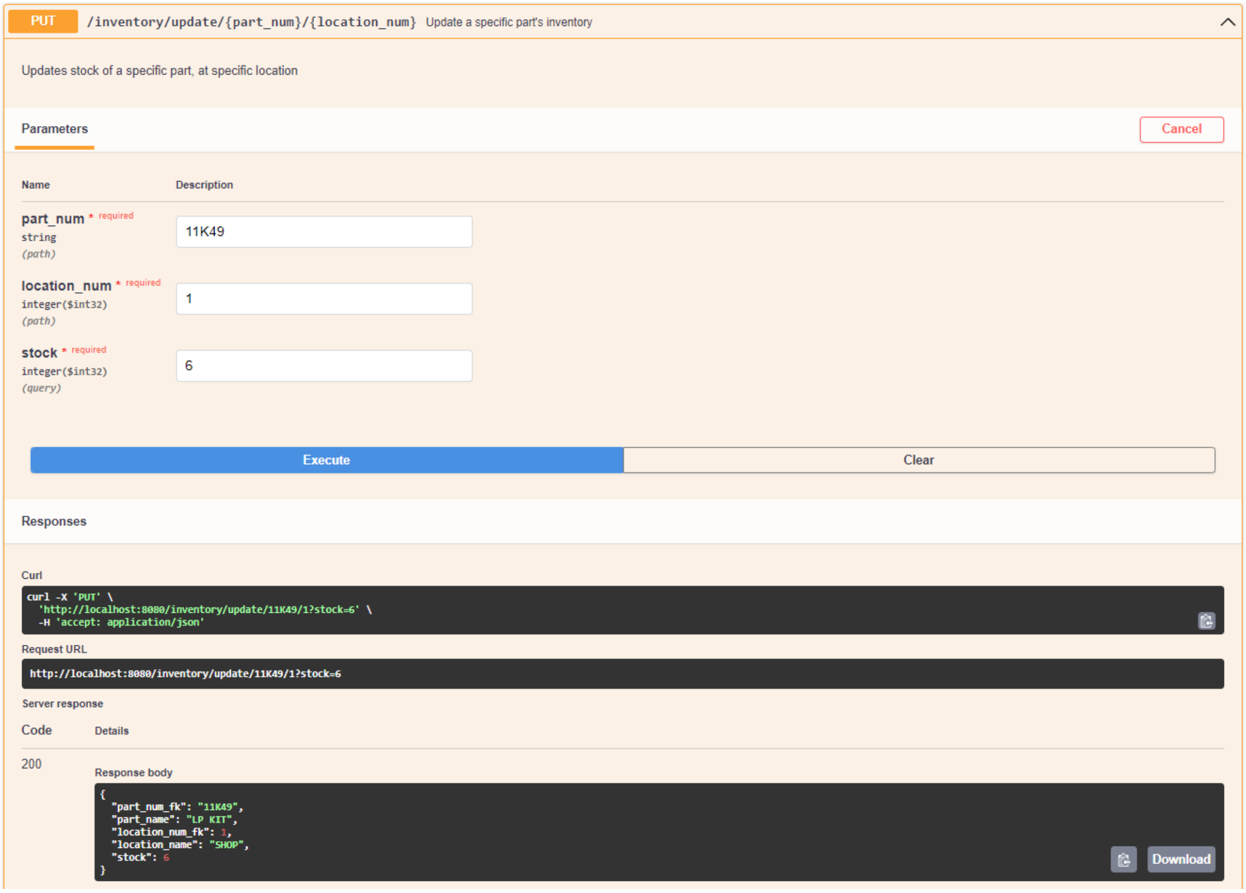
Final Project - HVAC Service Parts Inventory

GitHub URL: https://github.com/ANGRAYSON/Final-Project

Screenshots of Results of Running each path in Swagger:

Screenshots also under ‘Final Project Screenshots’ folder on GitHub. (There’s a lot)

Inventory:
Put:



/inventory/update/{part_num}/{location_num} Update a specific part's inventory

Updates stock of a specific part, at specific location

Parameters

Name	Description
part_num * required string (path)	11K49
location_num * required integer($int32) (path)	1
stock * required integer($int32) (query)	6

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:8080/inventory/update/11K49/1?stock=6' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/inventory/update/11K49/1?stock=6
```

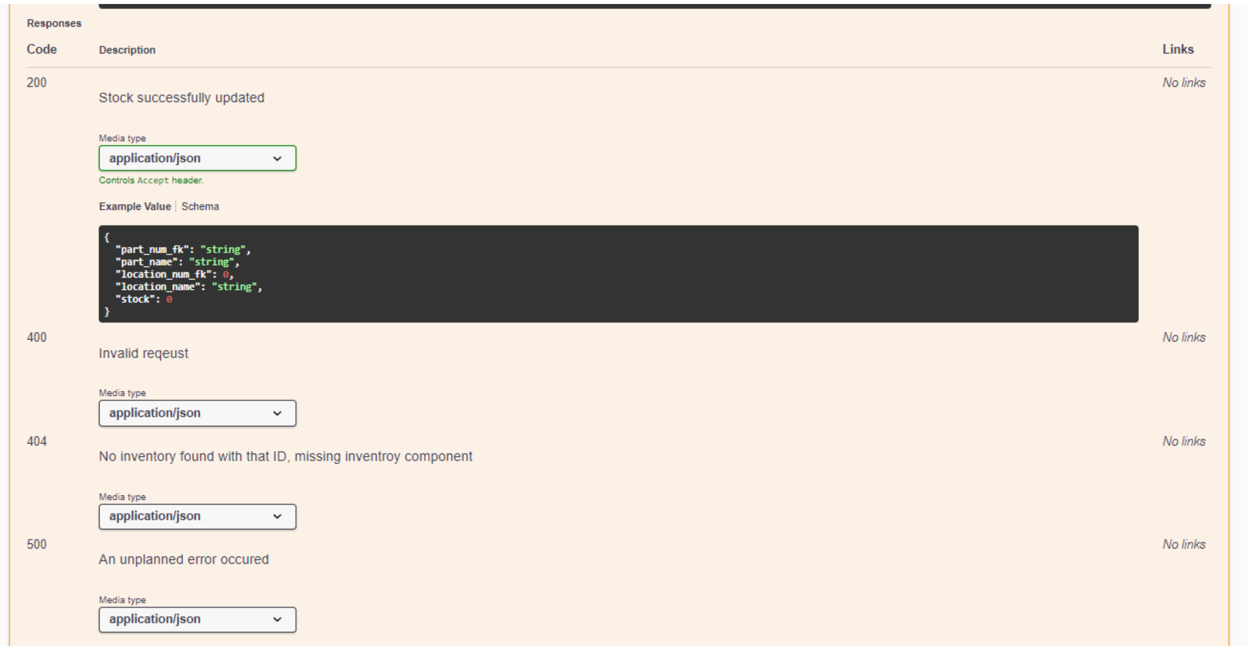
Server response

Code Details

200

Response body

```
{
  "part_num_fk": "11K49",
  "part_name": "LP KIT",
  "location_num_fk": 1,
  "location_name": "SHOP",
  "stock": 6
}
```



200

Stock successfully updated

Media type: application/json

Controls Accept header.

Example Value | Schema

```
{
  "part_num_fk": "string",
  "part_name": "string",
  "location_num_fk": 0,
  "location_name": "string",
  "stock": 0
}
```

No links

400

Invalid reqeust

Media type: application/json

No links

404

No inventory found with that ID, missing inventroy component

Media type: application/json

No links

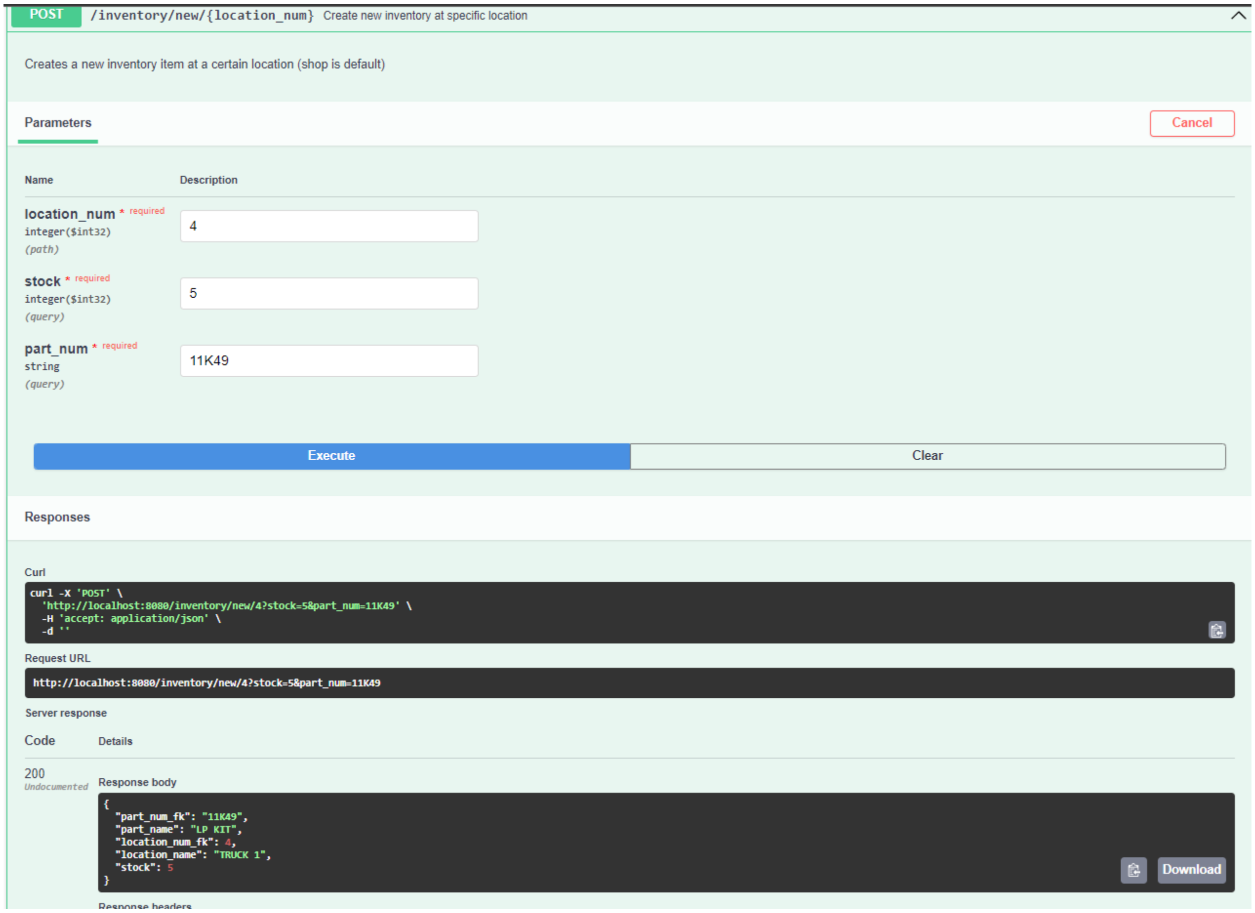
500

An unplanned error occured

Media type: application/json

No links

Post:



/inventory/new/{location_num} Create new inventory at specific location

Creates a new inventory item at a certain location (shop is default)

Parameters

Name	Description
location_num * required integer($int32) (path)	4
stock * required integer($int32) (query)	5
part_num * required string (query)	11K49

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/inventory/new/4?stock=5&part_num=11K49' \
  -H 'accept: application/json' \
  -d ''
```

Request URL

```
http://localhost:8080/inventory/new/4?stock=5&part_num=11K49
```

Server response

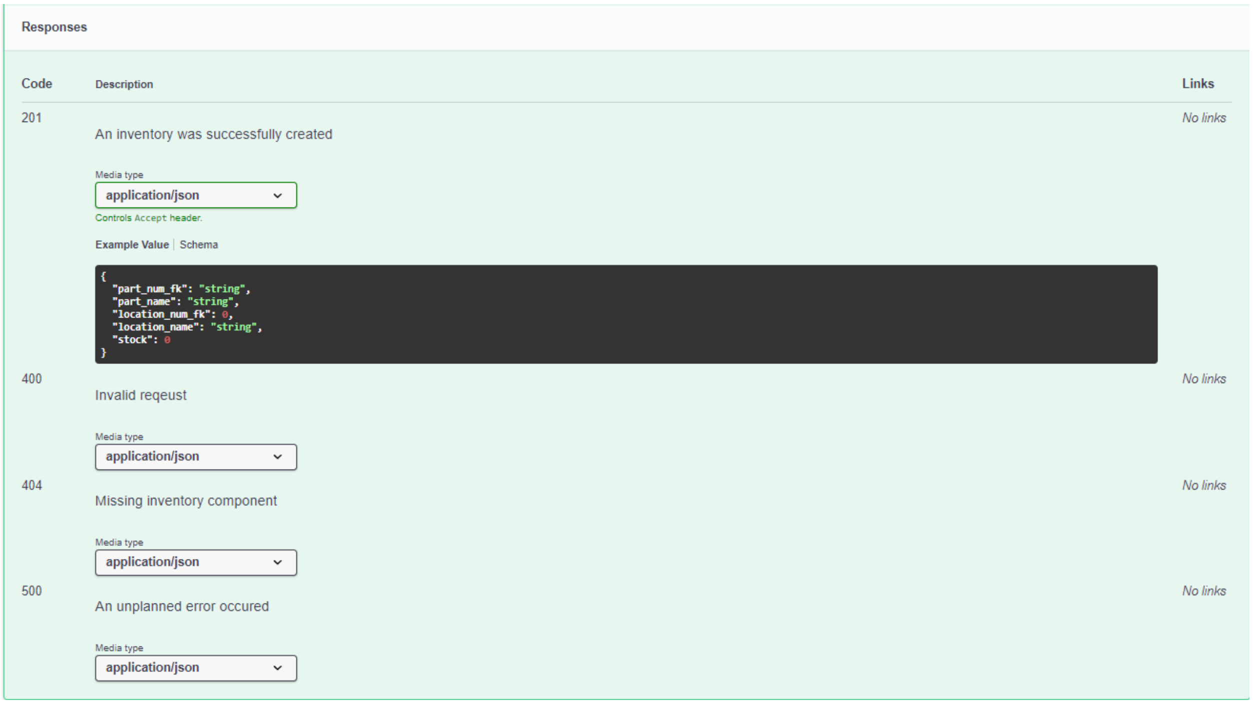
Code Details

200 *Undocumented*

Response body

```
{
  "part_num_fk": "11K49",
  "part_name": "LP KIT",
  "location_num_fk": 4,
  "location_name": "TRUCK 1",
  "stock": 5
}
```

Response headers



Responses

Code Description Links

201

An inventory was successfully created

No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "part_num_fk": "string",
  "part_name": "string",
  "location_num_fk": 0,
  "location_name": "string",
  "stock": 0
}
```

400

Invalid reqeust

No links

Media type

application/json

404

Missing inventory component

No links

Media type

application/json

500

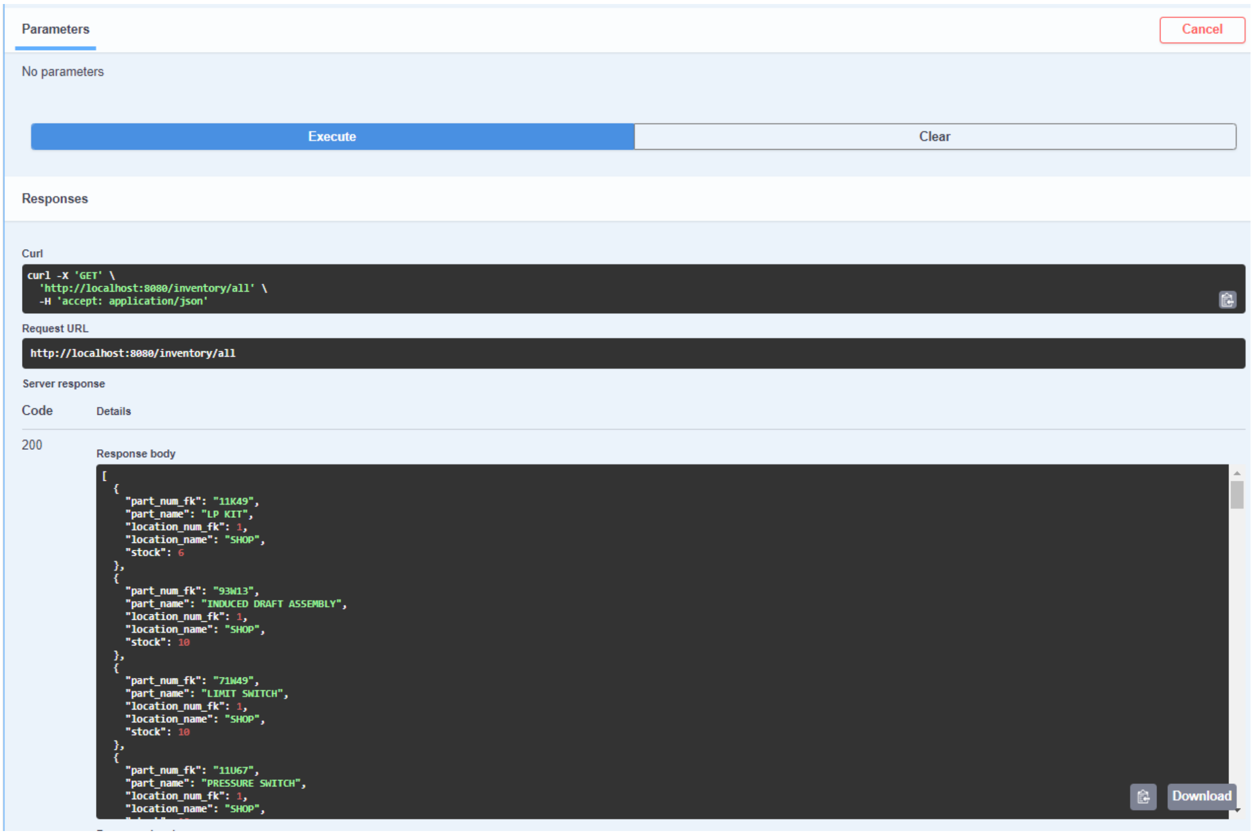
An unplanned error occured

No links

Media type

application/json

Get:



No parameters

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/inventory/all' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/inventory/all
```

