

Sachin Sirohi

API TOOL POSTMAN E-BOOK

HTTP Status Codes

httpstatuses.com is an easy to reference database of HTTP Status Codes with their definitions and helpful code references all in one place. Visit an individual status code via httpstatuses.com/code or browse the list below.

@ Share on Twitter  Add to Pinboard 

- **1xx Informational**
 - **100** Continue
 - **101** Switching Protocols
 - **102** Processing
- **2xx Success**
 - **200** OK
 - **201** Created
 - **202** Accepted
 - **203** Non-authoritative Information
 - **204** No Content
 - **205** Reset Content
 - **206** Partial Content
 - **207** Multi-Status
 - **208** Already Reported
 - **226** IM Used
- **3xx Redirection**
 - **300** Multiple Choices
 - **301** Moved Permanently
 - **302** Found
 - **303** See Other
 - **304** Not Modified
 - **305** Use Proxy
 - **307** Temporary Redirect
 - **308** Permanent Redirect
- **4xx Client Error**
 - **400** Bad Request
 - **401** Unauthorized
 - **402** Payment Required
 - **403** Forbidden
 - **404** Not Found
 - **405** Method Not Allowed
 - **406** Not Acceptable
 - **407** Proxy Authentication Required
 - **408** Request Timeout
 - **409** Conflict
 - **410** Gone

- **411** Length Required
 - **412** Precondition Failed
 - **413** Payload Too Large
 - **414** Request-URI Too Long
 - **415** Unsupported Media Type
 - **416** Requested Range Not Satisfiable
 - **417** Expectation Failed
 - **418** I'm a teapot
 - **421** Misdirected Request
 - **422** Unprocessable Entity
 - **423** Locked
 - **424** Failed Dependency
 - **426** Upgrade Required
 - **428** Precondition Required
 - **429** Too Many Requests
 - **431** Request Header Fields Too Large
 - **444** Connection Closed Without Response
 - **451** Unavailable For Legal Reasons
 - **499** Client Closed Request
 - **5xx Server Error**
 - **500** Internal Server Error
 - **501** Not Implemented
 - **502** Bad Gateway
 - **503** Service Unavailable
 - **504** Gateway Timeout
 - **505** HTTP Version Not Supported
 - **506** Variant Also Negotiates
 - **507** Insufficient Storage
 - **508** Loop Detected
 - **510** Not Extended
 - **511** Network Authentication Required
 - **599** Network Connect Timeout Error
-

HTTP Methods

HTTP Method	Request Has Body	Response Has Body	Safe	Idempotent
GET	Optional	Yes	Yes	Yes
HEAD	No	No	Yes	Yes
POST	Yes	Yes	No	No
PUT	Yes	Yes	No	Yes
DELETE	No	Yes	No	Yes
CONNECT	Yes	Yes	No	No
OPTIONS	Optional	Yes	Yes	Yes
TRACE	No	Yes	Yes	Yes
PATCH	Yes	Yes	No	No

CRUD - What is CRUD?

The acronym **CRUD** (**C**reate, **R**ead, **U**pdate, **D**elete) may be familiar to experienced programmers and UX designers, but most people who come across it may not fully understand what it means and how it relates to everyday websites and apps.

What is CRUD?

CRUD – again is:

- Create
- Read
- Update
- Delete

is shorthand for the basic functions that an application must conventionally be able to fulfill in order to be considered “complete.”

For example, let's consider a basic phone book app. In order to feel like a “finished” application, the user should be able to do at least four things: create new entries, access the information available for each entry, modify and save the existing information, and remove old and unnecessary entries from the book. In this example, the phone book is essentially functioning as a database, and it's no coincidence that the term **CRUD** is generally used to describe what an application with a database should conventionally do.

Because these four functions are so universal to programming, each of the elements of **CRUD** is represented in virtually every single programming language out there that deals with databases. A query language like SQL may call their Create function an **INSERT**, and a web application using **HTTP** may make a **POST** request, but both of these things are simply telling a database somewhere to produce a new entry. Same goes with reading, updating, and

deleting items in a database – these are universal necessities, and although different languages may handle them differently, you can always be assured that any application dealing with a table of information will need to have these four key capabilities.

REST vs CRUD

If you've ever dabbled in developing a website or an app, it's possible that you've come across another acronym that sounds awfully similar to what we're describing with CRUD. REST (Representational State Transfer), or RESTful programming, is another term that deals with how a developer might deal with the manipulation of information coming into and out of a program. However, these two concepts deal with very different levels of programming and actually don't compete against each other so much as complement one another.

CRUD, at its roots, is a basic and primitive way of manipulating information. It's what an application pretty much HAS to be able to do, rather than a way of going about doing it. REST, on the other hand, is a high-level way of creating, changing and removing information for the user. Specifically, REST refers to manipulating data through HTTP commands, and an ideology for how to streamline the relationship between the data that a user is manipulating on screen and the information that is saved on the server. A programmer can create a REST API that handles basic CRUD functions, but building a CRUD application in no way insinuates the use of RESTful programming.

Hopefully, this has been an insightful look at what CRUD means to programmers, how applications might handle those functions, and what exactly the difference is between CRUD and REST when it comes to creating an application.

Go for Dummy API Link

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Reqres - A hosted REST-API reader" and displays a REST API interface. The URL in the address bar is <https://reqres.in/>. The interface is divided into two main sections: "Request" and "Response".

Request: The URL `/api/users?page=2` is entered in the Request section.

Response: The status code **200** is shown above the JSON response. The response body is a JSON object representing a paginated list of users:

```
{
  "page": 2,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 7,
      "email": "michael.lawson@reqres.in",
      "first_name": "Michael",
      "last_name": "Lawson",
      "avatar": "https://reqres.in/img/users/7.avatar.jpg"
    },
    {
      "id": 8,
      "email": "lindsay.ferguson@reqres.in",
      "first_name": "Lindsay",
      "last_name": "Ferguson",
      "avatar": "https://reqres.in/img/users/8.avatar.jpg"
    },
    {
      "id": 9,
      "email": "tobias.funke@reqres.in",
      "first_name": "Tobias",
      "last_name": "Funke",
      "avatar": "https://reqres.in/img/users/9.avatar.jpg"
    },
    {
      "id": 10,
      "email": "byron.fields@reqres.in",
      "first_name": "Byron",
      "last_name": "Fields",
      "avatar": "https://reqres.in/img/users/10.avatar.jpg"
    }
  ]
}
```

The browser's taskbar at the bottom shows various pinned and open application icons, and the system tray indicates it's 12:56 on 15-02-2022, with a temperature of 21°C and weather as sunny.

URL Link for API after open in web browser with inspect

GET REQUEST

<https://reqres.in/api/users/?>

Open postman->goto collections->in collection click on add request-> get-> send

<https://reqres.in/api/users/2>

The screenshot shows the Postman application interface. The left sidebar has sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main area shows a 'My Workspace' section with a 'collection1' folder containing a 'New Request' item. The 'collection1 / New Request' panel is active, showing a GET request to 'https://reqres.in/api/users/2'. The 'Params' tab is selected, showing a single parameter 'Key' with value 'Value'. Below the request details, the 'Body' tab is selected, displaying the JSON response:

```
1 "data": {  
2     "id": 2,  
3     "email": "janet.weaver@reqres.in",  
4     "first_name": "Janet",  
5     "last_name": "Weaver",  
6     "avatar": "https://reqres.in/img/faces/2-image.jpg"  
7 },  
8 "support": {  
9     "url": "https://reqres.in/#support-heading",  
10    "text": "To keep ReqRes free, contributions towards server costs are appreciated!"  
11 }  
12 }  
13 }
```

The status bar at the bottom indicates: Status: 200 OK, Time: 315 ms, Size: 1.15 KB, Save Response.

<https://reqres.in/api/users/3>

The screenshot shows the Postman interface with a successful API call. The request method is GET, targeting the URL <https://reqres.in/api/users/3>. The response status is 200 OK, with a response time of 131 ms and a size of 1.14 KB. The response body is displayed in Pretty JSON format:

```
1 {  
2   "data": {  
3     "id": 3,  
4     "email": "emma.wong@reqres.in",  
5     "first_name": "Emma",  
6     "last_name": "Wong",  
7     "avatar": "https://reqres.in/img/faces/3-image.jpg"  
8   },  
9   "support": {  
10     "url": "https://reqres.in/#support-heading",  
11     "text": "To keep ReqRes free, contributions towards server costs are appreciated!"  
12   }  
13 }
```

Authorization

The screenshot shows the Postman interface with the Authorization tab selected for a GET request to <https://reqres.in/api/users/3>. The response status is 200 OK, with a response time of 131 ms and a size of 1.14 KB. The response body is the same as in the previous screenshot. The Authorization dropdown menu is open, showing various authentication types:

- Basic Auth
- Digest Auth
- OAuth 1.0
- OAuth 2.0
- Hawk Authentication
- AWS Signature
- NTLM Authentication
- Akamai EdgeGrid

User name- admin

Password- admin (****)

Goto header->Authorization

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

Invite Upgrade

My Workspace New Import collection1 New Collection GET New Request + ... No Environment

Collections APIs Environments Mock Servers Monitors Flows History

collection1 / New Request

GET https://reqres.in/api/users/3

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Headers Hide auto-generated headers

KEY VALUE DESCRIPTION Bulk Edit Presets

Authorization Basic YWRtaW46YWRtaW4=

Cache-Control no-cache

Postman-Token <calculated when request is sent>

Body Cookies Headers (19) Test Results

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 131 ms Size: 1.14 KB Save Response

Find and Replace Console

Type here to search

Capture requests and cookies Bootcamp Runner Trash

Got google= <https://www.base64decode.org/>

Convert authorization value - YWRtaW46YWRtaW4=

Check & update your Android ve SarkariResult.com :Sarkari Result API Testing with POSTMAN In Hindi Base64 Decode and Encode - On

base64decode.org

Apps Compress PDF to 5... CCS SKILL DEVELOP... SKILL DEVELOPM... YouTube SPSS data... Linked List - Delete... Download YouTube... Meerut Job viva whole sale co... Swati Caste Certific... Reading list

Decode from Base64 format

Simply enter your data then push the decode button.

