

Postman screenshot showing a successful API call to `http://localhost:4000/createShow`.

The request body is:

```
1 {
2     "show_id": "00000000-0000-0000-0000-000000000001",
3     "title": "Test Show",
4     "critics_score": 85,
5     "audience_score": 90,
6     "image_url": "https://example.com/image.jpg",
7     "media_url": "/tv/test_show",
8     "release_date": "Jan 1, 2025",
9     "is_video": true
10 }
```

The response status is **201 Created**, with a response body:

```
1 {
2     "message": "TV show created",
3     "show": {
4         "show_id": "00000000-0000-0000-0000-000000000001",
5         "title": "Test Show",
6         "critics_score": 85,
7         "audience_score": 90,
8         "image_url": "https://example.com/image.jpg",
9         "media_url": "/tv/test_show",
10        "release_date": "Jan 1, 2025",
11        "is_video": true
12    }
13 }
```

Postman screenshot showing an unsuccessful API call to `http://localhost:4000/createShow`.

The request body is identical to the one above:

```
1 {
2     "show_id": "00000000-0000-0000-0000-000000000001",
3     "title": "Test Show",
4     "critics_score": 85,
5     "audience_score": 90,
6     "image_url": "https://example.com/image.jpg",
7     "media_url": "/tv/test_show",
8     "release_date": "Jan 1, 2025",
9     "is_video": true
10 }
```

The response status is **400 Bad Request**, with a response body:

```
1 {
2     "error": "Validation failed",
3     "details": [
4         {
5             "message": "\"show_id\" is required",
6             "path": [
7                 "show_id"
8             ],
9             "type": "any.required",
10            "context": {
11                "label": "show_id",
12                "key": "show_id"
13            }
14        }
15    ]
16 }
```

Postman

File Edit View Window Help

Search Postman

PUT http://localhost:3000/updateShow/00000000-0000-0000-0000-000000000001

Save Share

Send

Params Authorization Headers (8) Body Scripts Settings

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

```
1 {
2   "title": "New Test Show",
3   "critics_score": 86
4 }
```

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create Collection

Response History

Click Send to get a response

Fri May 30 15:10:26

Postman

File Edit View Window Help

Search Postman

PUT http://localhost:4000/updateShow/00000000-0000-0000-0000-000000000001

Save Share

Send

Params Authorization Headers (8) Body Cookies Scripts Settings

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

```
1 {
2   "title": "New Test Show",
3   "critics_score": 86
4 }
```

Body Cookies Headers (8) Test Results

200 OK 1.16 s 549 B

```
1 {
2   "message": "Show updated successfully",
3   "updated": {
4     "show_id": "00000000-0000-0000-0000-000000000001",
5     "title": "New Test Show",
6     "critics score": 86,
7     "audience_score": 90,
8     "image_url": "https://example.com/image.jpg",
9     "media_url": "/tv/test_show",
10    "release_date": "Jan 1, 2025",
11    "is_video": true
12  }
13 }
```

Postbot Runner Start Proxy Cookies Vault Trash

Fri May 30 15:10:43

PUT http://localhost:4000/updateShow/00000000-0000-0000-0000-000000000001

Params Authorization Headers (8) Body Scripts Settings

Body (raw) JSON

```
1 {}
```

400 Bad Request 189 ms 446 B

```
3 "details": [
4   {
5     "message": "\"value\" must have at least 1 key",
6     "path": [],
7     "type": "object.min",
8     "context": {
9       "limit": 1,
10      "value": {},
11      "label": "value"
12    }
13  ]
14 }
```