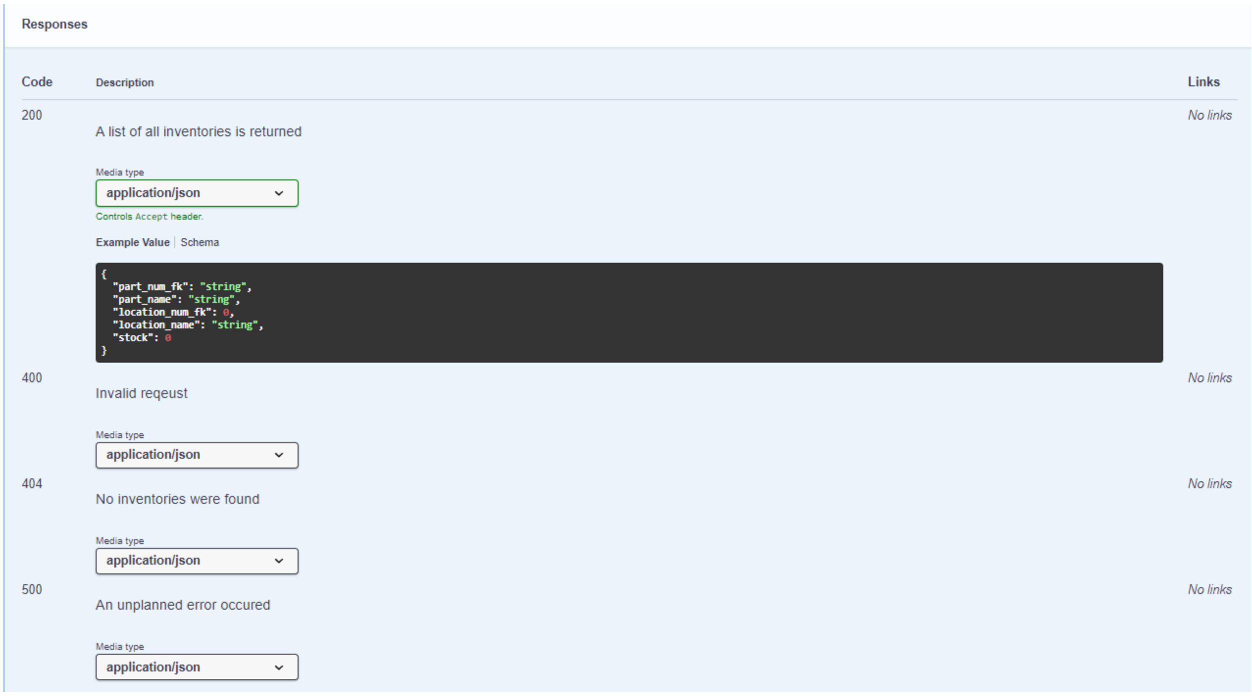
Server response

Code Details

200

Response body

```
[
  {
    "part_num_fk": "11K49",
    "part_name": "LP KIT",
    "location_num_fk": 1,
    "location_name": "SHOP",
    "stock": 6
  },
  {
    "part_num_fk": "93W13",
    "part_name": "INDUCED DRAFT ASSEMBLY",
    "location_num_fk": 1,
    "location_name": "SHOP",
    "stock": 10
  },
  {
    "part_num_fk": "71W49",
    "part_name": "LIMIT SWITCH",
    "location_num_fk": 1,
    "location_name": "SHOP",
    "stock": 10
  },
  {
    "part_num_fk": "11U67",
    "part_name": "PRESSURE SWITCH",
    "location_num_fk": 1,
    "location_name": "SHOP",
```



Responses

Code Description Links

200

A list of all inventories is returned

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "part_num_fk": "string",
  "part_name": "string",
  "location_num_fk": 0,
  "location_name": "string",
  "stock": 0
}
```

No links

400

Invalid reqeust

Media type

application/json

No links

404

No inventories were found

Media type

application/json

No links

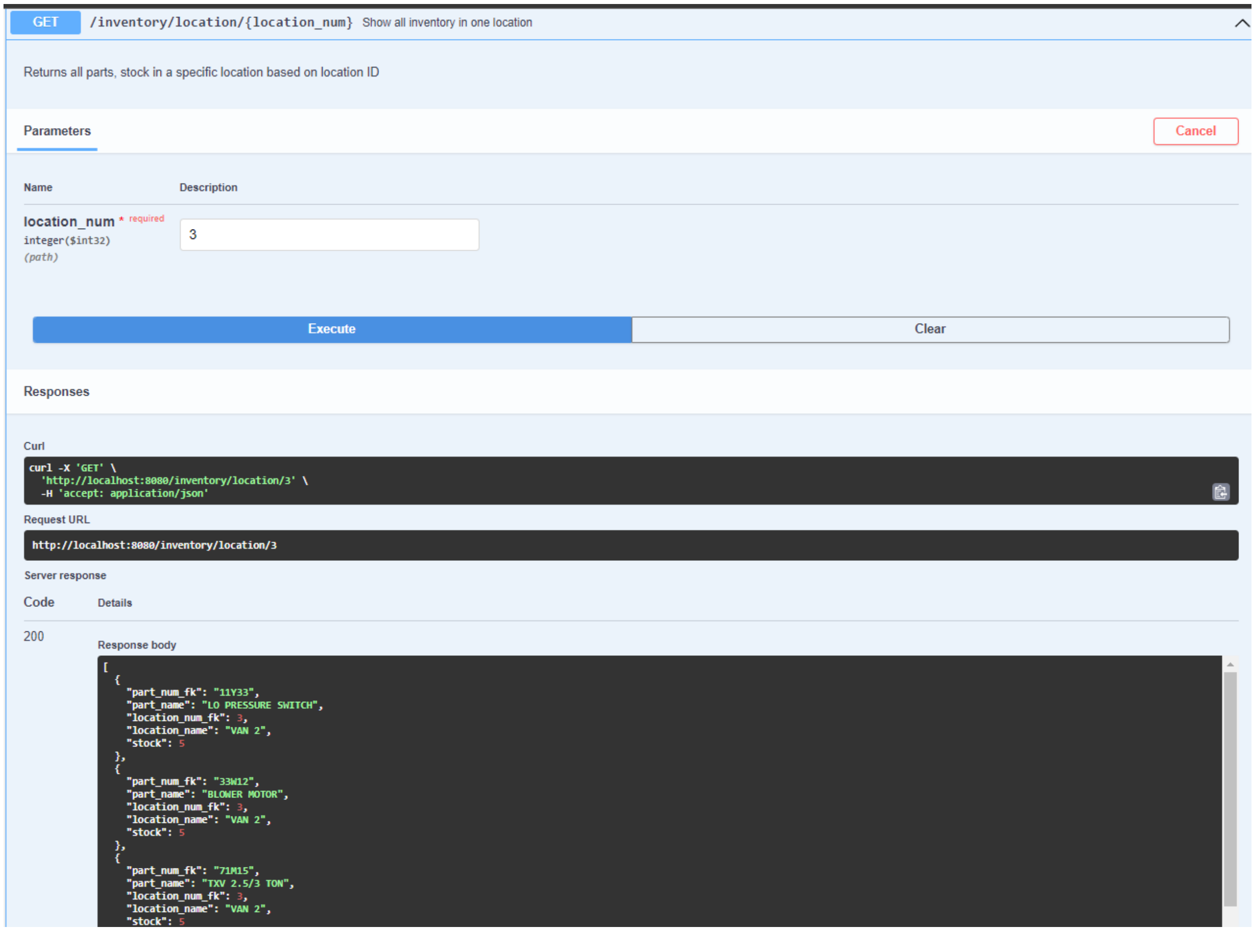
500

An unplanned error occured

Media type

application/json

No links



`/inventory/location/{location_num}` Show all inventory in one location

Returns all parts, stock in a specific location based on location ID

Parameters

Name	Description
location_num * required integer($int32) *(path)*	3

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/inventory/location/3' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/inventory/location/3
```

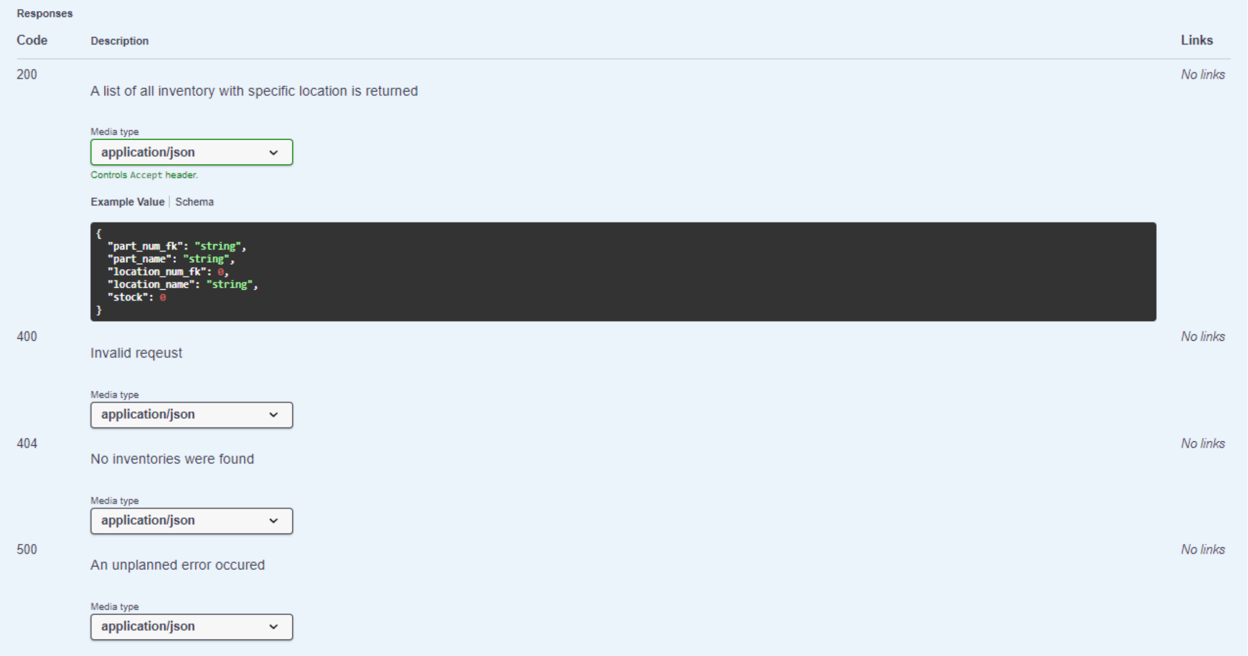
Server response

Code Details

200

Response body

```
[
  {
    "part_num_fk": "11Y33",
    "part_name": "LO PRESSURE SWITCH",
    "location_num_fk": 3,
    "location_name": "VAN 2",
    "stock": 5
  },
  {
    "part_num_fk": "33W12",
    "part_name": "BLOWER MOTOR",
    "location_num_fk": 3,
    "location_name": "VAN 2",
    "stock": 5
  },
  {
    "part_num_fk": "71M15",
    "part_name": "TXV 2.5/3 TON",
    "location_num_fk": 3,
    "location_name": "VAN 2",
    "stock": 5
```

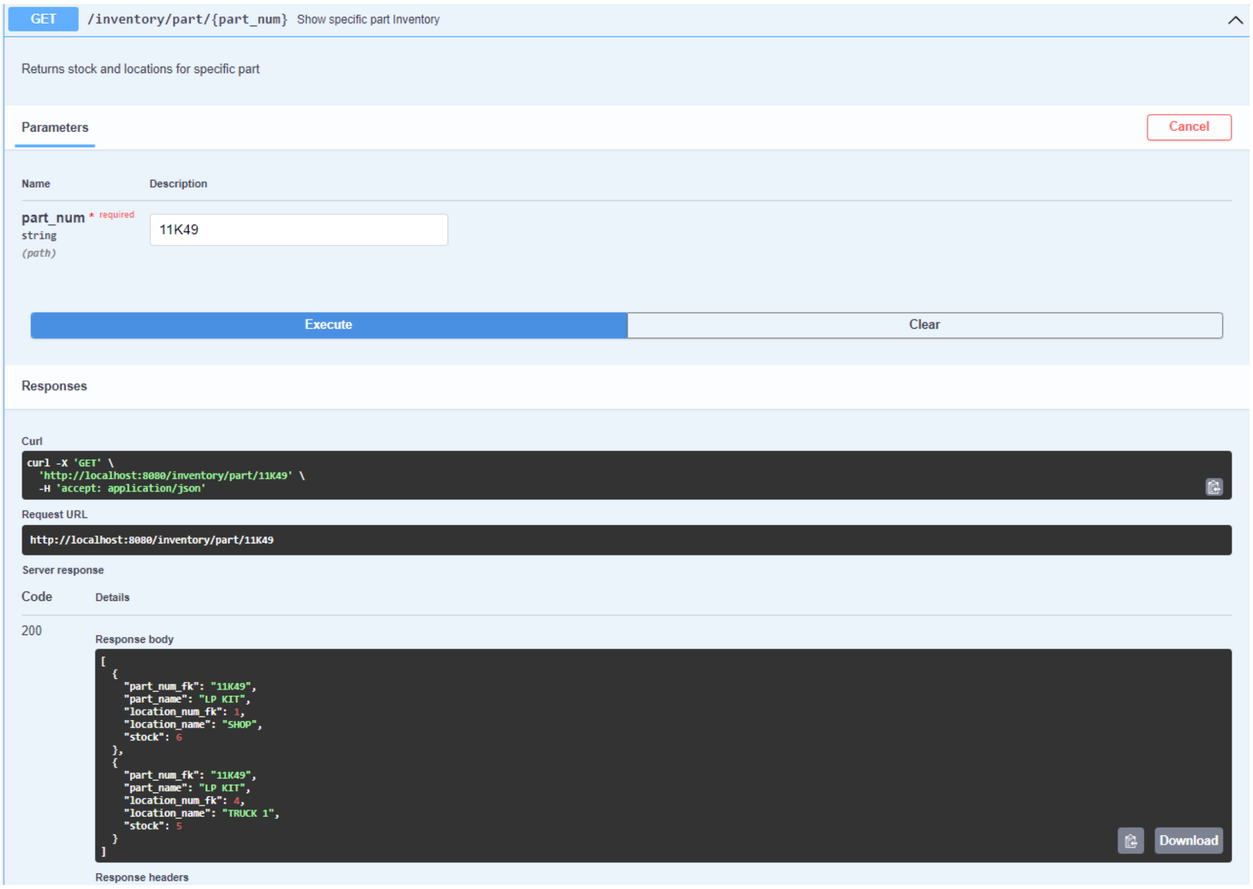


Responses

Code	Description	Links
200	A list of all inventory with specific location is returned Media type: application/json Controls Accept header.	*No links*
400	Invalid reqeust Media type: application/json	*No links*
404	No inventories were found Media type: application/json	*No links*
500	An unplanned error occured Media type: application/json	*No links*

Example value for code 200:

```
{
  "part_num_fk": "string",
  "part_name": "string",
  "location_num_fk": 0,
  "location_name": "string",
  "stock": 0
}
```



`/inventory/part/{part_num}` Show specific part Inventory

Returns stock and locations for specific part

Parameters

Name	Description
part_num * required string *(path)*	11K49

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/inventory/part/11K49' \
  -H 'accept: application/json'
```

Request URL

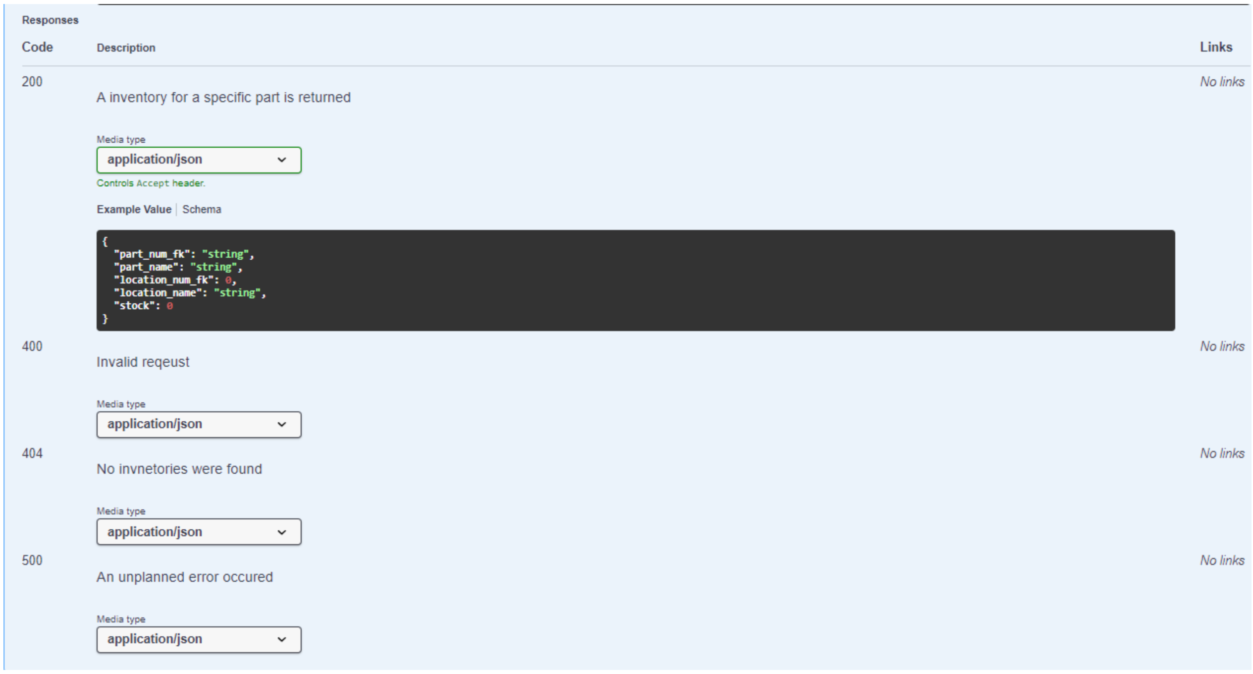
```
http://localhost:8080/inventory/part/11K49
```

Server response

Code	Details
200	Response body

