



GOVERNMENT OF INDIA
MINISTRY OF STATISTICS AND
PROGRAMME IMPLEMENTATION

API USER MANUAL FOR CONSUMER PRICE INDEX (CPI)

Disclaimer

This is an API Specifications created to facilitate the integraton of various Ministries/ Departments to NMDS Repository through APIs and manage Metadata in Standard Format. However, utmost care was taken to ensure all steps are explained to make it easier to understand the metadata management process, if any issues are found, same may be communicated with ASPD, MoSPI, and next version shall be released after incorporating the feedbacks.

Document History

Status	Version	Author	Change description	Date
Initial	Rev 1	DIID	Initial Document	22.10.2025
Approved	Rev 2	DIID	Approved Document	07.11.2025

Abbreviations

MoSPI – Ministry of Statistics and Programme Implementation

DIID – Data Informatics and Innovation Division

ASPD – Administrative Statistics and Policy Division

MWP – Metadata Web Portal

API – Application Programming Interface

JSON – JavaScript Object Notation

CPI – Consumer Price Index

UI – User Interface

XLS – Excel Spreadsheet Format

CSV – Comma Separated Values

URL – Uniform Resource Locator

GitHub – Code Repository Platform

Table of Contents

Disclaimer	2
Document History	3
Abbreviations	4
1 Introduction	6
2 Purpose of Document	7
3 User Registration	8
3.1 Using Swagger	8
3.1.1 Signup URL Access	8
3.1.2 Signup Screen	8
3.1.3 Try It Out Option	9
3.1.4 Request Body Details	9
3.2 Using Postman	11
3.2.1 Download Postman	11
3.2.2 Postman Startup Menu	11
3.2.3 Signup Process in Postman	12
4 Login Process	14
4.1 Generate Access Token in Postman	14
5 Accessing Inflation and Index Data from Swagger	16
5.1 Swagger UI Steps	16
5.1.1 API Screen Access	16
5.1.2 Parameter Entry	17
5.1.3 Execution and Response	18
5.2 Accessing Data Through Postman	19
6 Accessing Item Level Inflation and Index Data	21
6.1 Swagger UI Steps	21
6.1.1 API Screen Access	21
6.1.2 Parameter Entry	22
6.1.3 Execution and Response	23
6.2 Postman Steps	24
7 Accessing the API through Python and Curl	25

1. Introduction

The CPI API platform provides structured access to Consumer Price Index datasets released by the Ministry of Statistics and Programme Implementation. The platform supports authenticated data retrieval and delivers CPI Index and Inflation records in standard formats.

The Administrative Statistics and Policy Division manages the platform and ensures that users receive secure and stable API services for CPI data. The platform offers Group, Subgroup and Item level CPI data from January 2011 onward. Users follow a simple registration and login process to obtain an access token required for full data access.

New users must register once with a unique email ID on the API portal. After registration, the system issues an access token valid for 30 minutes. This token allows users to fetch complete datasets. Without a token, only the first ten records are returned.

The platform provides two tools for users. Swagger gives domain users a view of parameters and response structures. Postman helps technical users test and execute API requests with custom parameters. Both tools support JSON and CSV responses.

Users can download metadata sheets in XLS format from the API homepage. These sheets contain the codes needed for parameters like Year, Month, State, Group, Subgroup and Item.

Developers can also run the APIs using Python or Curl scripts available in the official GitHub repository. For further support or integration details, users may contact the ASPD Division of MoSPI.

2. Purpose of Document

This document guides users who work with the CPI API platform. It explains the process for registration, authentication and data access through the APIs hosted by the Ministry of Statistics and Programme Implementation. The document describes each API used to fetch CPI Index and Inflation data in standard formats.

The Administrative Statistics and Policy Division maintains the platform and provides technical support for its usage. The document helps users understand the parameters, response structures and execution steps through Swagger, Postman and script-based methods. Users are responsible for using valid credentials, applying correct parameters and ensuring proper handling of CPI data downloaded through the APIs.

