

4.1.1 GET /api/salespeople

Returns an array of JSON objects containing the *spid* and *name* of all salespeople.

GET

▼

http://localhost:8080/api/salespeople

Save

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

☐

Fullspec

☐

Fullspec

Key	Value	Description
-----	-------	-------------

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 260 ms

Size: 367 B

Save Response

Pretty

Raw

Preview

Visualize

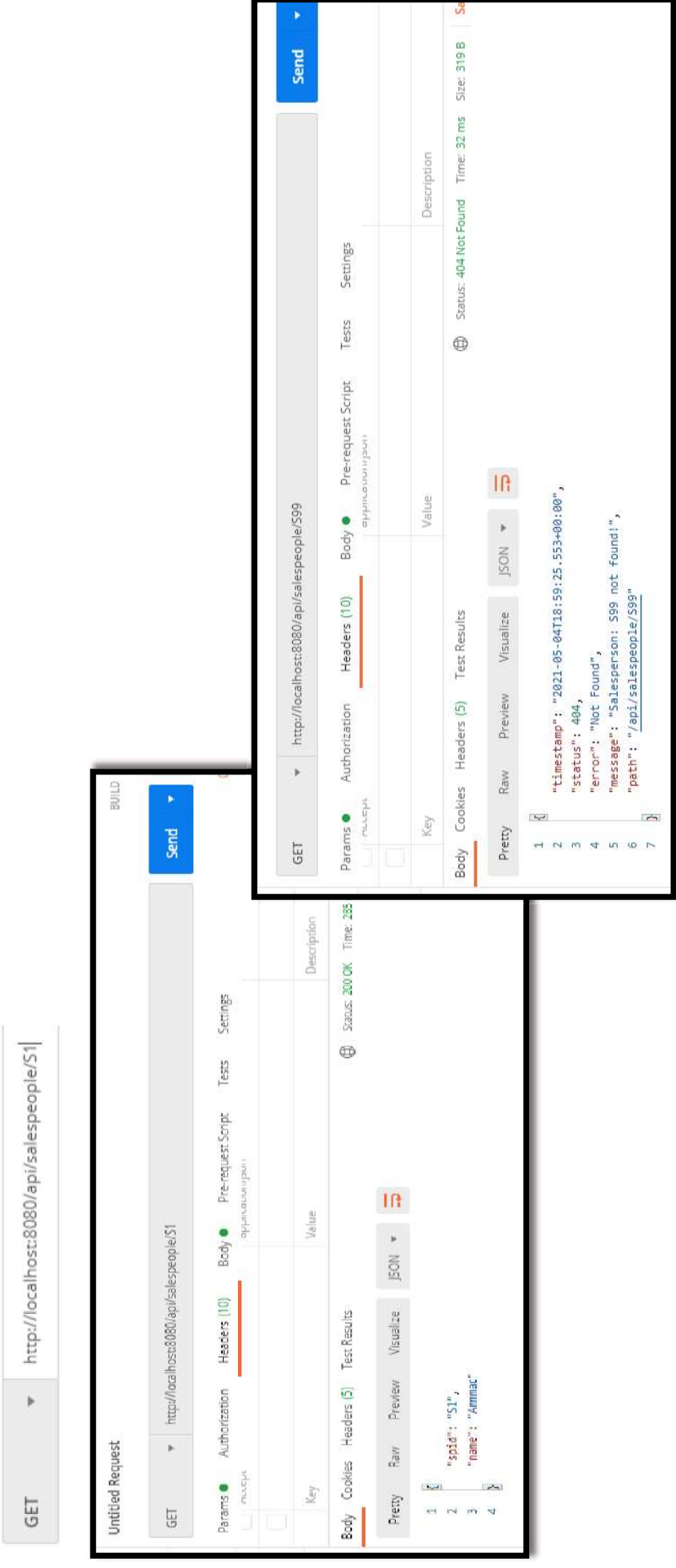
JSON

```
1  [
2    {
3      "spid": "S1",
4      "name": "Ammad"
5    },
6    {
7      "spid": "S2",
8      "name": "Mary Fitzgerald"
9    },
10   {
11     "spid": "S3",
12     "name": "Bill Flynn"
13   },
14   ]
```

4.1.2 GET /api/salespeople/{spid}

Returns a JSON object containing the *spid* and *name* of the salesperson whose *spid* is equal to the *spid* in the URL.

If no salesperson with the specified `spid` exists, a HTTP 404 response should be returned with the error message: "Salesperson {`spid`} not found." where {`spid`} is the `spid` in the URL.



```
"timestamp": "2021-05-04T18:59:25.553+00:00",
"status": 404,
"error": "Not Found",
"message": "Salesperson: S99 not found!",
"path": "/api/salespeople/S99"
```

4.1.3 DELETE /api/salespeople/{spid}

Deletes the salesperson whose *spid* is equal to the spid in the URL and returns a HTTP 200 response.

