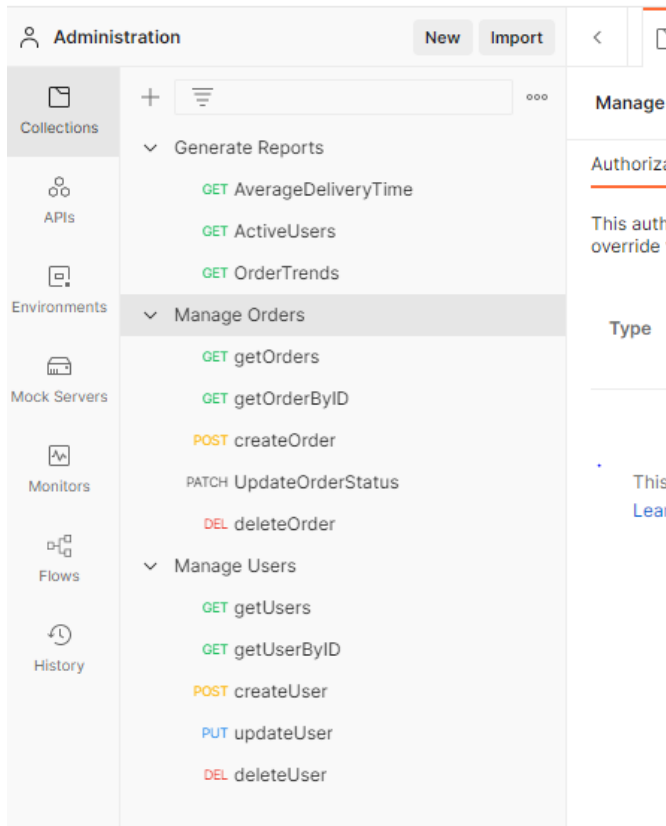
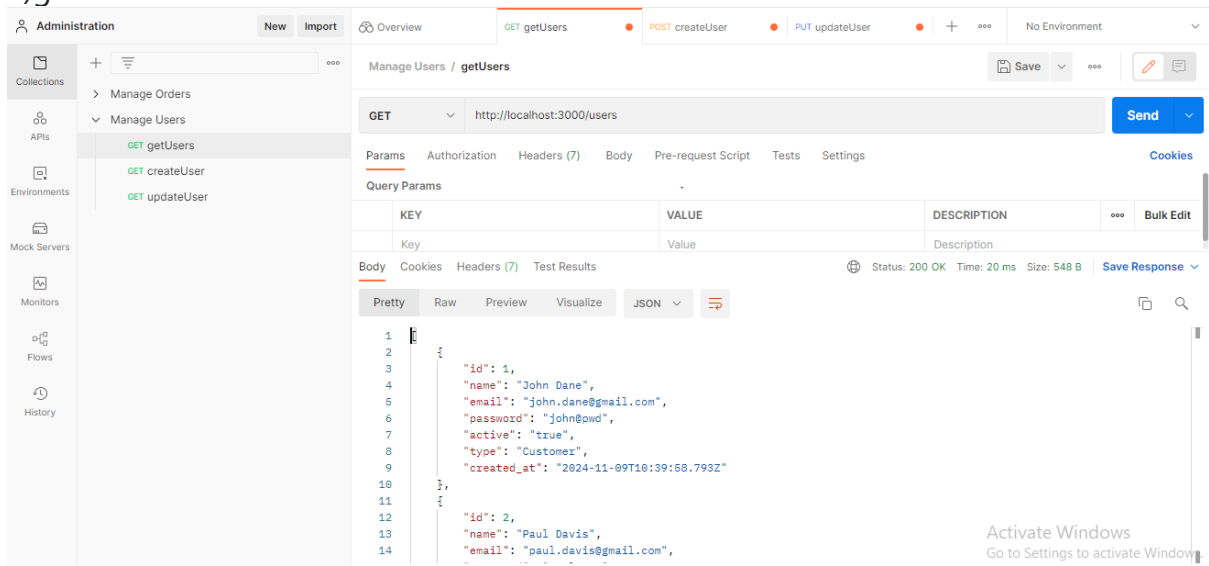