2. User Registration

2.1. Using Swagger, the steps for executing the APIs are mentioned below:

2.1.1. The users of the platform are required to sign up for using this API platform by accessing the **Base URL/api/users/users_signup**.

2.1.2. The below screen will open on accessing this URL

User Registration

[View MetaData](#) [View User Manual](#)

User Registration First time users can register here.

POST /api/users/users_signup

2.1.3. Click anywhere on the green tile shown above and then click on the “Try it out” button on the right as indicated below. The fields in the Request Body section will become editable.

The screenshot shows the "User Registration" API endpoint in a documentation tool. At the top, there's a green "POST" button and a URL field containing "/api/users/users_signup". Below this, there are sections for "Parameters" (which are empty) and "Request body" (which is marked as required). A "Try it out" button is located to the right of the request body section. Under the "Request body" section, there's a JSON example provided:

```
{  
  "username": "test@test.com",  
  "password": "Test@12345",  
  "organization": "Ministry of Statistics and Programme Implementation",  
  "purpose": "View/Download the Data",  
  "gender": "Male"  
}
```

2.1.4. Enter the details in the Request Body as below:

```
{  
  "username": "test@test.com",  
  "password": "Test@12345",  
  "organization": "Ministry of Statistics and Programme Implementation",  
  "purpose": "View/Download the Data",  
  "gender": "Male"  
}
```

2.1.4.1. A valid email is required in the username field. Only '@' special character is allowed in this field.

2.1.4.2. One email can be used for registration only once.

2.1.4.3. No special characters are allowed in the organization, purpose fields and gender.

2.1.4.4. Click on the Execute button. The API response code and description will be returned. response code of 200 indicates the execution was successful.

```
{
  "msg": "User registered successfully.",
  "statusCode": true,
  "response": {
    "id": 17,
    "gender": "Male",
    "username": "test7834@test.com",
    "organization": "Ministry of Statistics and Programme Implementation",
    "purpose": "View/Download the Data",
    "password": "VGVzdEAxMjM0NQ==",
    "role": "users",
    "createdBy": "",
    "isActive": "1",
    "updatedAt": "2024-04-15T21:50:14.098Z",
    "createdAt": "2024-04-15T21:50:14.098Z"
  }
}
```

2.2. Using Postman, the steps are mentioned below:

2.2.1. Download Postman from any web browser by visiting the link <https://www.postman.com/downloads/>

The screenshot shows the official Postman download page. At the top, there's a navigation bar with links for Product, Pricing, Enterprise, Resources and Support, and Public API Network. Below the navigation, a large orange button says "Windows 64-bit". To the right, there's a preview of the Postman interface showing an API collection named "Notion API" with several requests listed under "Databases". One request is highlighted: "GET https://api.notion.com/v1/databases/{id}/retrieve". The interface includes sections for Params, Headers, and Query Params.

2.2.2. After downloading Postman app, open the app sign in or use the “lightweight API Client” and you will see the following menu:

The screenshot shows the Postman application window. On the left, there's a sidebar with "My Workspace" containing collections, environments, and history. The main area shows an "Untitled Request" with a "GET" method. The "Params" tab is selected, showing a single parameter "Key" with value "Value". There are tabs for Authorization, Headers, Body, Pre-request Script, Tests, and Settings. A "Send" button is at the bottom right. A small cartoon character is in the bottom right corner of the interface.

3.2.3 Click the “+” button to add a request.

3.2.4 Expand the method button.

2.2.3. Use the signup API to register user for using the API platform, as described below.

2.2.3.1. Change the method to POST

2.2.3.2. Enter the **API BASE URL/api/users/usersignup** in the textbox

2.2.3.3. Select Body, raw radio button and select JSON from the dropdown where JSON is shown by default as highlighted in Yellow below

2.2.3.4. Enter the data for the signup user details as in step#2.1.4 but make sure the username is unique as shown below and click Send.

```

1 {
2   "username": "test789234@text.com",
3   "password": "Text12345",
4   "organization": "Ministry of Statistics and Programme Implementation",
5   "purpose": "View/Download the Data",
6   "gender": "Male"
7 }
  
