

Online Course Manager API Documentation

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

✖ General Rules

1. **No IDs in Body:** Never send id in the request body for POST or PUT. The server handles IDs automatically.
 2. **Strict Validation:** sending unknown fields (e.g., sending img instead of image) will result in a **400 Bad Request** error.
 3. **Data Types:** Please adhere strictly to the data types specified below (String, Number, etc.).
-

1 Categories (/category)

● Get All Categories

- **Method:** GET
- **Endpoint:** /category
- **Description:** Fetch all available categories.
- **Response:**

JSON

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

● Create Category

- **Method:** POST
- **Endpoint:** /category

- **Body:**

JSON

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

● Update Category

- **Method:** PUT
 - **Endpoint:** /category/:id
 - **Body:** {"name": "Updated Name"}
-

2 Courses (/courses)

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

- **Method:** GET
- **Endpoint:** /courses
- **Query Parameters (Optional):**
 - search: Search by **Title or Instructor**. (e.g., ?search=react)
 - category: Filter by **Category ID**. (e.g., ?category=1)
 - sort: Sort results.
 - price: Ascending (Low to High).
 - -price: Descending (High to Low).
 - Example: ?sort=-price

● Create Course

- **Method:** POST
- **Endpoint:** /courses
- **Body (JSON):**

JSON

```
{  
  "title": "React JS Masterclass",    // Required (min 3 chars)
```

```

"instructor": "John Doe",           // Required (min 3 chars)

"price": 150,                      // Required (Number, >= 0)

"duration": 20,                    // Required (Number, Positive)

"difficulty": "Beginner",          // Required (Enum: 'Beginner', 'Intermediate', 'Advanced', 'All Levels')

"categoryId": 1,                   // Required (Valid Integer ID)

"description": "Learn React form...", // Optional (String)

"image": "https://example.com/img.png", // Optional (String, Valid URL)

// Optional: You can add lessons directly while creating a course

"lessons": [
  {
    "title": "Intro",
    "content": "Video link...",
    "duration": "5 mins"
  }
]
}

```

● **Warning:** The image field is named image, NOT img. Sending img will cause a 400 Error.

● Update Course

- **Method:** PUT
- **Endpoint:** /courses/:id
- **Body:** Same structure as Create Course. You can send only the fields you want to update.

● Delete Course

- **Method:** DELETE
- **Endpoint:** /courses/:id

● Lessons (Sub-resources)

● Add Lesson to Course

- **Method:** POST
- **Endpoint:** /courses/:courseId/lessons
 - Example: /courses/1/lessons
- **Body:**

JSON

```
{
  "title": "Lesson 1: Setup", // Required (min 3 chars)
  "content": "https://youtu.be/...", // Required
  "duration": "10 mins" // Required
}
```

● Update Lesson

- **Method:** PUT
- **Endpoint:** /courses/:courseId/lessons/:lessonId
 - Example: /courses/1/lessons/5
- **Body:** Same as Add Lesson.

● Delete Lesson

- **Method:** DELETE
- **Endpoint:** /courses/:courseId/lessons/:lessonId

⚠ Troubleshooting Errors

Status Code	Potential Cause	Response Message Example	Solution
400 Bad Request	Sending extra fields (e.g., img, id).	"img" is not allowed	Remove the extra field from the frontend request.
400 Bad Request	Missing required field.	"title" is required	Ensure Formik validation prevents submission if empty.
400 Bad Request	Invalid Enum value.	"difficulty" must be one of [...]	Check <select> values in frontend (Case Sensitive).

Status Code	Potential Cause	Response Message Example	Solution
404 Not Found	Invalid ID.	Course not found	Verify the ID in the URL.