API Testing - Administration Service



1) Manage Users

a) getUsers



User X +						
Filters None Fields All Showing 2 of 2 Add record						
	id #	name A	email A	password A	active A	type A
	1	John Dane	john.dane@gmail.com	john@pwd	true	Customer
	2	Paul Davis	paul.davis@gmail.com	paul@123	true	Customer

b)getUserbyID

Administration

New

Import

Overview

GET getUsers

POST createUser

PUT updateUser

GET getUserByID

No Environment

Collections

Manage Orders

Manage Users

GET getUsers

GET getUserByID

GET createUser

GET updateUser

Environments

Mock Servers

Monitors

Flows

History

Manage Users / getUserByID

GET

http://localhost:3000/user/1

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 175 ms

Size: 391 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```

1  {
2    "id": 1,
3    "name": "John Dane",
4    "email": "john.dane@gmail.com",
5    "password": "john@pwd",
6    "active": "true",
7    "type": "Customer",
8    "created_at": "2024-11-09T10:39:58.793Z"
9  }
10
11

```

Activate Windows

Go to Settings to activate Windows.

c)createUser

Administration

New

Import

Overview

GET getUsers

POST createUser

PUT updateUser

No Environment

Collections

Manage Orders

Manage Users

GET getUsers

GET createUser

GET updateUser

Environments

Mock Servers

Monitors

Flows

History

Manage Users / createUser

POST

http://localhost:3000/user

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 71 ms

Size: 391 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```

1  {
2    "id": 2,
3    "name": "Paul Davis",
4    "email": "paul.davis@gmail.com",
5    "password": "paul@123",
6    "active": "true",
7    "type": "Customer",
8    "created_at": "2024-11-09T11:14:04.214Z"
9  }
10
11

```

Activate Windows

Go to Settings to activate Windows.

User X +						
Filters None Fields All Showing 2 of 2 Add record						
	id #	name A	email A	password A	active A	type A
	1	John Dane	john.dane@gmail.com	john@pwd	true	Customer
	2	Paul Davis	paul.davis@gmail.com	paul@123	true	Customer

d)updateUser

Administration | New | Import | Overview | GET getUsers | POST createUser | PUT updateUser | GET getUserById | + | No Environment

Manage Users / updateUser

PUT http://localhost:3000/user/2

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```
1 {
2   "name": "Paul Davis",
3   "email": "paul.davis@gmail.com",
4   "password": "paul@pwd",
5   "type": "Customer"
6 }
```

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 54 ms Size: 391 B Save Response

Pretty Raw Preview Visualize JSON

```
2 {
3   "id": 2,
4   "name": "Paul Davis",
5   "email": "paul.davis@gmail.com",
6   "password": "paul@pwd",
7   "active": "true",
8   "type": "Customer",
9   "created_at": "2024-11-09T11:14:04.214Z"
10 }
```

User						
Filters None Fields All Showing 2 of 2 Add record						
Id #	name A	email A	password A	active A	type A	created_at
1	John Dane	john.dane@gmail.com	john@pwd	true	Customer	2024-11-09T10:39:58.793Z
2	Paul Davis	paul.davis@gmail.com	paul@pwd	true	Customer	2024-11-09T11:14:04.214Z

e)deleteUser

Administration | New | Import | Overview | GET getUsers | POST createUser | PUT updateUser | GET getUserById | DEL deleteUser | + | No Environment

Manage Users / deleteUser

DELETE http://localhost:3000/user/1

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 17 ms Size: 389 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "John Dane",
4   "email": "john.dane@gmail.com",
5   "password": "john@pwd",
6   "active": "true",
7   "type": "Customer",
8   "created_at": "2024-11-09T10:39:58.793Z"
9 }
```

User						
Filters None Fields All Showing 1 of 1 Add record						
Id #	name A	email A	password A	active A	type A	created_at
2	Paul Davis	paul.davis@gmail.com	paul@pwd	true	Customer	2024-11-09T11:14:04.214Z

2) Manage Orders

a) getOrders

The screenshot shows the Postman interface with the 'getOrders' endpoint selected. The response is a JSON array of two order objects. The status is 200 OK, and the response size is 456 B.

```
1 {
2   {
3     "id": 1,
4     "name": "Pizza",
5     "state": "Preparing",
6     "deliverytime": "40 mins",
7     "created_at": "2024-11-09T11:53:00.877Z"
8   },
9   {
10    "id": 2,
11    "name": "Noodles",
12    "state": "Delivered",
13    "deliverytime": "20 mins",
14    "created_at": "2024-11-09T11:55:11.288Z"
15  }
16 }
```

