

Reading, Writing, and Parsing JSON Files in Python

Understanding & Fetching JSON Data



Ed Freitas

Software Developer // Automation Specialist

<https://edfreitas.me>

Overview

What to expect

Understanding JSON data

Fetch JSON data from URLs

Retrieve JSON data from web APIs



Reading, Writing, and Parsing JSON Files in Python

Version Check



Version Check

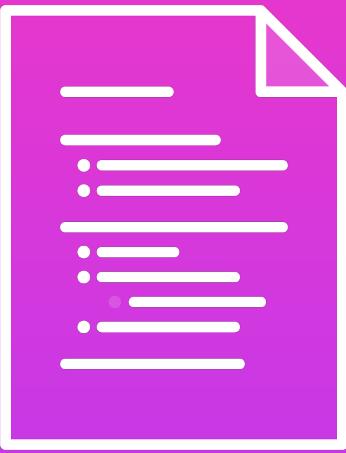


This course was created using:

- Python 3.11.2



Relevant Notes



Applicable also to:

- Any 3.X version of Python





What You Will Learn in This Course



Learning Topics

Fetch JSON Data from URLs

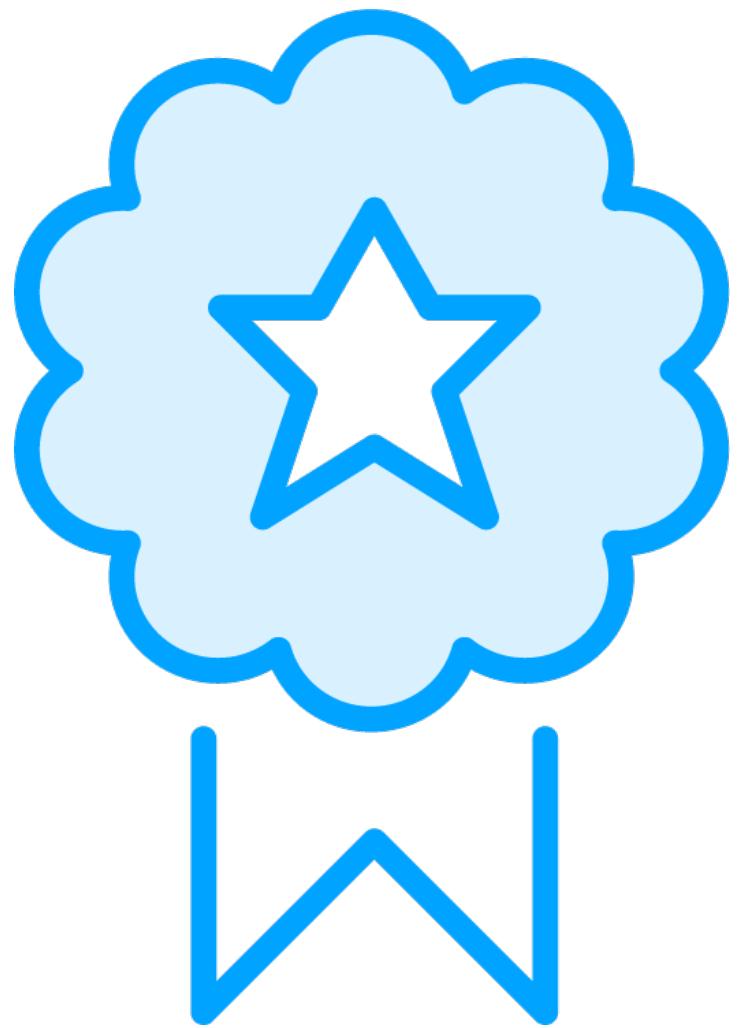
Read & Import JSON Files

Describe & Parse JSON Files

All Together



Learning Topics



Python
VS Code
JSON data
JSON files



| Scenario

Scenario Description

Wired Brain Coffee

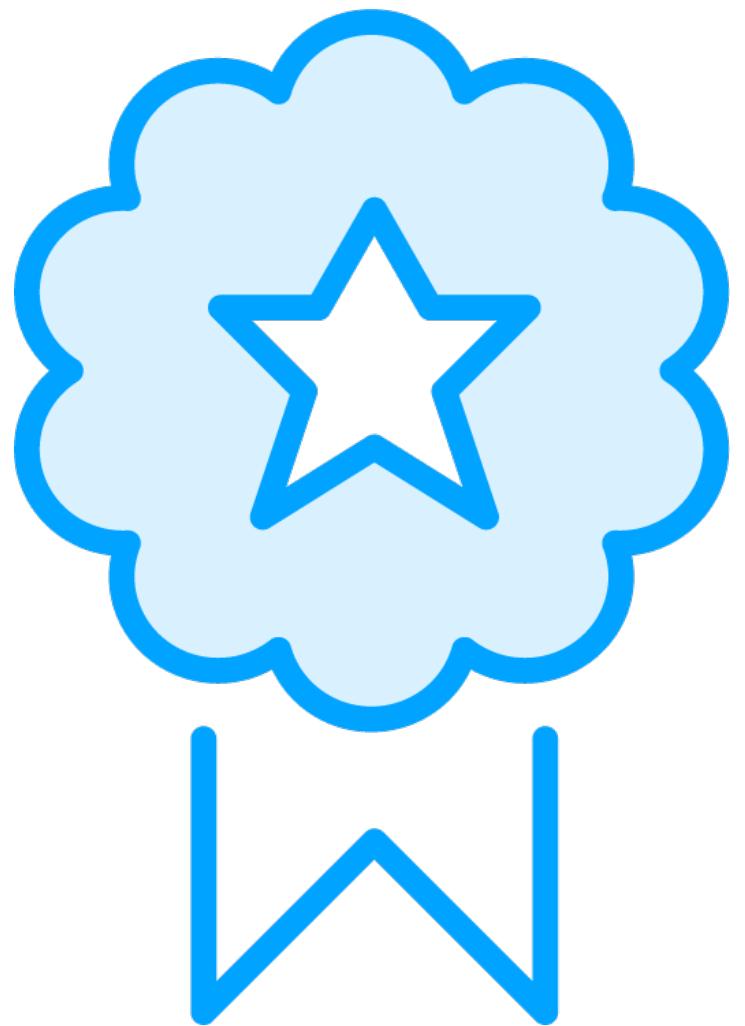
Python developer

Process JSON data from URLs

Process JSON data from files



Examples



Various examples

All-in-one example



Understanding JSON Data



JavaScript Object Notation

