Online Course Manager API Documentation

Base URL: http://localhost:3000/api

🛠 General Info

- **Content-Type:** application/json
- **CORS:** Enabled (You can request from localhost:5173 or any port).
- **Response Format:** All responses follow a standard structure known as JSend.

✅ Success Response Structure

JSON

```
{
  "status": "success",
  "results": 5, // (Optional) Only in GET all
  "data": { ... } // The actual data payload
}
```

❌ Error Response Structure

JSON

```
{
  "status": "error" | "fail",
  "message": "Error description here"
}
```

1️⃣ Categories (/category)

🟢 Get All Categories

- **Endpoint:** GET /category
- **Description:** Returns a list of all available categories.
- **Response:**

JSON

```
{
  "status": "success",
  "results": 3,
  "data": {
    "categories": [
      { "id": 1, "name": "Programming" },
      { "id": 2, "name": "Languages" }
    ]
  }
}
```

🟡 Create Category

- **Endpoint:** POST /category
- **Body (JSON):**

JSON

```
{
  "name": "Marketing" // Required (min 2 chars, max 50)
}
```

- **Response:** Returns the created category object.

🔵 Update Category

- **Endpoint:** PUT /category/:id
- **Description:** Updates the category name.
- **Body (JSON):**

JSON

```
{
  "name": "Digital Marketing"
}
```

🔴 Delete Category

- **Endpoint:** DELETE /category/:id
- **Response:** 200 OK with data: null.

2️⃣ Courses (/courses)

🟢 Get All Courses (Search, Filter, & Sort)

- **Endpoint:** GET /courses
- **Query Parameters (Optional):** You can combine them!
  - search: Search by **Title** OR **Instructor**.
    - Example: ?search=react
  - category: Filter by Category ID.
    - Example: ?category=1
  - sort: Sort by Price.
    - price: Low to High (Ascending).
    - -price: High to Low (Descending).
    - Example: ?sort=-price
- Full Example URL:

http://localhost:3000/api/courses?category=1&search=js&sort=price

- **Response:**

JSON

```
{
  "status": "success",
  "results": 2,
  "data": {
    "courses": [
   {
    "id": 1,
    "title": "React Guide",
    "instructor": "Andrew",
    "price": 100,
    "difficulty": "Intermediate",
    "categoryId": 1,
    "lessons": []
   }
  ]
 }
}
```

🟢 Get Single Course

- **Endpoint:** GET /courses/:id
- **Response:** Returns the single course object with its lessons.

🟡 Create Course

- **Endpoint:** POST /courses
- **Validation Rules:**
  - title: Required (3-100 chars).
  - instructor: Required (min 3 chars).
  - duration: Required (Positive Number).
  - difficulty: Required (Enum: 'Beginner', 'Intermediate', 'Advanced', 'All Levels').
  - categoryId: Required (Valid Integer ID).
  - price: Optional (Positive Number).
  - lessons: Optional (Array of objects).
- **Body Example (JSON):**

```json
JSON

{

  "title": "Advanced Node.js",

  "instructor": "Ahmed Khaled",

  "duration": 15,

  "difficulty": "Advanced",

  "categoryId": 1,

  "price": 200,

  "lessons": [

    {

      "title": "Event Loop",

      "content": "Deep dive into libuv",

      "duration": "20 mins"

    }

  ]

}
```

🔵 Update Course

- **Endpoint:** PUT /courses/:id
- **Description:** Updates any field of the course (supports partial updates).
- **Body:** Same structure as Create Course.

🔴 Delete Course

- **Endpoint:** DELETE /courses/:id
- **Response:** 200 OK with data: null.

⚠️ Common Errors (Status Codes)

- **400 Bad Request:** When Validation fails (e.g., missing title, wrong difficulty value).

    - *Payload:* {"status": "fail", "message": "\"title\" is required"}
- **404 Not Found:** When requesting an ID that doesn't exist.
    - *Payload:* {"status": "fail", "message": "No course found with ID: 99"}
- **500 Internal Server Error:** Something went wrong on the server.