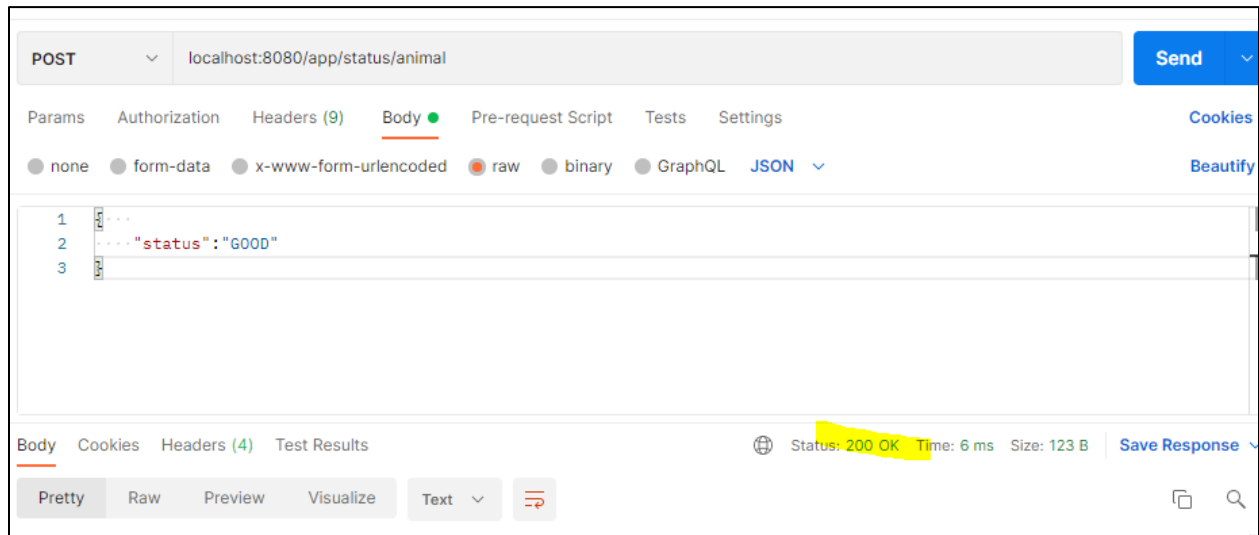


***All endpoints are working. This is a collection of Postman test for some endpoints.

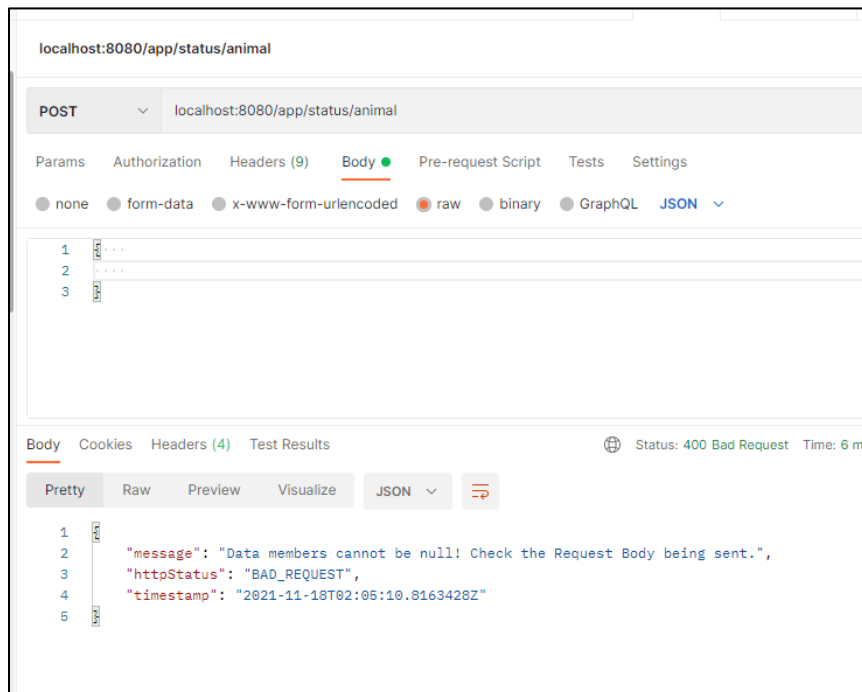
Status

POST – new Status object

Working

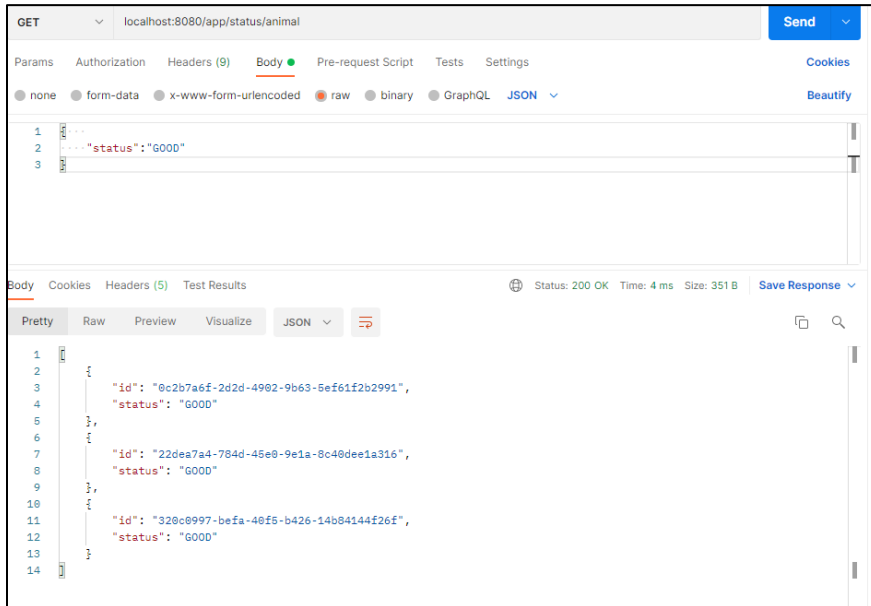


POST – invalid request body



I added another 2

GET- all



GET localhost:8080/app/status/animal

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

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

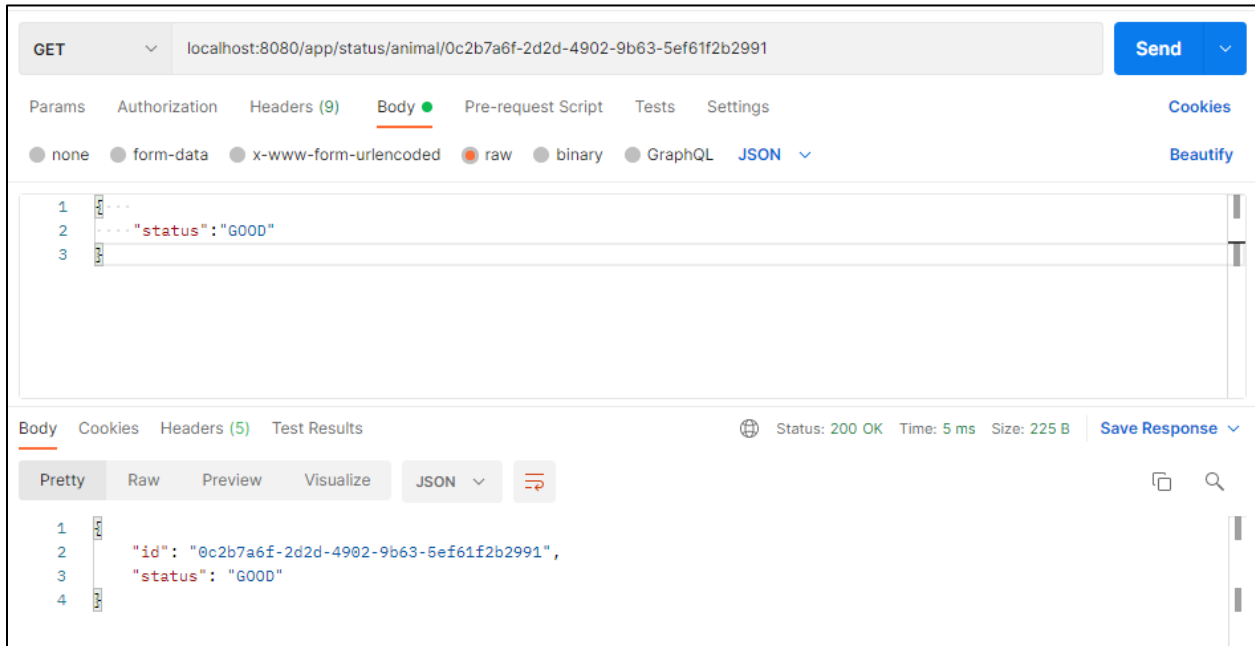
```
1 [
2   {
3     "status": "GOOD"
4   },
5   {
6     "id": "0c2b7a6f-2d2d-4902-9b63-5ef61f2b2991",
7     "status": "GOOD"
8   },
9   {
10    "id": "22dea7a4-784d-45e8-9e1a-8c48dee1a316",
11    "status": "GOOD"
12  },
13  {
14    "id": "320c0997-befa-40f5-b426-14b84144f26f",
15    "status": "GOOD"
16  }
17 ]
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 4 ms Size: 351 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "0c2b7a6f-2d2d-4902-9b63-5ef61f2b2991",
3   "status": "GOOD"
4 },
5 {
6   "id": "22dea7a4-784d-45e8-9e1a-8c48dee1a316",
7   "status": "GOOD"
8 },
9 {
10  "id": "320c0997-befa-40f5-b426-14b84144f26f",
11  "status": "GOOD"
12 }
13 ]
14 }
```