```
[
  {
    "part_num_fk": "11K49",
    "part_name": "LP KIT",
    "location_num_fk": 1,
    "location_name": "SHOP",
    "stock": 6
  },
  {
    "part_num_fk": "11K49",
    "part_name": "LP KIT",
    "location_num_fk": 4,
    "location_name": "TRUCK 1",
    "stock": 5
  }
]
```

Response headers



Responses

Code	Description	Links
200	A inventory for a specific part is returned	*No links*
400	Invalid reqeust	*No links*
404	No invnetories were found	*No links*
500	An unplanned error occured	*No links*

200 — Media type: application/json

Controls `Accept` header.

Example Value | Schema

```
{
  "part_num_fk": "string",
  "part_name": "string",
  "location_num_fk": 0,
  "location_name": "string",
  "stock": 0
}
```

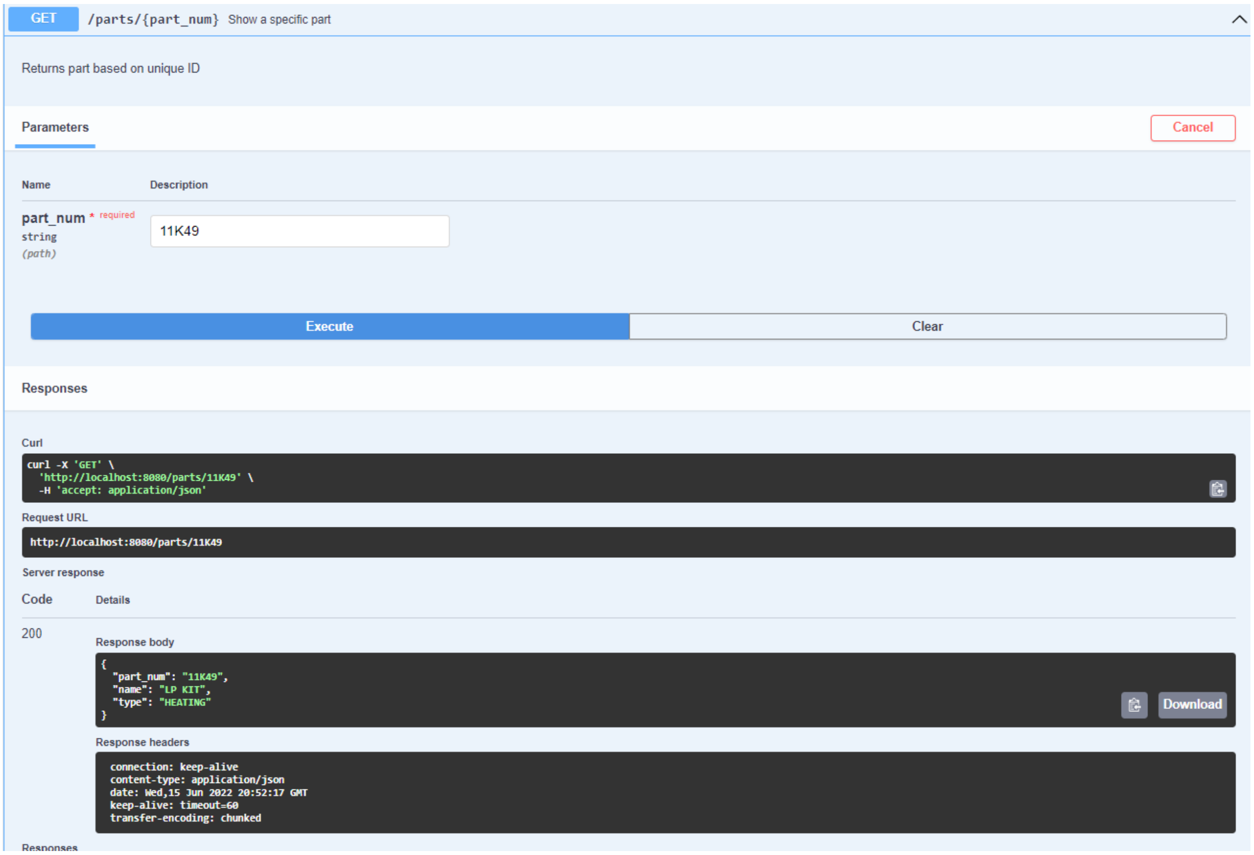
400 — Media type: application/json

404 — Media type: application/json

500 — Media type: application/json

Parts:

Get:



`/parts/{part_num}` Show a specific part

Returns part based on unique ID

Name	Description
part_num * required string *(path)*	11K49

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/parts/11K49' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/parts/11K49
```

Server response

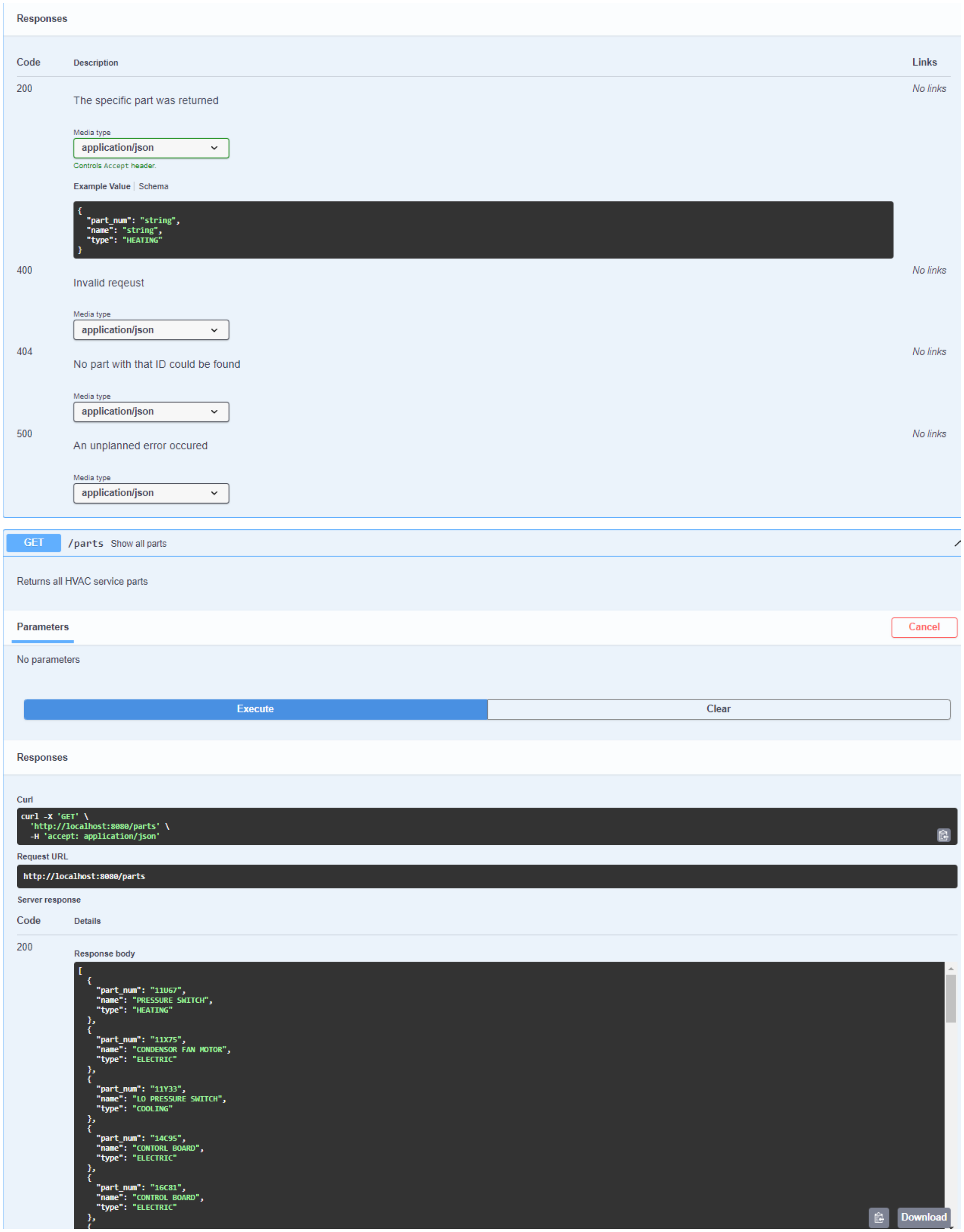
Code	Details
200	Response body

```
{
  "part_num": "11K49",
  "name": "LP KIT",
  "type": "HEATING"
}
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,15 Jun 2022 20:52:17 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
```

Responses



Responses

Code Description Links

200

The specific part was returned

No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "part_num": "string",
  "name": "string",
  "type": "HEATING"
}
```

400

Invalid reqeust

No links

Media type

application/json

404

No part with that ID could be found

No links

Media type

application/json

500

An unplanned error occured

No links

Media type

application/json

GET /parts Show all parts

Returns all HVAC service parts

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/parts' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/parts
```

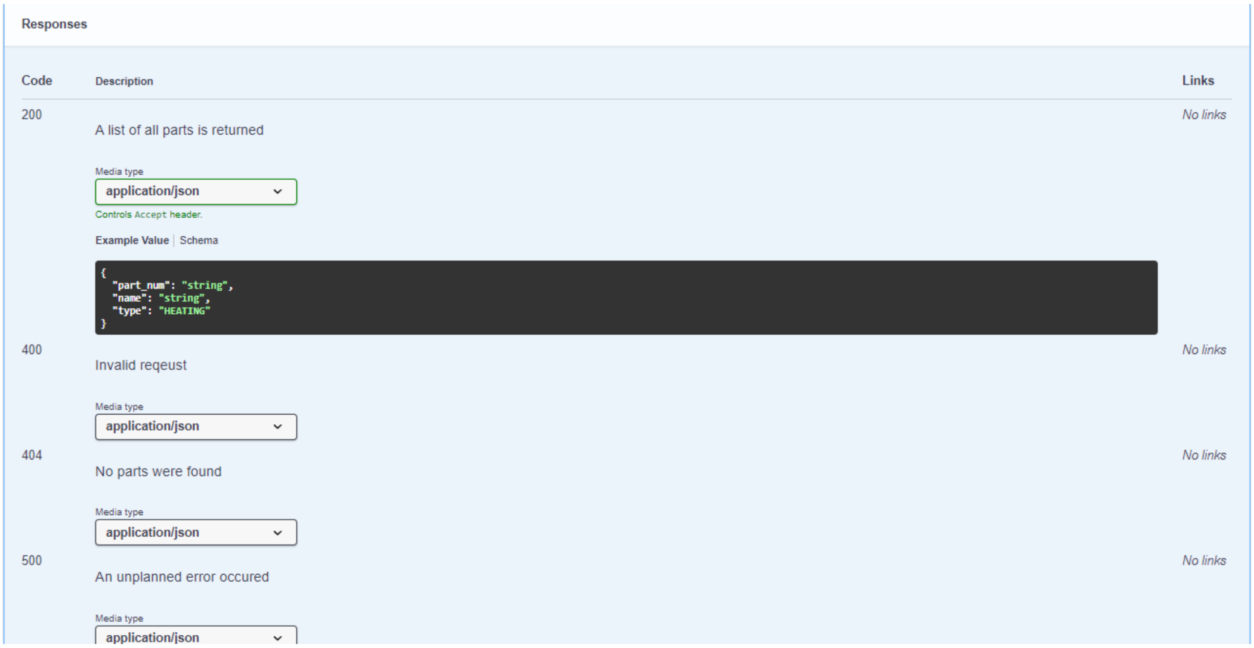
Server response

Code Details

200

Response body

```
[
  {
    "part_num": "11U67",
    "name": "PRESSURE SWITCH",
    "type": "HEATING"
  },
  {
    "part_num": "11X75",
    "name": "CONDENSOR FAN MOTOR",
    "type": "ELECTRIC"
  },
  {
    "part_num": "11Y33",
    "name": "LO PRESSURE SWITCH",
    "type": "COOLING"
  },
  {
    "part_num": "14C95",
    "name": "CONTORL BOARD",
    "type": "ELECTRIC"
  },
  {
    "part_num": "16C81",
    "name": "CONTROL BOARD",
    "type": "ELECTRIC"
  },
  {
```

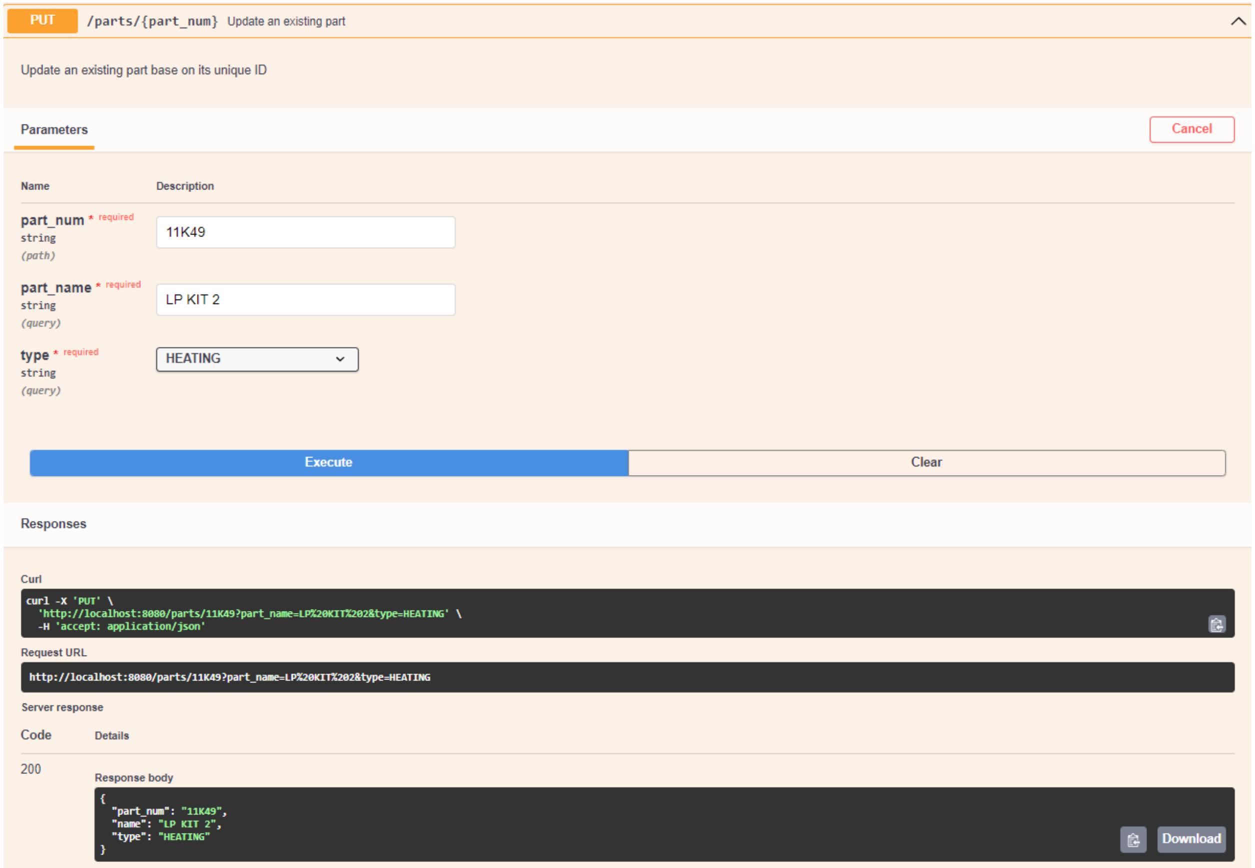


Responses

Code	Description	Links
200	A list of all parts is returned Media type: application/json Controls Accept header. Example Value | Schema	*No links*
400	Invalid reqeust Media type: application/json	*No links*
404	No parts were found Media type: application/json	*No links*
500	An unplanned error occured Media type: application/json	*No links*

```
{
  "part_num": "string",
  "name": "string",
  "type": "HEATING"
}
```

Put:



`/parts/{part_num}` Update an existing part

Update an existing part base on its unique ID

Parameters

Name	Description
part_num * required string *(path)*	11K49
part_name * required string *(query)*	LP KIT 2
type * required string *(query)*	HEATING

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:8080/parts/11K49?part_name=LP%20KIT%202&type=HEATING' \
  -H 'accept: application/json'
```

Request URL

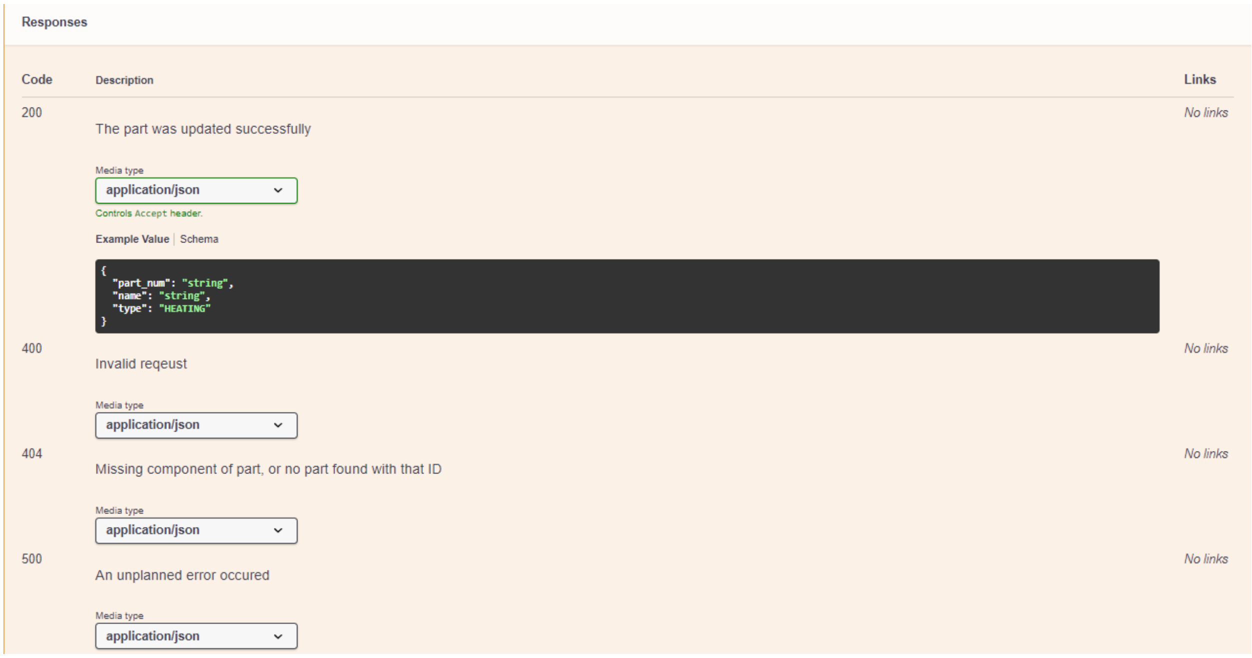
```
http://localhost:8080/parts/11K49?part_name=LP%20KIT%202&type=HEATING
```

Server response

Code: 200

Details: Response body