JSON (JavaScript Object Notation) is a text-based data format that is easy for people to read/write and easy for computers to parse/generate that uses key-value pairs to represent data.



JSON Sample

data.json

```
{  
  "sales": [  
    {  
      "id": 1,  
      "location": "London",  
      "date": "2021-01-01",  
      "sales": 100  
    },  
    {  
      "id": 2,  
      "location": "Paris",  
      "date": "2021-01-02",  
      "sales": 200  
    },  
  ]  
}
```



JSON Data Types

Strings

Numbers

Objects

Arrays

Booleans

Null



Nested Objects

data.json

```
{  
  "shop1" : {  
    "client" : "Juan del Amor",  
    "location" : "Spain",  
    "online" : true,  
    "amount" : 987  
  },  
  "shop2" : {  
    "client" : "Jane Smith",  
    "location" : "UK",  
    "online" : false,  
    "amount" : 432  
  },  
}
```



Nested Objects

data.json

```
{  
  "shop1" : {  
    "client" : {  
      "name" : "Juan del Amor",  
      "age" : 48  
    },  
    "location" : "Spain",  
    "online" : true,  
    "amount" : 987  
  },  
  "shop2" : {  
    "client" : {  
      "name" : "Jane Smith",  
      "age" : 32  
    },  
    "location" : "UK",  
    "online" : false,  
    "amount" : 432  
  },  
}
```