Get one by id



GET localhost:8080/app/status/animal/0c2b7a6f-2d2d-4902-9b63-5ef61f2b2991

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

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

```
1 {
2   "status": "GOOD"
3 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 5 ms Size: 225 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "0c2b7a6f-2d2d-4902-9b63-5ef61f2b2991",
3   "status": "GOOD"
4 }
```

GET one by Invalid id

GET localhost:8080/app/status/animal/0c2b7a6f-2d2d-4902-9b63-5

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

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

```
1 {
2   "status": "GOOD"
3 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 4 ms Size: 301 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T02:06:38.9043919Z"
5 }
```

PUT – UPDATE to bad status

PUT localhost:8080/app/status/animal/0c2b7a6f-2d2d-4902-9b63-5ef61f2b2991

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

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

```
1 {
2   "status": "BAD"
3 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 5 ms Size: 123 B Save Response

GET localhost:8080/app/status/animal/0c2b7a6f-2d2d-4902-9b63-5ef61f2b2991

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

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

```
1 {
2   "status": "BAD"
3 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 6 ms Size: 224 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "0c2b7a6f-2d2d-4902-9b63-5ef61f2b2991",
3   "status": "BAD"
4 }
```

PUT – invalid request body

PUT localhost:8080/app/status/animal/e104454b-a4e1-4d3d-9516-b431bd1d0ec8

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

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

Status: 400 Bad Request Time: 4 ms Size: 291 B Save Response

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Data members cannot be null: Check the Request Body being sent.",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T18:14:10.5279406Z"
5 }
```

PUT – invalid id

PUT localhost:8080/app/status/animal/e104454b-a4e1-4d3d-95

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

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

Status: 400 Bad Request Time: 5 ms Size: 301 B Save Response

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "bad"
3 }
4 }
```

```
1 {
2   "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T18:14:40.6297985Z"
5 }
```

Delete – invalid id

DELETE localhost:8080/app/status/animal/0a67b2a3-e2a5-47 Send

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

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

```
1 {
2   ...
3   ... "photoId": "15",
4   ... "dateUploaded": "2021-11-17 9:07PM",
5   ... "filepath": "src/resources/imag1000_.....png"
6 }
7
8
9
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 4 ms Size: 301 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T18:57:36.7024216Z"
5 }
```

DELETE

DELETE localhost:8080/app/status/animal/0c2b7a6f-2d2d-4902-9b63-5ef61f2b2991 Send

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

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

```
1 {
2   ...
3   ... "status": "BAD"
4 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 9 ms Size: 123 B Save Response

Pretty Raw Preview Visualize Text

```
1
```

GET localhost:8080/app/status/animal/ Send

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

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

```
1 {
2   ...
3   "status": "BAD"
}
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 4 ms Size: 289 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "22dea7a4-784d-45e0-9e1a-8c40dee1a316",
3   "status": "GOOD",
4 }
5 {
6   "id": "320c0997-befa-40f5-b426-14b84144f26f",
7   "status": "GOOD",
8 }
9 }
10
```