```

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize JSON

Status: 200 OK Time: 819 ms Size: 837 B Save as example

2.2.3.5. A success message will be sent as the response of the API indicating user signup is successful as shown above.

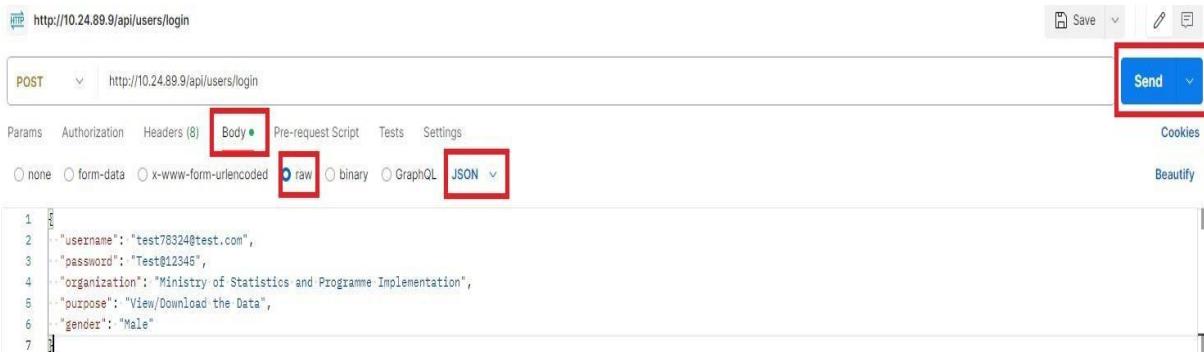
3. Login Process

- 3.1. The below steps describe the process to get the access token for the API through POSTMAN
- 3.2. Select “POST” and add the URL <http://api.mospi.gov/api/login>



- 3.3. Select Body, raw radio button and select JSON from the dropdown where JSON is shown by default as highlighted in Yellow below
- 3.4. Enter username and password as shown below using the same credentials used during signup process:

```
{  
  "username": "test78324@test.com",  
  "password": "Test@123434435"  
}
```



3.5. Click “Send”.



```
1 {  
2     "msg": "Login successful",  
3     "statusCode": true,  
4     "response": {  
5         "username": "test78324@test.com",  
6         "gender": "Male",  
7         "role": "users",  
8         "createdAt": "2024-04-15T21:54:41.932Z",  
9         "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJpc3MiOiJsb2dpbiIsInRlc3Q9dWlkIjoiMEdlbmRlcjI6Ik1hbGUiLCJyb2xlijojdXNlcnMilC3pZC16MTgsImlhCI6MTcxMzIxODY4MywiZXhwIjoxNzEzMjIwNDgzfQ.E-aTxU3BGb_i3ESx8JfjtX7lZv8ZAyb070dRuGbezy"}  
10    }  
11 }
```

- 3.6. Copy the “Token” value by clicking the “small double square” icon seen in the snapshot above.
- 3.7. Open a notepad and paste the value here. Copy only the value from the “token” element without the Quotes.
- 3.8. Use this token in the next sections for fetching the complete data **from Postman only** and NOT Swagger.

4. Accessing Inflation and Index data from Swagger

- 4.1. **Swagger UI:** Following are the steps for the API execution through the Swagger UI tool:
 - 4.1.1. Access the API **Base URL/cpi** on the web browser. The following page will load.



The screenshot shows the Swagger UI interface for the CPI API. At the top, it displays the title "Consumer Price Index (CPI)". Below the title, there is a note: "Swagger UI is used here as a visualization tool for API documentation and will return the first 10 records only." On the right side, there are two buttons: "View MetaData" and "View User Manual". The main content area is titled "Consumer Price Index and Inflation" and describes it as "All-India and State-wise Consumer Price Index and Inflation". It contains two expandable sections: "GET /api/getCPIIndex" and "GET /api/getItemIndex". The "GET /api/getCPIIndex" section is currently expanded, showing its details.