```
{
  "part_num": "11K49",
  "name": "LP KIT 2",
  "type": "HEATING"
}
```



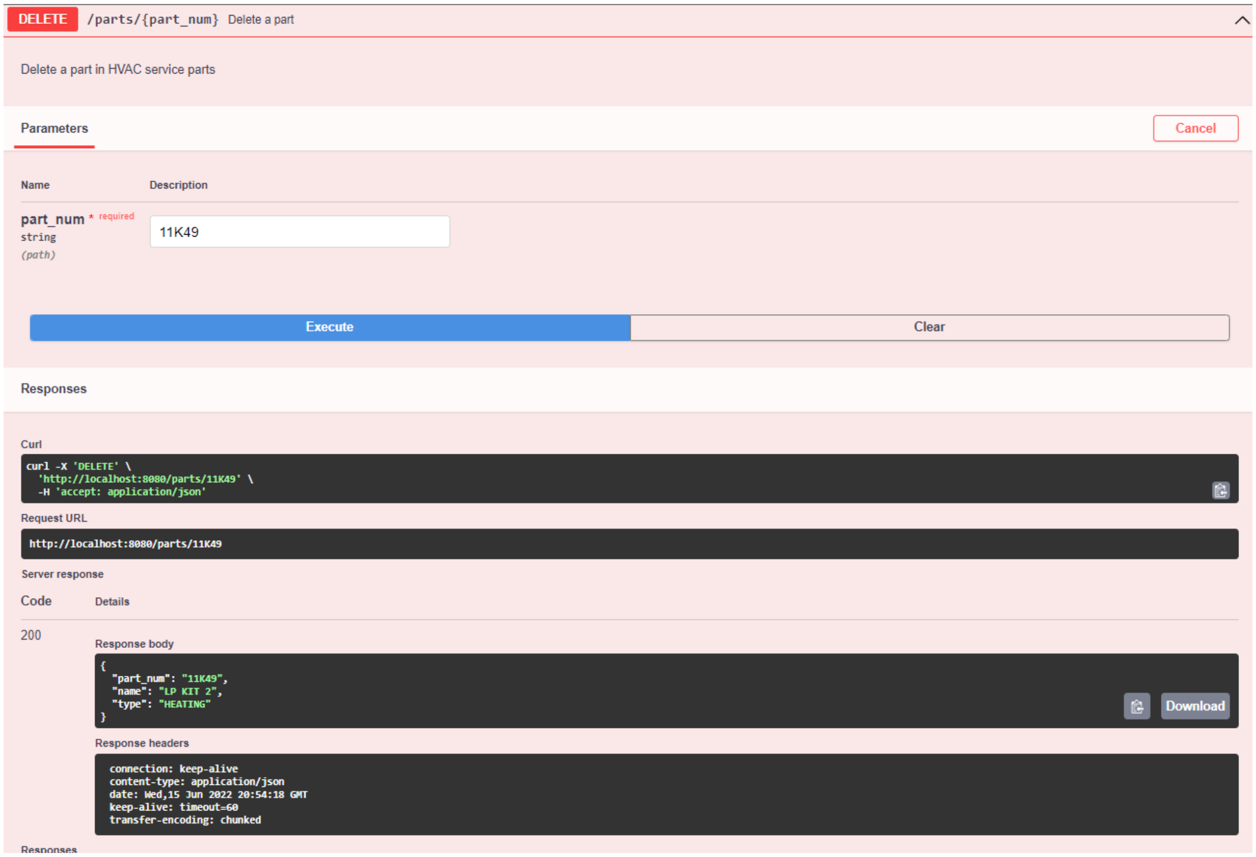
Responses

Code	Description	Links
200	The part was updated successfully Media type application/json Controls Accept header. Example Value | Schema	*No links*

```
{
  "part_num": "string",
  "name": "string",
  "type": "HEATING"
}
```

Code	Description	Links
400	Invalid reqeust Media type application/json	*No links*
404	Missing component of part, or no part found with that ID Media type application/json	*No links*
500	An unplanned error occured Media type application/json	*No links*

Delete:



DELETE `/parts/{part_num}` Delete a part

Delete a part in HVAC service parts

Parameters

Cancel

Name	Description
part_num * required string *(path)*	11K49

Execute | Clear

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:8080/parts/11K49' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/parts/11K49
```

Server response

Code	Details
200	Response body

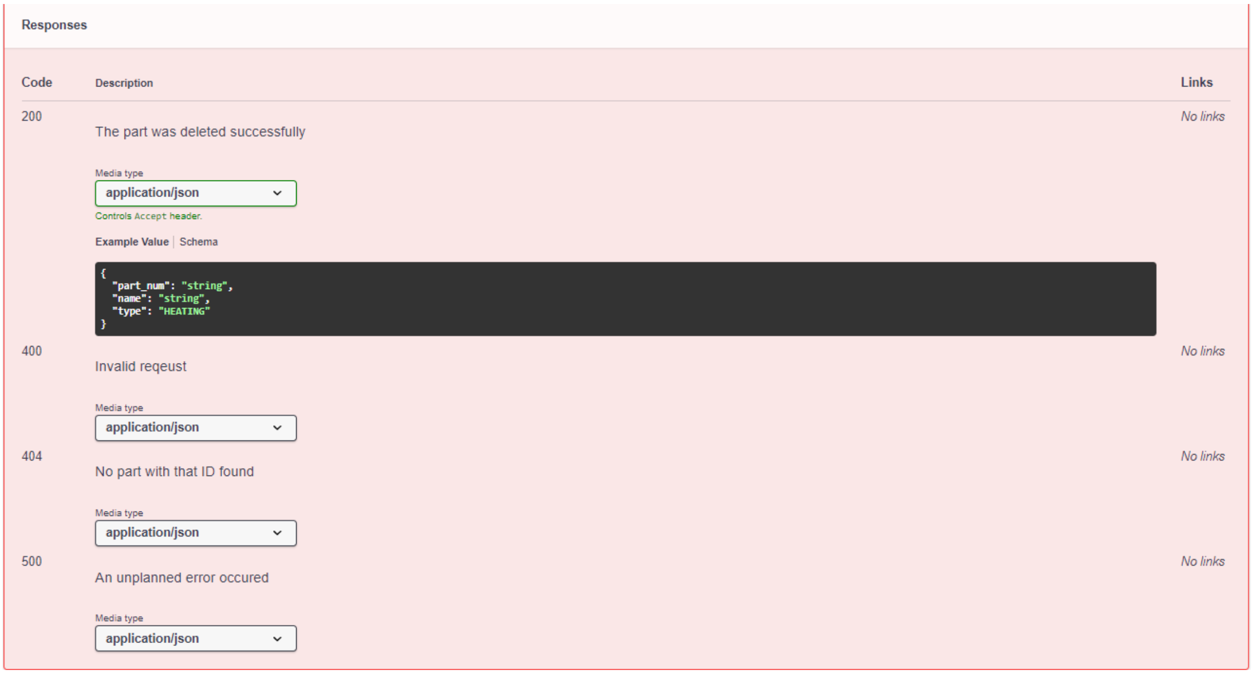
```
{
  "part_num": "11K49",
  "name": "LP KIT 2",
  "type": "HEATING"
}
```

Download

Response headers

```
connection: keep-alive
content-type: application/json
date: Wed,15 Jun 2022 20:54:18 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

Responses



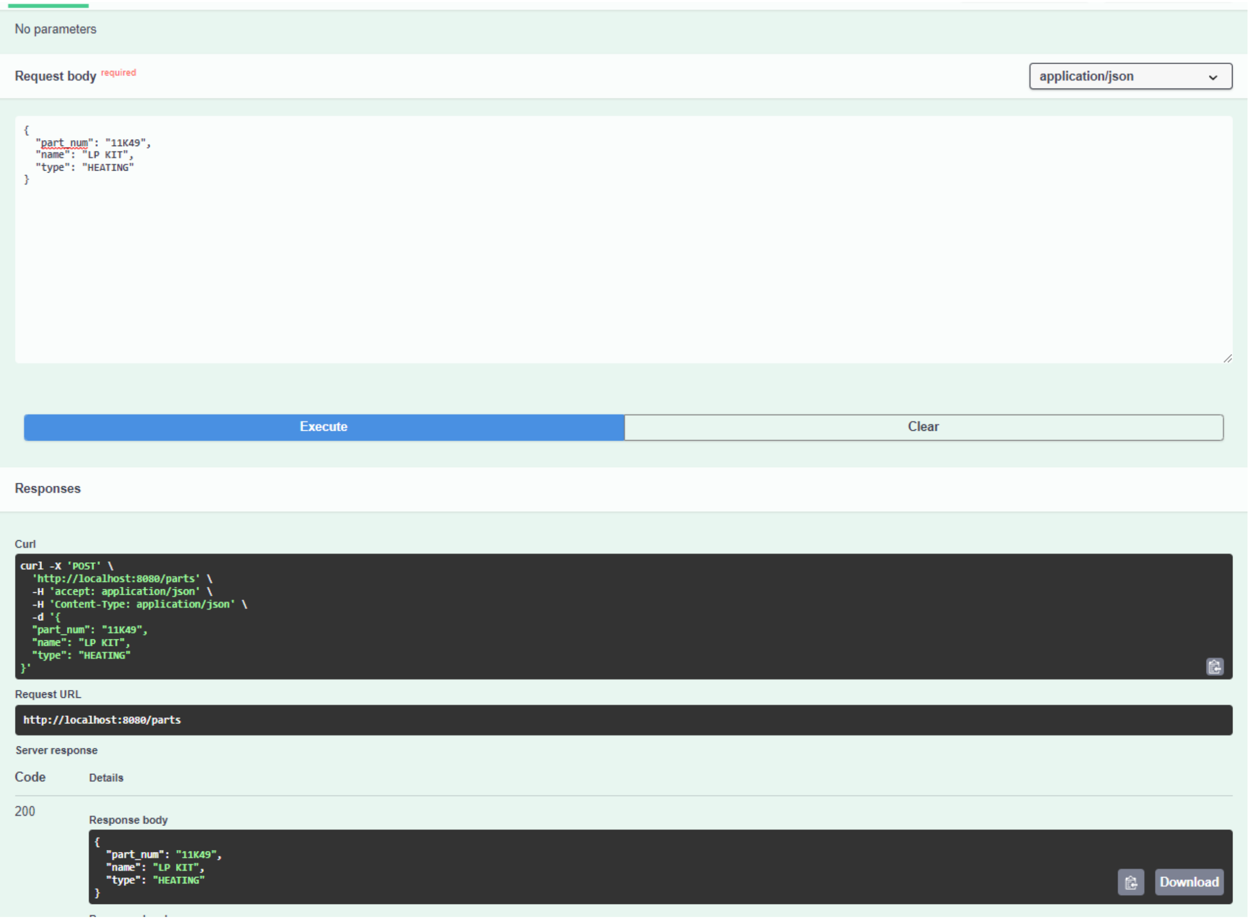
Responses

Code	Description	Links
200	The part was deleted successfully Media type application/json Controls Accept header. Example Value | Schema	*No links*

```
{
  "part_num": "string",
  "name": "string",
  "type": "HEATING"
}
```

Code	Description	Links
400	Invalid reqeust Media type application/json	*No links*
404	No part with that ID found Media type application/json	*No links*
500	An unplanned error occured Media type application/json	*No links*

Post:



No parameters

Request body required — application/json

```
{
  "part_num": "11K49",
  "name": "LP KIT",
  "type": "HEATING"
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/parts' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "part_num": "11K49",
  "name": "LP KIT",
  "type": "HEATING"
}'
```

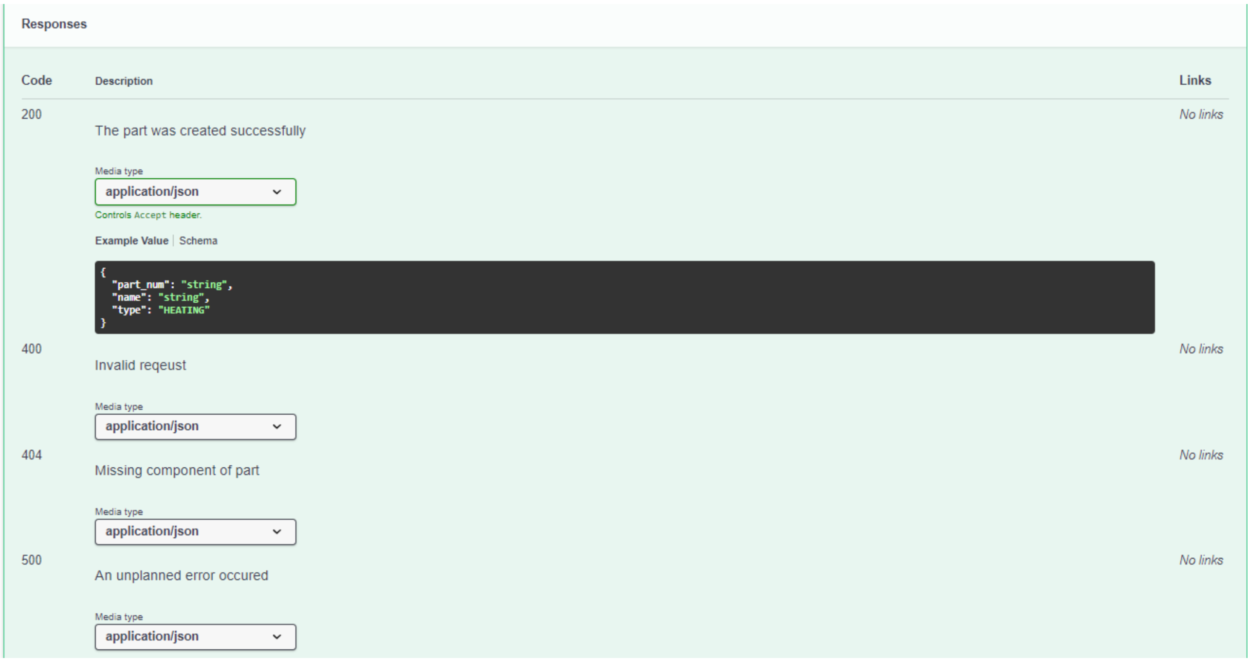
Request URL

```
http://localhost:8080/parts
```

Server response

Code	Details
200	Response body

```
{
  "part_num": "11K49",
  "name": "LP KIT",
  "type": "HEATING"
}
```



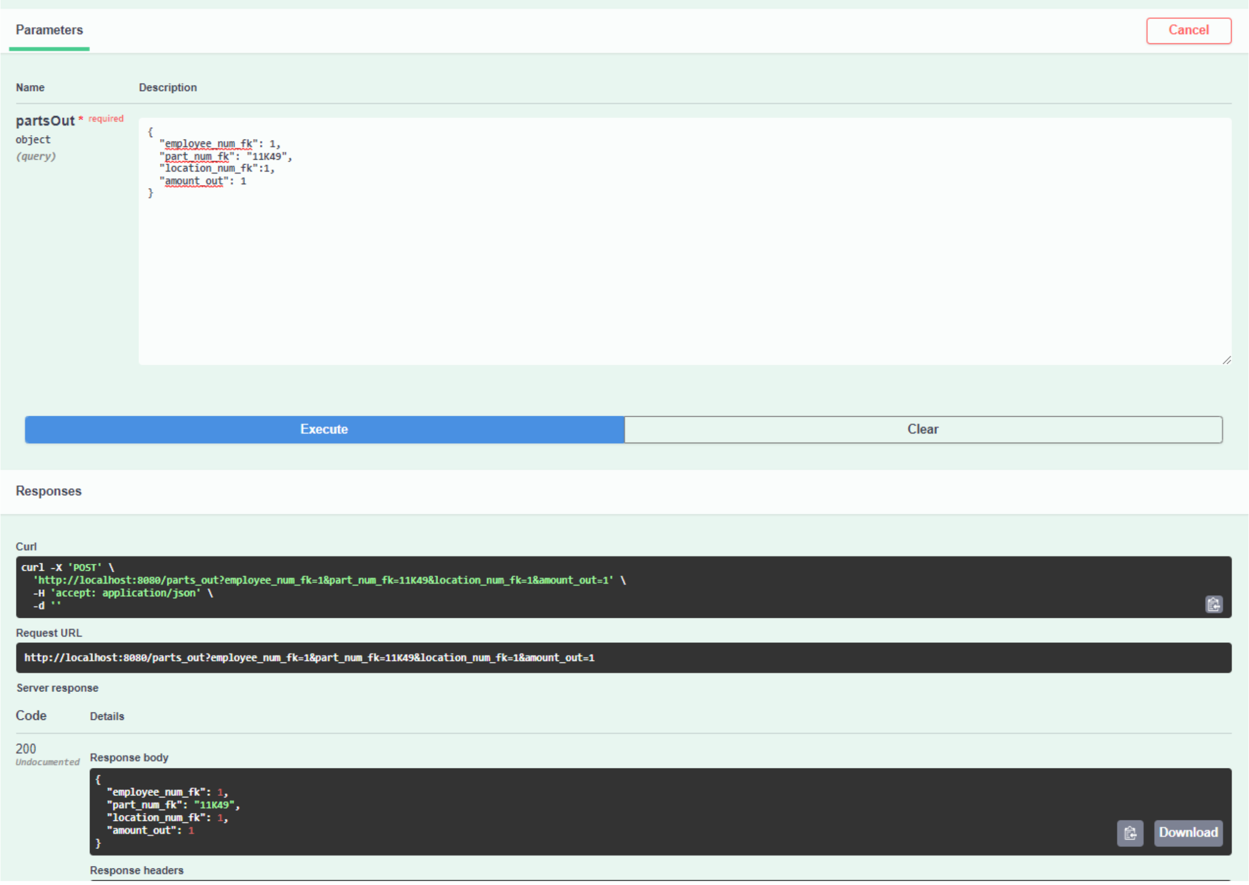
Responses

Code	Description	Links
200	The part was created successfully Media type: application/json Controls Accept header. Example Value | Schema	*No links*
400	Invalid reqeust Media type: application/json	*No links*
404	Missing component of part Media type: application/json	*No links*
500	An unplanned error occured Media type: application/json	*No links*