ONLY TWO REMAINING SO DELETE SUCCESSFUL

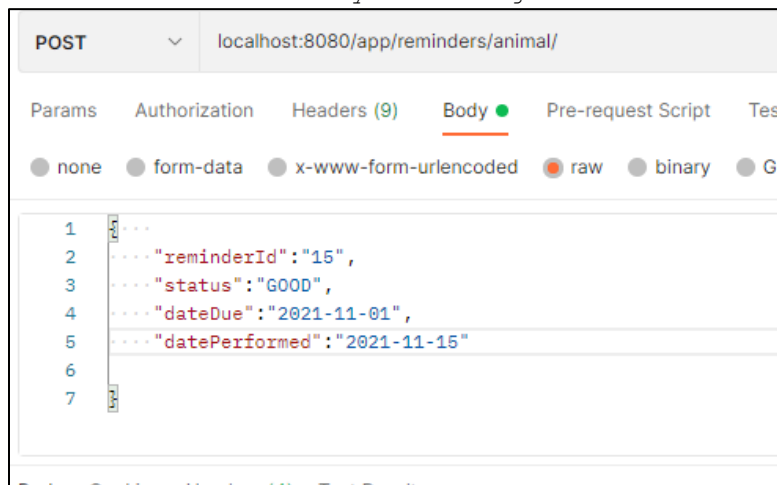
Reminders

Post – create new reminder

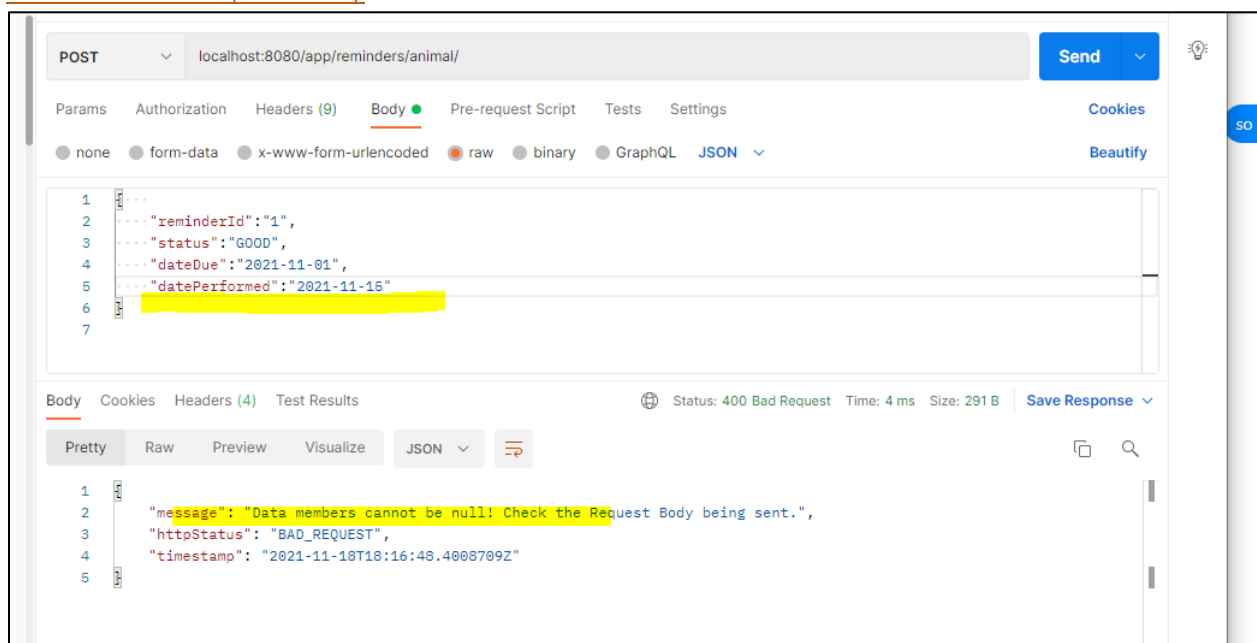
```
{  
  "reminderId": "15",  
  "status": "GOOD",  
  "dateDue": "2021-11-01",  
  "datePerformed": "2021-11-15",  
  "note": "removed splinter."  
}
```

Was successful.

When on data member entry is missing



POST – invalid request body



GET – all reminders

The screenshot shows a REST client interface with a GET request to `localhost:8080/app/reminders/animal/`. The response is a JSON array of two reminder objects, displayed in the 'Body' tab with the 'Pretty' format selected. The status is 200 OK, with a response time of 14 ms and a size of 482 B.

```
1 {
2   ...
3   ... "reminderId": "15",
4   ... "status": "GOOD",
5   ... "dateDue": "2021-11-01",
6   ... "datePerformed": "2021-11-15",
7   ... "note": "removed splinter."
8 }
```

```
1 {
2   "id": "f9e3fb48-8f48-487c-89c0-5305c79d8473",
3   "reminderId": "1",
4   "status": "GOOD",
5   "dateDue": "2021-11-01",
6   "datePerformed": "2021-11-15",
7   "note": "removed splinter."
8 },
9 {
10  "id": "32cc28ce-f5db-43d4-bb4e-a19852363228",
11  "reminderId": "15",
12  "status": "GOOD",
13  "dateDue": "2021-11-01",
14  "datePerformed": "2021-11-15",
15  "note": "removed splinter."
16 }
17 }
```

Get – by One ID

The screenshot shows a REST client interface with a GET request to `localhost:8080/app/reminders/animal/f9e3fb48-8f48-487c-89c0-5305c79d8473`. The response is a single JSON object representing a reminder, displayed in the 'Body' tab with the 'Pretty' format selected. The status is 200 OK, with a response time of 8 ms and a size of 312 B.

```
1 {
2   "id": "f9e3fb48-8f48-487c-89c0-5305c79d8473",
3   "reminderId": "1",
4   "status": "GOOD",
5   "dateDue": "2021-11-01",
6   "datePerformed": "2021-11-15",
7   "note": "removed splinter."
8 }
```


GET ONE – INVALID ID

The screenshot shows a REST client interface with a GET request to `localhost:8080/app/reminders/animal/9b5a3ad5-d612-4e8e-8f2d-68de...`. The request body is a JSON object:

```
1 {
2   "reminderId": "1",
3   "status": "GOOD",
4   "dateDue": "2021-11-01",
5   "datePerformed": "2021-11-15"
6 }
7
```

The response status is `400 Bad Request` with a time of `5 ms` and size of `300 B`. The response body is a JSON object:

```
1 {
2   "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T18:17:29.033409Z"
5 }
```

PUT – UPDATE TO NEW REMINDERID

The screenshot shows a REST client interface with a PUT request to `localhost:8080/app/reminders/animal/9b5a3ad5-d612-4e8e-8f2d-68debe066371`. The request body is a JSON object:

```
1 {
2   "reminderId": "105",
3   "status": "GOOD",
4   "dateDue": "2021-11-01",
5   "datePerformed": "2021-11-15",
6   "note": "removed splinter2222."
7 }
8
```

The response status is `200 OK` with a time of `6 ms` and size of `123 B`. The response body is empty.