- 4.1.2. Click the tile in BLUE as shown above to expand the API **/api/cpi/getCPIIndex**.

GET /api/cpi/getCPIIndex

Parameters

Name	Description
Base_year * required (query)	Available values : 2012, 2010 Default value : 2012 2012
Series * required (query)	Available values : Current_series, Back_series Default value : Current_series Current_series
Year string (query)	Enter the Year (format YYYY.Comma separated for multiple values) Year
Month string (query)	Enter the Month code (from 1-12.Comma separated for multiple values) Month
State_code string (query)	Enter the State code (from 1-36 & 99 for all india comma separated for multiple values) State_code
Group_code string (query)	Enter the Group code(Comma separated for multiple values) Group_code
Subgroup_code string (query)	Enter the Subgroup code(Comma separated for multiple values) Subgroup_code
Sector string (query)	Enter the Sector code (from 1-3.Comma separated for multiple values) Sector
Format * required string (query)	Select the Output format Available values : JSON, CSV Default value : JSON JSON

Try it out

4.1.3. Click “Try it out” to enter the following parameters of the API to query for the required data:

4.1.3.1. Base year: 2012,2010. Only a single value can be select for this parameter.

4.1.3.2. Series: Current series, Back series. Only a single value can be sent for this parameter.

4.1.3.3. Year: Comma separated multiple values in the Format YYYY is accepted in this parameter.

4.1.3.4. Month: Comma separated multiple values in the Format ‘1,4’ is accepted in this parameter. Refer the metadata sheet for the codes corresponding to the Month names.

4.1.3.5. State code: Comma separated multiple values Format ‘1,2’ etc, is accepted in this parameter. Refer the metadata sheet for the State names.

4.1.3.6. Group code: Comma separated multiple values in the Format ‘5,7’ etc. is accepted in this parameter. Refer the metadata sheet for the codes corresponding to the Group names.

4.1.3.7. Subgroup _code: Comma separated multiple values in the format ‘1.1.01, 3.1.01’ etc. is accepted in this parameter. Refer the metadata sheet for the codes corresponding to the Subgroup names.

4.1.3.8. Sector: Comma separated multiple values in the Format ‘1,2,3’ is accepted in this parameter. Refer the metadata sheet for the codes corresponding to the Subgroup names.

4.1.3.9. Format: CSV or JSON option can be mentioned for the data format returned from the API.

4.1.4. If no value is supplied for a particular parameter, then the API will not apply a filter for that parameter.

4.1.5. Click on ‘Execute’ for fetching the data with the applied parameter values.

4.1.6. The below is the response of the API in JSON and CSV formats. Click on the Download button to download the data in a file:

Server response

Code Details

200

Response body

```
{ "data": [ { "baseyear": "2010", "year": 2011, "month": "April", "state": "All India", "sector": "Combined", "group": "General", "subgroup": "", "index": "106.2", "inflation": null, "status": "F" }, { "baseyear": "2010", "year": 2011, "month": "April", "state": "All India", "sector": "Rural", "group": "General", "subgroup": "", "index": "107.5", "inflation": null, "status": "F" } ] }
```