PUT http://localhost:4000/updateShow/00000000-0000-0000-0000-000000000000

Params Authorization Headers (8) Body Scripts Settings

Body (raw) JSON

```
1 {
2   "critics_score": 100
3 }
```

404 Not Found 1.36 s 303 B

```
1 {
2   "error": "TV show not found"
3 }
```

Postman

File Edit View Window Help

PUT http://localhost:4000/updateShow/00000000-0000-0000-0000-000000000001

PUT http://localhost:4000/updateShow/00000000-0000-0000-0000-000000000001

Params Authorization Headers (8) Body Scripts Settings

Body raw JSON

```
1 {  
2   "critics_score": 120  
3 }
```

400 Bad Request 139 ms 511 B

Body Cookies Headers (8) Test Results

```
{ } JSON
```

1 {  
2 "error": "Validation failed",  
3 "details": [  
4 {  
5 "message": "\"critics\_score\" must be less than or equal to 100",  
6 "path": [  
7 "critics\_score"  
8 ],  
9 "type": "number.max",  
10 "context": {  
11 "limit": 100,  
12 "value": 120,  
13 "label": "critics\_score",  
14 }  
15 }  
16 ]  
17}

Postman

File Edit View Window Help

GET http://localhost:4000/getShow/00000000-0000-0000-0000-000000000001

GET http://localhost:4000/getShow/00000000-0000-0000-0000-000000000001

Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

200 OK 1.64 s 507 B

Body Cookies Headers (8) Test Results

```
{ } JSON
```

1 {  
2 "show": {  
3 "show\_id": "00000000-0000-0000-0000-000000000001",  
4 "title": "New Test Show",  
5 "critics\_score": 85,  
6 "audience\_score": 90,  
7 "image\_url": "https://example.com/image.jpg",  
8 "media\_url": "/tv/test\_show",  
9 "release\_date": "Jan 1, 2025",  
10 "is\_video": true  
11 }  
12 }

The screenshot shows the Postman application interface on a Mac OS X desktop. The main window displays a collection named "Your collection". A single GET request is listed with the URL `http://localhost:4000/getShow/00000000-0000-0000-0000-000000000000`. The "Params" tab is selected. The response status is 404 Not Found, with a response body containing the JSON object: 

```
1 {  
2   "error": "Show not found"  
3 }
```

The screenshot shows the Postman application interface on a Mac OS X desktop. The main window displays a collection named "My first collection", which contains two nested folders: "First folder inside collection" and "Second folder inside collection". A single GET request is listed with the URL `http://localhost:4000/getAllShows`. The "Headers" tab is selected. The response status is 200 OK, with a response body containing a JSON array of show objects: 

```
1 [  
2   {  
3     "show_id": "f53ab008-3afc-3454-adf5-0f9101eca583",  
4     "title": "Adults",  
5     "critics_score": 73,  
6     "audience_score": null,  
7     "image_url": "https://resizing.flixster.com/zd6SAPYoCUIK5EM1285gdyE14qew=/206x395/v2/https://resizing.flixster.com/-XZAfHZM39lwaGJIFWKA8fS0ak=/v3/t/assets/p29957001_b_v13_ab.jpg",  
8     "media_url": "/tv/adults",  
9     "release_date": "Latest Episode: May 28",  
10    "is_video": true  
11  },  
12  {
```

The screenshot shows the Postman application interface on a Mac OS X desktop. A collection named "Your collection" is selected in the left sidebar. In the main workspace, a DELETE request is being made to `http://localhost:4000/deleteShow/00000000-0000-0000-0000-000000000002`. The Headers tab shows "Content-Type: application/json". The Response section displays a 404 Not Found error with the JSON body: 

```
1 {  
2   "error": "Show not found"  
3 }
```

The screenshot shows the Postman application interface on a Mac OS X desktop, identical to the one above but with a successful response. The DELETE request to `http://localhost:4000/deleteShow/00000000-0000-0000-0000-000000000001` now returns a 200 OK status with the JSON body: 

```
1 {  
2   "message": "Show deleted successfully",  
3   "show": {  
4     "show_id": "00000000-0000-0000-0000-000000000001",  
5     "title": "New Test Show",  
6     "critics_score": 85,  
7     "audience_score": 90,  
8     "image_url": "https://example.com/image.jpg",  
9     "media_url": "/tv/test_show",  
10    "release_date": "Jan 1, 2025",  
11    "is_video": true  
12  }  
13 }
```