

API Documentation

The backend fetches data from MySQL database and uses RESTful API to provide data to the frontend.

Endpoints

Enrolments Endpoint

- **URL:** `/enrolments`
- **Method:** `GET`
- **Parameters:**
 - `limit` (int): The number of enrolments to return (default is 20).
 - `offset` (int): The value of offset (default is 0).
 - `userFilter` (string): Filter enrolments by the user's name.
 - `courseFilter` (string): Filter enrolments by the course description.
- **Responses:**
 - `200 OK`: Successful response with data
 - **Example:**

```
{
  "enrolments": [
    {
      "id": 1,
      "user": "Sam Zhang",
      "course": "Course 1",
      "completion_status": "completed"
    },
    {
      "id": 2,
      "user": "Jane Smith",
      "course": "Course 5",
      "completion_status": "in progress"
    }
  ],
  "totalCount": 2
}
```

- `400 Bad Request`: Invalid limit or offset, or bad request.
- `500 Internal Server Error`: Server error.

Users Endpoint

- **URL:** `/users`
- **Method:** `GET`
- **Parameters:**
 - `q` (string): The search keyword to filter users by name.
- **Responses:**

- **200 OK:** Successful response with data

- **Example:**

```
{
  "users": [
    {
      "user_id": 1,
      "first_name": "Sam",
      "surname": "Zhang"
    },
    {
      "user_id": 2,
      "first_name": "Jane",
      "surname": "Smith"
    }
  ]
}
```

- **400 Bad Request:** Bad request.
- **500 Internal Server Error:** Server error.

Courses Endpoint

- **URL:** `/courses`
- **Method:** `GET`
- **Parameters:**
 - `q` (string): The search keyword to filter courses by description.
- **Responses:**
 - **200 OK:** Successful response with data

- **Example:**

```
{
  "courses": [
    {
      "course_id": 1,
      "description": "Course 1"
    },
    {
      "course_id": 2,
      "description": "Course 2"
    }
  ]
}
```

- **400 Bad Request:** Bad request.
- **500 Internal Server Error:** Server error.