Order X +					
Filters None Fields All Showing 2 of 2 Add record					
Id #	name A	state A	deliverytime A	created_at	
1	Pizza	Preparing	40 mins	2024-11-09T11:53:00.877Z	
2	Noodles	Delivered	20 mins	2024-11-09T11:55:11.288Z	

b) getOrderbyID

The screenshot shows the Postman interface with the 'getOrderbyID' endpoint selected. The response is a JSON object for the order with ID 2. The status is 200 OK, and the response size is 347 B.

```
1 {
2   {
3     "id": 2,
4     "name": "Noodles",
5     "state": "Delivered",
6     "deliverytime": "20 mins",
7     "created_at": "2024-11-09T11:55:11.288Z"
8   }
9 }
```

c)createOrder

Administration New Import < DEL dele GET getC GET getC Manag PATCH Uj POST cre DEL dele > + ... No Environment

Collections + ...

- Manage Orders
 - GET getOrders
 - GET getOrderById
 - GET createOrder
 - GET UpdateOrderStatus
 - GET deleteOrder
- Manage Users
 - GET getUsers
 - GET getUserById
 - GET createUser
 - GET updateUser
 - GET deleteUser

Environments

Mock Servers

Monitors

Flows

History

Manage Orders / createOrder

POST http://localhost:3000/order Save Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```
1 {
2   "name": "Pizza",
3   "state": "Preparing",
4   "deliverytime": "40 mins"
5 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 434 ms Size: 344 B Save Response

Pretty Raw Preview Visualize JSON ...

```
1 {
2   "id": 1,
3   "name": "Pizza",
4   "state": "Preparing",
5   "deliverytime": "40 mins",
6   "created_at": "2024-11-09T11:53:00.877Z"
7 }
```

Activate Windows
Go to Settings to activate Windows.

Order X +					
Filters None Fields All Showing 1 of 1 Add record					
Id #	name A	state A	deliverytime A	created_at	
1	Pizza	Preparing	40 mins	2024-11-09T11:53:00.877Z	

d)updateOrderStatus

Administration New Import < DEL dele GET getC GET getC Manag PATCH Uj POST cre DEL dele > + ... No Environment

Collections + ...

- Manage Orders
 - GET getOrders
 - GET getOrderById
 - GET createOrder
 - PATCH UpdateOrderStatus
 - GET deleteOrder
- Manage Users
 - GET getUsers
 - GET getUserById
 - GET createUser
 - GET updateUser
 - GET deleteUser

Environments

Mock Servers

Monitors

Flows

History

Manage Orders / UpdateOrderStatus

PATCH http://localhost:3000/order/1 Save Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```
1 {
2   "state": "Rescheduled",
3   "deliverytime": "50 mins"
4 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 38 ms Size: 346 B Save Response

Pretty Raw Preview Visualize JSON ...

```
1 {
2   "id": 1,
3   "name": "Pizza",
4   "state": "Rescheduled",
5   "deliverytime": "50 mins",
6   "created_at": "2024-11-09T11:53:00.877Z"
7 }
```

Activate Windows
Go to Settings to activate Windows.

Order X +					
Filters None Fields All Showing 2 of 2 Add record					
Id #	name A	state A	deliverytime A	created_at	
1	Pizza	Rescheduled	50 mins	2024-11-09T11:53:00.877Z	
2	Noodes	Delivered	20 mins	2024-11-09T11:55:11.288Z	

Administration New Import

Collections + ...

- Manage Orders
 - GET getOrders
 - GET getOrderById
 - GET createOrder
 - PATCH UpdateOrderStatus
 - GET deleteOrder
- Manage Users
 - GET getUsers
 - GET getUserById
 - GET createUser
 - GET updateUser
 - GET deleteUser

Environments

Mock Servers

Monitors

Flows

History

Manage Orders / UpdateOrderStatus Save

PATCH ▼ http://localhost:3000/order/1

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

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

```
1 [PATCH]
2 ..... "state": "Cancelled",
3 ..... "deliverytime": "50 mins"
4 [BODY]
```