If no salesperson with the specified spid exists, a HTTP 200 response should still be returned.

If the salesperson with the specified spid exists, but has associated orders, a HTTP 422 response should be returned with the error message:

"salesperson {spid} can't be deleted. He/she has orders." where {spid} is the spid in the URL.

The screenshot displays a REST client interface for a DELETE request to `http://localhost:8080/api/salespeople/S1`. The request is configured with the method **DELETE**, a green status indicator, and the **Accept** header set to `application/json`. The **Body** tab is selected, showing a green dot and the text `application/json`. The **Test Results** section shows a status of **422 Unprocessable Entity (WebDAV) (RFC 4918)**. The **Body** tab is active, displaying the response in **JSON** format. The response is a 422 status with an error message indicating that the salesperson cannot be deleted due to associated orders.

Key	Value	Description	
Body	cookies	headers (5)	test results

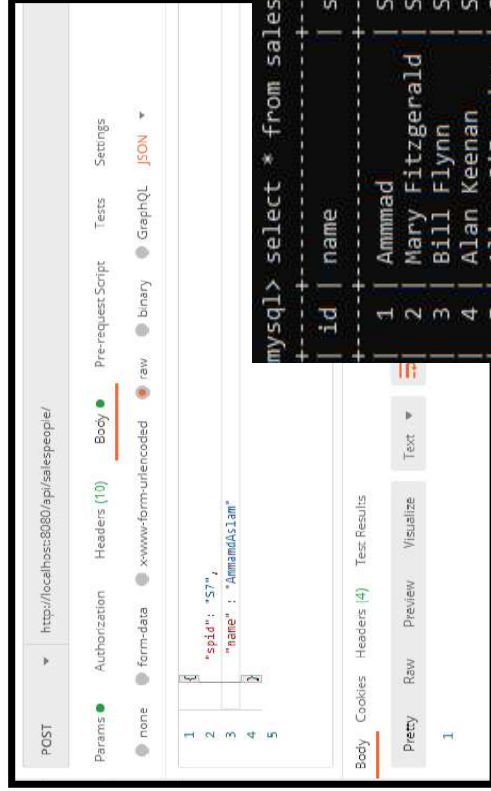
```
1 {
2   "timestamp": "2021-05-04T19:07:31.617+00:00",
3   "status": 422,
4   "error": "Unprocessable Entity",
5   "message": "Salesperson: S1 can't be deleted. He/she has orders",
6   "path": "/api/salespeople/S1"
7 }
```

4.1.4 POST /api/salespeople

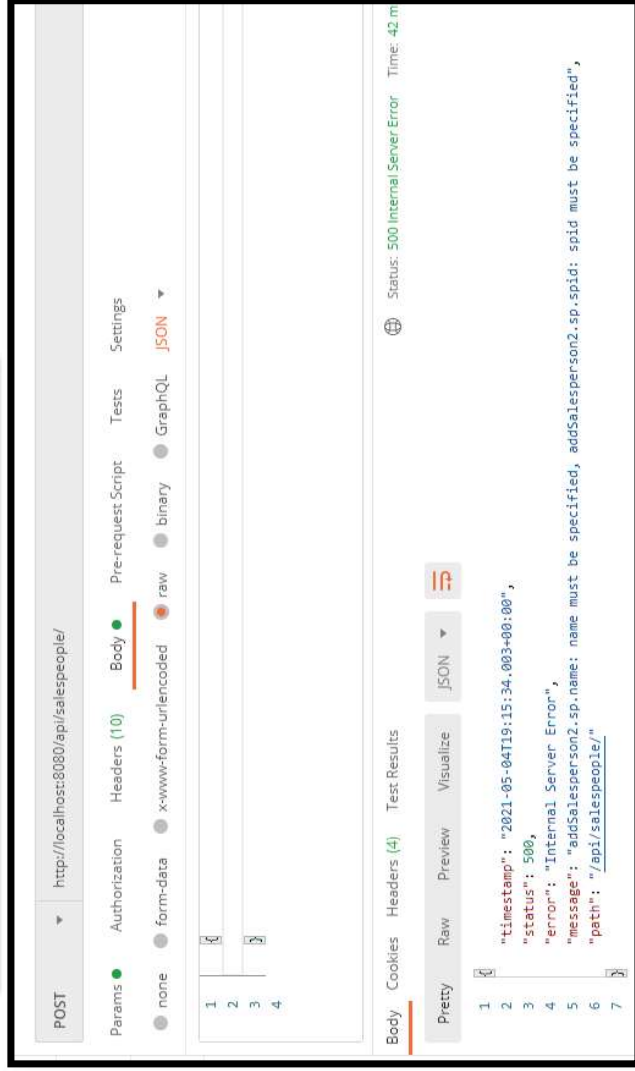
Persists the salesperson object specified in the request body to the database. The salesperson object must contain *spid* and *name*.