GET localhost:8080/app/reminders/animal/9b5a3ad5-d612-4e8e-8f2d-68debe066371 Send

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

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

```
1 {
2   ...
3   ... "reminderId": "105",
4   ... "status": "GOOD",
5   ... "dateDue": "2021-11-01",
6   ... "datePerformed": "2021-11-15",
7   ... "note": "removed splinter2222."
8 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 4 ms Size: 327 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "9b5a3ad5-d612-4e8e-8f2d-68debe066371",
3   "reminderId": "105",
4   "status": "GOOD",
5   "dateDue": "2021-11-01",
6   "datePerformed": "2021-11-15",
7   "note": "removed splinter2222."
8 }
```

UPDATED SUCESSFULLY

PUT – INVALID REQUEST BODY

PUT localhost:8080/app/reminders/animal/9b5a3ad5-d612-4e8e-8f2d-68debe066371 Send

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

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

```
1 {
2   ...
3   ... "reminderId": "105",
4   ... "status": "GOOD",
5   ... "dateDue": "2021-11-01",
6   ... "datePerformed": "2021-11-15"
7 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 5 ms Size: 291 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Data members cannot be null! Check the Request Body being sent.",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T18:19:22.7262684Z"
5 }
```

PUT – INVALID ID

PUT localhost:8080/app/reminders/animal/9b5a3ad5-d612-4e8 [redacted] Send

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

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

```
1 {
2   "reminderId": "105",
3   "status": "GOOD",
4   "dateDue": "2021-11-01",
5   "datePerformed": "2021-11-15",
6   "note": "removed splinter2222."
7 }
8
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 6 ms Size: 301 B Save Response ▾

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T18:19:40.7639078Z"
5 }
```

DELETE – INVALID ID

DELETE localhost:8080/app/reminders/animal/9b5a3ad5-d612-4e8 [redacted] Send

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

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

```
1 {
2   "reminderId": "105",
3   "status": "GOOD",
4   "dateDue": "2021-11-01",
5   "datePerformed": "2021-11-15",
6   "note": "removed splinter2222."
7 }
8
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 12 ms Size: 301 B Save Response ▾

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T18:20:02.8806545Z"
5 }
```

DELETE – by Id

DELETE localhost:8080/app/reminders/animal/f9e3fb48-8f48-487c-89c0-5305c79d8473

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

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

```
1 {
2   ...
3   "reminderId": "15",
4   "status": "GOOD",
5   "dateDue": "2021-11-01",
6   "datePerformed": "2021-11-15"
7 }
```

Status: 200 OK Time: 4 ms Size: 123 B Save Response

GET ALL ONLY RETURNS ONE REMINDER NOW

GET localhost:8080/app/reminders/animal

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

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

```
1 {
2   ...
3   "reminderId": "15",
4   "status": "GOOD",
5   "dateDue": "2021-11-01",
6   "datePerformed": "2021-11-15"
7 }
```

Body Cookies Headers (5) Test Results Status: 200

Pretty Raw Preview Visualize JSON

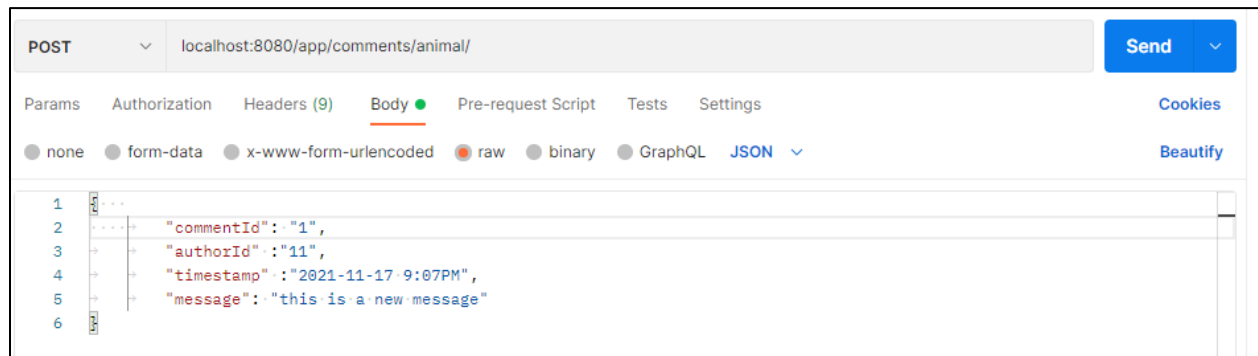
```
1 {
2   "id": "32cc28ce-f5db-43d4-bb4e-a19052363220",
3   "reminderId": "15",
4   "status": "GOOD",
5   "dateDue": "2021-11-01",
6   "datePerformed": "2021-11-15",
7   "note": "removed splinter."
8 }
```