YWRtaW46YWRtaW4=

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

Source character set: UTF-8

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE Decodes your data into the area below.

admin admin

Bonus tip: Bookmark us!

MIL-Spec Hardware and Swit MIL-Spec Capacitors and Fuses Tantalum Capacitors federalconnectors.com

Other tools

URL Decode URL Encode JSON Minify JSON Beautify JS Minify JS Beautify CSS Minify CSS Beautify

Sign Up Today
7 Keys of How You Can Increase Revenue in Business
Grow Your Business
Open >

Copy key copy as Curl(cmd)

W3Schools Online Web Tutorials x https://reqres.in/api/users/2 +

reqres.in/api/users/2

Apps Compress PDF to 5... CCS SKILL DEVELOP... SKILL DEVELOPM... YouTube SPSS data... Linked List - Delete... Download YouTube... Meerut Job viva whole sale co... Swati Caste Certific... » | Reading list

{"data": {"id": 2, "email": "janet.weaver@reqres.in", "first_name": "Janet", "last_name": "Weaver", "avatar": "https://reqres.in/img/faces/2-image.jpg"}, "support": {"url": "https://reqres.in/#support-heading", "text": "To keep ReqRes free, contributions towards server costs are appreciated!"}}

Open in Sources panel
Open in new tab

Clear browser cache
Clear browser cookies

Copy
Copy link address
Copy response
Copy as PowerShell
Copy as fetch
Copy as Node.js fetch
Copy as cURL (cmd)
Copy as cURL (bash)
Copy all as PowerShell
Copy all as fetch
Copy all as Node.js fetch
Copy all as cURL (cmd)
Copy all as cURL (bash)
Copy all as HAR

Memory Application Security Lighthouse Recorder

JS CSS Img Media Font Doc WS Wasm Manifest Other Has blocked cookies Blocked Requests 3rd-party requests

35 ms 40 ms 45 ms 50 ms 55 ms 60 ms 65 ms 70 ms 75 ms 80 ms 85 ms 90 ms 95 ms 100 ms 105 ms 110 ms

Status	Type	Initiator	Size	Time	Waterfall
200	document	Other	(disk cache)	28 ms	

Loaded: 85 ms Load: 82 ms

1 requests | 0 B transferred | 280 B resou

Type here to search

19°C Sunny ENG 11:25 15-02-2022

W3Schools Online Web Tutorials x https://reqres.in/api/users/2 +

reqres.in/api/users/2

Apps Compress PDF to 5... CCS SKILL DEVELOP... SKILL DEVELOPM... YouTube SPSS data... Linked List - Delete... Download YouTube... Meerut Job viva whole sale co... Swati Caste Certific... » | Reading list

{"data": {"id": 2, "email": "janet.weaver@reqres.in", "first_name": "Janet", "last_name": "Weaver", "avatar": "https://reqres.in/img/faces/2-image.jpg"}, "support": {"url": "https://reqres.in/#support-heading", "text": "To keep ReqRes free, contributions towards server costs are appreciated!"}}

Open in Sources panel
Open in new tab

Clear browser cache
Clear browser cookies

Copy
Copy link address
Copy response
Copy as PowerShell
Copy as fetch
Copy as Node.js fetch
Copy as cURL (cmd)
Copy as cURL (bash)
Copy all as PowerShell
Copy all as fetch
Copy all as Node.js fetch
Copy all as cURL (cmd)
Copy all as cURL (bash)
Copy all as HAR

Memory Application Security Lighthouse Recorder

JS CSS Img Media Font Doc WS Wasm Manifest Other Has blocked cookies Blocked Requests 3rd-party requests

35 ms 40 ms 45 ms 50 ms 55 ms 60 ms 65 ms 70 ms 75 ms 80 ms 85 ms 90 ms 95 ms 100 ms 105 ms 110 ms

Status	Type	Initiator	Size	Time	Waterfall
200	document	Other	(disk cache)	28 ms	

Loaded: 85 ms Load: 82 ms

1 requests | 0 B transferred | 280 B resou

Type here to search

19°C Sunny ENG 11:33 15-02-2022

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

Invite

Upgrade

My Workspace

Collections APIs Environments Mock Servers Monitors Flows History

New Import collection1 New Collection

GET New Request ● GET Untitled Request

GET https://reqres.in...

Save Send Cookies

Import

File Folder Link Raw text Code repository

Paste raw text

```
curl "https://reqres.in/api/users/2" --compressed
```

Continue

Find and Replace Console

Capture requests and cookies Bootcamp Runner Trash

19°C Sunny 11:26 15-02-2022

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

Invite

Upgrade

My Workspace

Collections APIs Environments Mock Servers Monitors Flows History

New Import collection1 New Collection

GET New Request ● GET Untitled Request

GET https://reqres.in...

Save Send Cookies

Import

Select files to import · 1/1 selected

NAME	FORMAT	IMPORT AS
https://reqres.in/api/users/2	Curl	Request

Cancel Import

Find and Replace Console

Capture requests and cookies Bootcamp Runner Trash

19°C Sunny 11:27 15-02-2022

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import collection1 New Collect... GET New Requ... ● GET Untitled Re... GET https://reqr... ● GET https://reqr... ● + ⚡ Save ⚡ No Environment

Collections APIs Environments Mock Servers Monitors Flows History

https://reqres.in/api/users/2

GET https://reqres.in/api/users/2

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Response

Click Send to get a response

Find and Replace Console

Capture requests and cookies Bootcamp Runner Trash

Type here to search

19°C Sunny 11:27 15-02-2022

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

New Import collection1 New Collect... GET New Requ... ● GET Untitled Re... GET https://reqr... ● GET https://reqr... ● + ⚡ Save ⚡ No Environment

Collections APIs Environments Mock Servers Monitors Flows History

https://reqres.in/api/users/2

GET https://reqres.in/api/users/2

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

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (19) Test Results

Pretty Raw Preview Visualize JSON

```
1 "data": {  
2     "id": 2,  
3     "email": "janet.weaver@reqres.in",  
4     "first_name": "Janet",  
5     "last_name": "Weaver",  
6     "avatar": "https://reqres.in/img/faces/2-image.jpg"  
7 },  
8 "support": {  
9     "url": "https://reqres.in/#support-heading",  
10    "text": "To keep ReqRes free, contributions towards server costs are appreciated!"  
11 }  
12 }  
13 }
```

Status: 200 OK Time: 101 ms Size: 1.15 KB Save Response

Find and Replace Console

Capture requests and cookies Bootcamp Runner Trash

Type here to search

19°C Sunny 11:28 15-02-2022

How to set content type header in json

Add Manually Content-Type= application/json

Send request into json and receive response into json

The screenshot shows the Postman interface. On the left, there's a sidebar with 'My Workspace' containing 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. The main area shows a collection named 'collection1' with a single request: 'GET https://reqres.in/api/users/2'. In the 'Headers' tab, 'Content-Type' is set to 'application/json'. The 'Body' tab contains a JSON object:

```
1 {  
2   "data": {  
3     "id": 2,  
4     "email": "janet.weaver@example.com",  
5     "first_name": "Janet",  
6     "last_name": "Weaver",  
7     "avatar": "https://reqres.in/img/faces/2-image.jpg"  
8   },  
9   "support": {  
10    "url": "https://reqres.in/#support-heading",  
11    "text": "To keep ReqRes free, contributions towards server costs are appreciated!"  
12  }  
13}
```

POST REQUEST

The screenshot shows a browser window with multiple tabs open. The active tab is 'Reqres - A hosted REST-API reader' at https://reqres.in. On the left, there's a sidebar with various API endpoints: 'LIST USERS', 'SINGLE USER', 'SINGLE USER NOT FOUND', 'LIST <RESOURCE>', 'SINGLE <RESOURCE>', 'SINGLE <RESOURCE> NOT FOUND', 'CREATE', 'UPDATE', 'UPDATE', and 'DELETE'. The 'CREATE' endpoint is highlighted. To the right, there are two panes: 'Request' and 'Response'. The 'Request' pane shows a JSON object:

```
{  
  "name": "morpheus",  
  "job": "leader"  
}
```

The 'Response' pane shows a JSON object:

```
{  
  "name": "morpheus",  
  "job": "leader",  
  "id": "166",  
  "createdAt": "2022-02-15T07:29:20.550Z"  
}
```

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

Collections APIs Environments Mock Servers Monitors Flows History

collection1 / https://reqres.in/api/users/

POST https://reqres.in/api/users

Params Auth Headers (10) Body Pre-req. Tests Settings Cookies Beautify

raw JSON

```
1
2 ... "name": "Sachin Sirohi",
3 ... "job": "Corporate Trainer"
4
```

Body 201 Created 483 ms 923 B Save Response

Pretty Raw Preview Visualize JSON

```
1
2 ...
3   "name": "Sachin Sirohi",
4   "job": "Corporate Trainer",
5   "id": "791",
6   "createdAt": "2022-02-15T07:39:58.172Z"
7
```

Code snippet Node.js - Axios

```
</>
1 var axios = require('axios');
2 var data = JSON.stringify({
3   "name": "Sachin Sirohi",
4   "job": "Corporate Trainer"
5 });
6
7 var config = {
8   method: 'post',
9   url: 'https://reqres.in/api/users',
10  headers: {
11    'Content-Type': 'application/json'
12  },
13  data : data
14 };
15
16 axios(config)
17 .then(function (response) {
18   console.log(JSON.stringify(response.data));
19 })
20 .catch(function (error) {
21   console.log(error);
22 });
23;
```

Find and Replace Console

Type here to search

21°C Sunny ENG 13:11 15-02-2022

For Check this user id is created or not

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

My Workspace

Collections APIs Environments Mock Servers Monitors Flows History

collection1 / https://reqres.in/api/users/

POST https://reqres.in/api/users/791

Params Auth Headers (10) Body Pre-req. Tests Settings Cookies Beautify

raw JSON

```
1
2 ...
3   "name": "Sachin Sirohi",
4   "job": "Corporate Trainer"
5
```

Body 201 Created 461 ms 911 B Save Response

Pretty Raw Preview Visualize JSON

```
1
2 ...
3   "name": "Sachin Sirohi",
4   "job": "Corporate Trainer",
5   "id": "791",
6   "createdAt": "2022-02-15T07:41:57.960Z"
7
```