Nested Arrays

data.json

```
{  
    "first_name" : "Sammy",  
    "last_name" : "McAllister",  
    "country" : "Ireland",  
    "shops" : [  
        {  
            "city" : "Dublin",  
            "amount" : "899"  
        },  
        {  
            "city" : "Cork",  
            "amount" : "121"  
        }  
    ]  
}
```

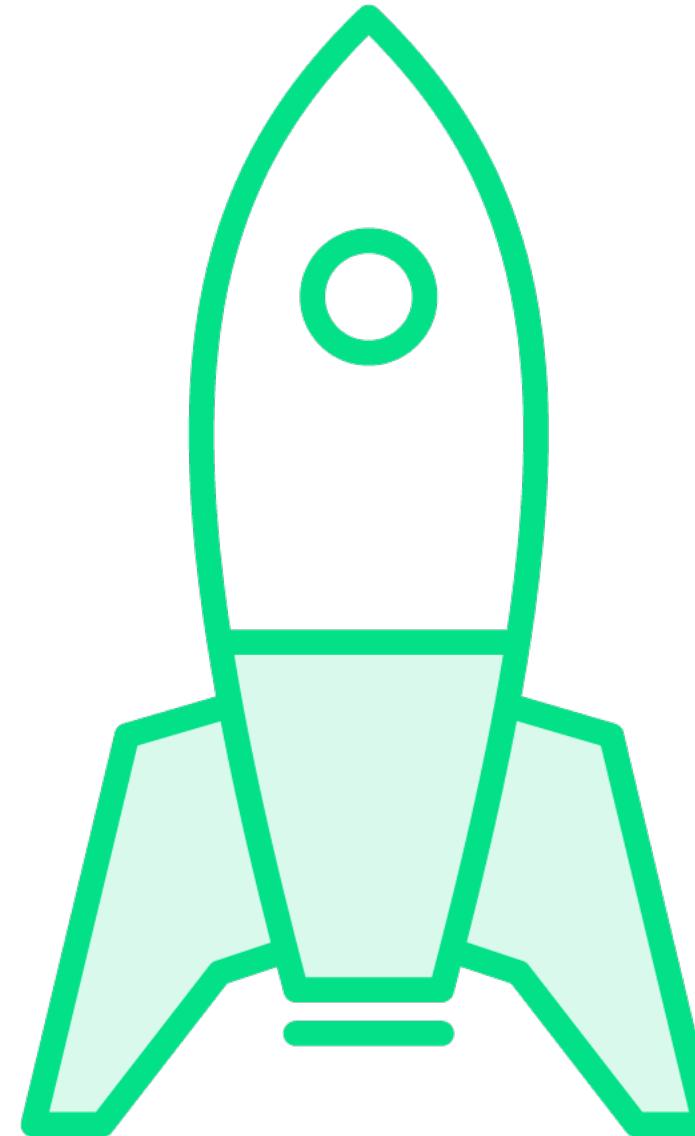


Combined Nesting

data.json

```
{  
    "first_name" : "Sammy",  
    "last_name" : "McAllister",  
    "shops" : [  
        {  
            "cities" : [  
                { "city" : "Dublin" },  
                { "city" : "Madrid" }  
            ],  
            "amount" : "899"  
        },  
        {  
            "cities" : [  
                { "city" : "Chicago" },  
                { "city" : "Miami" }  
            ],  
            "amount" : "121"  
        }  
    ]  
}
```



