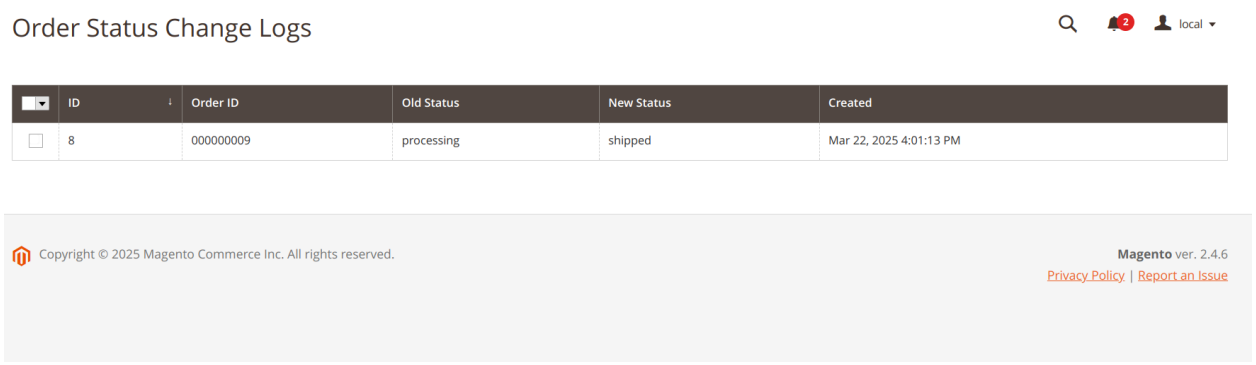


8. Order status logs

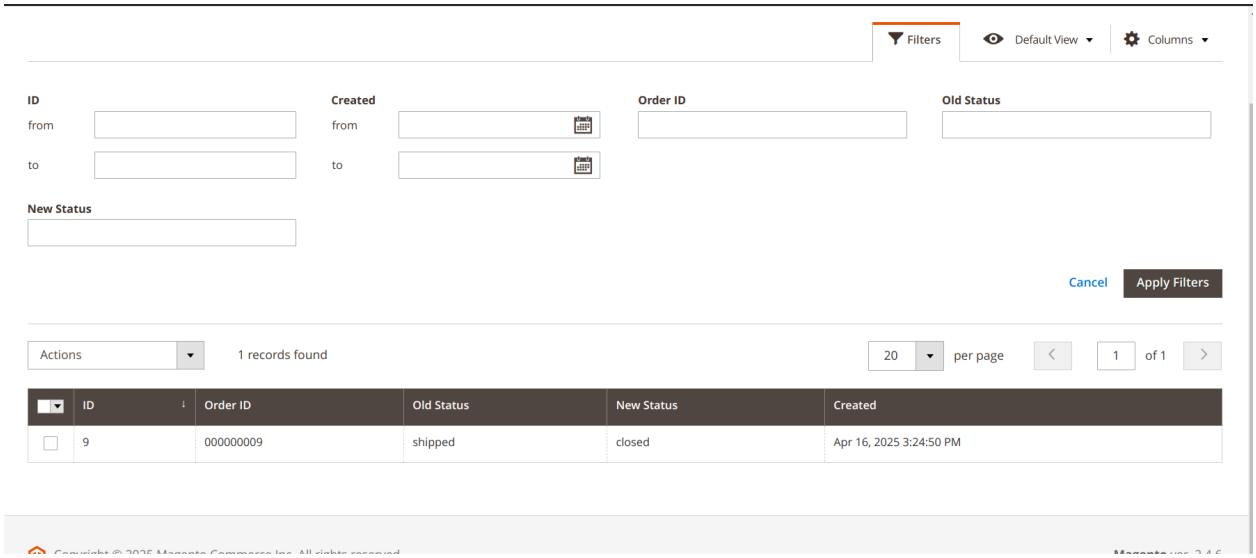
All the updates in the status for an order are logged in the database and can be viewed in the admin