Code snippet Node.js - Axios

```
</>
1 var axios = require('axios');
2 var data = JSON.stringify({
3   "name": "Sachin Sirohi",
4   "job": "Corporate Trainer"
5 });
6
7 var config = {
8   method: 'post',
9   url: 'https://reqres.in/api/users/791',
10  headers: {
11    'Content-Type': 'application/json'
12  },
13  data : data
14 };
15
16 axios(config)
17 .then(function (response) {
18   console.log(JSON.stringify(response.data));
19 })
20 .catch(function (error) {
21   console.log(error);
22 });
23;
```

Find and Replace Console

Type here to search

21°C Sunny ENG 13:12 15-02-2022

Put into Form Data use only if you want to send data into HTML Form

The screenshot shows the Postman application interface. On the left, the sidebar includes sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main workspace displays a collection named 'collection1' with a POST request to 'https://reqres.in/api/users'. The 'Body' tab is selected, showing a 'form-data' section with two fields: 'name' (Value: 'Ritu') and 'job' (Value: 'Teacher'). The response status is 201 Created, and the JSON response body is shown as:

```
1 "id": "57",
2 "createdAt": "2022-02-15T07:44:58.665Z"
```

On the right, there is a 'Code snippet' panel titled 'NodeJS - Axios' containing the following code:

```
1 var axios = require('axios');
2 var FormData = require('form-data');
3 var data = new FormData();
4 data.append('name', 'Ritu');
5 data.append('job', 'Teacher');
6
7 var config = {
8   method: 'post',
9   url: 'https://reqres.in/api/users',
10  headers: {
11    'Content-Type': 'application/json',
12    ...data.getHeaders()
13  },
14  data : data
15 };
16
17 axios(config)
18 .then(function (response) {
19   console.log(JSON.stringify(response.data));
20 })
21 .catch(function (error) {
22   console.log(error);
23 });
```

You can use pre-request like this

The screenshot shows the Postman application interface. The sidebar is identical to the previous one. The main workspace displays a collection named 'collection1' with a POST request to 'https://reqres.in/api/users'. The 'Pre-request Script' tab is selected, containing the following JavaScript code:

```
1 //login
2 //javascript
3 var name="Sachin"
4 var job="Faculty"
```

The response status is 201 Created, and the JSON response body is shown as:

```
1 "id": "57",
2 "createdAt": "2022-02-15T07:44:58.665Z"
```

A tooltip for the 'Pre-request scripts' tab provides information: 'Pre-request scripts are written in JavaScript, and are run before the request is sent.' It also links to 'Learn more about pre-request scripts', 'SNIPPETS', 'Get an environment variable', and 'Get a global variable'.

Postman Console- printing data whatever you type

The screenshot shows the Postman interface. On the left, the sidebar includes sections for Collections, APIs, Environment, Mock Servers, Monitors, Flows, and History. The main area displays a collection named "collection1" under the URL "https://reqres.in/api/users". A POST request is being prepared with the following body:

```
1 "name": "Sachin Sirohi",
2 "job": "Corporate Trainer",
3 "id": "452",
4 "createdAt": "2022-02-15T07:51:10.469Z"
```

The response status is 201 Created, time 470 ms, size 921 B. Below the body, there are tabs for Body, Cookies, Headers (15), and Test Results. The Body tab shows the JSON response:

```
1 {
2   "name": "Sachin Sirohi",
3   "job": "Corporate Trainer",
4   "id": "452",
5   "createdAt": "2022-02-15T07:51:10.469Z"
6 }
```

The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows the Postman interface with a different layout. The sidebar includes sections for Home, Workspaces, API Network, Reports, and Explore. The main area displays a collection named "Postman Console" under the URL "https://reqres.in/api/users". The interface shows a list of log entries:

Method	URL	Time
GET	https://reqres.in/api/users/2	200 315 ms
GET	https://reqres.in/api/users/3	200 131 ms
GET	https://reqres.in/api/users/2	200 114 ms
GET	https://reqres.in/api/users/2	200 101 ms
GET	https://reqres.in/api/users/2	200 160 ms
GET	https://reqres.in/api/users/2	200 82 ms
GET	https://reqres.in/api/users/2?name=Janet	200 499 ms
GET	https://reqres.in/api/users/2?name=Janet	200 35 ms
GET	https://reqres.in/api/users/2?name=Janets	200 445 ms
GET	https://reqres.in/api/users/2?names=amita	200 546 ms
GET	https://reqres.in/api/users/2?names=amita	200 37 ms
GET	https://reqres.in/api/users/2	200 61 ms
GET	https://reqres.in/api/users/2	200 53 ms
POST	https://reqres.in/api/users	201 747 ms
POST	https://reqres.in/api/users	201 485 ms
POST	https://reqres.in/api/users	201 468 ms
POST	https://reqres.in/api/users	201 479 ms
POST	https://reqres.in/api/users/970	201 483 ms
GET	https://reqres.in/api/users/970	404 468 ms

The bottom of the screen shows the Windows taskbar with various pinned icons.

Session 3

How to add cookie show expiry also

The screenshot shows the Postman application interface. A modal window titled "MANAGE COOKIES" is open for the URL <https://reqres.in>. The modal contains a list of cookies from the domain reqres.in, specifically "Cookie_4", "Cookie_1", and "Cookie_3". Below the list is a text input field labeled "Type a domain name" with "reqres.in" typed into it. At the bottom of the modal, there is a "Save" button.

This screenshot shows the same Postman interface and "MANAGE COOKIES" dialog as the previous one, but with a new cookie added. The list now includes "Cookie_4" along with "Cookie_1" and "Cookie_3". A new cookie entry for "Cookie_4" is visible at the bottom of the list, showing the value "name=sachin;Cookie_4=value; Path=/; Expires=Wed, 15 Feb 2023 15:34:06 GMT;". The "Save" button is still present at the bottom of the modal.

Share Collection

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

SachinCollection New Collection [CONFLICT] GET Ne... [CONFLICT] POST ht... No Environment

Invite Save Send Cookies Beautify

My Workspace Collections APIs Environments Mock Servers Monitors Flows History

collection1 / https://reqres.in/api/users/

POST Params Body Pretty

With people Via Run in Postman Via JSON link

Generate a shareable public link to a static snapshot of your collection. All your generated links are available in your profile.

This link does not update automatically when changes are made to the collection. Check out the Postman API to access your synced collections.

<https://www.getpostman.com/collections/8c0436647ba4ea978e84> Update Link

status: 201 Created Time: 1614 ms Size: 913 B Save Response

Find and Replace Console Capture requests and cookies Bootcamp Runner Trash

Type here to search

24°C Haze ENG 15-02-2022

Share SachinCollection

Take care when sharing

Anyone who has this link can access the collection - even people outside your team or organization.

Export Collection

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

SachinCollection New Collection [CONFLICT] GET Ne... [CONFLICT] POST ht... No Environment

Invite Save Send Cookies Beautify

My Workspace Collections APIs Environments Mock Servers Monitors Flows History

collection1 / https://reqres.in/api/users/

POST Params Body Pretty

Skip exporting! Sharing just got quicker & easier.

Share this collection with people directly and work together.

Share collection

New Collection will be exported as a JSON file. Export as:

Collection v2

Collection v2.1 (recommended)

Learn more about collection formats

Cancel Export

"createdAt": "2022-02-15T15:32:22.946Z"

Find and Replace Console Capture requests and cookies Bootcamp Runner Trash

Type here to search

24°C Haze ENG 15-02-2022

Import Collection by file

The screenshot shows the Postman interface with the 'Import' dialog open. The dialog title is 'Import' and it says 'Select files to import · 1/1 selected'. There is a table with one row:

NAME	FORMAT	IMPORT AS
SachinCollection	Postman Collection v2.1	Collection

At the bottom right of the dialog are 'Cancel' and 'Import' buttons.

Monitor Collection

The screenshot shows the Postman interface with the 'Create a monitor' dialog open. The dialog title is 'Create a monitor'.

Form fields include:

- Monitor name: SachinCollection
- Collection: SachinCollection
- Collection tag: CURRENT
- Environment: None
- Run this monitor:
 - Week Timer
 - Every Day
 - 9:00 PM
- Regions:
 - Automatically select region (radio button)
 - Manually select region (radio button)
- Receive email notifications for run failures and errors (checkbox): sachihi47@gmail.com

At the bottom right of the dialog are 'Cancel' and 'Create' buttons.

Run Monitor Collection after show this

We can also check geolocation and along with www.runscope.com by geolocation also

The screenshot shows the Postman interface with the following details:

- Header:** Postman, File, Edit, View, Help, Home, Workspaces, API Network, Reports, Explore, Search Postman.
- Left Sidebar:** My Workspace (Collections, APIs, Environments, Mock Servers, Monitors, Flows, History).
- Central Area:** A failed monitor collection named "SachinCollection" is displayed as "UNHEALTHY". It shows a timeline from 0.06s to 0.10s. A red bar indicates a failure at 9:14 PM on Feb 15. Filter options include All Requests, Type: All, Run result: All, US (East), and Clear Filters.
- Bottom Status:** 0 failed tests, 3 errors, across 1 region 09:14 PM, 15 Feb 2022.
- System Tray:** Region US (East), Capture requests and cookies, Bootcamp, Runner, Trash, 21:14, 15-02-2022.

View Documentation and publish

The screenshot shows the Postman interface with the following details:

- Header:** Postman, File, Edit, View, Help, Home, Workspaces, API Network, Reports, Explore, Search Postman.
- Left Sidebar:** My Workspace (Collections, APIs, Environments, Mock Servers, Monitors, Flows, History).
- Central Area:** The "SachinCollection" documentation page is shown. It includes sections for "SachinCollection" (description: "Make things easier for your teammates with a complete collection description."), "GET New Request" (description: "Make things easier for your teammates with a complete request description."), and "GET https://reqres.in/api/users/" (description: "Make things easier for your teammates with a complete request description"). A "JUMP TO" sidebar lists "Introduction", "GET New Request", and "GET https://reqres.in/api/users/".
- Bottom Status:** Capture requests and cookies, Bootcamp, Runner, Trash, 21:21, 15-02-2022.