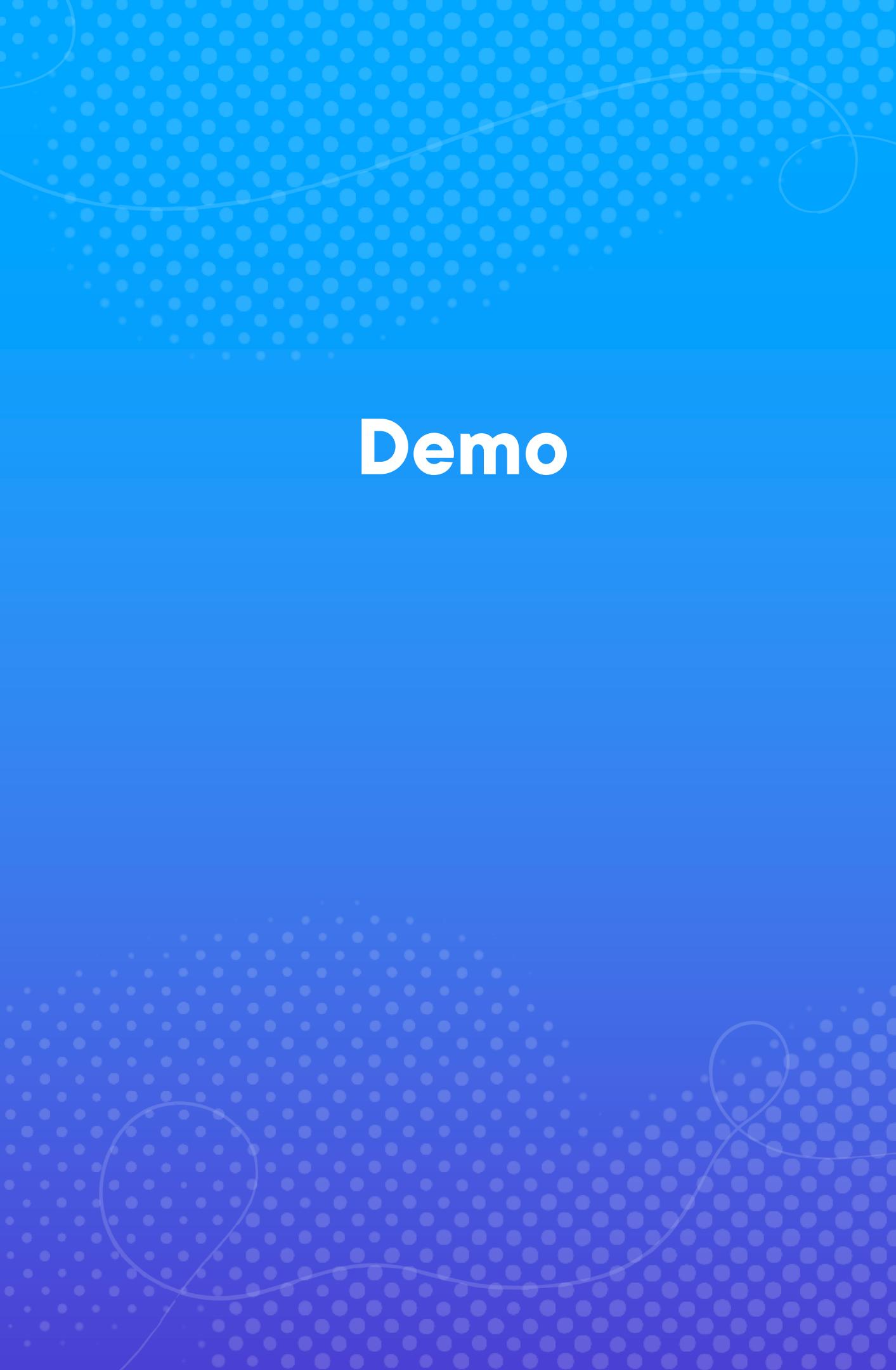


Nesting can go on and on...

