

Book Search Engine API Documentation

Base URL

<http://your-api-domain.com/api/v1/books>

Endpoints

1. Create a Book

Endpoint: **POST** /

Description: Creates a new book entry.

Request Body:

```
{
  "title": "string",
  "author": "string",
  "publishedYear": "number",
  "genre": "string"
}
```

Responses:

201 CREATED

```
{
  "success": true,
  "statusCode": 201,
  "message": "Book created successfully"
}
```

•

409 CONFLICT (If the book already exists)

```
{
  "success": false,
  "statusCode": 409,
  "message": "Book already exists"
}
```

-

400 BAD REQUEST (On failure)

```
{
  "success": false,
  "statusCode": 400,
  "message": "Book creation failed",
  "error": "Error message"
}
```

-

2. Get All Books

Endpoint: `GET /`

Description: Retrieves all books from the database.

Responses:

200 OK

```
{
  "success": true,
  "statusCode": 200,
  "message": "Books retrieved successfully",
  "data": [
    {
      "_id": "string",
      "title": "string",
      "author": "string",
      "publishedYear": "number",
      "genre": "string"
    }
  ]
}
```

-

400 BAD REQUEST (On failure)

```
{
  "success": false,
  "statusCode": 400,
  "message": "Failed to retrieve books",
}
```

```
"error": "Error message"
}
```

-

3. Search Book by Title

Endpoint: `GET /bookSearch?title={title}`

Description: Searches for books based on title.

Parameters:

- `title` (string, required): The title of the book to search for.

Responses:

200 OK

```
{
  "success": true,
  "statusCode": 200,
  "message": "Books retrieved successfully",
  "data": [
    {
      "_id": "string",
      "title": "string",
      "author": "string",
      "publishedYear": "number",
      "genre": "string"
    }
  ]
}
```

-

400 BAD REQUEST (On failure)

```
{
  "success": false,
  "statusCode": 400,
  "message": "Search failed",
  "error": "Error message"
}
```

-

