

{ api }

TheTestingAcademy

JSON Schema Validation

Postman



POSTMAN



Scrolltest.com



Let's Start ...

In this Tutorial...

- What is JSON ?
- What is JSON Schema ?
- Why JSON Schema Validation required?
- Basic JSON Schema example
- How to do JSON Schema Validation in Postman Step by step
- Tiny Validator for JSON Schema v4

What is JSON?



- JSON stands for JavaScript Object Notation
- JSON is a lightweight format for storing and transporting data
- JSON is often used when data is sent from a server to a web page
- JSON is "self-describing" and easy to understand

```
{  
  - data: {  
    id: 2,  
    email: "janet.weaver@reqres.in",  
    first_name: "Janet",  
    last_name: "Weaver",  
    avatar: "https://s3.amazonaws.com/uifaces/faces/twitter/josephstein/128.jpg"  
  }  
}
```

What JSON Schema POSTMAN

JSON Schema is a contract for your JSON document that defines the expected data types and format of each field in the response

It describes your existing data format with clear, human- and machine-readable documentation for complete structural validation, useful for automated testing and validating client-submitted data.

Why JSON Schema Validation required?

1. Using JSON Schema to construct a model of your API response makes it easier to validate your API is returning the data it should.
2. Monitor your API responses, ensuring they adhere to a specified format.
3. Get alerted when breaking changes occur.

Basic JSON Schema example

```
[  
  {  
    "firstName": "Pramod",  
    "lastName": "Dutta",  
    "age": 2  
  }  
]
```

Basic JSON Schema example

1. The top-level type key tells us the API returns an "array".
2. The items describes the shape of each array entry (each user).
3. The firstName and lastName fields are strings, while age is an integer.

Basic JSON Schema example

```
{ "$schema": "https://json-schema.org/draft-04/schema#",  
  "type": "array",  
  "items":  
    {  
      "type": "object",  
      "Properties":  
        {  
          "firstName": { "type": "string" },  
          "lastName": { "type": "string" },  
          "age": { "type": "integer" }  
        }  
      }  
    }  
}
```

How to do JSON Schema Validation in Postman

Step by step

1. Generate JSON Schema
2. <https://jsonschema.net/#/>
3. - > Object Assertions and check REQD Properties. We do this so that every element in the JSON becomes mandatory.
4. Number Assertions and check Use number, not integer for all numeric instances
5. Configure and Test using Postman

Tiny Validator for JSON Schema v



1. Use json-schema draft v4 to validate simple values and complex objects using a rich validation vocabulary
2. <https://github.com/geraintluff/tv4>
3. `var valid = tv4.validate(data, schema);`

Click to Subscribe



Scrolltest.com
TheTestingAcademy.com



Thanks for Watching...