Comments

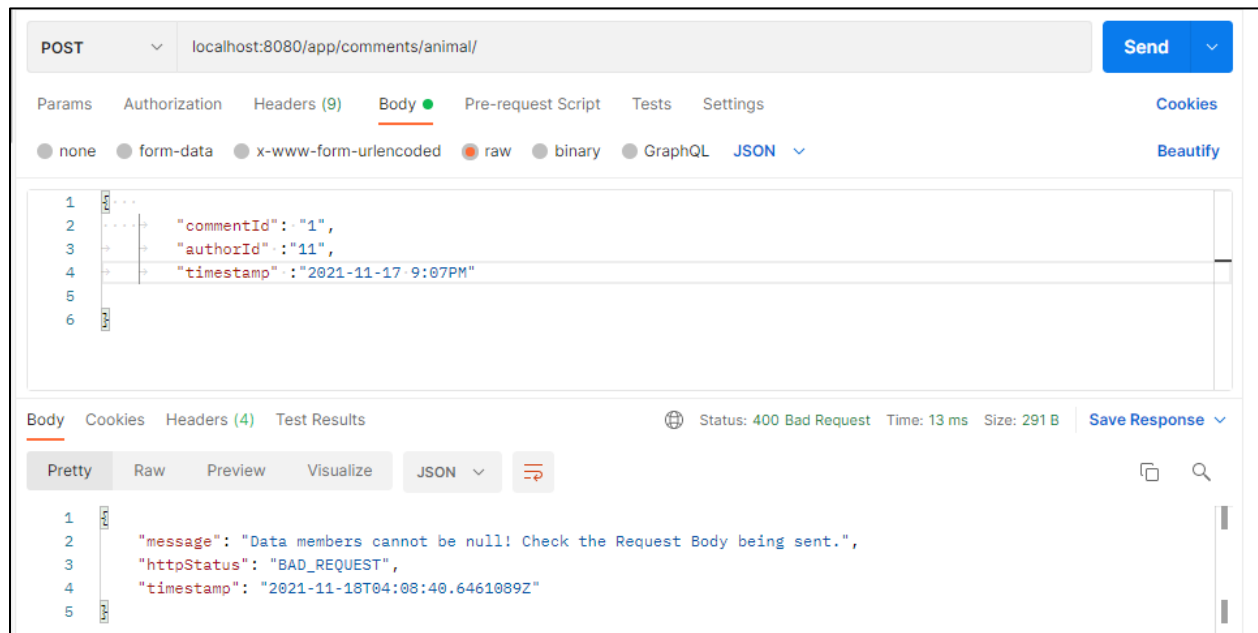
POST – create new comment

Working

```
{  
  "commentId": "1",  
  "authorId": "11",  
  "timestamp": "2021-11-17 9:07PM",  
  "message": "this is a new message"  
}
```



Missing a data member



GET – all comments

GET localhost:8080/app/comments/animal/ Send

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

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

```
1 {
2   ...
3   "commentId": "3",
4   "authorId": "15",
5   "timestamp": "2021-11-17 9:07PM",
6   "message": "this is a new message"
7 }
```

body Cookies Headers (5) Test Results Status: 200 OK Time: 9 ms Size: 453 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "0a5dd3a1-158d-4577-aac6-b74b46cccd42",
3   "commentId": "1",
4   "authorId": "11",
5   "timestamp": "2021-11-17 9:07PM",
6   "message": "this is a new message"
7 },
8 {
9   "id": "4f972765-f250-405e-8a9c-260d8d0c2e3b",
10  "commentId": "3",
11  "authorId": "15",
12  "timestamp": "2021-11-17 9:07PM",
13  "message": "this is a new message"
14 }
15 }
```

GET – for one animal

GET localhost:8080/app/comments/animal/0a5dd3a1-158d-4577-aac6-b74b46cccd42 Send

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

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

```
1 {
2   ...
3   "commentId": "3",
4   "authorId": "15",
5   "timestamp": "2021-11-17 9:07PM",
6   "message": "this is a new message"
7 }
```

body Cookies Headers (5) Test Results Status: 200 OK Time: 13 ms Size: 307 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "0a5dd3a1-158d-4577-aac6-b74b46cccd42",
3   "commentId": "1",
4   "authorId": "11",
5   "timestamp": "2021-11-17 9:07PM",
6   "message": "this is a new message"
7 }
```

Get – invalid id

The screenshot shows a REST client interface with a GET request to `localhost:8080/app/comments/animal/51e02891-ff87-40d4-8e [redacted]`. The request body is a JSON object:

```
1 {
2   "commentId": "12",
3   "authorId": "105",
4   "timestamp": "2021-11-17 9:07PM",
5   "message": "this is a new message222"
6 }
```

The response status is `400 Bad Request` with a time of `4 ms` and size of `300 B`. The response body is:

```
1 {
2   "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T18:22:00.950934Z"
5 }
```