If either of these attributes are not present a HTTP 500 response should be returned with an appropriate error message.

If a salesperson with the specified spid already exists, a HTTP 422 response should be returned with the following error message: "salesperson: {spid} already exists" where {spid} is the spid in the request body.



```
mysql> select * from salesperson;
+----+-----+-----+
| id | name | spid |
+----+-----+-----+
| 1 | Ammad | S1 |
| 2 | Mary Fitzgerald | S2 |
| 3 | Bill Flynn | S3 |
| 4 | Alan Keenan | S4 |
| 5 | Alice O'Rourke | S5 |
| 6 | AA | S6 |
| 7 | AmmamAslam | S7 |
+----+-----+-----+
7 rows in set (0.00 sec)
```



4.1.1.5 PUT /api/salespeople/{spid}

Updates the salesperson with the *spid* specified in the URL with the attributes specified in the request body. The request body should only contain the *name* attribute.

If any other attributes are present, a HTTP 500 response should be returned with an appropriate error message.

If no salesperson with the specified *spid* exists, a HTTP 404 response should be returned with the error message: "Salesperson {spid} not found." where {spid} is the *spid* in the URL.

PUT <http://localhost:8080/api/salespeople/S1>

PUT <http://localhost:8080/api/salespeople/S1>

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1 {

2 "spid": "S1",

3 "name": "NewAmmamdAslam"

4 }

5

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize

JSON

1 {

2 "timestamp": "2021-05-04T21:17:58.521+00:00",

3 "status": 500,

4 "error": "Internal Server Error",

5 "message": "update.sp.spid: spid must not be provided ",

6 "path": "/api/salespeople/S1"

7 }

Status:

```
mysql> select * from salesperson;
+----+-----+-----+
id | name | spid |
+----+-----+-----+
1 | Ammmad | S1 |
2 | Mary Fitzgerald | S2 |
3 | Bill Flynn | S3 |
4 | Alan Keenan | S4 |
5 | Alice O'Rourke | S5 |
6 | AA | S6 |
+----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> select * from salesperson;
+----+-----+-----+
id | name | spid |
+----+-----+-----+
1 | NewAmmamdAslam | S1 |
2 | Mary Fitzgerald | S2 |
3 | Bill Flynn | S3 |
4 | Alan Keenan | S4 |
5 | Alice O'Rourke | S5 |
6 | AA | S6 |
+----+-----+-----+
6 rows in set (0.00 sec)
```

Untitled Request

PUT

http://localhost:8080/api/salespeople/S1

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

1

{

2

"name" : "NewAmmamdAS1am"

3

}

4

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"spid": "S1",

3

"name": "NewAmmamdAS1am"

4

}

4.2 Order API endpoints

4.2.1 GET /api/orders

Returns an array of JSON objects containing the *oid*, *orderDate*, *orderSalesperson.spid*, *orderSalesperson.name*, *orderProduct.pid*, *orderProduct.product*, *orderProduct.quantity*, and *orderProduct.orderable* attributes of all orders.

GET ▼ http://localhost:8080/api/orders

The screenshot shows a REST client interface with the following components:

- Request Bar:** Method: GET, URL: http://localhost:8080/api/orders
- Request Body:** Params (0), Authorization, Headers (8), Body (selected), Pre-request Script, Tests, Settings. The Body tab shows: none, form-data, x-www-form-urlencoded, raw, binary, GraphQL.
- Response Bar:** Body (selected), Cookies, Headers (5), Test Results, Test Results. A message states: "This request does not have a body".
- Response Body:** Pretty, Raw, Preview, Visualize, JSON (selected). The JSON response is:

```
{
  "oid": "0200",
  "orderDate": "2019-12-01",
  "orderSalesperson": {
    "spid": "S1",
    "name": "NewAmmaAslam"
  },
  "orderProduct": {
    "pid": "TV",
    "product": "TV",
    "quantity": 4,
    "orderable": "true",
    "productOrders": {
```


4.2.2 GET /api/specificOrders?year=<year>&qty=<quantity>

Return the *oid*, *orderDate*, *orderSalesperson.spid*, *orderSalesperson.name*, *orderProduct.pid*, *orderProduct.product*, *orderProduct.quantity*, and *orderProduct.orderable* attributes of all orders where the order year = year and order quantity > qty.

or example, `/api/specificOrders?year=2020&qty=4` returns all details of orders in the year 2020 that had an order quantity of greater than 4.

