

Introduction

Data Model Explained

Coming soon (Drafted)

Project Structure Explained

Coming soon (Not Started)

Quick Start

Install Nacos

It's very simple, see here: <https://nacos.io/en-us/docs/quick-start.html>

Please run it in standalone mode for non-cluster setup.

Default username and password are "nacos".

Default port is 8848

Install MySQL (version > 5.6)

run the sql script below:

```
cd ${anyware_home}/sql
mysql -uroot -p < anyware_product.sql
mysql -uroot -p < anyware_ware.sql
mysql -uroot -p < anyware_order.sql
```

Install the warehouse application

Clone the repo

```
git clone https://github.com/beiji-ma/anyware.git
```

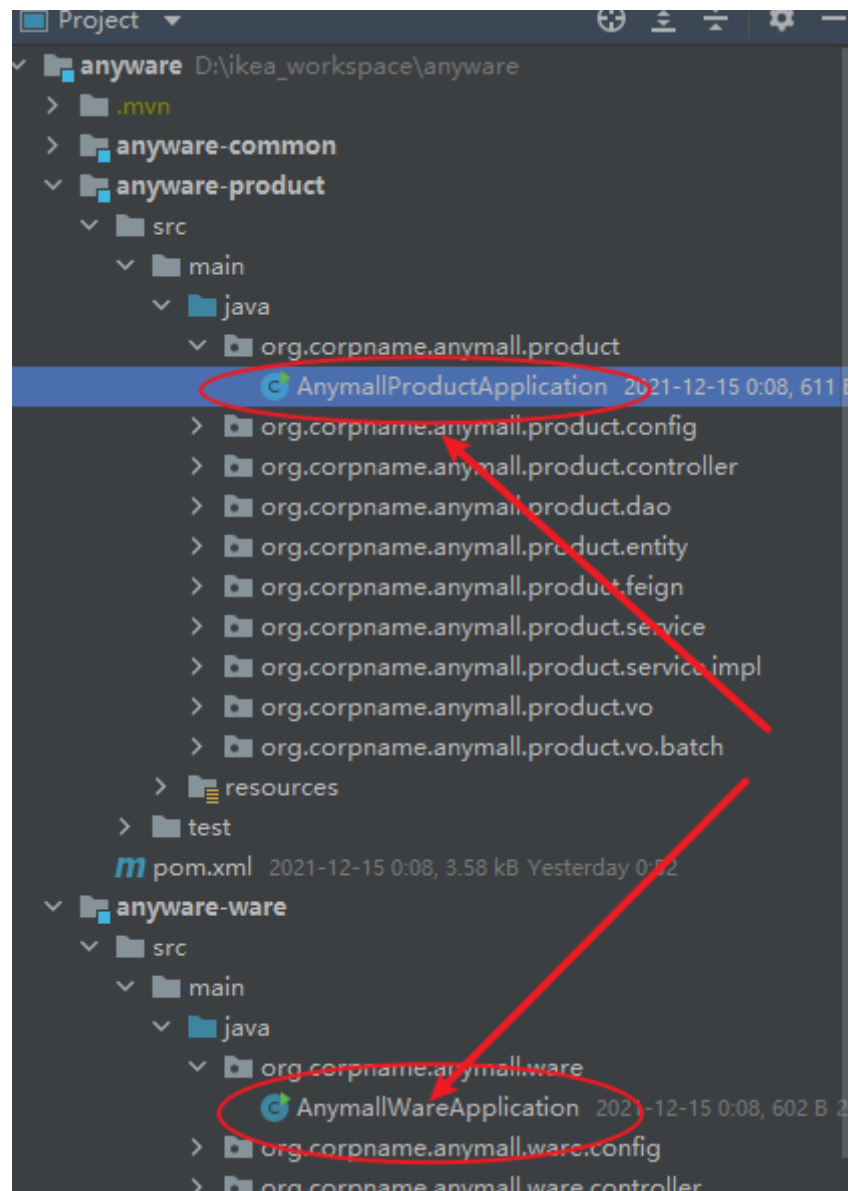
Update the IP address in each application.yml/application-dev.yml if you deploy the Nacos and the app on the different host server.

```
spring:
  datasource:
    username: root
    password: 123456
    url: jdbc:mysql://localhost:3306/anyware_war
    driver-class-name: com.mysql.cj.jdbc.Driver
  cloud:
    nacos:
      discovery:
        server-addr: 127.0.0.1:8848
```