PUT – update a comment

The screenshot shows a REST client interface with a PUT request to `localhost:8080/app/comments/animal/0a5dd3a1-158d-4577-aac6-b74b46cccd42`. The request body is a JSON object:

```
1 {
2   "commentId": "3",
3   "authorId": "15",
4   "timestamp": "2021-11-17 9:07PM",
5   "message": "2nd new message"
6 }
```

The screenshot shows the response to the PUT request, which is a `200 OK` status with a time of `5 ms` and size of `301 B`. The response body is a JSON object:

```
1 {
2   "id": "0a5dd3a1-158d-4577-aac6-b74b46cccd42",
3   "commentId": "3",
4   "authorId": "15",
5   "timestamp": "2021-11-17 9:07PM",
6   "message": "2nd new message"
7 }
```

PUT WITH INVALID ID

The screenshot shows a REST client interface with a PUT request to `localhost:8080/app/comments/animal/0a5dd3a1-158d-4577-aac`. The request body is a JSON object with the following fields:

```
1 {
2   "commentId": "3",
3   "authorId": "15",
4   "timestamp": "2021-11-17 9:07PM",
5   "message": "2nd new message"
6 }
```

The response status is `400 Bad Request`. The response body is a JSON object:

```
1 {
2   "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T04:11:00.09245012"
5 }
```

PUT WITH MISSING DATA MEMBERS

The screenshot shows a REST client interface with a PUT request to `localhost:8080/app/comments/animal/dbf362c1-4fa6-412a-9b5f-dd341372e386`. The request body is a JSON object with the following fields:

```
1 {
2   "commentId": "1",
3   "authorId": "11",
4   "timestamp": "2021-11-17 9:07PM"
5 }
```

The response status is `400 Bad Request`. The response body is a JSON object:

```
1 {
2   "message": "Data members cannot be null! Check the Request Body being sent.",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T04:13:42.48606762"
5 }
```


DELETE

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

- Method:** DELETE
- URL:** localhost:8080/app/comments/animal/dbf362c1-4fa6-412a-9b5f-dd341372e386
- Body:** A JSON object with the following fields:

```
{  "commentId": "1",  "authorId": "11",  "timestamp": "2021-11-17 9:07PM",  "message": "2ND MESSAGE"}
```
- Status:** 200 OK
- Time:** 4 ms
- Size:** 123 B

DELETE WITH INCORRECT ID

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

- Method:** DELETE
- URL:** localhost:8080/app/comments/animal/dbf362c1-4fa6-41
- Body:** A JSON object with the following fields:

```
{  "commentId": "1",  "authorId": "11",  "timestamp": "2021-11-17 9:07PM",  "message": "2ND MESSAGE"}
```
- Status:** 400 Bad Request
- Time:** 4 ms
- Size:** 301 B
- Response Body:** A JSON object with the following fields:

```
{  "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",  "httpStatus": "BAD_REQUEST",  "timestamp": "2021-11-18T04:14:48.1368459Z"}
```

Photos

POST – create a new photo

```
{  
  
  "photoId" : "12",  
  "dateUploaded" : "2021-11-17 9:07PM",  
  "filepath" : "src/resources/imag2.png"  
}
```

The screenshot shows a REST client interface with a POST request to `localhost:8080/app/photos/animal/`. The request body is a JSON object: `{ "photoId": "12", "dateUploaded": "2021-11-17 9:07PM", "filepath": "src/resources/imag2.png" }`. The response status is `200 OK` with a time of 4 ms and a size of 123 B. The response body is empty.

Post - MISSING DATA MEMBER

The screenshot shows a REST client interface with a POST request to `localhost:8080/app/photos/animal/`. The request body is a JSON object: `{ "photoId": "12", "dateUploaded": "2021-11-17 9:07PM" }`. The response status is `400 Bad Request` with a time of 4 ms and a size of 291 B. The response body is a JSON object: `{ "message": "Data members cannot be null! Check the Request Body being sent.", "httpStatus": "BAD_REQUEST", "timestamp": "2021-11-18T04:37:32.2365143Z" }`.

GET – ALL PHOTOS

The screenshot shows a REST client interface with a GET request to `localhost:8080/app/photos/animal/`. The request body is a JSON object with the following fields:

```
1 {
2   ...
3   "photoId": "12",
4   "dateUploaded": "2021-11-17 9:07PM",
5   "filepath": "src/resources/imag2.png"
6 }
```

The response is displayed in the "Body" tab, showing a JSON array of two photo objects:

```
1 [
2   {
3     "id": "7bf34847-1537-4151-94bb-5a2d8be9bd5b",
4     "photoId": "11",
5     "dateUploaded": "2021-11-17 9:07PM",
6     "filepath": "src/resources/imag1.png"
7   },
8   {
9     "id": "00f46c35-cfb2-4775-b1f4-0f211397a3f9",
10    "photoId": "12",
11    "dateUploaded": "2021-11-17 9:07PM",
12    "filepath": "src/resources/imag2.png"
13  }
14 ]
```

Response status: 200 OK, Time: 8 ms, Size: 431 B.