 Download

Server response

Code Details

200

Response body

```
baseyear,year,month,state,sector,group,subgroup,index,inflation,status
2012,2013,April,All India,Urban,General,,105.7,,F
2012,2013,April,All India,Rural,General,,106.4,,F
2012,2013,April,All India,Combined,General,,106.1,,F
2012,2013,April,All India,Urban,Food and Beverages,Meat and Fish,113.4,,F
2012,2013,April,All India,Urban,Food and Beverages,Fruits,109.5,,F
2012,2013,April,All India,Urban,Food and Beverages,Milk and Products,104.7,,F
2012,2013,April,All India,Urban,Food and Beverages,Cereals and Products,114.6,,F
2012,2013,April,All India,Urban,Food and Beverages,"Prepared Meals, Snacks, Sweets etc.",110.6,,F
2012,2013,April,All India,Urban,Food and Beverages,Spices,103.5,,F
2012,2013,April,All India,Urban,Food and Beverages,Oils and Fats,102.1,,F
2012,2013,April,All India,Urban,Food and Beverages,Sugar and Confectionery,102.0,,F
2012,2013,April,All India,Urban,Food and Beverages,Pulses and Products,104.6,,F
2012,2013,April,All India,Urban,Food and Beverages,Vegetables,109.7,,F
2012,2013,April,All India,Urban,Food and Beverages,,108.8,,F
2012,2013,April,All India,Urban,Food and Beverages,Egg,106.0,,F
2012,2013,April,All India,Urban,Food and Beverages,Non-alcoholic Beverages,108.2,,F
2012,2013,April,All India,Combined,Food and Beverages,Fruits,108.9,,F
2012,2013,April,All India,Combined,Food and Beverages,Oils and Fats,104.4,,F
2012,2013,April,All India,Rural,Food and Beverages,Sugar and Confectionery,104.2,,F
2012,2013,April,All India,Combined,Food and Beverages,Milk and Products,105.7,,F
```

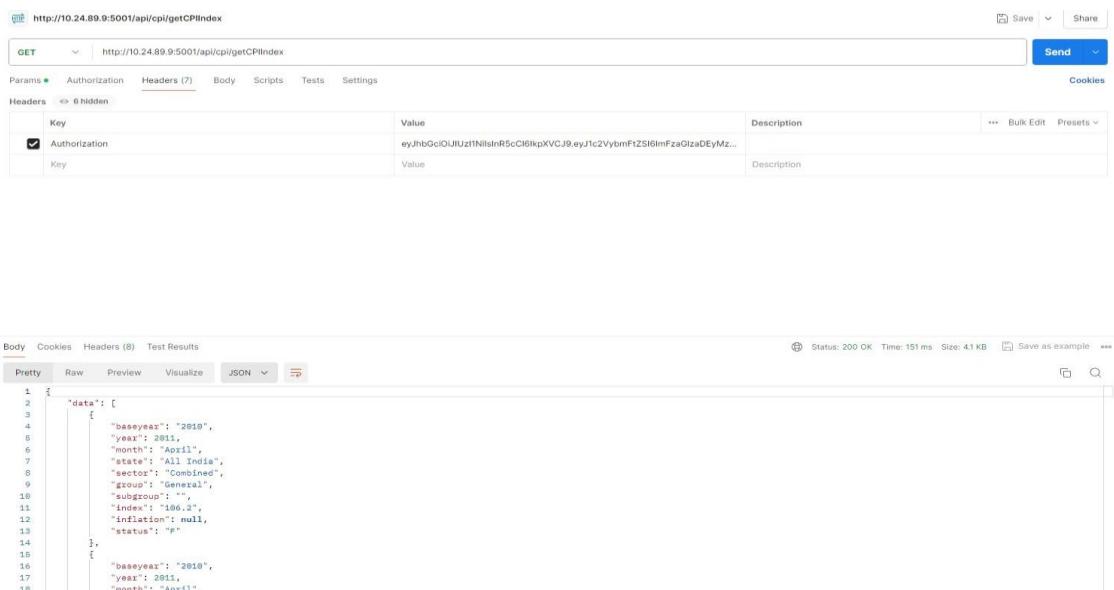
 Download

4.2. Postman: Following are the steps for the API execution through Postman tool:

4.2.1. The below steps describe the process to get the access token for the API through POSTMAN

4.2.1.1. Select “GET”, enter the API **Base URL/api/cpi/getCPIIndex** in the URL field.

4.2.1.2. Fill the Authorization access token (refer Section 3.4 above to generate the token) and add it in the Headers tab as shown below and click on Send:



The screenshot shows the Postman interface with a GET request to `http://10.24.89.9:5001/api/cpi/getCPIIndex`. The Headers tab has an `Authorization` key set to a long token value. The Body tab displays the JSON response from the API, which contains two data objects with fields like baseyear, year, month, state, sector, group, subgroup, index, inflation, and status.

```
1 {
2   "data": [
3     {
4       "baseyear": "2010",
5       "year": 2011,
6       "month": "April",
7       "state": "All India",
8       "sector": "Combined",
9       "group": "General",
10      "subgroup": "",
11      "index": "106.2",
12      "inflation": null,
13      "status": "F"
14    },
15    {
16      "baseyear": "2010",
17      "year": 2011,
18      "month": "April",
19    }
20  ]
21 }
```

4.2.1.3. If no parameters are given then the complete data is fetched for the API as shown in the snapshot above.

4.2.1.4. Now enter the required parameters in the Params tab along with Authorization token under Header's tab and click Send as per the below screenshot. The data as per the query parameters is fetched in the JSON/CSV format.

The screenshot shows a REST API testing interface. At the top, the URL is set to `http://10.24.89.9:5001/api/cpi/getCPIIndex?Base_year=2010&Series=Current_series&Year=2013`. In the 'Params' tab, four parameters are selected: 'Key', 'Base_year' (value: 2010), 'Series' (value: Current_series), and 'Year' (value: 2013). The 'Headers' tab contains an 'Authorization' header with the value 'Token 1234567890'. The 'Body' tab displays the JSON response:

```
1  {
2   "data": [
3     {
4       "baseyear": "2010",
5       "year": 2013,
6       "month": "April",
7       "state": "All India",
8       "sector": "All Sector",
9       "group": "General",
10      "subgroup": "",
11      "index": "128.1",
12      "inflation": "9.39",
13      "status": "Up"
14    },
15    {
16      "baseyear": "2010",
17      "year": 2013,
18      "month": "April"
```

5. Accessing Item level Inflation and Index data from Swagger

- 5.1. **Swagger UI:** Following are the steps for the API execution through the Swagger UI tool:
5.1.1. Access the API on the web browser: **Base URL/cpi**. The following page will load



- 5.1.2. Click the tile in BLUE as shown above to expand the API **/api/cpi/getItemIndex**

The screenshot shows the expanded API parameters for the '/api/cpi/getItemIndex' endpoint. The title at the top is 'GET /api/cpi/getItemIndex'. On the right side, there is a 'Cancel' button. The main area is titled 'Parameters' and contains four input fields:

- Base_year** (query): A dropdown menu set to '2010'.
- Year** (query): An input field containing 'Year'.
- Month** (query): An input field containing 'Month'.
- Item** (query): An input field containing 'Item'.

Below these fields is a 'Format' section with a dropdown menu set to 'JSON'.

- 5.1.2.1. Click "Try it out" to enter the following parameters of the API to query for the required data:
5.1.2.2. Base_year:2010,2012, Only single value can be select for this parameter.
5.1.2.3. Year: Comma separated multiple values in the Format YYYY is accepted in this parameter.
5.1.2.4. Month: Comma separated multiple values in the Format '1,4' is accepted in this parameter. Refer the metadata sheet for the codes corresponding to the Month names.
5.1.2.5. Item code: Comma Separated multiple values in the Format '1.1.01.1.1.02.X, 1.1.01.1.1.01.P' is accepted in this parameter. Refer to the metadata sheet for the codes corresponding to the Item codes.
5.1.2.6. Format: CSV or JSON option can be mentioned for the data format returned from the API.
5.1.2.7. Click on 'Execute' for fetching the data with the applied parameter values.
5.1.2.8. The below is the response of the API in JSON and CSV formats respectively. Click on the Download button to download the data in a file:

Server response

Code	Details
200	<p>Response body</p> <pre>{ "baseyear": "2010", "year": 2011, "month": "January", "item": "wheat/atta-pds", "index": "102.1", "inflation": null, "status": "F" }, { "baseyear": "2010", "year": 2011, "month": "January", "item": "wheat/atta-other sources", "index": "103.4", "inflation": null, "status": "F" }, { "baseyear": "2010", "year": 2011, "month": "January", "item": "Maida" }</pre> <p>Copy Download</p>