Collection Runner->click on run

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main area displays a collection named "SachinCollection" under "My Workspace". The collection contains three requests: "GET New Request", "GET https://reqres.in/api/users/", and "POST https://reqres.in/api/users". The "POST" request is highlighted. The status bar at the bottom shows system information like weather (24°C Haze), battery level (21:34), and date (15-02-2022).

Authentication

- Basic Auth - Base64 (username and password)
- Digest authentication - User, Pass with Encryption
- Bearer Tokens - They are predominant type of access token used with OAuth 2.0. Opaque string, not intended to have any meaning to clients using it.
- JWT - Structured tokens

Jwt- json web token

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main area displays a collection named "SachinCollection" with a GET request to "https://reqres.in/api/users/2". The "Authorization" tab is selected in the request settings. A dropdown menu is open, showing options like "Inherit auth from parent", "No Auth", "API Key", "Bearer Token", "Basic Auth", "Digest Auth", "OAuth 1.0", "OAuth 2.0", "Hawk Authentication", "AWS Signature", "NTLM Authentication", and "Akamai EdgeGrid". The status bar at the bottom indicates "Status: 200 OK Time: 100 ms Size: 1.15 KB".

Go to fake api also see this command

Install Node Js Software

When Node js is install npm is automatically install no need to install

Install Json Server PowerShell= `npm install -g json-server@v0.12.2`

Open cmd check ->`node -v`

Check npm on cmd type->`npm`

Go to fake api

<https://www.npmjs.com/package/json-server>

CRUD with Local Dummy Server.

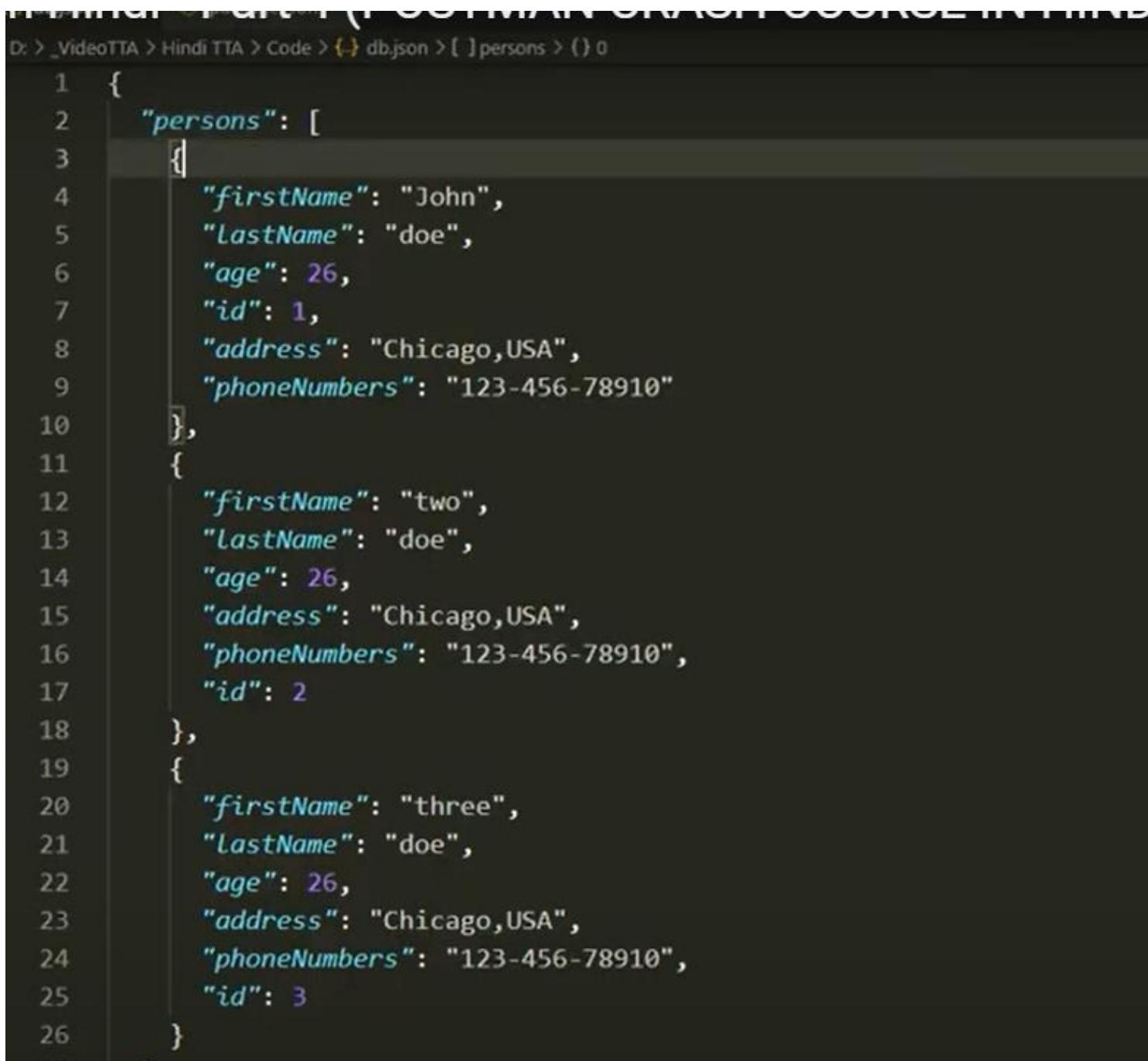
- Install NodeJS.org
- Install JSON Server
- `npm i -g json-server@v0.12.2`
- Create db.json
- Run Collection
- `Json-server -w db.json`

```
*person*
{
  "firstName": "John",
  "lastName": "doe",
  "age": 26,
  "id": 1,
  "address": "Chicago,USA",
  "phoneNumbers": "123-456-7890"
}
```

`npm install -g json-server@v0.12.2`

```
json-server --watch db.json
```

Create db.json on vs code and save any location



The screenshot shows the contents of a db.json file in a code editor. The file is a JSON object with a single key-value pair. The value is an array of three objects, each representing a person with properties: firstName, LastName, age, id, address, and phoneNumbers. The persons are numbered 1, 2, and 3.

```
D:\_VideoTTA > Hindi TTA > Code > db.json > [ ]persons > {} 0
1 {
2   "persons": [
3     {
4       "firstName": "John",
5       "LastName": "doe",
6       "age": 26,
7       "id": 1,
8       "address": "Chicago,USA",
9       "phoneNumbers": "123-456-78910"
10    },
11    {
12      "firstName": "two",
13      "LastName": "doe",
14      "age": 26,
15      "address": "Chicago,USA",
16      "phoneNumbers": "123-456-78910",
17      "id": 2
18    },
19    {
20      "firstName": "three",
21      "LastName": "doe",
22      "age": 26,
23      "address": "Chicago,USA",
24      "phoneNumbers": "123-456-78910",
25      "id": 3
26    }
27 }
```

Change cmd path where db.json file is store and after run the - json-server --watch db.json

```
C:\Windows\system32\cmd.exe - "node" "C:\Users\Sachin sirohi\AppData\Roaming\npm\node_modules\json-server\bin\index.js" -w db.json
C:\Users\Sachin sirohi>node -v
v16.14.0
C:\Users\Sachin sirohi>npm
npm <command>
Usage:
  npm install      install all the dependencies in your project
  npm install <foo> add the <foo> dependency to your project
  npm test         run this project's tests
  npm run <foo>   run the script named <foo>
  npm <command> -h quick help on <command>
  npm -l          display usage info for all commands
  npm help <term> search for help on <term> (in a browser)
  npm help npm    more involved overview (in a browser)

All commands:
  HPL  access, adduser, audit, bin, bugs, cache, ci, completion,
M125 config, dedupe, deprecate, diff, dist-tag, docs, doctor,
edit, exec, explain, explore, find-dupes, fund, get, help,
hook, init, install, install-ci-test, install-test, link,
ll, login, logout, ls, org, outdated, owner, pack, ping,
pkg, prefix, profile, prune, publish, rebuild, repo,
restart, root, run-script, search, set, set-script,
SACH shrinkwrap, star, stars, start, stop, team, test, token,
SIRO uninstall, unpublish, unstar, update, version, view, whoami

Specify configs in the ini-formatted file:
  C:\Users\Sachin sirohi\.npmrc
Or on the command line via: npm <command> --key=value

More configuration info: npm help config
Configuration fields: npm help 7 config

npm@8.3.1 C:\Program Files\nodejs\node_modules\npm
LibreOffice: C:\Users\Sachin sirohi>npm i -g json-server@v0.12.2
7.2 npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
McAfee added 230 packages, and audited 231 packages in 24s
AntiVirus
  4 packages are looking for funding
```

```
C:\Windows\system32\cmd.exe - "node" "C:\Users\Sachin sirohi\AppData\Roaming\npm\node_modules\json-server\bin\index.js" -w db.json
Specify configs in the ini-formatted file:
  C:\Users\Sachin sirohi\.npmrc
Or on the command line via: npm <command> --key=value

More configuration info: npm help config
Configuration fields: npm help 7 config

npm@8.3.1 C:\Program Files\nodejs\node_modules\npm
C:\Users\Sachin sirohi>npm i -g json-server@v0.12.2
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
added 230 packages, and audited 231 packages in 24s
  4 packages are looking for funding
    run `npm fund` for details
  11 moderate severity vulnerabilities

  To address issues that do not require attention, run:
    npm audit fix

  To address all issues (including breaking changes), run:
    npm audit fix --force

  Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 8.3.1 => 8.5.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.5.0
npm notice Run npm install -g npm@8.5.0 to update!
npm notice

C:\Users\Sachin sirohi>json-server -w db.json
\(^_~)/ hi!

Oops, db.json doesn't seem to exist
Creating db.json with some default data

Loading db.json
Done.

Resources
http://localhost:3000/posts
http://localhost:3000/categories
```

```
C:\Windows\system32\cmd.exe - "node" "C:\Users\Sachin sirohi\AppData\Roaming\npm\node_modules\json-server\bin\index
Oops, db.json doesn't seem to exist
Creating db.json with some default data

Loading db.json
Done

Resources
http://localhost:3000/posts
http://localhost:3000/comments
http://localhost:3000/profile

Home
http://localhost:3000

Type s + enter at any time to create a snapshot of the database
Watching...

GET /db 200 6.895 ms - 243
GET /__rules 404 5.951 ms - 2
GET /posts 200 13.814 ms - 77
HP LGET /db 304 0.695 ms - -
M125^C
C:\Users\Sachin sirohi>
C:\Users\Sachin sirohi>
SACHC:\Users\Sachin sirohi>d:
SIRO
D:>cd D:\Postman API
D:\Postman API>json-server -w db.json
R x64 4 \{^_^\}/ hi!

Loading db.json
Done

Resources
http://localhost:3000/persons

Home
http://localhost:3000

Type s + enter at any time to create a snapshot of the database
Watching...
```