GET – ONE ANIMAL'S PHOTOS

The screenshot shows a REST client interface with a GET request to `localhost:8080/app/photos/animal/7bf34847-1537-4151-94bb-5a2d8be9bd5b`. The request body is a JSON object with the following fields:

```
1 {
2   ...
3   "photoId": "12",
4   "dateUploaded": "2021-11-17 9:07PM"
5 }
```

The response is displayed in the "Body" tab, showing a JSON object with the following fields:

```
1 {
2   "id": "7bf34847-1537-4151-94bb-5a2d8be9bd5b",
3   "photoId": "11",
4   "dateUploaded": "2021-11-17 9:07PM",
5   "filepath": "src/resources/imag1.png"
6 }
```

Response status: 200 OK, Time: 9 ms, Size: 296 B.

GET - INVALID ID

GET localhost:8080/app/photos/animal/7bf34847-

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

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

```
1 {
2   ...
3   "photoId": "12",
4   "dateUploaded": "2021-11-17 9:07PM"
5 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 4 ms Size: 301 B

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T04:38:39.7341637Z"
5 }
```

PUT – update a Photo

PUT localhost:8080/app/photos/animal/7bf34847-1537-4151-94bb-5a2d8be9bd5b **Send**

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

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

```
1 {
2   ...
3   "photoId": "12",
4   "dateUploaded": "2021-11-17 9:07PM",
5   "filepath": "***NEWPATH**/imag1.png"
6 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 6 ms Size: 123 B **Save Response**

Pretty Raw Preview Visualize **Text**

Put with invalid id

The screenshot shows a REST client interface with a PUT request to `localhost:8080/app/photos/animal/7bf34847-1537-41`. The request body is a JSON object with `photoId`, `dateUploaded`, and `filepath` fields. The response status is 400 Bad Request, and the response body contains an error message indicating the ID is invalid.

```
PUT localhost:8080/app/photos/animal/7bf34847-1537-41 Send
```

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

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

```
1 {
2   ...
3   "photoId": "12",
4   "dateUploaded": "2021-11-17 9:07PM",
5   "filepath": "***NEWPATH__/imag1.png"
6 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 7 ms Size: 301 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T04:40:14.733221Z"
5 }
```

PUT – missing data

The screenshot shows a REST client interface with a PUT request to `localhost:8080/app/photos/animal/7bf34847-1537-4151-94bb-5a2d8be9bd5b`. The request body is a JSON object with `photoId` and `dateUploaded` fields, but it is missing the `filepath` field. The response status is 400 Bad Request, and the response body contains an error message indicating that data members cannot be null.

```
PUT localhost:8080/app/photos/animal/7bf34847-1537-4151-94bb-5a2d8be9bd5b Send
```

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

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

```
1 {
2   ...
3   "photoId": "12",
4   "dateUploaded": "2021-11-17 9:07PM"
5 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 10 ms Size: 290 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Data members cannot be null! Check the Request Body being sent.",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T04:39:53.184239Z"
5 }
```

DELETE – invalid id

The screenshot shows a REST client interface for a DELETE request. The URL is `localhost:8080/app/photos/animal/7bf34847-1537-4151-94bb-5a2d8be9bd5b`. The request body is a JSON object with the following fields:

```
1 { ...
2   ...
3   "photoId": "12",
4   "dateUploaded": "2021-11-17 9:07PM",
5   "filepath": "***NEWPATH__/imag1.png"
6 }
```

The response status is **400 Bad Request**. The response body is a JSON object:

```
1 {
2   "message": "Invalid Id. Please check that the Id is a valid Id for an existing Animal",
3   "httpStatus": "BAD_REQUEST",
4   "timestamp": "2021-11-18T04:41:44.9733614Z"
5 }
```

DELETE – valid id

The screenshot shows a REST client interface for a DELETE request. The URL is `localhost:8080/app/photos/animal/7bf34847-1537-4151-94bb-5a2d8be9bd5b`. The request body is a JSON object with the following fields:

```
1 { ...
2   ...
3   "photoId": "12",
4   "dateUploaded": "2021-11-17 9:07PM",
5   "filepath": "***NEWPATH__/imag1.png"
6 }
```

The response status is **200 OK**. The response body is empty.

GET localhost:8080/app/photos/animal/ Send

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

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

```
1 {
2   ...
3   "photoId": "12",
4   "dateUploaded": "2021-11-17 9:07PM",
5   "filepath": "***NEWPATH__/imag1.png"
6 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 4 ms Size: 298 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "00f46c35-cfb2-4775-b1f4-0f211397a3f9",
3   "photoId": "12",
4   "dateUploaded": "2021-11-17 9:07PM",
5   "filepath": "src/resources/imag2.png"
6 }
```

Only one remaining

BOTTOM OF DOCUMENT