```
{
  "part_num": "string",
  "name": "string",
  "type": "HEATING"
}
```

Parts Out:

Post:



Parameters

Name	Description
partsOut * required object *(query)*	

```
{
  "employee_num_fk": 1,
  "part_num_fk": "11K49",
  "location_num_fk":1,
  "amount_out": 1
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/parts_out?employee_num_fk=1&part_num_fk=11K49&location_num_fk=1&amount_out=1' \
  -H 'accept: application/json' \
  -d ''
```

Request URL

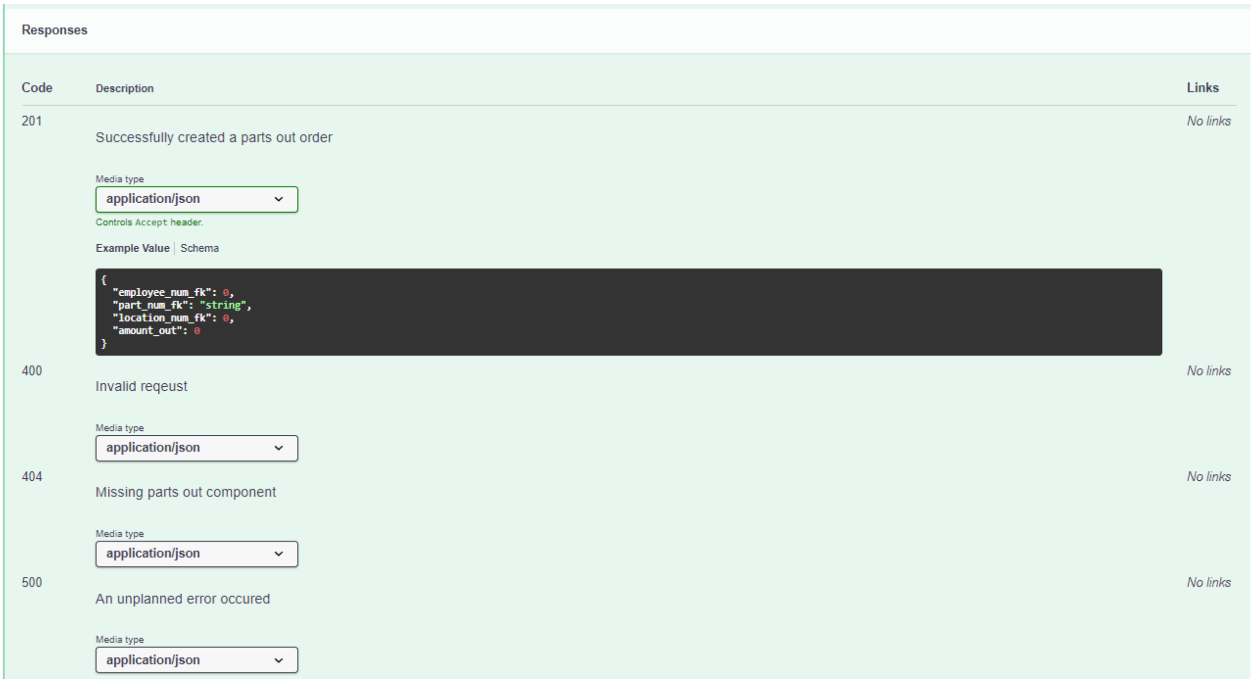
```
http://localhost:8080/parts_out?employee_num_fk=1&part_num_fk=11K49&location_num_fk=1&amount_out=1
```

Server response

Code	Details
200 *Undocumented*	Response body

```
{
  "employee_num_fk": 1,
  "part_num_fk": "11K49",
  "location_num_fk": 1,
  "amount_out": 1
}
```

Response headers



Responses

Code | Description | Links

201

Successfully created a parts out order

No links

Media type: application/json

Controls Accept header.

Example Value | Schema

```
{
  "employee_num_fk": 0,
  "part_num_fk": "string",
  "location_num_fk": 0,
  "amount_out": 0
}
```

400

Invalid reqeust

No links

Media type: application/json

404

Missing parts out component

No links

Media type: application/json

500

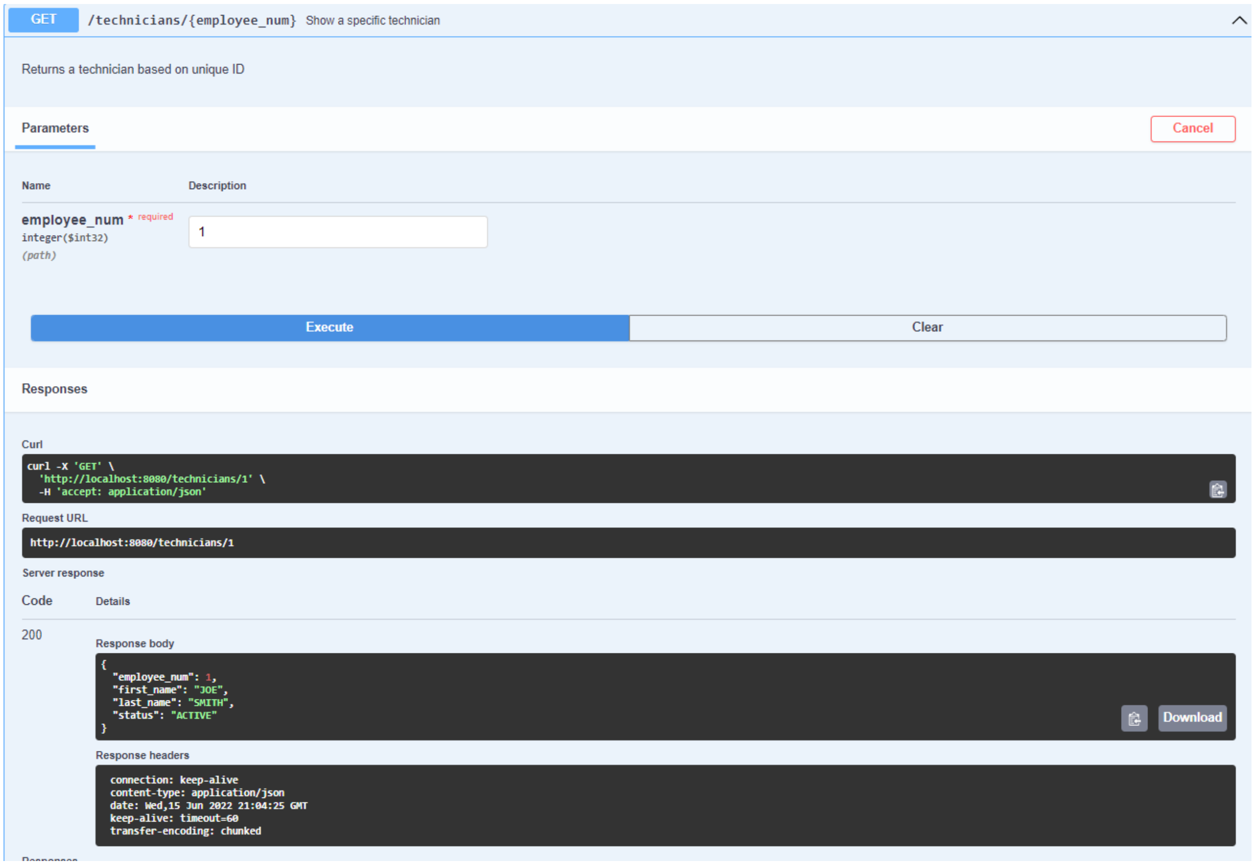
An unplanned error occured

No links

Media type: application/json

Technicians:

Get:



GET /technicians/{employee_num} Show a specific technician

Returns a technician based on unique ID

Parameters

Name | Description

employee_num * required
integer($int32)
(path)

1

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/technicians/1' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/technicians/1
```

Server response

Code | Details

200

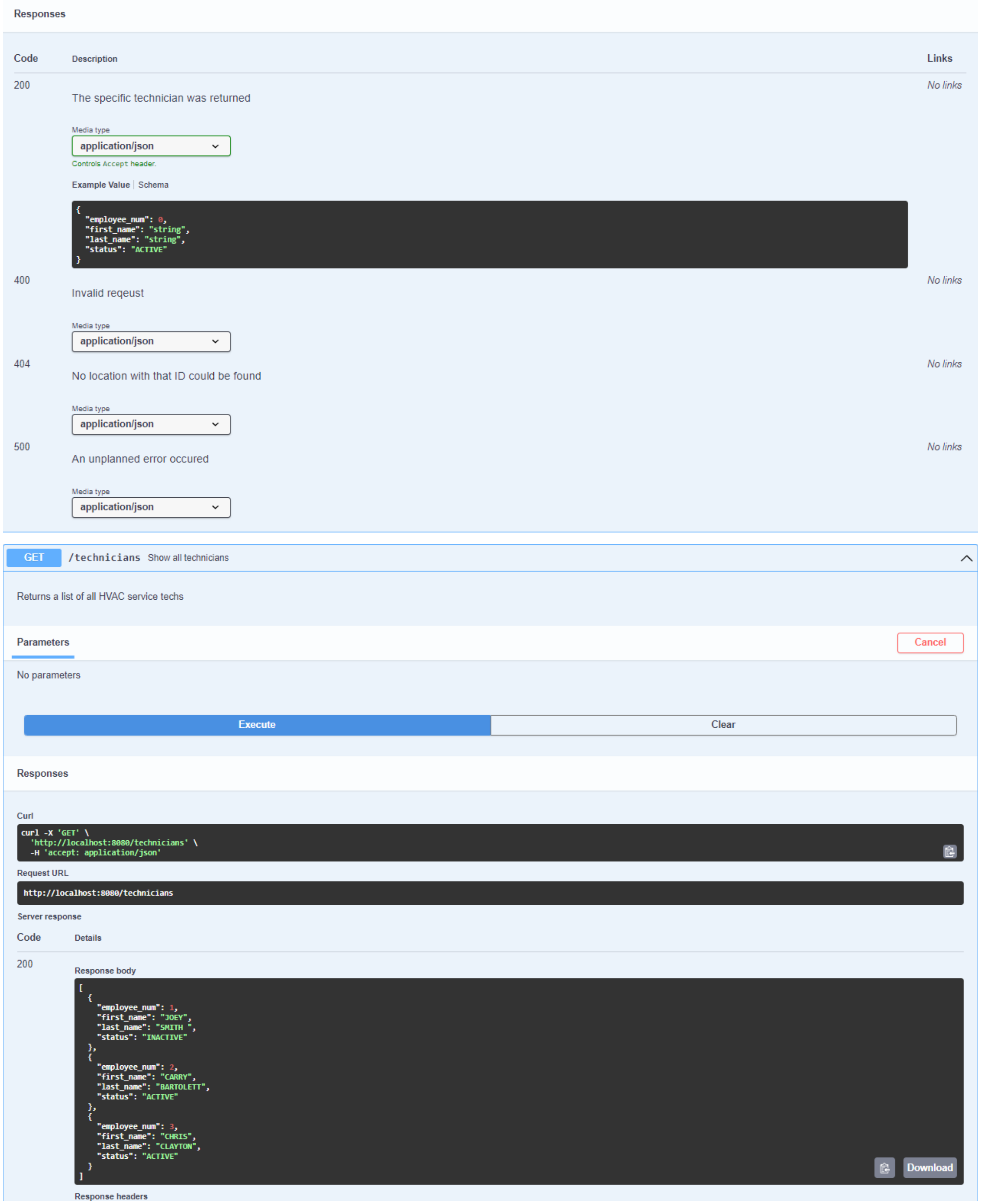
Response body

```
{
  "employee_num": 1,
  "first_name": "JOE",
  "last_name": "SMITH",
  "status": "ACTIVE"
}
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,15 Jun 2022 21:04:25 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
```

Responses



Responses

Code	Description	Links
200	The specific technician was returned	*No links*

Media type

application/json

Controls `Accept` header.

Example Value | Schema

```
{
  "employee_num": 0,
  "first_name": "string",
  "last_name": "string",
  "status": "ACTIVE"
}
```

400	Invalid reqeust Media type application/json	*No links*
404	No location with that ID could be found Media type application/json	*No links*
500	An unplanned error occured Media type application/json	*No links*

GET `/technicians` Show all technicians

Returns a list of all HVAC service techs

Parameters

No parameters

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/technicians' \
  -H 'accept: application/json'
```

Request URL

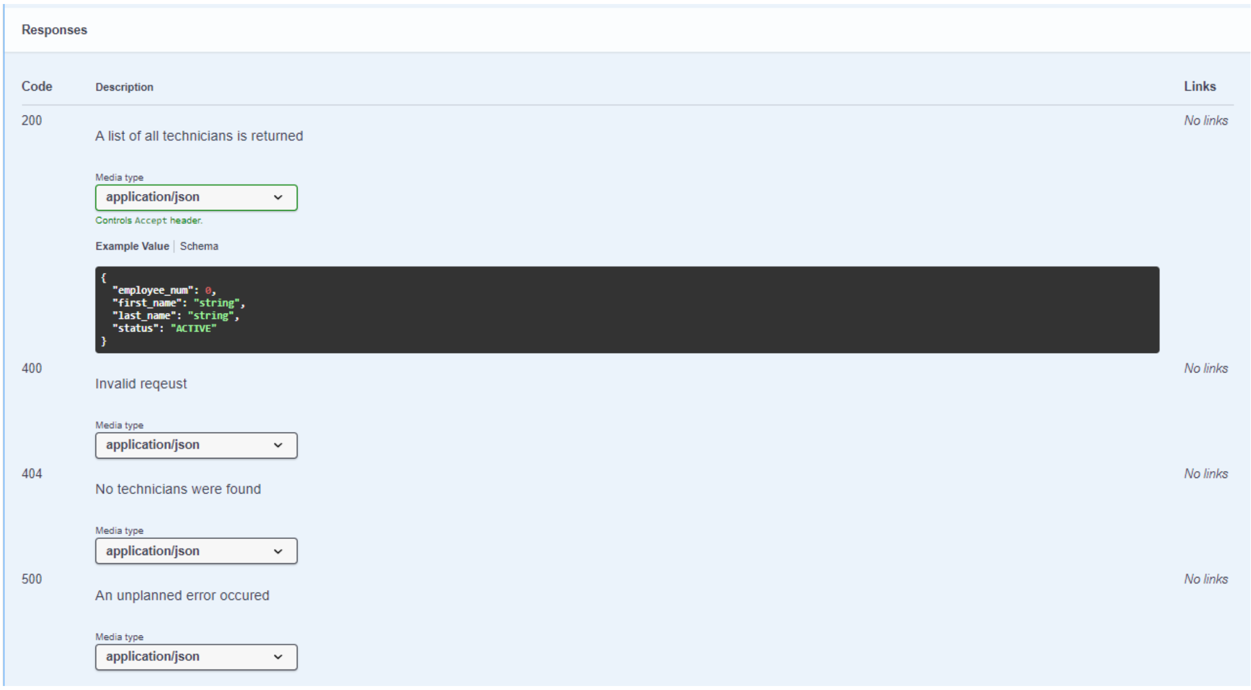
```
http://localhost:8080/technicians
```

Server response

Code	Details
200	Response body

```
[
  {
    "employee_num": 1,
    "first_name": "JOEY",
    "last_name": "SMITH ",
    "status": "INACTIVE"
  },
  {
    "employee_num": 2,
    "first_name": "CARRY",
    "last_name": "BARTOLETT",
    "status": "ACTIVE"
  },
  {
    "employee_num": 3,
    "first_name": "CHRIS",
    "last_name": "CLAYTON",
    "status": "ACTIVE"
  }
]
```

Response headers

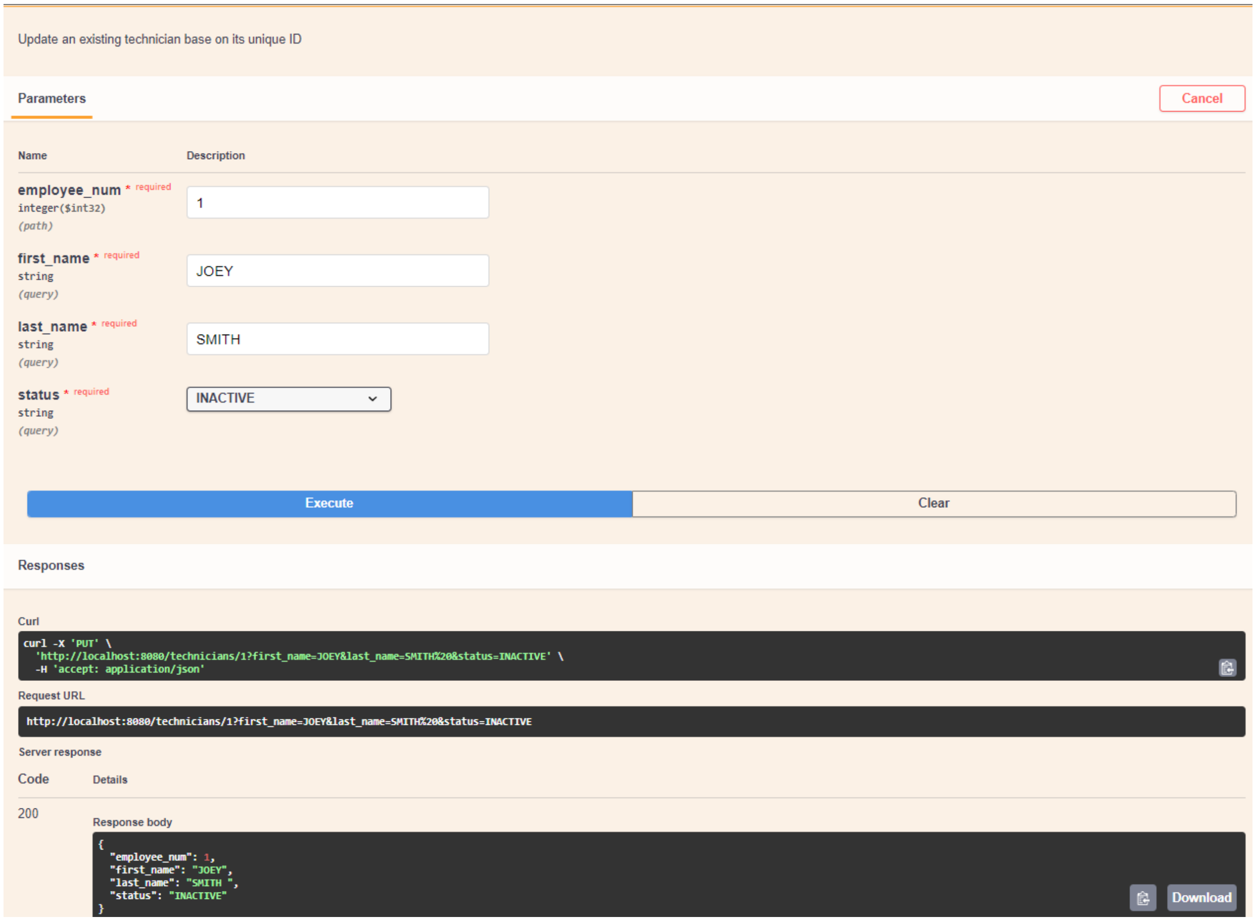


Responses

Code	Description	Links
200	A list of all technicians is returned Media type application/json Controls Accept header. Example Value | Schema	No links
400	Invalid reqeust Media type application/json	No links
404	No technicians were found Media type application/json	No links
500	An unplanned error occured Media type application/json	No links