You need update the pom.xml to remove the setup related with nexus:

```
<distributionManagement>
  <snapshotRepository>
    <id>matrixone_snapshot</id>
    <url>http://localhost:8081/nexus/content/repositories/matrixone_snapshot</url>
  </snapshotRepository>
  <repository>
    <id>matrixone_release</id>
    <url>http://localhost:8081/nexus/content/repositories/matrixone_release</url>
  </repository>
</distributionManagement>
```

Startup 2 modules below:



Congratulations if you could see the 2 service below from the console of Nacos:

Service Name	Group Name	Cluster Count	Instance Count	Healthy Instance Count	Operation
anyware-product	DEFAULT_GROUP	1	1	1	Details Code Example Delete
anyware-ware	DEFAULT_GROUP	1	1	1	Details Code Example Delete

Import the sample data:

Inventory.json

Inventory DB / BatchAdd

POST localhost:10010/ware/waresku/

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> file	inventory.json ×

Product.json

Inventory DB / BatchAddProduct

POST localhost:10020/product/product/

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> file	products.json ×

Test:

Case 1.1: list all Inventory information

GET localhost:10010/ware/waresku/

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 [{
2   ... "product_id": "33",
3   ... "art_id": "1",
4   ... "amount_of": 4
5 }]
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "success",
3   "code": 0,
4   "page": {
5     "totalCount": 4,
6     "pageSize": 10,
7     "totalPage": 1,
8     "currPage": 1,
9     "list": [
10      {
11        "id": 1,
12        "name": "leg",
13        "stock": 12,
14        "art_id": 1
15      },
16      {
17        "id": 2,
18        "name": "screw",
19        "stock": 17,
20        "art_id": 2
21      },
22      {
23        "id": 3,
24        "name": "seat",
25        "stock": 2,
26        "art_id": 3
27      },
28      {
29        ...
30      }
31    ]
32   }
33 }
```

Case 1.2: Update stock

Inventory / Update Stock

PUT `localhost:10010/inventory/99?stock=1111`

Params **Authorization** Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
<input checked="" type="checkbox"/> stock	1111

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "name": "bottom",  
3   "stock": 1111,  
4   "art_id": 99  
5 }
```

Case 1.3: Add Article(Inventory Item)

Inventory / Add Inventory Item

POST `localhost:10010/ware/waresku/`

Params **Authorization** Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded **☒ raw** ☐ binary ☐ GraphQL JSON

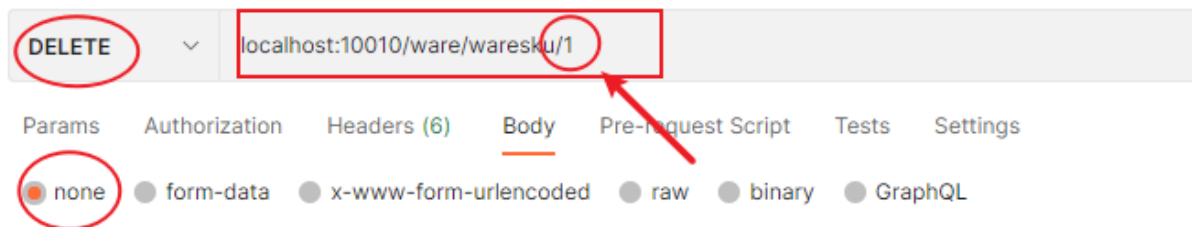
```
1 {  
2   "name": "bottom",  
3   "stock": 100,  
4   "art_id": 99  
5 }
```