Open the web browser and type <http://localhost:3000/> after click person

<http://localhost:3000/persons>

Go to postman and create a request type above link

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

SachinCollection / JsonServerLocalAPI

GET http://localhost:3000/persons

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

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "firstName": "John",
3   "lastName": "doe",
4   "age": 26,
5   "id": 1,
6   "address": "Chicago,USA",
7   "phoneNumbers": "123-456-78910"
8 },
9 {
10   "firstName": "two",
11   "lastName": "doe",
12   "age": 26,
13   "address": "Chicago,USA",
14   "phoneNumbers": "123-456-78910",
15   "id": 2
16 }
```

Capture requests and cookies Bootcamp Runner Trash 21:28 23°C Haze ENG 16-02-2022

Create new person record by post request

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

SachinCollection / JsonServerLocalAPIPost

POST http://localhost:3000/persons

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

Body

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

Beautify

```
1 {
2   "firstName": "Sachin",
3   "lastName": "Sirohi",
4   "age": 30,
5   "id": 17,
6   "address": "Meerut",
7   "phoneNumbers": "123-456-78910"
8 }
```

Body Cookies Headers (15) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "firstName": "Sachin",
3   "lastName": "Sirohi",
4   "age": 30,
5   "id": 17,
6   "address": "Meerut",
7   "phoneNumbers": "123-456-78910"
8 }
```

Capture requests and cookies Bootcamp Runner Trash 21:33 23°C Haze ENG 16-02-2022

Postman

SachinCollection / JsonServerLocalAPIPost

POST http://localhost:3000/persons

Body (raw)

```
{
  "firstName": "four",
  "lastName": "ji",
  "age": 26,
  "address": "Chicago,USA",
  "phoneNumbers": "123-456-78910",
  "id": 13
},
{
  "firstName": "four",
  "lastName": "ji",
  "age": 26,
  "address": "Chicago,USA",
  "phoneNumbers": "123-456-78910",
  "id": 14
},
{
  "firstName": "four",
  "lastName": "ji",
  "age": 26,
  "address": "Chicago,USA",
  "phoneNumbers": "123-456-78910",
  "id": 15
},
{
  "firstName": "Lucky",
  "lastName": "ji",
  "age": 26,
  "address": "Chicago,USA",
  "phoneNumbers": "123-456-78910",
  "id": 16
},
{
  "firstName": "Sachin",
  "lastName": "Sirohi",
  "age": 30,
  "id": 17,
  "address": "Meerut",
  "phoneNumbers": "123-456-78910"
}
]
```

Body Cookies Headers (15) Test Results

Pretty Raw Preview Visualize JSON

Find and Replace Console

Learning Postma....docx db.json Show all

23°C Haze ENG 16-02-2022 21:34

Patch Request change partial update into json file behalf of id

You can update only one record also

Postman

SachinCollection / JsonServerLocalAPIPatch

PATCH http://localhost:3000/persons/1

Body (raw)

```
{
  "firstName": "Swati",
  "lastName": "Chaudhary",
  "age": 26,
  "id": 1,
  "address": "Chicago,USA",
  "phoneNumbers": "123-456-78910"
}
```

Body Cookies Headers (13) Test Results

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 60 ms Size: 536 B Save Response

Find and Replace Console

Capture requests and cookies Bootcamp Runner Trash

23°C Haze ENG 16-02-2022 21:40

Partial Update

The screenshot shows the Postman interface with a collection named "SachinCollection / JsonServerLocalAPIPatch". A PATCH request is selected with the URL `http://localhost:3000/persons/1`. The "Body" tab is active, showing a JSON payload with only the "firstName" field updated to "Gita". The response status is 200 OK, indicating a successful partial update.

```
1
2   ...
3   ...
4   ...
5   ...
6   ...
7   ...
8   ...
```

"firstName": "Gita",
"lastName": "Chaudhary",
"age": 26,
"id": 1,
"address": "Chicago,USA",
"phoneNumbers": "123-456-78910"

Put Request demand all value to supply but we can change one or more according to your need.

Put also update values whole

The screenshot shows the Postman interface with a collection named "SachinCollection / JsonServerLocalAPIPut". A PUT request is selected with the URL `http://localhost:3000/persons/1`. The "Body" tab is active, showing a JSON payload with all fields updated to new values. The response status is 200 OK, indicating a successful full update.

```
1
2   ...
3   ...
4   ...
5   ...
6   ...
7   ...
8   ...
```

"firstName": "Anamika",
"lastName": "Chaudhary",
"age": 26,
"id": 1,
"address": "Chicago,USA",
"phoneNumbers": "123-456-78910"

Delete the record behalf of id doesn't show anything but show only status

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing collections like 'SachinCollection / JsonServerLocalAPIDelete'. The main area displays a DELETE request to 'http://localhost:3000/persons/1'. The 'Body' tab is selected, showing the message 'This request does not have a body'. Below the request, the response section shows a status of '200 OK' with a detailed description: 'Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action.' At the bottom, there's a search bar and a taskbar with various icons.

Integration Scenario with the help of test case

Post-Get-Patch-Del-Get-Post-Get work in flow

The screenshot shows an integration flow in Postman. The top navigation bar includes 'GET Get Single', 'Postman - LI...', 'GET New Request', 'POST Create Person', and 'PATCH Up...'. The main view shows a sequence of requests: a POST to 'http://localhost:3000/persons' with a test script, followed by a GET to 'http://localhost:3000/persons/1'. The test script under the first request contains the following code:

```
1 pm.test('verify get all persons', function(){
2     var responseBody = pm.response.json();
3     console.log(responseBody);
4     pm.globals.set("pid", responseBody["id"]);
5     pm.environment.set("pid", responseBody["id"]);
6});
```

API Integration Tests With Postman

Go to API <https://cat-fact.herokuapp.com/facts>

The screenshot shows a web browser window with the URL <https://cat-fact.herokuapp.com/docs/>. The page title is "API Documentation". Below it, a sub-section titled "Daily cat facts!" features a small cat icon. A button labeled "View on GitHub" is visible. The background is dark blue.

API Documentation

Base URL for all endpoints <https://cat-fact.herokuapp.com>

The response time will likely be a few seconds long on the first request, because this app is running on a free Heroku dyno. Subsequent requests will behave as normal.

Endpoints

/facts Retrieve and query facts

The screenshot shows the Postman application interface. On the left, the sidebar includes sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main workspace shows a collection named "SachinCollection / IntegrationTestingWithTestCase". Inside this collection, there is a single GET request to "https://cat-fact.herokuapp.com/facts". The "Params" tab is selected, showing a single parameter "Key" with the value "Value". The "Body" tab shows a JSON response with status 200 OK, time 1787 ms, and size 2.21 KB. The response body is displayed in a code editor-like view:

```
1 {
2   "status": {
3     "verified": true,
4     "sentCount": 1
5   },
6   "_id": "58e00800aac31001185ed07",
7   "user": "58e007480aac31001185ecf",
8   "text": "Wikipedia has a recording of a cat meowing, because why not?",
9   "__v": 0,
10   "source": "user",
11 }
```

Copy Paste postman API json code into online json formatter <https://jsonformatter.org/>

The screenshot shows the JSON formatter.org website with two code editors. The left editor contains the original JSON code from Postman:

```
1 {
2   "status": {
3     "verified": true,
4     "sentCount": 1
5   },
6   "_id": "58e008800aac31001185ed07",
7   "user": "58e007480aac31001185ecef",
8   "text": "Wikipedia has a recording of a cat meowing, because why not?",
9   "__v": 0,
10  "source": "user",
11  "updatedAt": "2020-08-23T20:20:01.611Z",
12  "type": "cat",
13  "createdAt": "2018-03-06T21:20:03.505Z",
14  "deleted": false,
15  "used": false
16 },
17 }
18 {
19   "status": {
20     "verified": true,
21     "sentCount": 1
22   },
23   "_id": "58e008630aac31001185ed01",
24   "user": "58e007480aac31001185ecef",
25 }
```

The right editor shows the same JSON code after being formatted by the tool:

```
1 {
2   "status": {
3     "verified": true,
4     "sentCount": 1
5   },
6   "_id": "58e008800aac31001185ed07",
7   "user": "58e007480aac31001185ecef",
8   "text": "Wikipedia has a recording of a cat meowing, because why not?",
9   "__v": 0,
10  "source": "user",
11  "updatedAt": "2020-08-23T20:20:01.611Z",
12  "type": "cat",
13  "createdAt": "2018-03-06T21:20:03.505Z",
14  "deleted": false,
15  "used": false
16 },
17 }
18 {
19   "status": {
20     "verified": true,
21     "sentCount": 1
22   },
23   "_id": "58e008630aac31001185ed01",
24   "user": "58e007480aac31001185ecef",
25 }
```

A banner at the bottom of the page reads "Fast and simple data warehousing from day one" with an AWS logo.

See object minimize click side in number 2

The screenshot shows the JSON formatter.org website again, but this time the second object's "status" object is collapsed, indicated by a small icon next to the brace on line 19. The rest of the JSON code remains the same as in the previous screenshot.