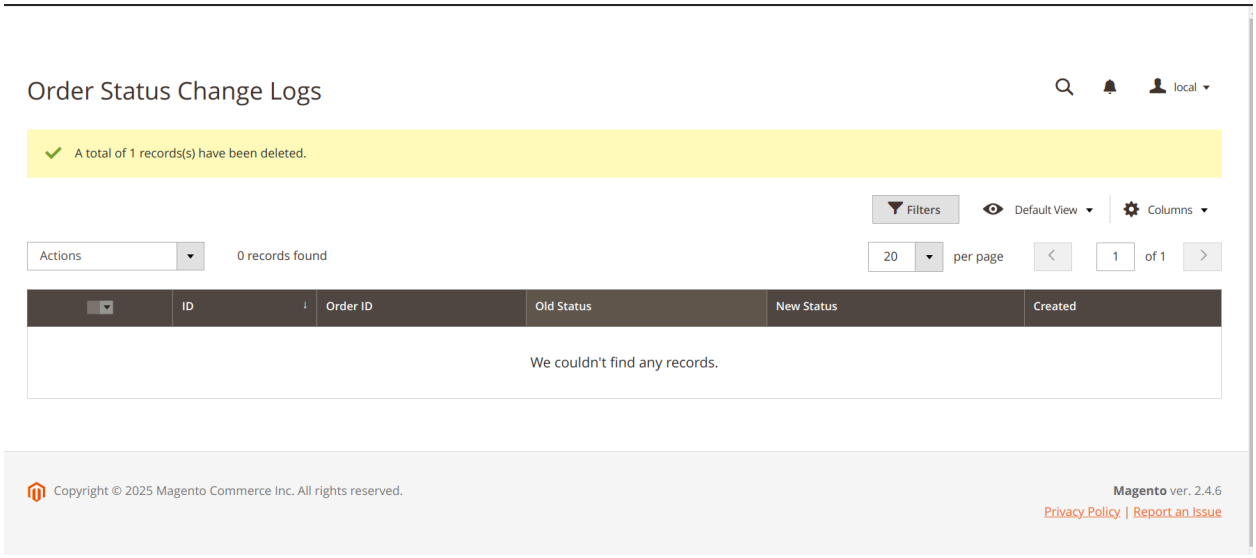
Goto Sales >> Order Status Change Logs





Order status change logs with mass action and filters

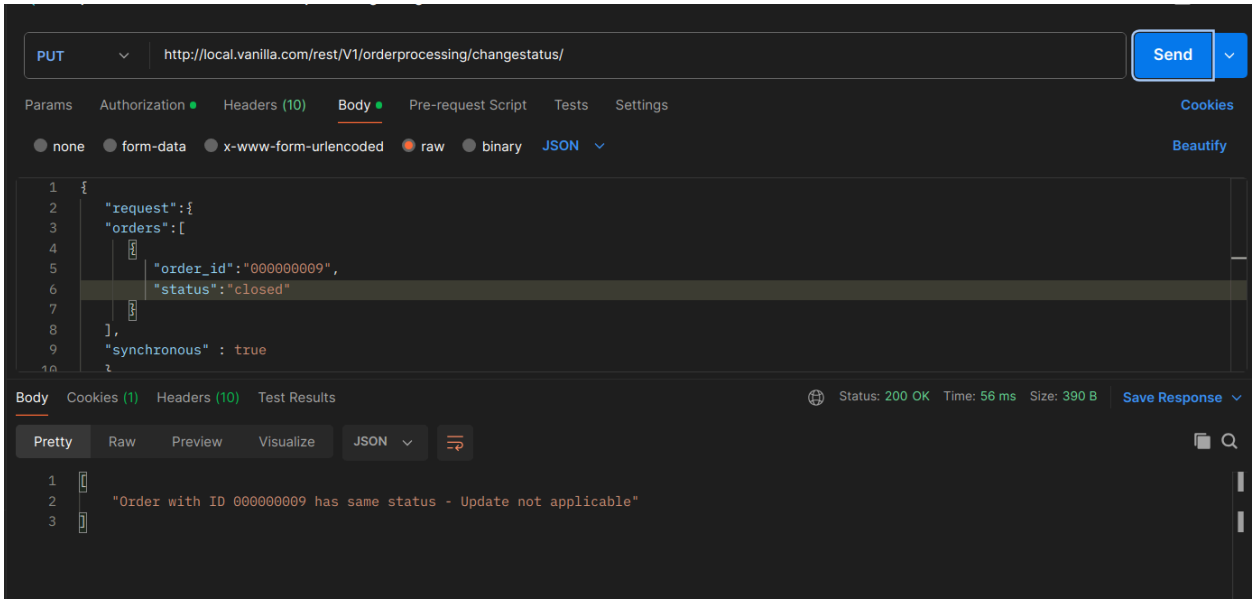




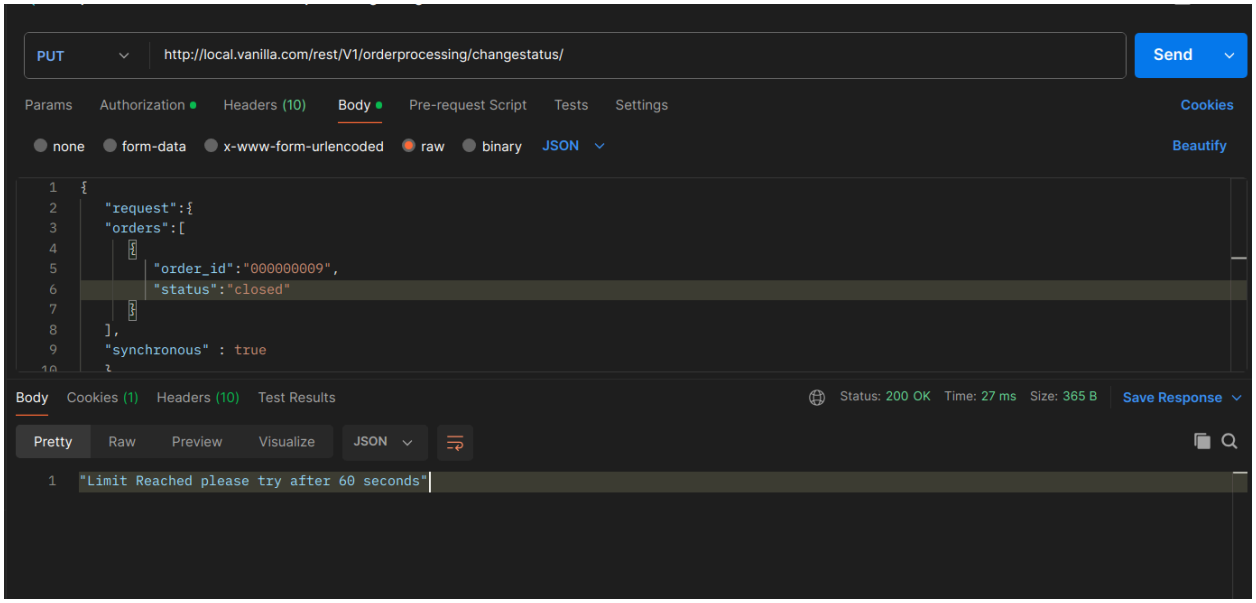
9. Rate Limiter