```
{
  "employee_num": 0,
  "first_name": "string",
  "last_name": "string",
  "status": "ACTIVE"
}
```

Put:



Update an existing technician base on its unique ID

Parameters

Name	Description
employee_num * required integer($int32) (path)	1
first_name * required string (query)	JOEY
last_name * required string (query)	SMITH
status * required string (query)	INACTIVE

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:8080/technicians/1?first_name=JOEY&last_name=SMITH%20&status=INACTIVE' \
  -H 'accept: application/json'
```

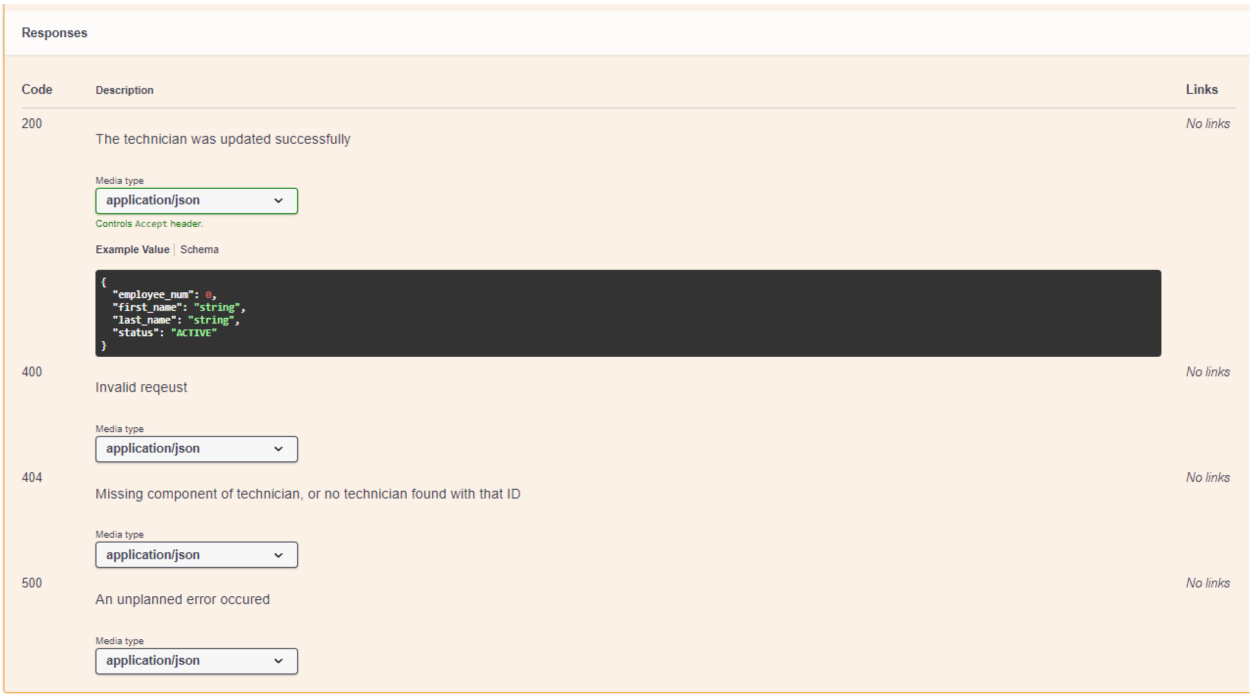
Request URL

```
http://localhost:8080/technicians/1?first_name=JOEY&last_name=SMITH%20&status=INACTIVE
```

Server response

Code	Details
200	Response body

```
{
  "employee_num": 1,
  "first_name": "JOEY",
  "last_name": "SMITH ",
  "status": "INACTIVE"
}
```



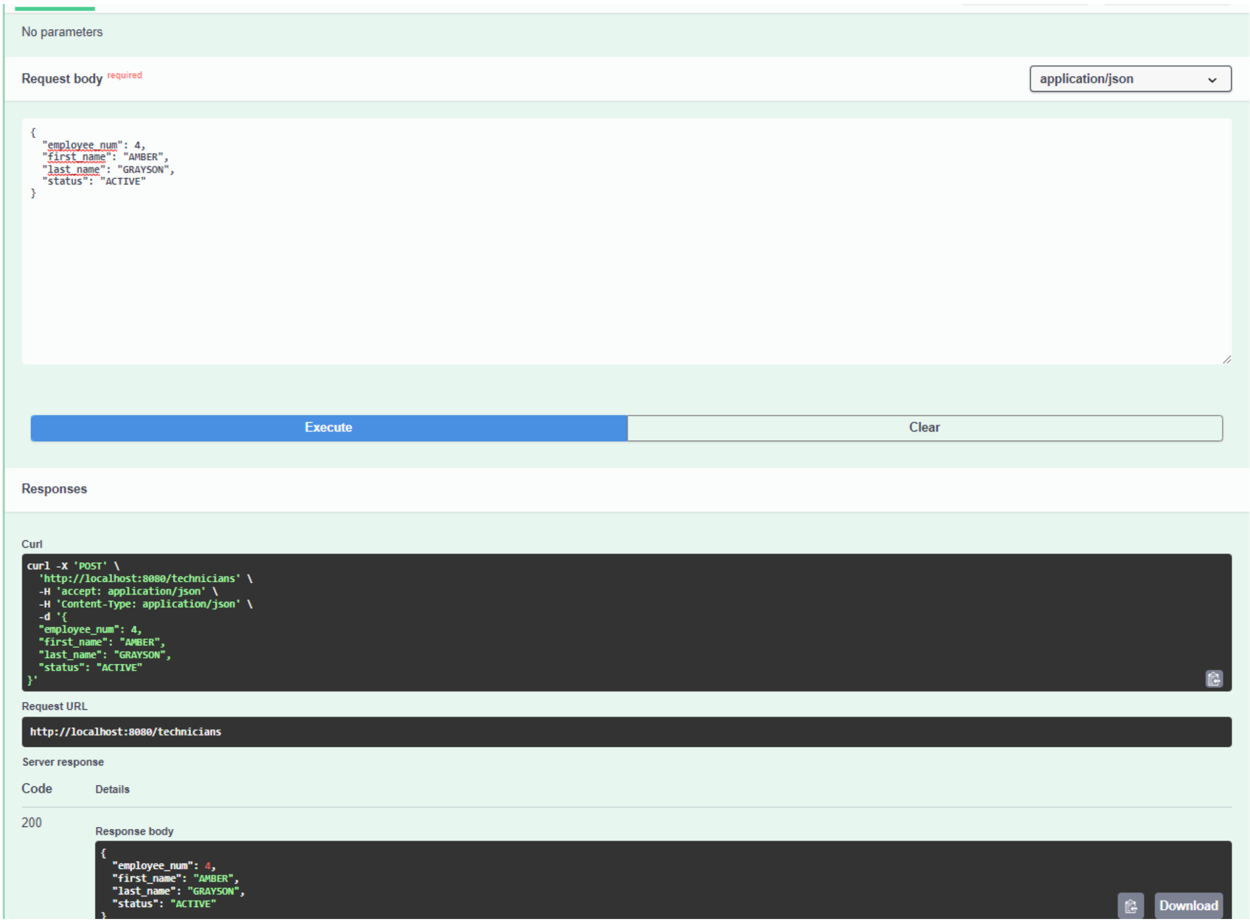
Responses

Code	Description	Links
200	The technician was updated successfully Media type application/json Controls Accept header. Example Value | Schema	*No links*

```
{
  "employee_num": 0,
  "first_name": "string",
  "last_name": "string",
  "status": "ACTIVE"
}
```

Code	Description	Links
400	Invalid reqeust Media type application/json	*No links*
404	Missing component of technician, or no technician found with that ID Media type application/json	*No links*
500	An unplanned error occured Media type application/json	*No links*

Post:



No parameters

Request body required

application/json

```
{
  "employee_num": 4,
  "first_name": "AMBER",
  "last_name": "GRAYSON",
  "status": "ACTIVE"
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/technicians' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "employee_num": 4,
  "first_name": "AMBER",
  "last_name": "GRAYSON",
  "status": "ACTIVE"
}'
```

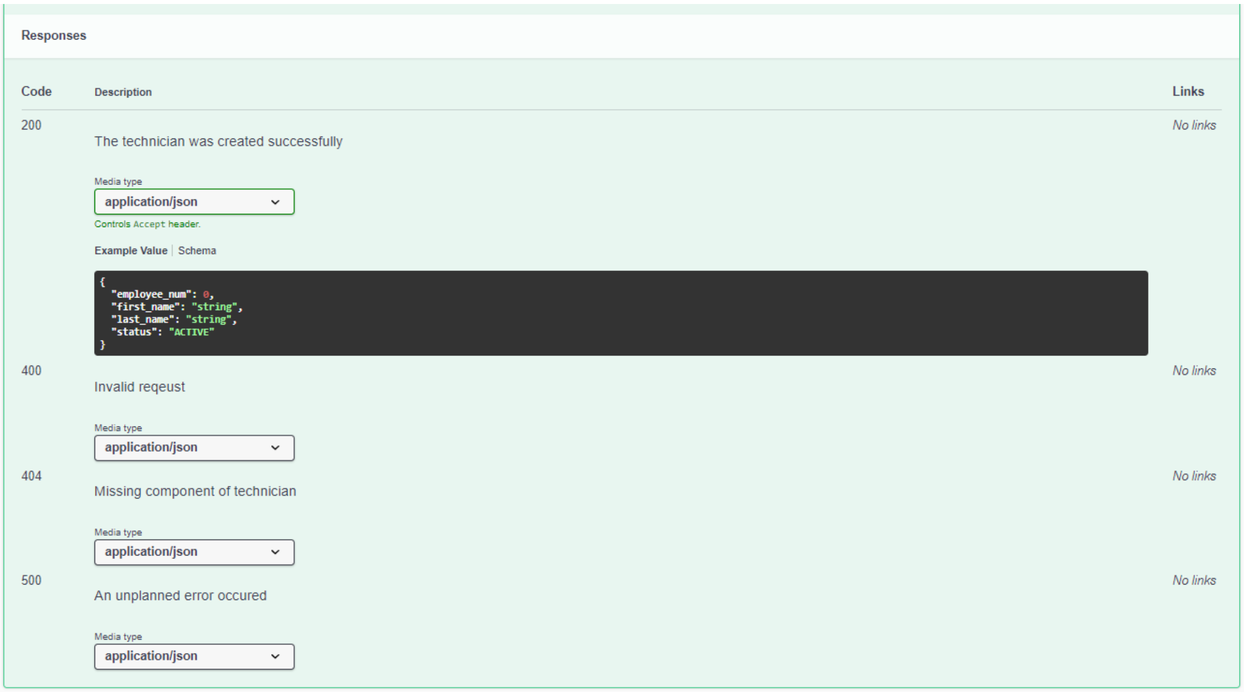
Request URL

```
http://localhost:8080/technicians
```

Server response

Code	Details
200	Response body

```
{
  "employee_num": 4,
  "first_name": "AMBER",
  "last_name": "GRAYSON",
  "status": "ACTIVE"
}
```



Responses

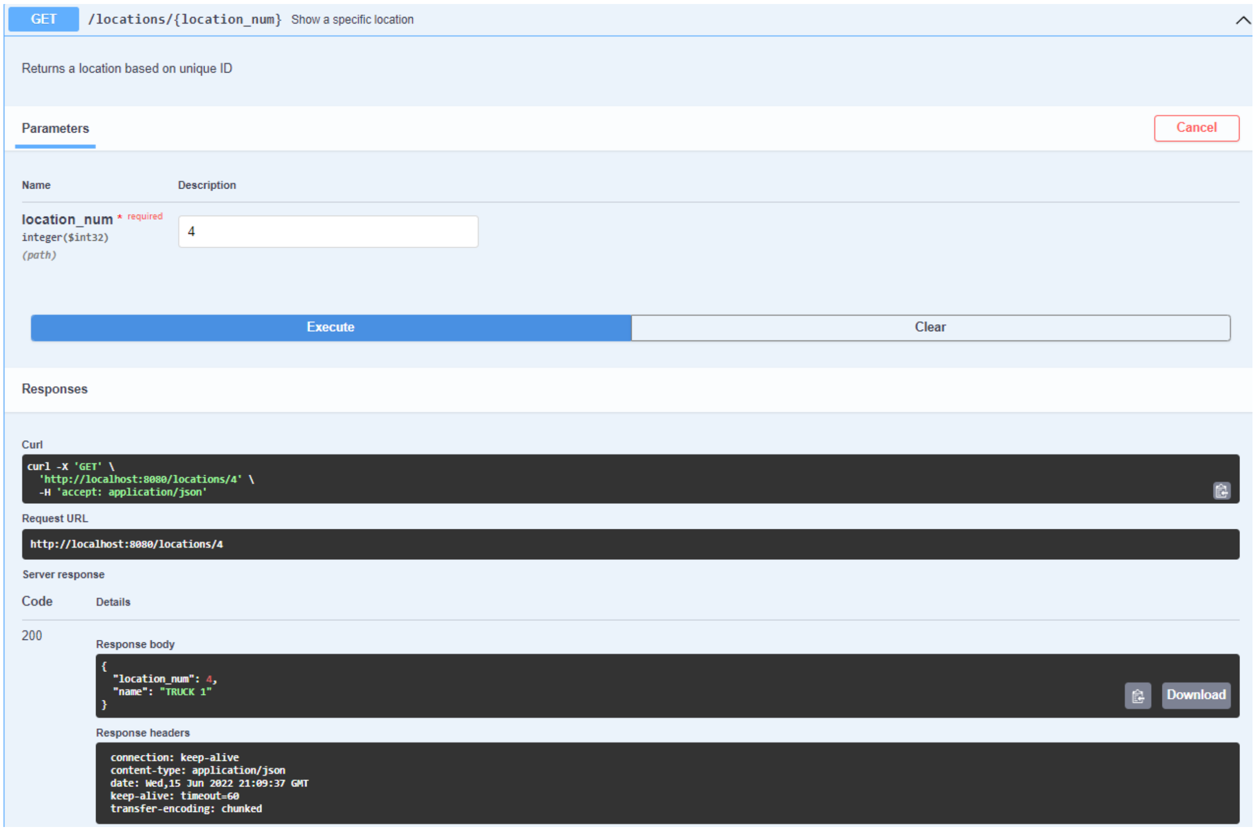
Code	Description	Links
200	The technician was created successfully Media type application/json Controls Accept header. Example Value | Schema	No links
400	Invalid reqeust Media type application/json	No links
404	Missing component of technician Media type application/json	No links
500	An unplanned error occured Media type application/json	No links

Example value for 200:

```
{
  "employee_num": 0,
  "first_name": "string",
  "last_name": "string",
  "status": "ACTIVE"
}
```

Locations:

Get:



`/locations/{location_num}` Show a specific location

Returns a location based on unique ID

Parameters

Name	Description
location_num * required integer($int32) *(path)*	4

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/locations/4' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/locations/4
```

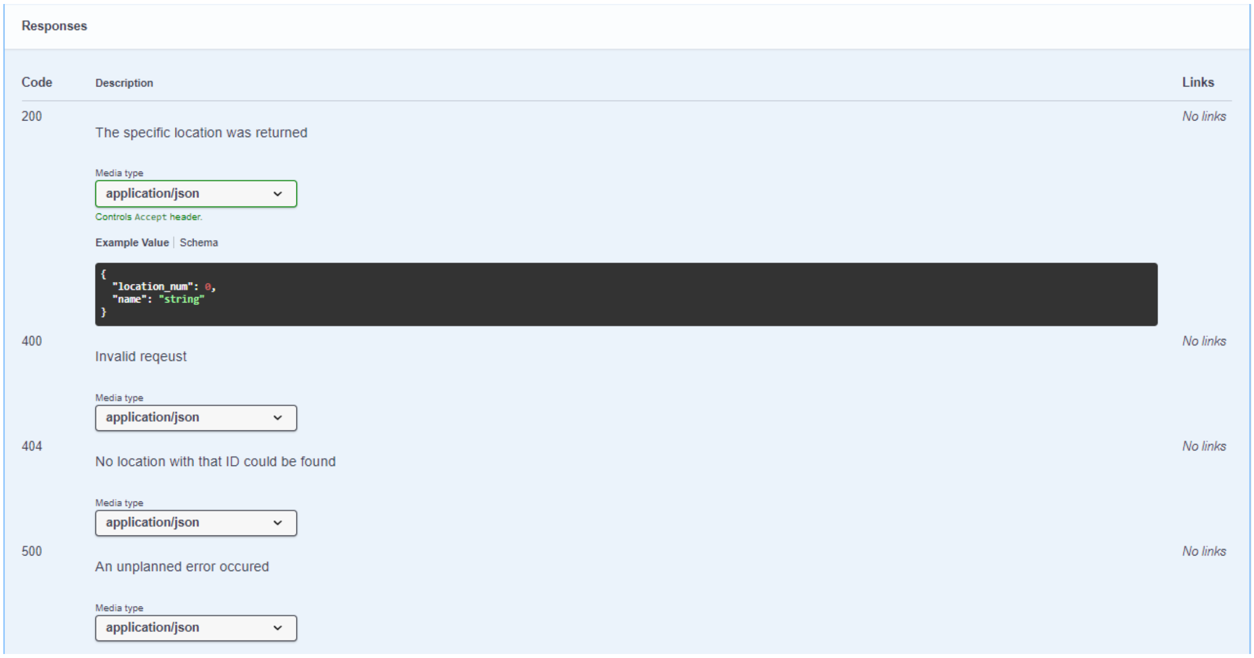
Server response

Code	Details
200	Response body

