

Спецификация Open Api Swagger для <https://editor.swagger.io/>

openapi: "3.0.0"
info:
 version: "1.0.0"
 title: "Library Management API"
 description: "API for managing books and authors."

paths:
 /books:
 post:
 summary: "Add a new book"
 requestBody:
 required: true
 content:
 application/json:
 schema:
 \$ref: "#/components/schemas/Book"
 responses:
 "201":
 description: "Book successfully created."
 get:
 summary: "Get all books"
 parameters:
 - name: genre
 in: query
 description: "Filter by genre"
 required: false
 schema:
 type: string
 - name: page
 in: query
 description: "Page number for pagination"
 required: false
 schema:

type: integer
- name: limit
in: query
description: "Limit records per page"
required: false
schema:
type: integer
responses:
"200":
description: "Successful response"
content:
application/json:
schema:
type: array
items:
\$ref: "#/components/schemas/Book"

/books/{id}:
get:
summary: "Get a book by ID"
parameters:
- name: id
in: path
required: true
schema:
type: integer
responses:
"200":
description: "Successful response"
content:
application/json:
schema:
\$ref: "#/components/schemas/Book"
put:
summary: "Update a book by ID"
parameters:

- name: id
 - in: path
 - required: true
 - schema:
 - type: integer

requestBody:

- required: true
- content:
 - application/json:
 - schema:
 - \$ref: "#/components/schemas/Book"

responses:

- "200":
 - description: "Book successfully updated."

delete:

- summary: "Delete a book by ID"

parameters:

- name: id
 - in: path
 - required: true
 - schema:
 - type: integer

responses:

- "204":
 - description: "Book successfully deleted."

/authors:

post:

- summary: "Add a new author"

requestBody:

- required: true
- content:
 - application/json:
 - schema:
 - \$ref: "#/components/schemas/Author"

responses:

"201":
description: "Author successfully created."

get:

summary: "Get all authors"

responses:

"200":
description: "Successful response"
content:
application/json:
schema:
type: array
items:
\$ref: "#/components/schemas/Author"

/authors/{id}:

get:

summary: "Get an author by ID"

parameters:

- name: id
in: path
required: true
schema:
type: integer

responses:

"200":
description: "Successful response"
content:
application/json:
schema:
\$ref: "#/components/schemas/Author"

put:

summary: "Update an author by ID"

parameters:

- name: id
in: path
required: true

schema:
 type: integer
requestBody:
 required: true
 content:
 application/json:
 schema:
 \$ref: "#/components/schemas/Author"
responses:
 "200":
 description: "Author successfully updated."

delete:
 summary: "Delete an author by ID"
 parameters:
 - name: id
 in: path
 required: true
 schema:
 type: integer
 responses:
 "204":
 description: "Author successfully deleted."

components:
 schemas:
 Book:
 type: object
 properties:
 id:
 type: integer
 title:
 type: string
 genre:
 type: string
 year:
 type: integer

authorId:
 type: integer
authorBooksCount:
 type: integer

Author:
 type: object
properties:
 id:
 type: integer
 firstName:
 type: string
 lastName:
 type: string
 dob:
 type: string
 booksWritten:
 type: integer