Body Cookies Headers (3) Test Results

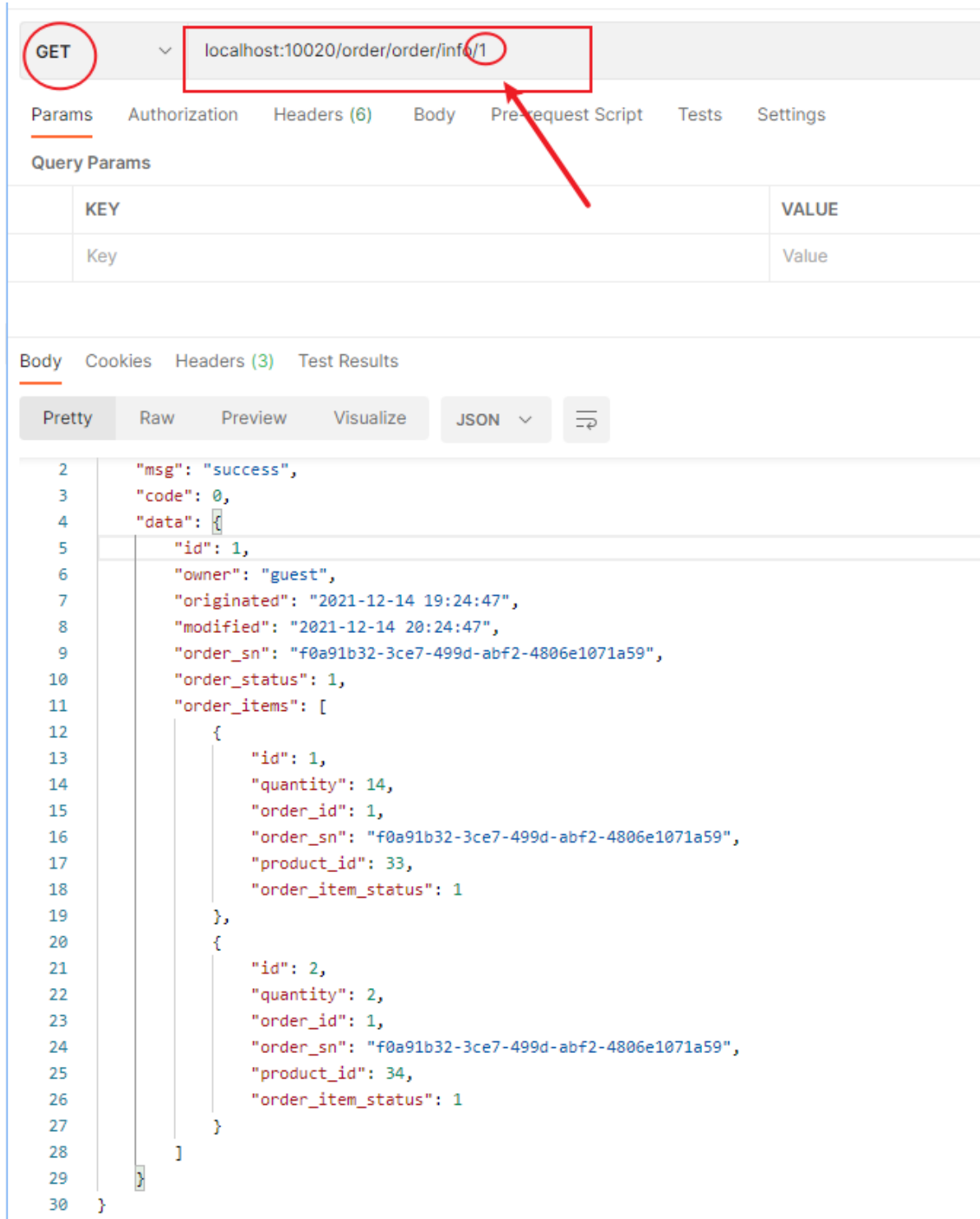
Pretty Raw Preview Visualize JSON

```
1 {  
2   "msg": "success",  
3   "code": 0  
4 }
```

Case 1.4: Delete Article (Inventory Item)



Case 2.1: Query Order



Case 2.2: Create Order

Order DB / Create Order

POST localhost:10020/order/order/

Params Authorization Headers (8) **Body** Pre-request Script T

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary

```
1 [{
2   ... "product_id": "33",
3   ... "quantity": "14"
4 },
5 {
6   ... "product_id": "34",
7   ... "quantity": "2"
8 }]
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "success",
3   "code": 0
4 }
```

