

Example of API calls with results
Commands:
Add Order, BookOrder

- result contains
 - "state" (current state, after update)
 - "success"
 - "changes" - changes from this operation

```
body Cookies Headers JS Test Results
Pretty Raw Preview Viewsource JSDOM -
1 {
2   "state": {}
3   "currentUser": "adm",
4   "apiKey": null,
5   "accessToken": null,
6   "deviceId": null,
7   "auth": false,
8   "resetPasswordToken": "2022-06-09T13:13:47.4685317+02:00",
9   "passwordResetToken": "2022-06-09T13:13:47.4685317+02:00",
10  "loggedIn": false,
11  "rememberMe": true,
12  "cancelUrl": null,
13  "cancelReason": null,
14  "cancelReasonMsg": null,
15  "cancelReasonCode": 0,
16  "passwordResetCode": 0,
17  "validationMessages": null,
18  "id": "",
19  "value": "2022-06-06_RoleNameID or other visual-identifier if wanted"
20 }
21 {
22   "success": true,
23   "changes": []
24 }
25 {
26   "request": {
27     "userId": "2022-06-06_RoleNameID or other visual-identifier id wanted",
28     "username": "adm",
29     "password": "1qaz!@WSXx",
30     "rememberMe": "2022-06-09T13:13:47.4685317+02:00",
31   },
32   "eventTypes": ["ui.closeModal"]
33 }
34 }
```

eventbus LAB / Orders / **book order**

POST <https://localhost:4432/orders/book>

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

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

```
1 {
2   "orderId": "f4432d3c1f",
3   "productId": 999,
4   "productIdAmount": 0,
5   "currency": "USD",
6   "discountAmount": 0,
7   "codeBookedAt": "2022-06-04T09:18:39.742Z"
8 }
```

```

1  {
2    "state": {
3      "customerid": "456",
4      "title": " ",
5      "amount": 999,
6      "currency": "USD",
7    },
8    "outstanding": {
9      "amount": 999,
10     "currency": "USD",
11   },
12   "discount": {
13     "amount": 0,
14     "currency": "USD",
15   },
16   "mail": false,
17   "creditCardNumber": "2022-06-06734-30-47,46602-02-00",
18   "creditCardExpiry": "2022-06-06734-30-46,46602-02-00",
19   "house": true,
20   "creditCardLabel": "2022-06-06734-30-46,46602-02-00",
21   "cancelled": false,
22   "cancelLabel": null,
23   "cancelReason": null,
24   "cancelType": null,
25   "paymentProcess": {},
26   "invoiceStatus": null,
27   "invoiceNumber": null,
28   "id": " ",
29   "url": "2022-06-06_SomeCustomerId-or-other-visual-identifier-if-needed"
30 },
31 {
32   "success": true,
33   "messages": [
34     {
35       "event": {
36         "message": "2022-06-06_SomeCustomerId-or-other-visual-identifier-if-needed",
37         "creditCardNumber": "2022-06-06734-30-46,46602-02-00",
38         "creditCardExpiry": "2022-06-06734-30-46,46602-02-00",
39         "house": true,
40         "creditCardLabel": "2022-06-06734-30-46,46602-02-00",
41         "cancelled": true,
42         "cancelLabel": " ",
43         "cancelReason": " ",
44       }
45     }
46   ]
47 }

```

EVENT STORE

Dashboard
Stream Browser
Projections
Query
Persistent Subscriptions
Admin
Users
Log Out

Event Stream 'Order-2022-06-06_SomeCustomerId-or-other-visual-identifier-if-wanted'

Pause

Edit ACL

Add Event

Delete

Query

Back

self

first

previous

metadata

Event #	Name	Type	Created Date	
1	1@Order-2022-06-06_SomeCustomerId-or-other-visual-identifier-if-wanted	V1.OrderBooked	2022-06-06 16:36:09	JSON
Data				
<pre>{ "orderId": "2022-06-06_SomeCustomerId-or-other-visual-identifier-if-wanted", "orderBookedDate": "2022-06-06T16:36:06.9554481+02:00", "orderPrice": 999, "prepaidAmount": 0, "discountAmount": 0, "outstandingAmount": 999, "currency": "SEK" }</pre>				
Metadata				
<pre>{ "trace-id": "87717f667e1f90e4dab7ee9b7b6ca596", "span-id": "a807d4fabde27b0d", "parent-span-id": "f728bbd9a9dedf7c" }</pre>				
0	0@Order-2022-06-06_SomeCustomerId-or-other-visual-identifier-if-wanted	V1.OrderAdded	2022-06-06 16:33:48	JSON
Data				
<pre>{ "orderId": "2022-06-06_SomeCustomerId-or-other-visual-identifier-if-wanted", "customerId": "456", "orderCreatedDate": "2022-06-06T16:33:47.4656317+02:00" }</pre>				
Metadata				
<pre>{ "trace-id": "252d467d0aaab42e4abe74a54f95500", "span-id": "d3da8766f218da29", "parent-span-id": "948d7ce19cc44bca" }</pre>				

"drilling" into each event, or clicking on the `JSON` to expand row, we can see the event details (i.e. "changes" to the Aggregate state)

With **Swagger** you will see your APIs (command and query) with routes and expected data (json body)

[illegible]