Both year and qty parameters must be supplied to this endpoint and their values must always be integers.

GET	▼	http://localhost:8080/api/specificOrders?year=2020&qty=4			
Params	●	Authorization	Headers (6)	Body	Pre-request Script

Untitled Request					
GET	▼	http://localhost:8080/api/specificOrders?year=2020&qty=4			
Params	●	Authorization	Headers (8)	Body	Pre-request Script
●	none	●	form-data	●	x-www-form-urlencoded
●	raw	●	binary	●	GraphQL
This request does not have a body					
Body	Cookies	Headers (5)	Test Results		
Pretty	Raw	Preview	Visualize	JSON	
<pre>1 { 2 "oid": "O202", 3 "orderDate": "2020-05-16", 4 "orderSalesperson": { 5 "spid": "S3", 6 "name": "Bill Flynn" 7 }, 8 "orderProduct": { 9 "pid": "US8", 10 "product": "4GB USB", 11 "quantity": 0, 12 "orderable": "true", 13 "productOrders": [14</pre>					

GET

▼

http://localhost:8080/api/specificOrders?year=2020

Params

Authorization

Headers (6)

Body

Pre-request Script

Body

Cookies

Headers (4)

Test Results

GET

▼

http://localhost:8080/api/specificOrders?year=2020

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

Body

Cookies

Headers (4)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

{

"timestamp": "2021-05-04T21:25:37.617+00:00",

"status": 400,

"error": "Bad Request",

"message": "Required int parameter 'qty' is not present",

"path": "/api/specificOrders"

}

4.2.3 POST /api/orders

Persists the order object specified in the request body to the database. The order object must contain the *oid*, *orderDate*, *orderQuantity*, *orderSalesperson.spid* and *orderProduct.pid* attributes.

If any of these attributes is not present a HTTP 500 response should be returned with an appropriate error message. (NOTE: *orderQuantity* should be > 0).

POST

▼

http://localhost:8080/api/orders

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL

JSON

1

2

3

4

5

6

7

8

{

"oid": "0206",

"orderDate": "2021-03-19",

"orderSalesperson": {

"spid": "51"

},

"orderQuantity": 1

}

Body

Cookies

Headers (4)

Test Results

● Pretty

● Raw

● Preview

● Visualize

JSON

1

2

3

4

5

6

{

"timestamp": "2021-05-04T21:28:57.895+00:00",

"status": 500,

"error": "Internal Server Error",

"message": "addOrder.order.orderProduct: orderproduct.o.orderproduct: order product must be provided",

"path": "/api/orders"

}

Status: 500 Internal Server Error

If *orderSalesperson.spid* is not a valid salesperson, a HTTP 422 response should be returned with the following error message: "Salesperson {orderSalesperson.spid} does not exist" where {orderSalesperson.spid} is the salesperson spid in request body.

POST

http://localhost:8080/api/orders

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

{

"oid": "0206",

"orderDate": "2021-03-19",

"orderSalesperson": {

"spid": "S99"

},

"orderProduct": {

"pid": "TV"

},

"orderQuantity": 1

}

Body

Cookies

Headers (4)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

{

"timestamp": "2021-05-04T22:06:05.547+00:00",

"status": 400,

"error": "Bad Request",

"message": "Salesperson: S99does not exist",

"path": "/api/orders"

}

Status: 400

POST

http://localhost:8080/api/orders

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

{

"oid": "0216",

"orderDate": "2021-03-19",

"orderSalesperson": {

| "spid": "51"

},

"orderProduct": {

| "pid": "ABC"

},

"orderQuantity": 1

}

Body

Cookies

Headers (5)

Test Results

Status: 422 Unprocessable Entity

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

"timestamp": "2021-05-04T22:07:43.509+00:00",

"status": 422,

"error": "Unprocessable Entity",

"message": "Product: ABC does not exist",

"path": "/api/orders"

1

2

3

4

5

6

7

8

9

10

11

12

13

{

"oid": "0206",

"orderDate": "2021-03-19",

"orderSalesperson": {

| "spid": "51"

},

"orderProduct": {

| "pid": "TV"

},

"orderQuantity": 7

}

Body

Cookies

Headers (4)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

"timestamp": "2021-05-04T21:39:32.996+00:00",

"status": 400,

"error": "Bad Request",

"message": "Stock on hand: 5 is less than order quantity: = 7",

"path": "/api/orders"