Test Case click on Test->Status Code: Code in 200

The screenshot shows the Postman application interface. On the left, the sidebar includes 'My Workspace' with 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'Flows'. Under 'Collections', 'SachinCollection' is expanded, showing various API requests like 'New Request', 'GET https://reqres.in/api/users...', etc. A specific test case named 'IntegrationTestingWithTestCase' is selected. The main panel displays a GET request to 'https://cat-fact.herokuapp.com/facts'. The 'Tests' tab is active, containing the following JavaScript code:

```
1 pm.test("Status code is 200", function () {  
2   pm.response.to.have.status(200);  
3 });
```

The 'Body' tab shows the response JSON, which includes fields such as '_id', 'user', 'text', '_v', 'source', 'updatedAt', 'type', 'createdAt', 'deleted', and 'used'. The status bar at the bottom indicates 'Status: 200 OK'.

Check test case result -> click on send->see Test Results

This screenshot shows the same Postman interface after the test case has been run. The 'Test Results' tab is now active, indicating 1/1 result. The status is 'PASS' and the message is 'Status code is 200'. The rest of the interface is identical to the previous screenshot, showing the collection tree and the successful request details.

Json Schema Validator format ->click on->use Tiny validator for JSON Data

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Current Collection:** SachinCollection / IntegrationTestingWithTestCase
- Request:** GET https://cat-fact.herokuapp.com/facts
- Test Script (Tests tab):**

```
1 pm.test("Status code is 200", function () {
2     pm.response.to.have.status(200);
3 });
4 var schema = {
5     "items": [
6         {
7             "type": "boolean"
8         }
9     ];
10 var data1 = [true, false];
11 var data2 = [true, 123];
12
13 pm.test('Schema is valid', function () {
14     pm.expect(tv4.validate(data1, schema)).to.be.true;
15     pm.expect(tv4.validate(data2, schema)).to.be.true;
16 });

```
- Test Results:** 1/1 Passed
- Bottom Status Bar:** Status: 200 OK Time: 1163 ms Size: 2.09 KB Save Response
- Taskbar:** Find and Replace, Console, Type here to search, Start button, Task View, Edge, File Explorer, Mail, Google Chrome, Windows File Explorer, Taskbar icons, Weather (22°C Haze), Date (20-02-2022), Time (18:36).

Convert Json to json Schema online -><https://www.liquid-technologies.com/online-json-to-schema-converter>

Copy post man json API Code and paste online and generate json schema and copy paste json schema into postman

Var schema={ };

The screenshot shows the liquid-technologies.com/online-json-to-schema-converter website with the following details:

- Left Sidebar:** Free Online JSON Tools (JSON Formatter, JSON Validator, JSON Validator, JSON to JSON Schema, JSON Schema to JSON), XML Schema Tutorials (Elements and Attributes, Conventions and Recommendations, Extending Existing Types, Namespaces, Groups and Any Types).
- Input Area:** JSON code:

```
22     },
23     "_id": "58e008630aac31001185ed01",
24     "user": "58e007480aac31001185ecf",
25     "text": "When cats grimace, they are usually \"taste-scenting.\" They have an extra organ that, with some breathi
26
```
- Options:** Options button
- Inferred JSON Schema:**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "array",
  "items": [
    {
      "type": "object",
      "properties": {
        "status": {
          "type": "object",
          "properties": {
            "verified": {
              "type": "boolean"
            },
            "sentCount": {
              "type": "integer"
            }
          }
        },
        "required": [

```
- Bottom Status Bar:** Liquid Technologies Web Site uses cookies. Learn more. Close, Taskbar icons, Weather (22°C Haze), Date (20-02-2022), Time (18:46).

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

SachinCollection / IntegrationTestingWithTestCase

GET https://cat-fact.herokuapp.com/facts

Params Authorization Headers (8) Body Pre-request Script Tests Settings

```
1 pm.test("Status code is 200", function () {  
2   pm.response.to.have.status(200);  
3 });  
4 var schema =  
5  
6 "$$schema": "http://json-schema.org/draft-04/schema#",  
7 "type": "array",  
8 "items": [  
9   {  
10     "type": "object",  
11     "properties": {  
12       "status": {  
13         "type": "object",  
14         "properties": {  
15           "verified": {  
16             "type": "boolean"  
17           },  
18         },  
19       },  
20     },  
21   },  
22 ],  
23   "sentCount": 1,  
24 },  
25   "_id": "58e008800aac31001185ed07",  
26 }];  
27  
28 pm.expect(tv4.validate(data1, schema)).to.be.true;  
29 pm.expect(tv4.validate(data2, schema)).to.be.true;  
30  
31 pm.test('Schema is valid', function () {  
32   pm.expect(tv4.validate(data1, schema)).to.be.true;  
33   pm.expect(tv4.validate(data2, schema)).to.be.true;  
34 });  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
18
```

After remove data1 and data2

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing sections for Collections, APIs, Environments, Mock Servers, Monitors, and Flows. The main area displays a collection named 'SachinCollection' with several requests like 'GET New Request', 'GET https://reqres.in/api/users/', etc. A specific test case titled 'IntegrationTestingWithTestCase' is selected. The 'Tests' tab is active, showing the following JavaScript code:

```
314     "text",
315     "__v",
316     "source",
317     "updatedAt",
318     "type",
319     "createdAt",
320     "deleted",
321     "used"
322   ]
323 }
324 ]
325 };
326
327 pm.test('Schema is valid', function () {
328   pm.expect(tv4.validate(data1, schema)).to.be.true;
329   pm.expect(tv4.validate(data2, schema)).to.be.true;
330});
```

Below the code, the 'Body' tab is selected, showing the response body as:

```
1 [
2   {
3     "status": "OK",
4     "verified": true,
5     "sentCount": 1
6   },
7   "_id": "5be000800aac31001185ed07",
```

The status bar at the bottom indicates: Status: 200 OK, Time: 1163 ms, Size: 2.09 KB, Save Response, Capture requests and cookies, Bootcamp, Runner, Trash, 22°C Haze, 18:49, ENG, 20-02-2022.

Write test case

```
pm.test('Sachin Sirohi Test Case', function ()
{
  var result=tv4.validateResult(pm.response.json(),schema);
  if (!result.valid)
  {
    console.log(result)
  }
  pm.expect(result.valid).to.be.true;
});
```

The screenshot shows the Postman interface with a collection named 'SachinCollection' containing a test case for a GET request to <https://cat-fact.herokuapp.com/facts>. The 'Tests' tab displays the following JavaScript code:

```

319 "createdAt",
320 "deleted",
321 "used"
322 ]
323 }
324 }
325 };
326
327 pm.test('Sachin Sirohi Test Case', function () {
328 {
329     var result=tv4.validateResult(pm.response.json(),schema);
330     if (!result.valid)
331     {
332         console.log(result);
333     }
334     pm.expect(result.valid).to.be.true;
335 });

```

The 'Test Results' section shows two successful tests:

- PASS Status code is 200
- PASS Sachin Sirohi Test Case

The status bar at the bottom indicates a successful response: Status: 200 OK.

If we change `"sentCount": {`

`"type": "integer"`

}

To `"sentCount": {`

`"type": "string"`

}

Show test case fail because we can't supply string value into integer

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

SachinCollection / IntegrationTestingWithTestCase

GET https://cat-fact.herokuapp.com/facts

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Test scripts are written in JavaScript, and are run after the response is received.

Learn more about tests scripts

SNIPPETS Response body: JSON value check

Response body: Is equal to a string

Response headers: Content-Type header check

Response time is less than 200ms

Status code: Successful POST request

Status code: Code name has string

Response body: Convert XML body to a JSON Object

Use Tiny Validator for JSON data

Body Cookies (1) Headers (9) Test Results (1/2)

All Passed Skipped Failed

PASS Status code is 200

FAIL Sachin Sirohi Test Case | AssertionError: expected false to be true

Find and Replace Console

Capture requests and cookies Bootcamp Runner Trash

Type here to search

22°C Haze ENG 19:01 20-02-2022

See error in console below postman window invalid type error

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

SachinCollection / IntegrationTestingWithTestCase

GET https://cat-fact.herokuapp.com/facts

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Test scripts are written in JavaScript, and are run after the response is received.

Learn more about tests scripts

SNIPPETS Response time is less than 200ms

Status code: Successful POST request

Status code: Code name has string

Body Cookies (1) Headers (9) Test Results (1/2)

All Passed Skipped Failed

PASS Status code is 200

All Logs Clear

200 | 1231 ms

GET https://cat-fact.herokuapp.com/facts

+ toString: "[Function: toString]", error: {}, missing: [0]...
 toString: "[Function: toString]"
+ error: {}
 type: "Error"
 name: "ValidationError"
 message: "Invalid type: number (expected string)"
 missing: [0]
 valid: false

GET https://cat-fact.herokuapp.com/facts

Find and Replace Console

Capture requests and cookies Bootcamp Runner Trash

Type here to search

22°C Haze ENG 19:06 20-02-2022

After you can run API Test Case

The screenshot shows the Postman application interface. On the left, the sidebar lists collections, APIs, environments, mock servers, monitors, flows, and history. The main workspace displays a collection named "SachinCollection / IntegrationTestingWithTestCase". A specific test case is selected, showing a GET request to "https://cat-fact.herokuapp.com/facts". The "Tests" tab is active, containing a code snippet in JavaScript:

```
137 "properties": {  
138   "status": {  
139     "type": "object",  
140     "properties": {  
141       "verified": {  
142         "type": "boolean"  
143       },  
144       "sentCount": {  
145         "type": "integer"  
146       }  
147     }  
148   }  
149 }  
150 }  
151 }  
152 }  
153 }  
154 }  
155 }  
156 }  
157 }  
158 }  
159 }  
160 }  
161 }  
162 }  
163 }  
164 }  
165 }  
166 }  
167 }  
168 }  
169 }  
170 }  
171 }  
172 }  
173 }  
174 }  
175 }  
176 }  
177 }  
178 }  
179 }  
180 }  
181 }  
182 }  
183 }  
184 }  
185 }  
186 }  
187 }  
188 }  
189 }  
190 }  
191 }  
192 }  
193 }  
194 }  
195 }  
196 }  
197 }  
198 }  
199 }  
200 }  
201 }  
202 }  
203 }  
204 }  
205 }  
206 }  
207 }  
208 }  
209 }  
210 }  
211 }  
212 }  
213 }  
214 }  
215 }  
216 }  
217 }  
218 }  
219 }  
220 }  
221 }  
222 }  
223 }  
224 }  
225 }  
226 }  
227 }  
228 }  
229 }  
230 }  
231 }  
232 }  
233 }  
234 }  
235 }  
236 }  
237 }  
238 }  
239 }  
240 }  
241 }  
242 }  
243 }  
244 }  
245 }  
246 }  
247 }  
248 }  
249 }  
250 }  
251 }  
252 }  
253 }  
254 }  
255 }  
256 }  
257 }  
258 }  
259 }  
260 }  
261 }  
262 }  
263 }  
264 }  
265 }  
266 }  
267 }  
268 }  
269 }  
270 }  
271 }  
272 }  
273 }  
274 }  
275 }  
276 }  
277 }  
278 }  
279 }  
280 }  
281 }  
282 }  
283 }  
284 }  
285 }  
286 }  
287 }  
288 }  
289 }  
290 }  
291 }  
292 }  
293 }  
294 }  
295 }  
296 }  
297 }  
298 }  
299 }  
300 }  
301 }  
302 }  
303 }  
304 }  
305 }  
306 }  
307 }  
308 }  
309 }  
310 }  
311 }  
312 }  
313 }  
314 }  
315 }  
316 }  
317 }  
318 }  
319 }  
320 }  
321 }  
322 }  
323 }  
324 }  
325 }  
326 }  
327 }  
328 }  
329 }  
330 }  
331 }  
332 }  
333 }  
334 }  
335 }  
336 }  
337 }  
338 }  
339 }  
340 }  
341 }  
342 }  
343 }  
344 }  
345 }  
346 }  
347 }  
348 }  
349 }  
350 }  
351 }  
352 }  
353 }  
354 }  
355 }  
356 }  
357 }  
358 }  
359 }  
360 }  
361 }  
362 }  
363 }  
364 }  
365 }  
366 }  
367 }  
368 }  
369 }  
370 }  
371 }  
372 }  
373 }  
374 }  
375 }  
376 }  
377 }  
378 }  
379 }  
380 }  
381 }  
382 }  
383 }  
384 }  
385 }  
386 }  
387 }  
388 }  
389 }  
390 }  
391 }  
392 }  
393 }  
394 }  
395 }  
396 }  
397 }  
398 }  
399 }  
400 }  
401 }  
402 }  
403 }  
404 }  
405 }  
406 }  
407 }  
408 }  
409 }  
410 }  
411 }  
412 }  
413 }  
414 }  
415 }  
416 }  
417 }  
418 }  
419 }  
420 }  
421 }  
422 }  
423 }  
424 }  
425 }  
426 }  
427 }  
428 }  
429 }  
430 }  
431 }  
432 }  
433 }  
434 }  
435 }  
436 }  
437 }  
438 }  
439 }  
440 }  
441 }  
442 }  
443 }  
444 }  
445 }  
446 }  
447 }  
448 }  
449 }  
450 }  
451 }  
452 }  
453 }  
454 }  
455 }  
456 }  
457 }  
458 }  
459 }  
460 }  
461 }  
462 }  
463 }  
464 }  
465 }  
466 }  
467 }  
468 }  
469 }  
470 }  
471 }  
472 }  
473 }  
474 }  
475 }  
476 }  
477 }  
478 }  
479 }  
480 }  
481 }  
482 }  
483 }  
484 }  
485 }  
486 }  
487 }  
488 }  
489 }  
490 }  
491 }  
492 }  
493 }  
494 }  
495 }  
496 }  
497 }  
498 }  
499 }  
500 }  
501 }  
502 }  
503 }  
504 }  
505 }  
506 }  
507 }  
508 }  
509 }  
510 }  
511 }  
512 }  
513 }  
514 }  
515 }  
516 }  
517 }  
518 }  
519 }  
520 }  
521 }  
522 }  
523 }  
524 }  
525 }  
526 }  
527 }  
528 }  
529 }  
530 }  
531 }  
532 }  
533 }  
534 }  
535 }  
536 }  
537 }  
538 }  
539 }  
540 }  
541 }  
542 }  
543 }  
544 }  
545 }  
546 }  
547 }  
548 }  
549 }  
550 }  
551 }  
552 }  
553 }  
554 }  
555 }  
556 }  
557 }  
558 }  
559 }  
560 }  
561 }  
562 }  
563 }  
564 }  
565 }  
566 }  
567 }  
568 }  
569 }  
570 }  
571 }  
572 }  
573 }  
574 }  
575 }  
576 }  
577 }  
578 }  
579 }  
580 }  
581 }  
582 }  
583 }  
584 }  
585 }  
586 }  
587 }  
588 }  
589 }  
590 }  
591 }  
592 }  
593 }  
594 }  
595 }  
596 }  
597 }  
598 }  
599 }  
600 }  
601 }  
602 }  
603 }  
604 }  
605 }  
606 }  
607 }  
608 }  
609 }  
610 }  
611 }  
612 }  
613 }  
614 }  
615 }  
616 }  
617 }  
618 }  
619 }  
620 }  
621 }  
622 }  
623 }  
624 }  
625 }  
626 }  
627 }  
628 }  
629 }  
630 }  
631 }  
632 }  
633 }  
634 }  
635 }  
636 }  
637 }  
638 }  
639 }  
640 }  
641 }  
642 }  
643 }  
644 }  
645 }  
646 }  
647 }  
648 }  
649 }  
650 }  
651 }  
652 }  
653 }  
654 }  
655 }  
656 }  
657 }  
658 }  
659 }  
660 }  
661 }  
662 }  
663 }  
664 }  
665 }  
666 }  
667 }  
668 }  
669 }  
670 }  
671 }  
672 }  
673 }  
674 }  
675 }  
676 }  
677 }  
678 }  
679 }  
680 }  
681 }  
682 }  
683 }  
684 }  
685 }  
686 }  
687 }  
688 }  
689 }  
690 }  
691 }  
692 }  
693 }  
694 }  
695 }  
696 }  
697 }  
698 }  
699 }  
700 }  
701 }  
702 }  
703 }  
704 }  
705 }  
706 }  
707 }  
708 }  
709 }  
710 }  
711 }  
712 }  
713 }  
714 }  
715 }  
716 }  
717 }  
718 }  
719 }  
720 }  
721 }  
722 }  
723 }  
724 }  
725 }  
726 }  
727 }  
728 }  
729 }  
730 }  
731 }  
732 }  
733 }  
734 }  
735 }  
736 }  
737 }  
738 }  
739 }  
740 }  
741 }  
742 }  
743 }  
744 }  
745 }  
746 }  
747 }  
748 }  
749 }  
750 }  
751 }  
752 }  
753 }  
754 }  
755 }  
756 }  
757 }  
758 }  
759 }  
760 }  
761 }  
762 }  
763 }  
764 }  
765 }  
766 }  
767 }  
768 }  
769 }  
770 }  
771 }  
772 }  
773 }  
774 }  
775 }  
776 }  
777 }  
778 }  
779 }  
780 }  
781 }  
782 }  
783 }  
784 }  
785 }  
786 }  
787 }  
788 }  
789 }  
790 }  
791 }  
792 }  
793 }  
794 }  
795 }  
796 }  
797 }  
798 }  
799 }  
800 }  
801 }  
802 }  
803 }  
804 }  
805 }  
806 }  
807 }  
808 }  
809 }  
810 }  
811 }  
812 }  
813 }  
814 }  
815 }  
816 }  
817 }  
818 }  
819 }  
820 }  
821 }  
822 }  
823 }  
824 }  
825 }  
826 }  
827 }  
828 }  
829 }  
830 }  
831 }  
832 }  
833 }  
834 }  
835 }  
836 }  
837 }  
838 }  
839 }  
840 }  
841 }  
842 }  
843 }  
844 }  
845 }  
846 }  
847 }  
848 }  
849 }  
850 }  
851 }  
852 }  
853 }  
854 }  
855 }  
856 }  
857 }  
858 }  
859 }  
860 }  
861 }  
862 }  
863 }  
864 }  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }  
871 }  
872 }  
873 }  
874 }  
875 }  
876 }  
877 }  
878 }  
879 }  
880 }  
881 }  
882 }  
883 }  
884 }  
885 }  
886 }  
887 }  
888 }  
889 }  
890 }  
891 }  
892 }  
893 }  
894 }  
895 }  
896 }  
897 }  
898 }  
899 }  
900 }  
901 }  
902 }  
903 }  
904 }  
905 }  
906 }  
907 }  
908 }  
909 }  
910 }  
911 }  
912 }  
913 }  
914 }  
915 }  
916 }  
917 }  
918 }  
919 }  
920 }  
921 }  
922 }  
923 }  
924 }  
925 }  
926 }  
927 }  
928 }  
929 }  
930 }  
931 }  
932 }  
933 }  
934 }  
935 }  
936 }  
937 }  
938 }  
939 }  
940 }  
941 }  
942 }  
943 }  
944 }  
945 }  
946 }  
947 }  
948 }  
949 }  
950 }  
951 }  
952 }  
953 }  
954 }  
955 }  
956 }  
957 }  
958 }  
959 }  
960 }  
961 }  
962 }  
963 }  
964 }  
965 }  
966 }  
967 }  
968 }  
969 }  
970 }  
971 }  
972 }  
973 }  
974 }  
975 }  
976 }  
977 }  
978 }  
979 }  
980 }  
981 }  
982 }  
983 }  
984 }  
985 }  
986 }  
987 }  
988 }  
989 }  
990 }  
991 }  
992 }  
993 }  
994 }  
995 }  
996 }  
997 }  
998 }  
999 }  
999 }
```

Click only last integration testing test case

The screenshot shows the Postman application interface. On the left, the sidebar lists collections, APIs, environments, mock servers, monitors, flows, and history. The main workspace displays a collection named "SachinCollection". In the center, the "RUN ORDER" section shows a list of requests:

- GET New Request
- GET https://reqres.in/api/users/
- POST https://reqres.in/api/users
- GET JsonServerLocalAPI
- POST JsonServerLocalAPIPost
- PATCH JsonServerLocalAPIPatch
- PUT JsonServerLocalAPIPut
- GET JsonServerLocalAPIDelete
- GET IntegrationTestingWithTestCase

The last item, "GET IntegrationTestingWithTestCase", has a checked checkbox next to it. To the right of the list, there are settings for "Iterations" (set to 1), "Delay" (set to 0 ms), and "Data" (with a "Select File" button). Below these settings is a "Run SachinCollection" button. At the bottom of the workspace, it says "No logs yet" and "Send a request to view its details in the console." The bottom bar includes a search bar, pinned icons for various applications, and system status indicators.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

SachinCollection No Environment, just now

All Tests Passed (2) Failed (0)

Iteration 1

GET IntegrationTestingWithTestCase https://cat-fact.herokuapp.com/facts /IntegrationTestingWithTestCase 200 OK 1217 ms 2.138 KB

Pass Status code is 200

Pass Sachin Sirohi Test Case

Find and Replace Console

All Logs Clear

GET https://cat-fact.herokuapp.com/facts 200 1217 ms

Capture requests and cookies Bootcamp Runner Trash

22°C Haze ENG 19:15 20-02-2022

Add environment variable

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

SachinCollection [CONFLICT] [CONFLICT] [CONFLICT] [CONFLICT] [CONFLICT] [CONFLICT] [CONFLICT] [CONFLICT]

Local Variables / Creat a Person

POST Enter request URL

Params Authorization Headers (8) Body Pre-request Script

Query Params

KEY	VAL
Key	Value

Environment

No active Environment

An environment is a set of variables that allow you to switch the context of your requests.

Globals

No global variables

Global variables are a set of variables that are always available in a workspace.

Use variables to reuse values and protect sensitive data

Store sensitive data in variable type secret to keep its values masked on the screen. Learn more about variable type

Work with the current value of a variable to prevent sharing sensitive values with your team. Learn more about variable values

Find and Replace Console

GET https://cat-fact.herokuapp.com/facts 200 1217 ms

Capture requests and cookies Bootcamp Runner Trash

22°C Haze ENG 19:45 20-02-2022

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

Invite Upgrade

My Workspace

Collections

APIs Environments Mock Servers Monitors Flows History

sachinvariable

VARIABLE	TYPE	INITIAL VALUE	CURRENT VALUE	Persist All	Reset All
url	default	localhost:3000	localhost:3000		
Add a new variable					

Use variables to reuse values and protect sensitive data

Store sensitive data in variable type secret to keep its values masked on the screen. Learn more about variable type

Work with the current value of a variable to prevent sharing sensitive values with your team. Learn more about variable values

Find and Replace Console

All Logs Clear

GET https://cat-fact.herokuapp.com/facts 200 | 1217 ms



Set environment variable into put request type http://{{ select environment, environment priority is high as compare to normal variable }}

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

Invite Upgrade

My Workspace

Collections

APIs Environments Mock Servers Monitors Flows History

Local Variables / Create a Person

POST http://{{

Params	Authorization	Query Params	KEY	Cookies
GUID	Stimestamp	SisoTimestamp	RandomInt	RandomPhoneNum1
				RandomPhoneNum1

Send

Response

All Logs Clear

GET https://cat-fact.herokuapp.com/facts 200 | 1217 ms



If localhost path is not set set into environment variable ->

C:\Program Files\nodejs

Open cmd-> node -v

Open cmd-> json-server --watch db.json

Go to db.json file because persons json data there