JSON is a compact and flexible format

Best suits our requirements





Demo

Fetch JSON Data from URLs



Retrieving JSON from a URL is straightforward



Steps

Used the requests library

Send a get request to the URL

Containing the JSON data

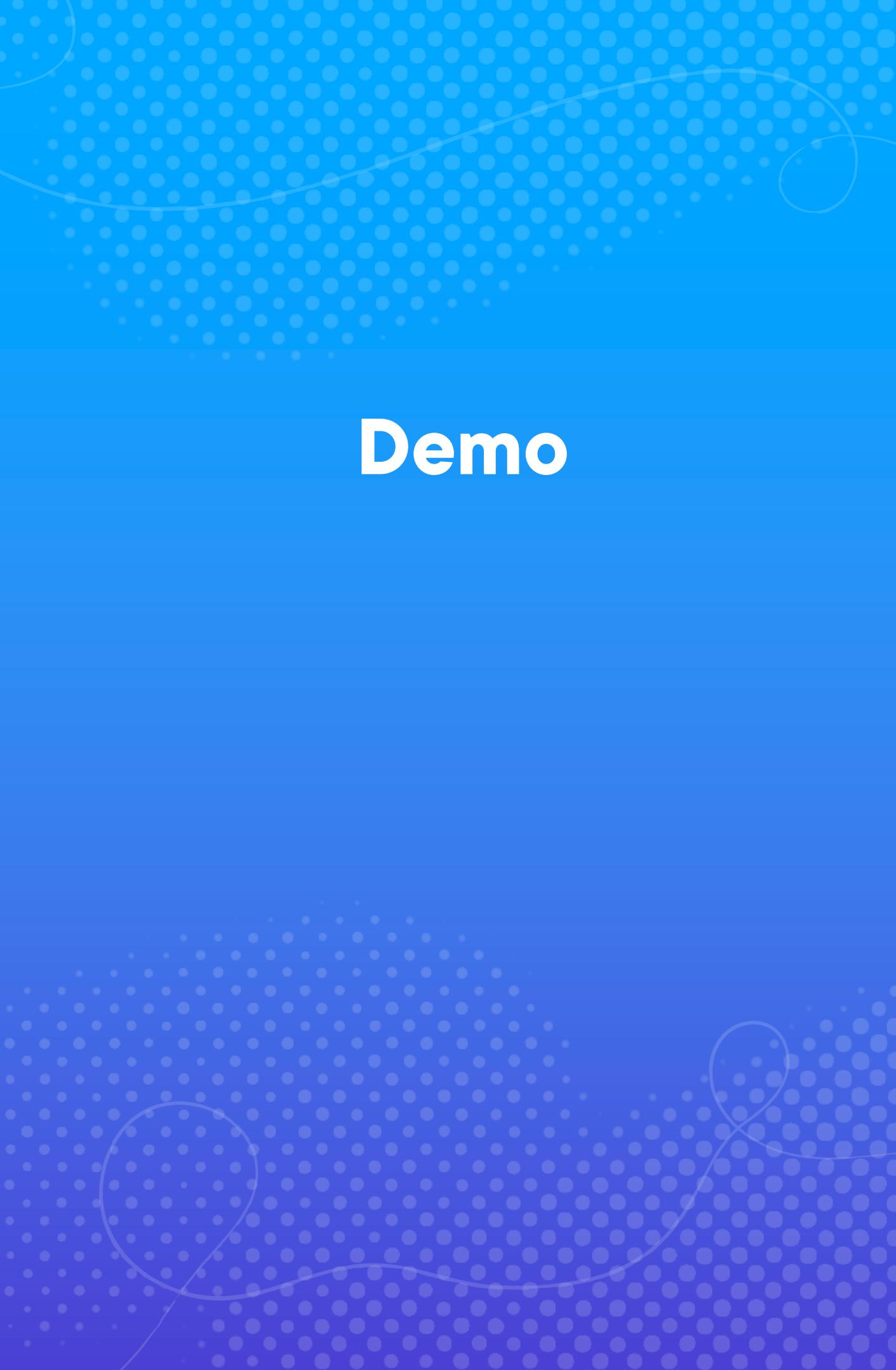


Steps

Checked if the
request was OK

Printed the results
obtained

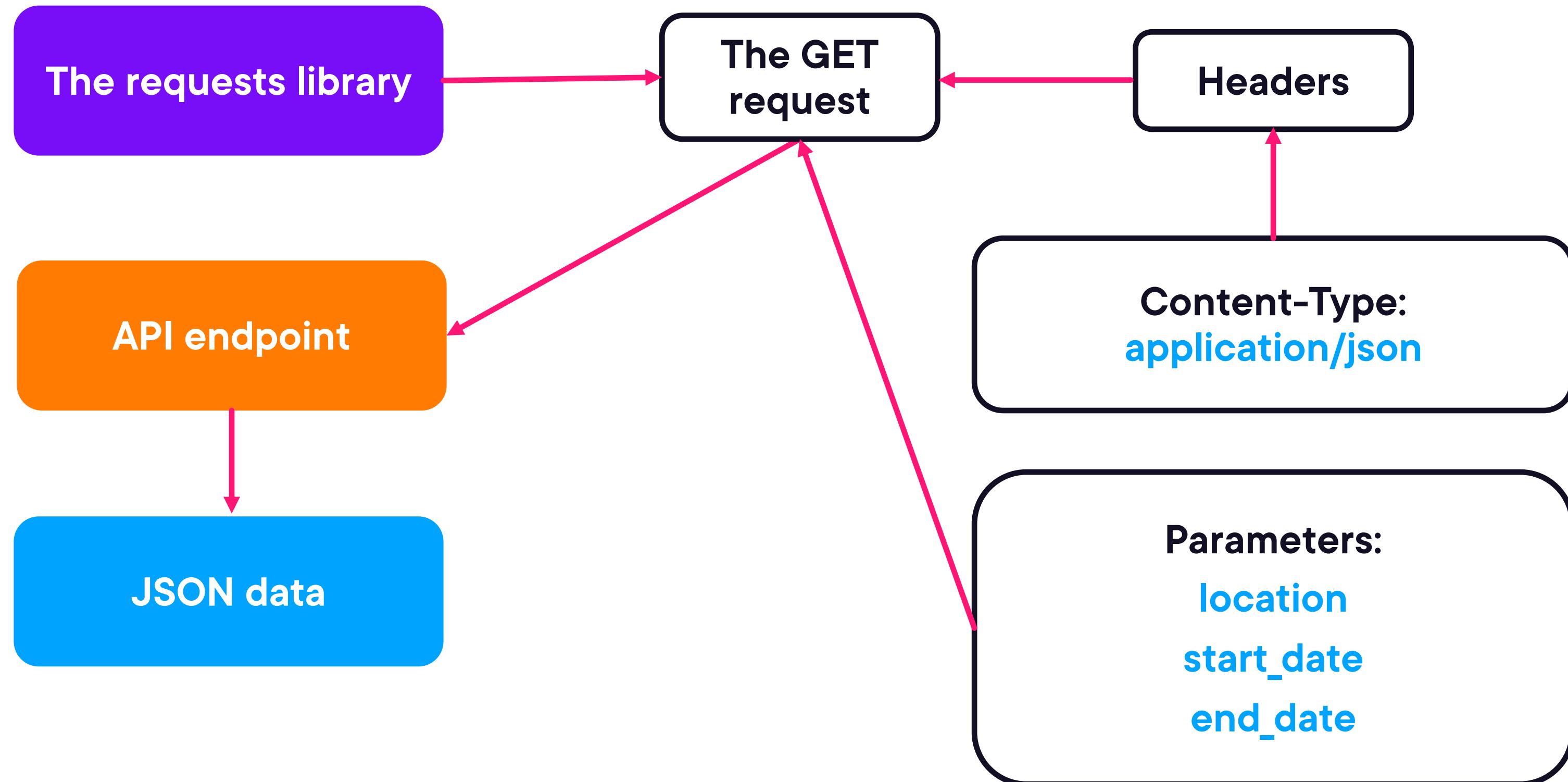




Demo

Retrieve JSON Data from web APIs





Sample Recap



Similar steps to fetching a URL

Headers

Parameters



{JSON} Placeholder

<https://jsonplaceholder.typicode.com>



Sample Limitations

We used headers
We used parameters
Not a proper web API
JSON data through a URL



Summary

What to expect

Understanding JSON data

Fetch JSON data from URLs

Retrieve JSON data from web APIs



Up Next:

Importing & Parsing JSON Files