```
{
  "location_num": 4,
  "name": "TRUCK 1"
}
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,15 Jun 2022 21:09:37 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
```



Responses

Code	Description	Links
200	The specific location was returned	*No links*
400	Invalid reqeust	*No links*
404	No location with that ID could be found	*No links*
500	An unplanned error occured	*No links*

200

Media type: application/json

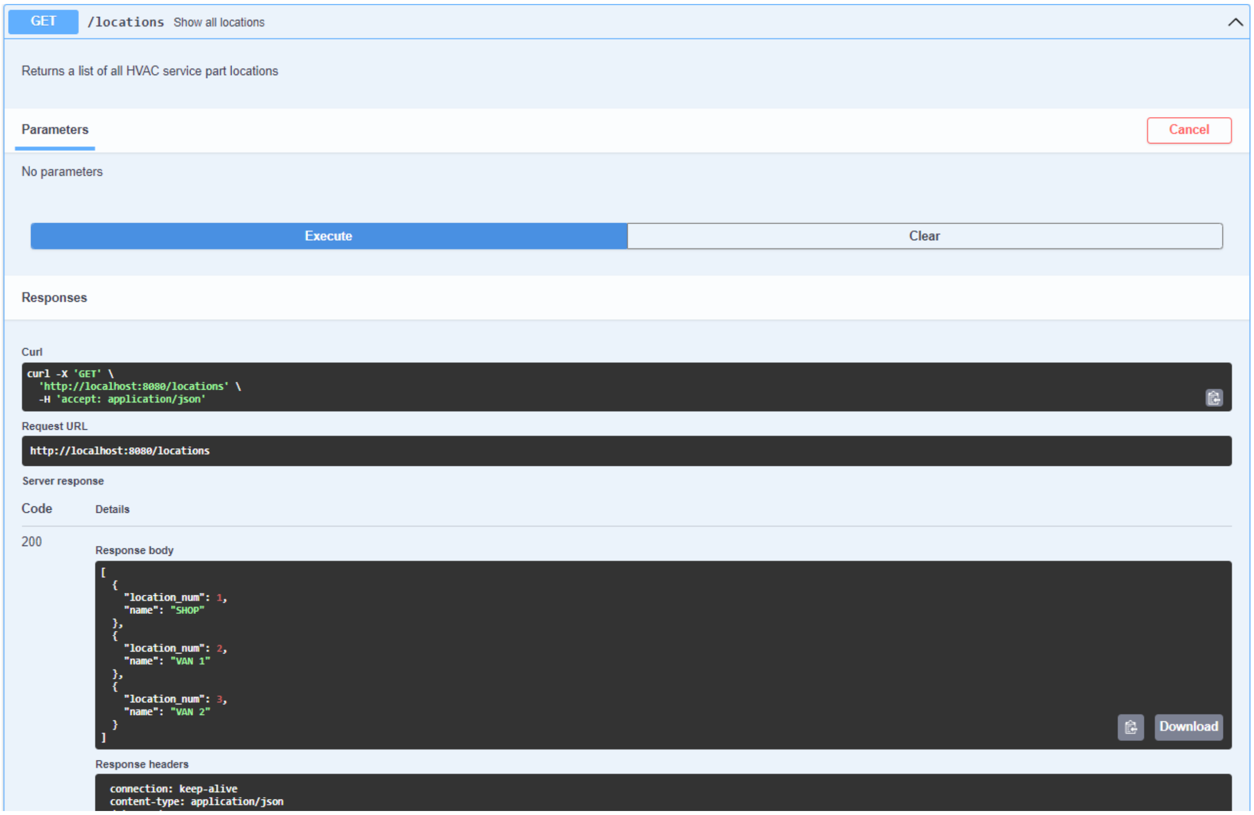
Controls Accept header.

Example Value | Schema

```
{
  "location_num": 0,
  "name": "string"
}
```

400, 404 and 500

Media type: application/json



/locations Show all locations

Returns a list of all HVAC service part locations

Parameters

No parameters

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/locations' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/locations
```

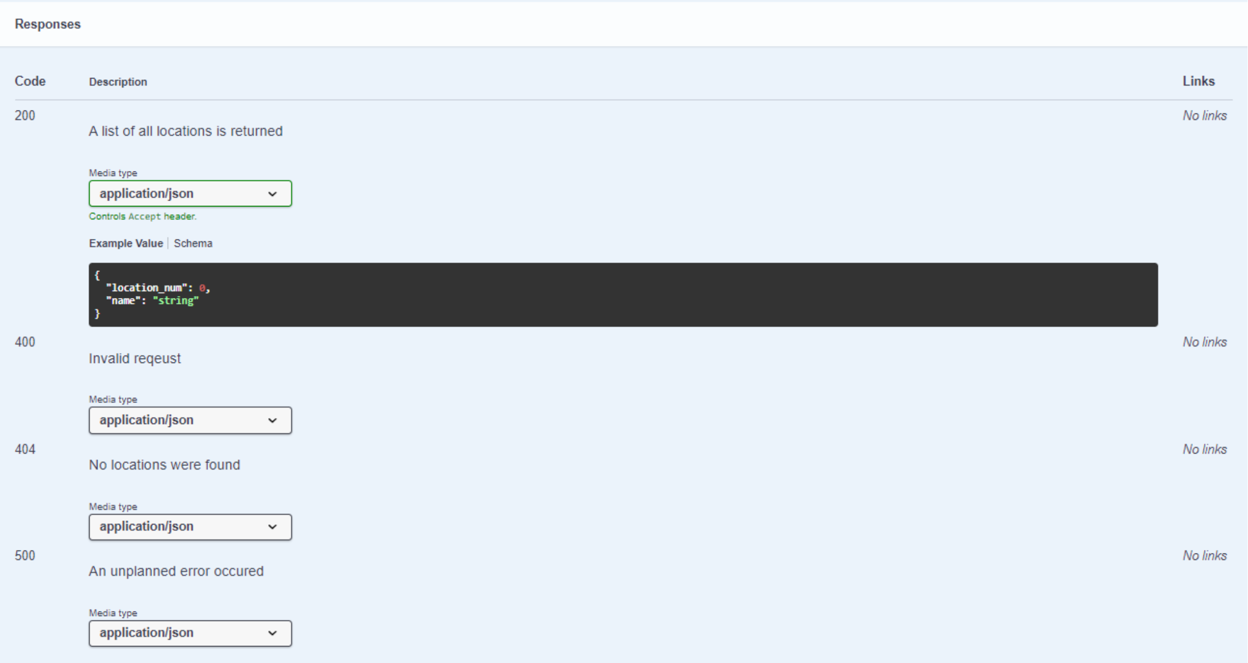
Server response

Code	Details
200	Response body

```
[
  {
    "location_num": 1,
    "name": "SHOP"
  },
  {
    "location_num": 2,
    "name": "VAN 1"
  },
  {
    "location_num": 3,
    "name": "VAN 2"
  }
]
```

Response headers

```
 connection: keep-alive
 content-type: application/json
```

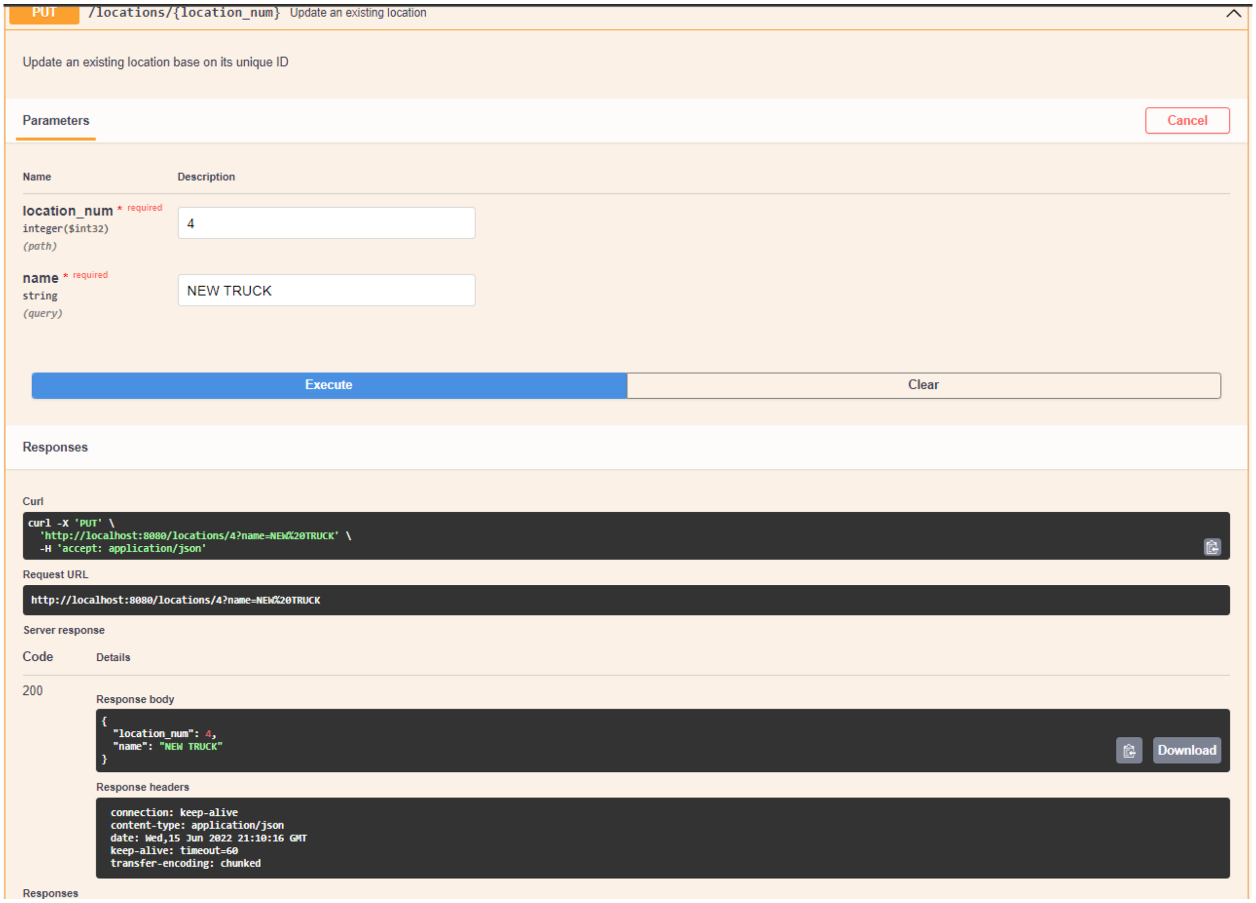


Responses

Code	Description	Links
200	A list of all locations is returned Media type: application/json Controls Accept header. Example Value | Schema	No links
400	Invalid reqeust Media type: application/json	No links
404	No locations were found Media type: application/json	No links
500	An unplanned error occured Media type: application/json	No links

```
{
  "location_num": 0,
  "name": "string"
}
```

Put:



/locations/{location_num} Update an existing location

Update an existing location base on its unique ID

Parameters

Name	Description
location_num * required integer($int32) *(path)*	4
name * required string *(query)*	NEW TRUCK

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:8080/locations/4?name=NEW%20TRUCK' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/locations/4?name=NEW%20TRUCK
```

Server response

Code: 200

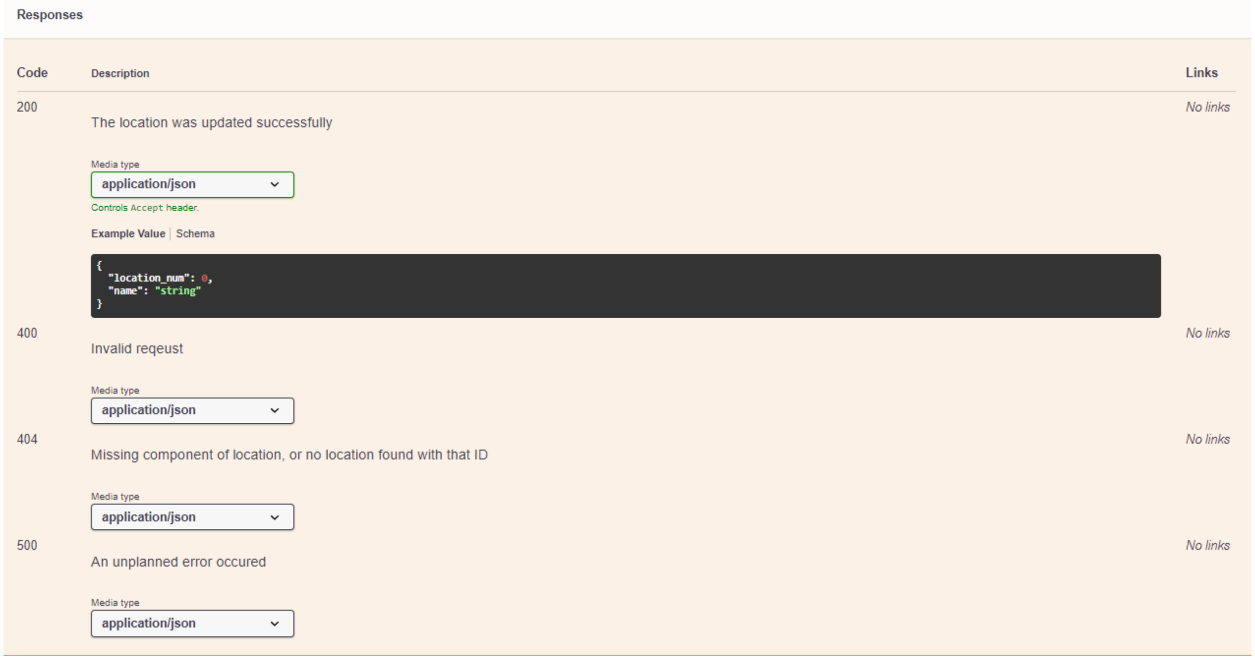
Response body

```
{
  "location_num": 4,
  "name": "NEW TRUCK"
}
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,15 Jun 2022 21:10:16 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
```

Responses

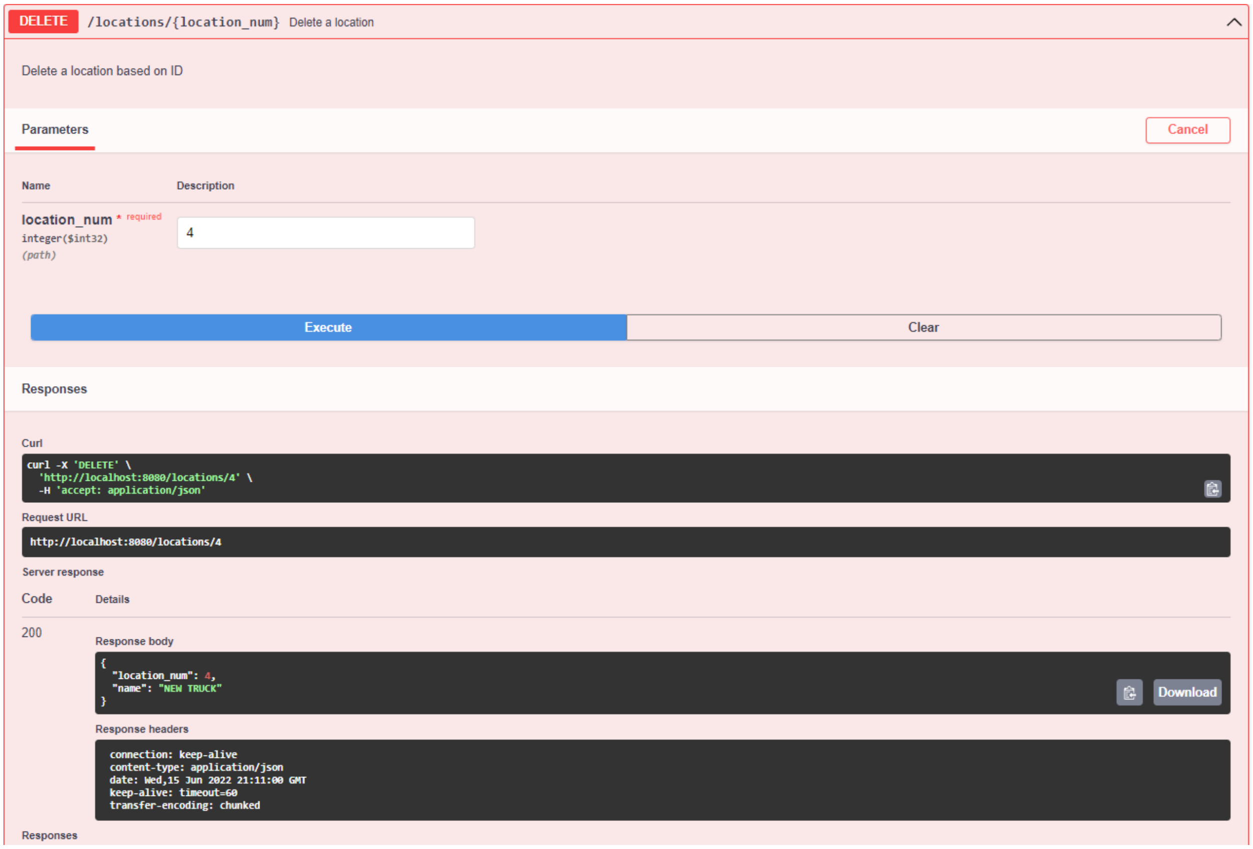


Responses

Code	Description	Links
200	The location was updated successfully Media type application/json Controls Accept header. Example Value | Schema	*No links*
400	Invalid reqeust Media type application/json	*No links*
404	Missing component of location, or no location found with that ID Media type application/json	*No links*
500	An unplanned error occured Media type application/json	*No links*

```
{
  "location_num": 0,
  "name": "string"
}
```

Delete:



/locations/{location_num} Delete a location

Delete a location based on ID

Name	Description
location_num * required integer($int32) *(path)*	4

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:8080/locations/4' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/locations/4
```