```
C:\Windows\system32\cmd.exe - "node" "C:\Users\Sachin sirohi\AppData\Roaming\npm\node_mod
Microsoft Windows [Version 10.0.19043.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sachin sirohi>d:

D:\>cd D:\Postman API

D:\Postman API>json-server --watch db.json

\{^_}/ hi!

Loading db.json
Done

Resources
http://localhost:3000/persons

Home
http://localhost:3000

Type s + enter at any time to create a snapshot of the database
Watching...
```

The screenshot shows the Postman application interface. On the left, the sidebar includes 'My Workspace', 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. The main area displays a 'Local Variables / Creat a Person' section with a 'POST' request to 'http://{{url}}'. The 'Params' tab is selected, showing a single parameter 'Key' with 'Value'. Below the request, the 'Body' tab is selected, showing a JSON payload:

```
1 { "firstName": "Amit",  
2   "lastName": "Sirohi",  
3   "age": 20,  
4   "id": 20,  
5   "address": "Meerut",  
6   "phoneNumbers": "123-456-78910"  
7 }
```

The response status is 'Status: 404 Not Found' with a 'Time: 20 ms' and a 'Size: 426 B'. The bottom status bar shows 'Capture requests and cookies', 'Bootcamp', 'Runner', 'Trash', 'ADI Testing with Postman', '22°C Haze', 'ENG', and the date '20-02-2022'.

Is should be unique

The screenshot shows the Postman application interface. The setup is identical to the previous one, but the response has changed. The 'Body' tab now shows a successful response with a status of 'Status: 201 Created' and a 'Time: 57 ms'. The JSON response body is identical to the one sent:

```
1 { "firstName": "Amit",  
2   "lastName": "Sirohi",  
3   "age": 20,  
4   "id": 20,  
5   "address": "Meerut",  
6   "phoneNumbers": "123-456-78910"  
7 }
```

The bottom status bar shows 'Capture requests and cookies', 'Bootcamp', 'Runner', 'Trash', 'ADI Testing with Postman', '22°C Haze', 'ENG', and the date '20-02-2022'.

You can check of json file google chrome also

```

[{"firstName": "four", "lastName": "ji", "age": 26, "address": "Chicago,USA", "phoneNumbers": "123-456-78910", "id": 14}, {"firstName": "four", "lastName": "ji", "age": 26, "address": "Chicago,USA", "phoneNumbers": "123-456-78910", "id": 15}, {"firstName": "Lucky", "lastName": "ji", "age": 26, "address": "Chicago,USA", "phoneNumbers": "123-456-78910", "id": 16}, {"firstName": "Sachin", "lastName": "Sirohi", "age": 30, "id": 17, "address": "Meerut", "phoneNumbers": "123-456-78910"}, {"firstName": "Amit", "lastName": "Sirohi", "age": 20, "id": 18, "address": "Meerut", "phoneNumbers": "123-456-78910"}, {"firstName": "Amit", "lastName": "Sirohi", "age": 20, "id": 19, "address": "Meerut", "phoneNumbers": "123-456-78910"}]

```

Set test case of local and environment variable because we don't want to access id value other bcz it can be common.

Click on response body: json value check

Local Variables / Create a Person

POST http://([url])/persons

Tests

```

1 pm.test("Your test name", function () {
2     var jsonData = pm.response.json();
3     pm.expect(jsonData.value).to.eql(100);
4 });

```

Body

```

1 {
2     "firstName": "Amit",
3     "lastName": "Sirohi",
4     "age": 20,
5     "id": 19,
6     "address": "Meerut",
7     "phoneNumbers": "123-456-78910"
8 }

```

Click on set variable id should be unique

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

Local Variables / Creat a Person

POST http://{{url}}/persons

Params Authorization Headers (9) Body Pre-request Script

```
1 pm.test("Your test name", function () {  
2     var jsonData = pm.response.json();  
3     pm.environment.set("id", jsonData.id);  
4 });
```

sachinvariable

VARIABLE	INITIAL VALUE	CURRENT VALUE
url	localhost:3000	localhost:3000
id		21

Globals

No global variables

Global variables are a set of variables that are always available in a workspace.

① Use variables to reuse values and protect sensitive data

Store sensitive data in variable type secret to keep its values masked on the screen. [Learn more about variable type](#)

Work with the current value of a variable to prevent sharing sensitive values with your team. [Learn more about variable values](#)

Body Cookies Headers (15) Test Results (1/1)

Pretty Raw Preview Visualize JSON

```
1 {"firstName": "Amit",  
2 "lastName": "Sirohi",  
3 "age": 20,  
4 "id": 21,  
5 "address": "Meerut",  
6 "phoneNumbers": "123-456-78910"
```

Find and Replace Console

Type here to search

Capture requests and cookies Bootcamp Runner Trash

22°C Haze ENG 21:08 20-02-2022

Get value by id

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

Local Variables / getsingleperson

GET http://localhost:3000/persons/21

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

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (13) Test Results

Pretty Raw Preview Visualize JSON

```
1 {"firstName": "Amit",  
2 "lastName": "Sirohi",  
3 "age": 20,  
4 "id": 21,  
5 "address": "Meerut",  
6 "phoneNumbers": "123-456-78910"
```

Save Send Cookies

Status: 200 OK Time: 17 ms Size: 528 B Save Response

Find and Replace Console

Type here to search

Capture requests and cookies Bootcamp Runner Trash

22°C Haze ENG 21:10 20-02-2022

Get value by environment variable id

The screenshot shows the Postman application interface. On the left, the sidebar includes sections for Home, Workspaces, API Network, Reports, and Explore. The 'My Workspace' section is active, showing collections like 'Local Variables' and 'SachinCollection'. Under 'SachinCollection', there are several requests: 'Create a Person' (POST), 'getsingleperson' (GET), 'New Request' (GET), and others for JSON Server Local API (POST, PATCH, PUT, DELETE) and an integration test case ('IntegrationTestingWithTestCase').

The main workspace displays a 'Local Variables / getsingleperson' collection. A specific GET request is selected, which details the method as 'GET' and the URL as 'http://localhost:3000/persons/({id})'. The 'Description' panel states: 'This is the request meta data which shows up in Postman documentation. These are not sent with your HTTP request.' The 'Params' tab is selected, showing a table for 'Query Params' with columns for KEY, VALUE, DESCRIPTION, and Bulk Edit.

Below the request details, the 'Body' tab is active, showing a JSON response with the following data:

```
1  {
2      "firstName": "Amit",
3      "lastName": "Sirohi",
4      "age": 20,
5      "id": 21,
6      "address": "Meerut",
7      "phoneNumbers": "123-456-78910"
8 }
```

The status bar at the bottom indicates: Status: 200 OK | Time: 32 ms | Size: 528 B | Save Response. The bottom navigation bar includes icons for Find and Replace, Console, Capture requests and cookies, Bootcamp, Runner, Trash, and a search bar.