Body Cookies Headers (7) Test Results ⊕ Status: 200 OK Time: 17 ms Size: 344 B

Pretty Raw Preview Visualize **JSON** ▼ ⌵

```
1 [PATCH]
2 {
3   "id": 1,
4   "name": "Pizza",
5   "state": "Cancelled",
6   "deliverytime": "50 mins",
7   "created_at": "2024-11-09T11:53:00.877Z"
8 }
```

Activate Windows
Go to Settings

Find and Replace Console Cookies Capture requests Bootcamp

Order × +					
⌂ Filters None Fields All Showing 2 of 2 Add record					
<input type="checkbox"/>	id #	name A	state A	deliverytime A	created_at 📅
<input type="checkbox"/>	1	Pizza	Cancelled	50 mins	2024-11-09T11:53:00.877Z
<input type="checkbox"/>	2	Noodles	Delivered	20 mins	2024-11-09T11:55:11.288Z

e)deleteOrder

Administration New Import

Collections + ...

- Manage Orders
 - GET getOrders
 - GET getOrderById
 - GET createOrder
 - PATCH UpdateOrderStatus
 - GET deleteOrder
- Manage Users
 - GET getUsers
 - GET getUserById
 - GET createUser
 - GET updateUser
 - GET deleteUser

Environments

Mock Servers

Monitors

Flows

History

Manage Orders / deleteOrder Save ⌵ 🗨

DELETE ▼ http://localhost:3000/order/2 Send

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

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (7) Test Results ⊕ Status: 200 OK Time: 16 ms Size: 345 B Save Response ▼

Pretty Raw Preview Visualize **JSON** ▼ ⌵

```
1 [DELETE]
2 {
3   "id": 2,
4   "name": "Noodles",
5   "state": "Delivered",
6   "deliverytime": "20 mins",
7   "created_at": "2024-11-09T11:55:11.288Z"
8 }
```

Activate Windows
Go to Settings to activate Windows.

Order × +					
⌂ Filters None Fields All Showing 1 of 1 Add record					
<input type="checkbox"/>	id #	name A	state A	deliverytime A	created_at 📅
<input type="checkbox"/>	1	Pizza	Delivered	40 mins	2024-11-09T12:11:43.617Z

3) Generate Reports

a) Average Delivery Time

The screenshot shows the REST client interface with the 'AverageDeliveryTime' report selected. The URL is `http://localhost:3000/averagedeliverytime` and the method is GET. The response status is 200 OK, with a time of 185 ms and a size of 235 B. The response body is displayed in the 'Pretty' view.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```

1  25

```

b) Order Trends

The screenshot shows the REST client interface with the 'OrderTrends' report selected. The URL is `http://localhost:3000/orderrends` and the method is GET. The response status is 200 OK, with a time of 17 ms and a size of 447 B. The response body is displayed in the 'Pretty' view.

```

1  [
2    {
3      "id": 4,
4      "name": "Noodles",
5      "state": "Preparing",
6      "deliverytime": "10",
7      "created_at": "2024-11-09T16:07:00.149Z"
8    },
9    {
10     "id": 1,
11     "name": "Pizza",
12     "state": "Delivered",
13     "deliverytime": "40",
14     "created_at": "2024-11-09T12:11:43.617Z"
15   }
16 ]

```

c) Active Users

The screenshot displays the Postman API client interface. On the left, the 'Administration' sidebar is visible, showing a tree structure with categories like 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. The 'Generate Reports' section is expanded, listing various endpoints such as 'GET AverageDeliveryTime', 'GET ActiveUsers', 'GET OrderTrends', 'Manage Orders', 'getOrders', 'getOrderByID', 'createOrder', 'UpdateOrderStatus', 'deleteOrder', 'Manage Users', 'getUsers', 'getUserByID', 'createUser', 'updateUser', and 'deleteUser'.

The main workspace shows a GET request to the endpoint `http://localhost:3000/activeusers`. The 'Query Params' table is empty. The 'Body' tab is selected, showing the response in 'Pretty' format. The response is a JSON object with the following structure:

```
1 {
2   "id": 2,
3   "name": "Paul Davis",
4   "email": "paul.davis@gmail.com",
5   "password": "paul@pwd",
6   "active": "true",
7   "type": "Customer",
8 }
```

The status bar at the bottom indicates a successful response with status 200 OK, time 789 ms, and size 393 B. A 'Save Response' button is visible. An 'Activate Windows' watermark is present in the bottom right corner.