Structure

A-Merch

- **AM-1. Authorization and authentication**
  - 1.1. Log in
  - 1.2. Log out
  - 1.3. Restore password
  - 1.4. Registration
- **AM-2. Items management**
  - 2.1. Create item
  - 2.2. View item
  - 2.3. Search item
  - 2.4. Edit item
  - 2.5. Delete item
- **AM-3. Requests management (HR)**
  - 3.1. Create order
  - 3.2. Get user's orders
  - 3.3. Update order
  - 3.4. Delete order
  - 3.5. Close order
  - 3.6/4.4. Add comment to order
- **AM-4. Requests management (MDM)**
  - 4.1. View the order
  - 4.2. Accept the order
  - 4.3. Reject the order
  - 4.5. Change the order status
- **AM-5. Warehouse management**
  - 5.1. View items at the warehouse

- **Create a new order with incorrect or missing "quantity" field data**
  - POST empty field
  - POST 0 (zero)
  - POST null
  - POST number more than 127
  - POST number less than -128
  - POST not an integer, contains "," (for example: 19,99)
  - POST cyrillic letters (for example: Тест)
- with incorrect or missing "size" field data
- with incorrect or missing "color" field data
- with incorrect or missing "merchId" field data
- 3.2. Get user's orders
- 3.3. Update order
- **3.4. Delete order**
  - DEL Delete order as HR
  - DEL Delete order as MDM
  - DEL Delete order as Lead
  - DEL Delete order as HR with non-existent orderId number
  - DEL Delete order as HR with invalid orderId
  - DEL Delete order without token
- 3.5. Close order
- 3.6/4.4. Add comment to order

- **AM-4. Requests management (MDM)**
  - 4.1. View the order
  - **4.2. Accept the order**
    - **Accept the order (MDM)**
      - POST Login as MDM
      - GET Finding user's orders by status
      - PUT Update status

Run results

A-Merch - Run results

Run Again Automate Run ⌄ + New Run ⬆ Export Results

◉ Run on Today, 12:50:46 · View all runs

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	QA	1	6s 302ms	3	193 ms

All Tests   Passed (3)   Failed (0)   Skipped (0)   View Summary

Iteration 1   1

POST **Login as MDM**   {{url}}/auth/login   / AM-4. Requests management (MDM) / 4.2. ...   200 OK   300 ms   648 B

  Pass   Verify status code is 200

GET **Finding user's order...**   {{url}}/orders/filter?page=1&size=2&status=...   / AM-4. R...   200 OK   132 ms   2.777 KB

  Pass   Verify status code is 200

PUT **Update status**   {{url}}/orders/status   / AM-4. Requests management (MDM) / 4.2...   200 OK   146 ms   403 B

  Pass   Verify status code is 200

Environment

QA

VARIABLE	INITIAL VALUE	CURRENT VALUE
url		htt⬚
token_HR		eyJhbGciOiJIUzI1NiJ⬚ YW1lcmNoaHJAbWF⬚ iJIUilsImlhdCI6MTY2⬚ IjoxNjY2Nzk0ODU3f⬚
token_MDM		eyJhbGciOiJIUzI1NiJ⬚ YW1lcmNob1RtQG1⬚ oiTURNIiwiaWF0Ijoxl⬚

Swagger

order-controller

PUT `/orders` Updating order

POST `/orders` Creating order

PUT `/orders/status` Update status

POST `/orders/partial-closing` Close request partially

POST `/orders/closing/{orderId}` Close order completely by id

GET `/orders/{orderId}` Finding order by id

DELETE `/orders/{orderId}` Delete order by orderId

GET `/orders/filter` Finding user's orders by status

GET `/orders/closed` Finding user's orders

GET `/orders/closed/statuses` Getting status for closed orders

GET `/orders/active` Finding user's orders

GET `/orders/active/statuses` Getting status for active orders

notification-controller

PUT `/notifications/status` Update notification status

GET `/notifications` Get all notifications

GET `/notifications/user` Get notifications by user