



# Getting started

Getting started with Open Library API

## 1. Base URL

```
https://openlibrary.org
```

## 2. Authentication

- **Authentication Required:** No
- **Rate Limiting:** None for basic usage. For large-scale requests, contact [info@openlibrary.org](mailto:info@openlibrary.org)

## 3. List of APIs

1. Search
2. Author

## 4. Endpoints

### 4.1 Search

**Description:** Search books, authors, ISBN, category

**Endpoint:**

```
GET /search.json?q={query}
```

**Table 1: Parameters:**

Name	Type	Required	Description	Example
q	string	Yes	Search	q=harry+potter
page	int	No	Page number	q=sapiens&page=2

**Example requests:**

```
https://openlibrary.org/search.json?q=harry+potter
```

```
https://openlibrary.org/search.json?q=twain
```

```
https://openlibrary.org/search.json?q=science
```

```
https://openlibrary.org/search.json?q=9780385533225
```

```
https://openlibrary.org/search.json?q=harry+potter&page=1
```

**Example Response:**

```
{
  "numFound": 245,
  "start": 0,
  "docs": [
    {
      "title": "Harry Potter and the Philosopher's Stone",
      "author_name": ["J. K. Rowling"],
      "first_publish_year": 1997,
      "isbn": ["9780747532699"]
    }
  ]
}
```



```
]
}
```

## 4.2 Authors

**Description:** Search author names by using search api

**Endpoint:** GET /search/authors.json

**Table 1: Parameters:**

Name	Type	Required	Description	Example
q	string	Yes	Search(author)	q=twain

### Example requests:

```
https://openlibrary.org/search/authors.json?q=j%20k%20rowling
```

```
{
  numFound: 1,
  start: 0,
  numFoundExact: true,
  docs: [
    {
      key: "OL23919A",
      text: [...],
      type: "author",
      name: "J. K. Rowling",
      alternate_names: [...],
      birth_date: "31 July 1965",
      top_work: "Harry Potter and the Philosopher's Stone",
      work_count: 162,
      top_subjects: [...],
      _version_: 1702166143068799000
    }
  ]
}
```

## 5. Usage Guidelines

1. Respect rate limits by adding delayed between large batches of request
2. Attribute Open Library in your application

## 6. Error Handling

Status Code	Meaning
200	Success
404	Not Found
500	Server Error

## 7. Example Integration

### 7.1 JavaScript

```
fetch("https://openlibrary.org/search.json?q=harry+potter")
  .then(response => response.json())
  .then(data => console.log(data.docs))
  .catch(error => console.error(error));
```

### 7.2 Python



```
#Request library

import requests

url = "https://openlibrary.org/search.json?q=harry+potter"

response = requests.get(url)

if response.status_code == 200:
    data = response.json()
    print(data["docs"])
else:
    print("Error:", response.status_code)
```

### 7.3 Java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class OpenLibraryExample {
    public static void main(String[] args) {
        try {
            String urlString = "https://openlibrary.org/search.json?q=harry+potter";
            URL url = new URL(urlString);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("GET");

            BufferedReader in = new BufferedReader(
                new InputStreamReader(conn.getInputStream()));
            String inputLine;
            StringBuilder content = new StringBuilder();

            while ((inputLine = in.readLine()) != null) {
                content.append(inputLine);
            }

            in.close();
            conn.disconnect();

            System.out.println(content.toString());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

### 7.4 PHP

```
<?php
$url = "https://openlibrary.org/search.json?q=harry+potter";
$response = file_get_contents($url);

if ($response !== FALSE) {
    $data = json_decode($response, true);
    print_r($data["docs"]);
} else {
    echo "Error fetching data.";
}
?>
```

### 7.5 Node.js (Axios)



```
const axios = require("axios");

axios.get("https://openlibrary.org/search.json?q=harry+potter")
  .then(response => {
    console.log(response.data.docs);
  })
  .catch(error => {
    console.error(error);
  });
```

## 8. References

1. [Open Library API](#)