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



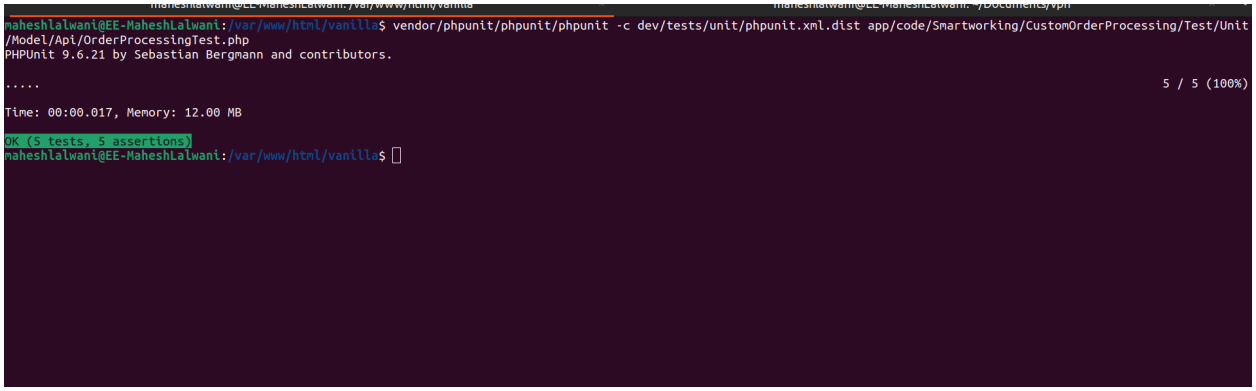
2. Rate Limit reached



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



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