Case 2.3: Ware Order Task

GET localhost:10010/ware/ordertask/

Params Authorization Headers (6) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE
Key	Value

dy Cookies Headers (3) Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "msg": "success",
3   "code": 0,
4   "page": {
5     "totalCount": 3,
6     "pageSize": 10,
7     "totalPage": 1,
8     "currPage": 1,
9     "list": [
10      {
11        "id": 1,
12        "orderId": 1,
13        "orderSn": "f0a91b32-3ce7-499d-abf2-4806e1071a59",
14        "taskStatus": 1,
15        "taskComment": "say something, god.",
16        "originated": "2021-12-15 00:26:53",
17        "modified": "2021-12-15 00:26:53"
18      },
19      {
20        "id": 2,
21        "orderId": 1,
22        "orderSn": "f0a91b32-3ce7-499d-abf2-4806e1071a59",
```

Case 3.1: Available stock for each product

GET

localhost:10020/product/product/stock/

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

KEY	VALUE
Key	Value

BodyCookiesHeaders (3)Test Results

PrettyRawPreviewVisualizeJSON

```
1 {
2   "msg": "success",
3   "code": 0,
4   "page": {
5     "totalCount": 3,
6     "pageSize": 10,
7     "totalPage": 1,
8     "currPage": 1,
9     "list": [
10      {
11        "name": "Dining Chair",
12        "stock": 2,
13        "product_id": 33
14      },
15      {
16        "name": "Dinning Table",
17        "stock": 1,
18        "product_id": 34
19      },
20      {
21        "name": "Spis",
22        "stock": 0,
23        "product_id": 35
24      }
25    ]
26  }
27 }
```

Case 4.1: Lock Stock for Order (Incompleted)

Product DB / Lock Order Ware Order Task

GET

localhost:10020/order/order/lock/1

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE
Key	Value

Body

Cookies

Headers (3)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "msg": "success",
3   "code": 0,
4   "data": {
5     "originated": "2021-12-15 02:27:35",
6     "modified": "2021-12-15 02:27:35",
7     "order_id": 1,
8     "order_sn": "f0a91b32-3ce7-499d-abf2-4806e1071a59",
9     "task_status": 1,
10    "task_comment": "say something, god.",
11    "products": [
12      {
13        "quantity": 14,
14        "originated": "2021-12-15 02:27:35",
15        "modified": "2021-12-15 02:27:35",
16        "order_id": 1,
17        "product_id": 33,
18        "task_item_status": 1,
19        "articles": [
20          {
21            "amountOf": 4,
22            "originated": "2021-12-15 02:27:35",
```

Case 5.1: Query Product Article Relation

GET localhost:10020/product/productarticlerelation/

Params Authorization Headers (6) Body Pre-request Script Tests Settings

ody Cookies Headers (3) Test Results

Pretty Raw Preview Visualize JSON ↕

```
30 product_id": 34,  
31 "art_id": 2,  
32 "amount_of": 8  
33 },  
34 {  
35 "id": 6,  
36 "product_id": 34,  
37 "art_id": 4,  
38 "amount_of": 1  
39 },  
40 {  
41 "id": 7,  
42 "product_id": 35,  
43 "art_id": 1,  
44 "amount_of": 3  
45 },  
46 {  
47 "id": 8,  
48 "product_id": 35,  
49 "art_id": 2,  
50 "amount_of": 11  
51 },  
52 {  
53 "id": 9,  
54 "product_id": 35,  
55 "art_id": 5,  
56 "amount_of": 2  
57 },  
58 {  
59 "id": 19,  
60 "product_id": 33,  
61 "art_id": 1,  
62 "amount_of": 4  
63 }  
64 ]  
65 }  
66 }
```

5.2: Create Product Article Relation

ProductArticleRelation / Create Product Article Relation

POST localhost:10020/product/productarticlerelation/

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▾

```
1 {
2   ... "product_id": "5",
3   ... "art_id": "1",
4   ... "amount_of": "4"
5 }
```

5.3: Update Product Article Relation

ProductArticleRelation / Update Product Article

PUT localhost:10020/product/productarticlerelation/1

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▾

```
1 {
2   ... "id": 1,
3   ... "amount_of": "4"
4 }
```

5.4: Delete Product Article

ProductArticleRelation / Delete Product Article

DELETE localhost:10020/product/productarticlerelation/1

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value