Server response

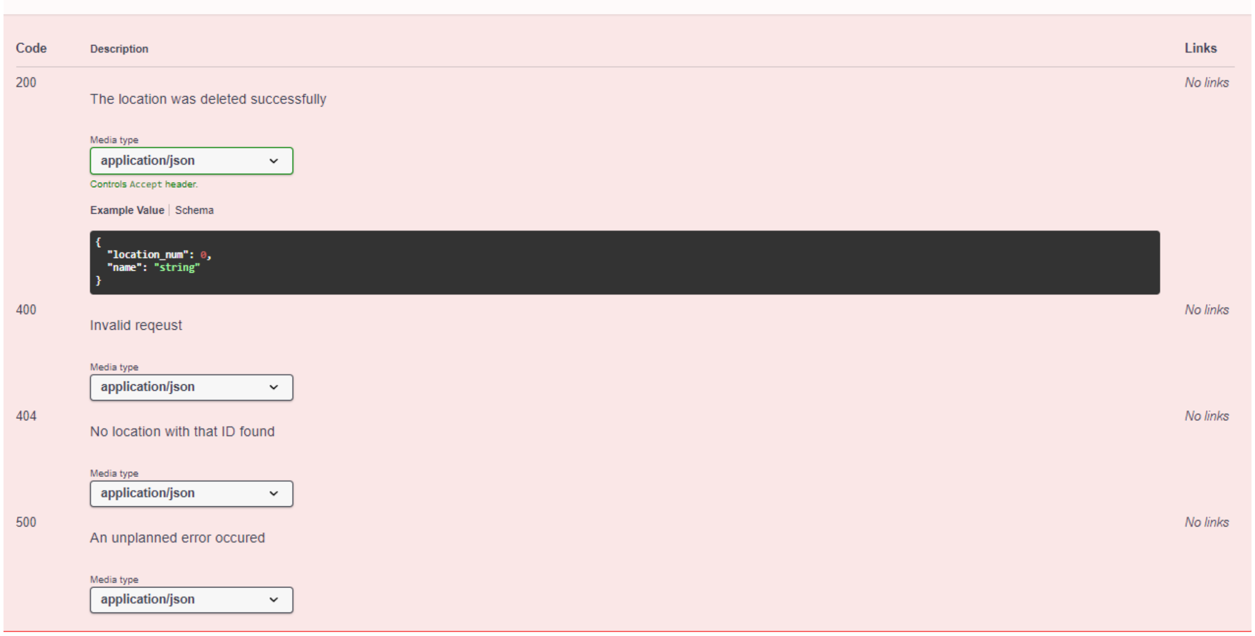
Code	Details
200	Response body

```
{
  "location_num": 4,
  "name": "NEW TRUCK"
}
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Wed,15 Jun 2022 21:11:00 GMT
keep-alive: timeout=60
transfer-encoding: chunked
```

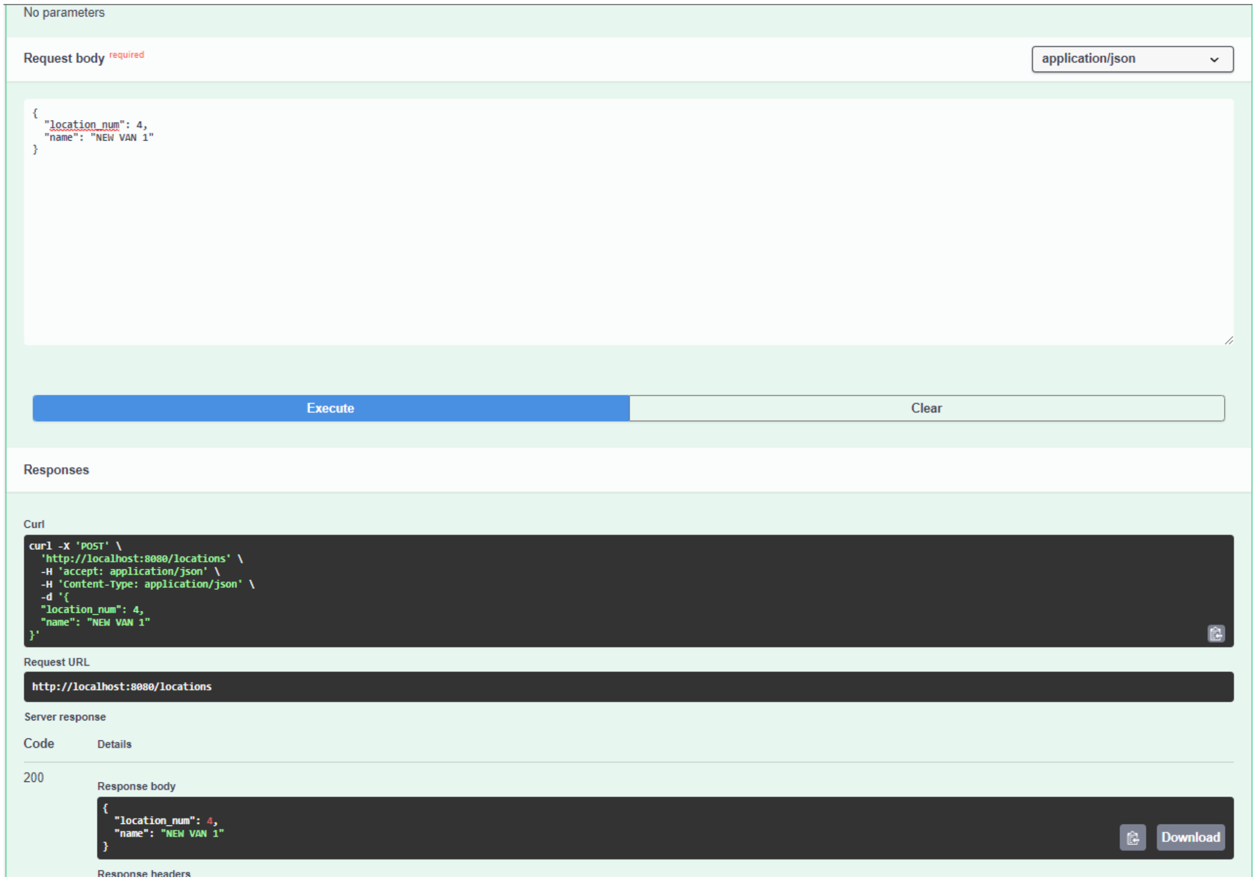
Responses



Code	Description	Links
200	The location was deleted successfully Media type: application/json Controls Accept header. Example Value | Schema	*No links*
400	Invalid reqeust Media type: application/json	*No links*
404	No location with that ID found Media type: application/json	*No links*
500	An unplanned error occured Media type: application/json	*No links*

```
{
  "location_num": 0,
  "name": "string"
}
```

Post:



No parameters

Request body required — application/json

```
{
  "location_num": 4,
  "name": "NEW VAN 1"
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/locations' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "location_num": 4,
  "name": "NEW VAN 1"
}'
```

Request URL

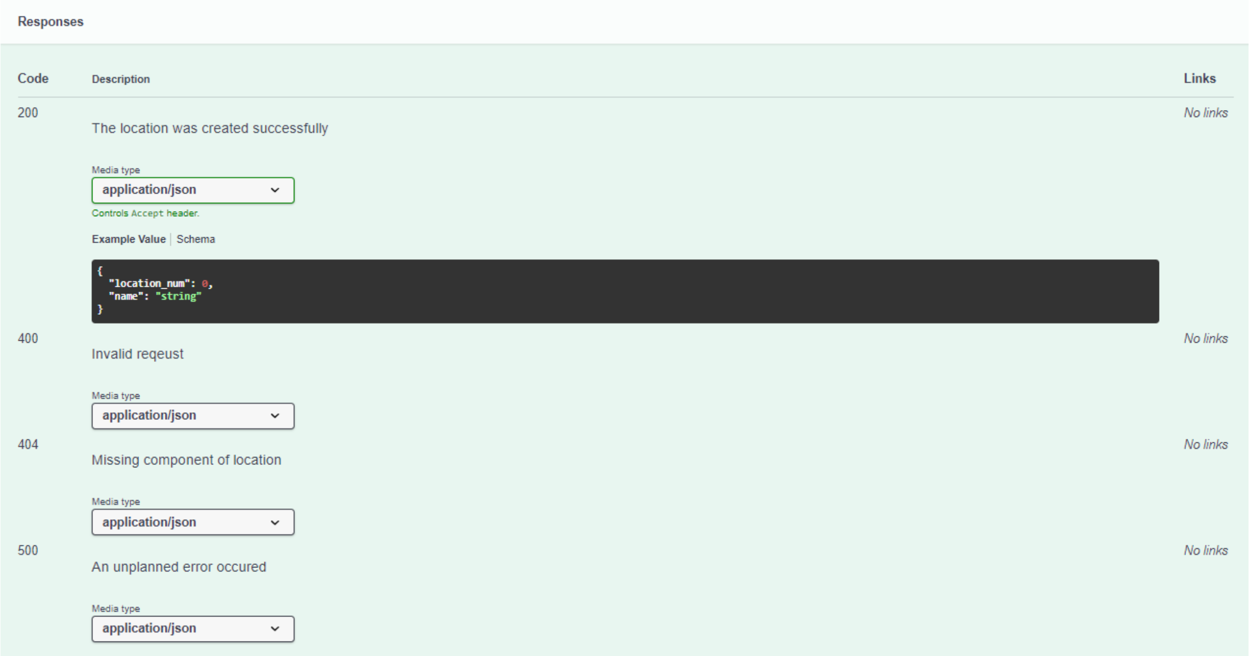
```
http://localhost:8080/locations
```

Server response

Code	Details
200	Response body

```
{
  "location_num": 4,
  "name": "NEW VAN 1"
}
```

Response headers

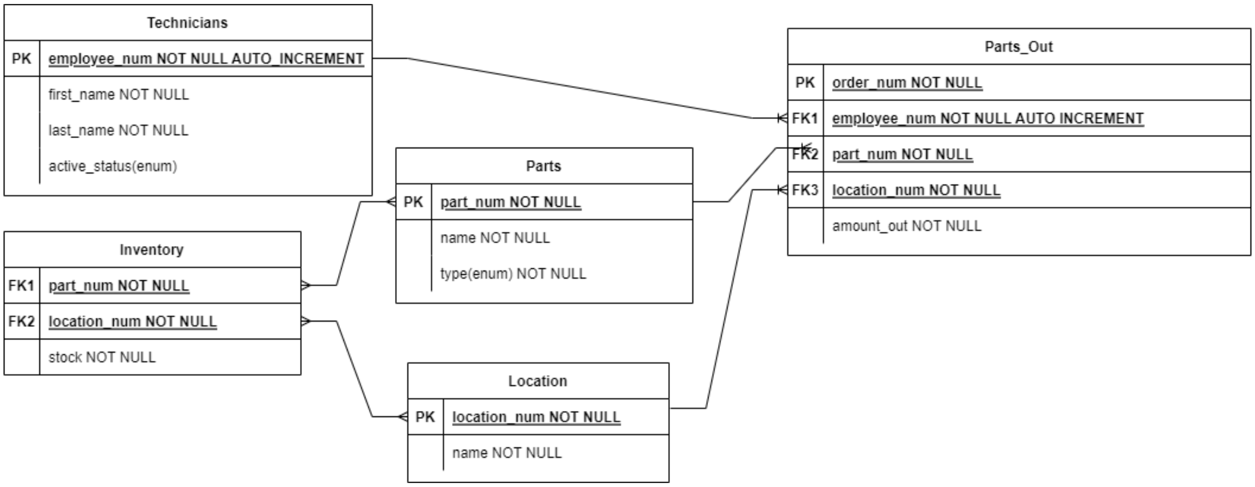


Responses

Code	Description	Links
200	The location was created successfully Media type application/json Controls Accept header. Example Value | Schema	*No links*
400	Invalid reqeust Media type application/json	*No links*
404	Missing component of location Media type application/json	*No links*
500	An unplanned error occured Media type application/json	*No links*

```
{
  "location_num": 0,
  "name": "string"
}
```

ERD



Technicians	
PK	employee_num NOT NULL AUTO_INCREMENT
	first_name NOT NULL
	last_name NOT NULL
	active_status(enum)

Parts_Out	
PK	order_num NOT NULL
FK1	employee_num NOT NULL AUTO INCREMENT
FK2	part_num NOT NULL
FK3	location_num NOT NULL
	amount_out NOT NULL

Parts	
PK	part_num NOT NULL
	name NOT NULL
	type(enum) NOT NULL

Inventory	
FK1	part_num NOT NULL
FK2	location_num NOT NULL
	stock NOT NULL

Location	
PK	location_num NOT NULL
	name NOT NULL

Web API Documentation:

Inventory: inventory as many URLS do to that fact that part numbers are strings not integers. Spring was unhappy about that so each one got its own URL to avoid errors.

Url: /inventory/update/{part_num}/{location_num}

- **Allowed requests**: PUT
- **Path variables:** part number, and location. Both required
- **Additional input:** stock. Also required

- **Excepted:** A user can update a parts stock at the specified location. Example: I can update the stock for part number 11K49 to be 2 at the shop location. This would most likely be done after an inventory count.
- **Returns:** This will return the updated part when it is done. Http status code 200
- **Errors:** If no part number, location number, or stock is supplied, or the part number or location doesn't exist, it will give a 400 status, meaning a bad request was sent. If there is no part at that location to update it will give a 404 status meaning the part was not found there. All errors put out messages explaining to the user what went wrong.

Url: /inventory/new/{location_num}

- **Allowed requests:** POST
- **Path Variables:** location number
- **Additional input:** stock, part number
- **Expected:** A user can create an inventory (add a part) to a location. Example: I can add 4 of part number 93W13 to my new truck.
- **Returns:** The newly created inventory when it's done
- **Errors:** If no location, stock, or part is supplied it will give a 400 bad request status. If the part or location does not exist it will have a 404 not found status. If that part already exists at the location it will throw a runtime error letting you know that you should update instead of create. All errors put out messages explaining to the user what went wrong.

Url: /inventory/part/{part_num}

- **Allowed requests:** GET
- **Path Variables:** part number
- **Additional input:** none
- **Expected:** A user can get the inventory for a specific part for all locations. Example. I can see all the locations and the inventory for part number 11K49.
- **Returns:** Returns a list of the part and its locations that it currently has stock.
- **Errors:** If no part number is supplied it will give a 400 bad request response. If the part does not exist it will give a 404 not found response. Both errors have messages stating the same.

Url: /inventory/location/{location_num}

- **Allowed requests:** GET
- **Path Variables:** location number
- **Additional input:** none
- **Expected:** A user can get all parts and their stock in a specific location. Example: I can look at what parts I have and how many I have of each in truck number 2.
- **Returns:** A list of parts and their stock at that location
- **Errors:** If no location number is supplied it will give a 400 bad request response. If the location does not exist it will give a 404 not found response. Both errors have messages stating the same.

Url: /inventory/all

- **Allowed requests:** GET
- **Path Variables:** none
- **Additional input:** none

- **Expected:** A user can look at all parts, locations, and their stock. Example: I can look at all parts in stock and where they are at.
- **Returns:** A list of parts, locations, and their stock.
- **Errors:** If no inventories are found at all, it will give a 404 not found response. There is a message stating there are no inventories.

Parts

Url: /parts/{part_num}

- **Allowed requests:** GET, PUT, DELETE
- **Path Variables:** part number for all requests
- **Additional input:** part name and type required for PUT request
- **Expected:** A user can get a specific part. A user can update the name and type of a specific part. A user can delete a specific part.
- **Returns:** GET returns the part with the matching part number. PUT returns the updated part. DELETE returns the deleted part.
- **Errors:** If the part number is not given, or if part name and type are not given for the PUT request, a 400 bad request status is given. If the part number does not exist, a 404 not found status is given. Both have messages with details.

Url: /parts

- **Allowed requests:** GET, POST
- **Path Variables:** none
- **Additional input:** request body - part number, part name, and part type
- **Expected:** A user can look at all parts. A user can create a new part.
- **Returns:** A list of all parts, or the new part created.
- **Errors:** If there are no parts found it will throw a 404 not found status. If any part of the request body is invalid(null or not given) it will give a 400 bad request status. Both have messages with detail.

Parts Out

Url: /parts_out

- **Allowed requests:** POST
- **Path Variables:** none
- **Additional input:** request body - employee number, part number, location number, and amount
- **Expected:** A user can create a part out order with a specific employee, part, location the part is coming from, and how many they are taking. Example: I am employee number 1. I am taking 4 of part number 93W13 from the shop.
- **Returns:** A created parts out order.
- **Errors:** If any part of the request body is null or not given, it will give a 400 bad request status. If the employee, part, or location does not exist it will give a 404 not found status. If the part doesn't have any stock at the location it will also give a 404 not found status. All have messages with more detail about how to fix it when it happens.

Technicians

Url: /technicians/{employee_num}

- **Allowed requests:** GET, PUT
- **Path Variables:** employee number
- **Additional input:** first name, last name, status for PUT request
- **Expected:** A user can look at a specific technician based on the employee number. A user can update the status or name of an existing tech.
- **Returns:** Both will return a technician. Either the one searched for or the one updated.
- **Errors:** If the employee number, or the first name, last name, or status, is not given you will get a 400 bad request status. If an employee with that number doesn't exist you will get a 404 not found status. Both have detailed messages.

Url: /technicians

- **Allowed requests:** GET, POST
- **Path Variables:** none
- **Additional input:** request body - employee number, first name, last name, status
- **Expected:** A user can look at all technicians. A user can create a new technician.
- **Returns:** A list of techs, or the newly created tech.
- **Errors:** If any items in the request body are not given you will get a 400 bad request error. If no techs are available to show you will get a 404 not found error.

Locations

Url: /locations{location_num}

- **Allowed requests:** GET, PUT, DELETE
- **Path Variables:** location number
- **Additional input:** name for PUT request
- **Expected:** A user can see a specific location based on the number. A user can update a location's name based on its number. A user can delete a location based on its number.
- **Returns:** All return a location.
- **Errors:** If location number, or name, is not given it will give you a 400 bad request status. If the location does not exist it will give you 404 not found status.

Url: /locations

- **Allowed requests:** GET, POST
- **Path Variables:** none
- **Additional input:** request body - location number, name
- **Expected:** A user can look at all locations. A user can create a new location.
- **Returns:** A list of all locations. A newly created location.
- **Errors:** If location number or name is not given you will get a 400 bad request status. If there are no locations you will get a 404 not found status.