Server response

Code	Details
200	<p>Response body</p> <pre>baseyear,year,month,item,index,inflation,status 2012,2014,January,Rice - PDS,90.5,,F 2012,2014,January,Rice - Other Sources,123.7,,F 2012,2014,January,Wheat - PDS,106.0,,F 2012,2014,January,Muri,117.5,,F 2012,2014,January,Other Rice Products,118.9,,F 2012,2014,January,Wheat/ Atta - PDS,89.8,,F 2012,2014,January,Wheat/ Atta - Other Sources,122.7,,F 2012,2014,January,Maida,120.9,,F 2012,2014,January,"Sugar (Refined)",118.4,,F 2012,2014,January,"Bread (Bakery)",111.3,,F 2012,2014,January,Dried (bakery),117.9,,F 2012,2014,January,"Biscuits, Chocolates, etc.",107.8,,F 2012,2014,January,Other Cereals,195.0,,F 2012,2014,January,Cereal Substitutes: Tapioca, etc.,141.5,,F 2012,2014,January,Jowar & its Products,106.0,,F 2012,2014,January,Bajra & its Products,123.5,,F 2012,2014,January,Ragi & its Products,117.5,,F 2012,2014,January,Small Millets & their Products,122.5,,F 2012,2014,January,Ragi & its Products,130.9,,F 2012,2014,January,Grinding Charges,117.9,,F</pre> <p>Copy Download</p>

5.2. Postman: Following are the steps for the API execution through the Postman tool:

5.2.1. Select “GET”, enter the API **Base URL/api/cpi/getItemIndex** in the URL field.

5.2.2. Fill the Authorization access token (refer Section 3.4 above for the steps to generate the token) and add it in the Headers tab shown below and click Send:

The screenshot shows the Postman interface with a GET request to `http://10.24.89.9:5001/api/cpi/getItemIndex?Base_year=2012&Format=JSON`. The Headers tab contains an `Authorization` header with a value of `eyJhbGciOiJUzI1NiIsInR5cCI6IkpXVCJ9.eyJtC2VybmlFZSI6ImFzaGlzaDeyMz...Value`. The Body tab displays the JSON response:

```

1   {
2     "data": [
3       {
4         "baseyear": "2012",
5         "year": 2014,
6         "month": "January",
7         "item": "Rice - PDS",
8         "index": "90.5",
9         "inflation": null,
10        "status": "F"
11      },
12      {
13        "baseyear": "2012",
14        "year": 2014,
15        "month": "January",
16        "item": "Rice - Other Sources",
17        "index": "123.7",
18        "inflation": null,
19      }
20    ]
}

```

5.2.3. If no parameters are given then the complete data is fetched for the API shown above.

5.2.4. Enter the required parameters in the Params tab along with the authorization token under Headers tab and click Send. The data as per the query parameters is fetched in the JSON/CSV format shown below.

http://10.24.89.9:5001/api/cpi/getItemIndex?Base_year=2012&Format=JSON

Save Share

GET http://10.24.89.9:5001/api/cpi/getItemIndex?Base_year=2012&Format=JSON

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
Base_year	2012	
Format	JSON	
Key	Value	Description

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 i "data": [
2   {
3     "baseyear": "2012",
4     "year": 2014,
5     "month": "January",
6     "item": "Rice - PDS",
7     "index": "98.5",
8     "inflation": null,
9     "status": "F"
10   },
11   {
12     "baseyear": "2012",
13     "year": 2014,
14     "month": "January",
15     "item": "Rice - Other Sources",
16     "index": "123.7",
17     "inflation": null,
18   }
]
```

Status: 200 OK Time: 100 ms Size: 2.73 KB Save as example

6. Accessing the API through Python and Curl

6.1. The Python and CURL scripts for executing the API is available in the below GitHub repo:
https://github.com/CCSPIDev/mospi_api_platform.git
