openapi_week6.yaml Explore

Week Six Assignment Problem Statement 1.0.0 OAS3

openapi_week6.yaml

In this assignment, you have to create a RESTful API, database models using Flask-RESTful and flask-SQLAlchemy. We list below instructions to be followed in preparing and submitting the solution.

General instructions:

1. Submit a single .zip file containing all your submission files and folders, the name of which should be "roll_number.zip". E.g.: 21f1000000.zip
2. The folder structure inside the zip file should be as follows:
   I. The Python program must be written inside a file named "app.py". This file must reside inside the root folder.
   II. The database file named "api_database.sqlite3". You are not required to submit this database file with your submission.
3. You should not keep any code inside the scope of the condition " if __name__ == '__main__' " except run() call.
4. Allowed Python packages: flask, flask-sqlalchemy, flask-restful, and any standard Python3 package.
5. The database URI must be the same as: app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///api_database.sqlite3'.
6. You should not use create_all() call in your program to create a database.

Course Table Schema

Column Name	Column Type	Constraints
course_id	Integer	Primary Key, Auto Increment
course_name	String	Not Null
course_code	String	Unique, Not Null
course_description	String	

Student Table Schema

Column Name	Column Type	Constraints
student_id	Integer	Primary Key, Auto Increment
roll_number	String	Unique, Not Null
first_name	String	Not Null
last_name	String	

Enrollment Table Schema

Column Name	Column Type	Constraints
enrollment_id	Integer	Primary Key, Auto Increment
student_id	Integer	Foreign Key (student.student_id), Not Null
course_id	Integer	Foreign Key (course.course_id), Not Null

Error Codes

Resource	Error Code	Message
Course	COURSE001	Course Name is required
Course	COURSE002	Course Code is required
Student	STUDENT001	Roll Number required
Student	STUDENT002	First Name is required
Enrollment	ENROLLMENT001	Course does not exist
Enrollment	ENROLLMENT002	Student does not exist.

Servers

http://127.0.0.1:5000/

default

GET `/api/course/{course_id}`

Operation to Read course resource.

Parameters

Try it out

Name	Description
course_id * required `integer` *(path)*	201

Responses

Code	Description	Links
200	Request Successful Media type application/json Controls `Accept` header. **Example Value** Schema	*No links*

```
{
  "course_id": 201,
  "course_name": "Maths1",
  "course_code": "MA101",
  "course_description": "Course Description Example"
}
```

Code	Description	Links
404	Course not found	*No links*
500	Internal Server Error	*No links*

PUT `/api/course/{course_id}`

Operation to update the course resource.

Parameters

Try it out

Name	Description
course_id * required `integer` *(path)*	201

Request body

application/json

Example Value Schema

```
{
  "course_name": "Maths1",
  "course_code": "MA101",
  "course_description": "Course Description Example"
}
```

Responses

Code Description Links

200

Successfuly updated

No links

Media type

application/json

Controls `Accept` header.

Example Value Schema

```
{
  "course_id": 201,
  "course_name": "Maths1",
  "course_code": "MA101",
  "course_description": "Course Description Example"
}
```

400

Bad request

No links

Media type

application/json

Example Value Schema

```
{
  "error_code": "string",
  "error_message": "string"
}
```

404

Course not found

No links

500

Internal Server Error

No links

DELETE `/api/course/{course_id}`

Operation to delete the course resource

Parameters

Try it out

Name	Description
course_id * required integer *(path)*	201

Responses

Code	Description	Links
200	Successfully Deleted	*No links*
404	Course not found	*No links*
500	Intenal Server Error	*No links*

POST `/api/course`

Operation to create the course resource

Parameters

Try it out

No parameters

Request body

application/json

Example Value Schema

```
{
  "course_name": "Maths1",
  "course_code": "MA101",
  "course_description": "Course Description Example"
}
```

Responses

<table>
<tr><th>Code</th><th>Description</th><th>Links</th></tr>
<tr><td>201</td><td>Successfully Created

Media type
application/json
Controls <code>Accept</code> header.

<b>Example Value</b> Schema<pre>{
  "course_id": 201,
  "course_name": "Maths1",
  "course_code": "MA101",
  "course_description": "Course Description Example"
}</pre></td><td><i>No links</i></td></tr>
<tr><td>400</td><td>Bad request

Media type
application/json

<b>Example Value</b> Schema<pre>{
  "error_code": "string",
  "error_message": "string"
}</pre></td><td><i>No links</i></td></tr>
<tr><td>409</td><td>course_code already exist</td><td><i>No links</i></td></tr>
<tr><td>500</td><td>Internal Server Error</td><td><i>No links</i></td></tr>
</table>

GET `/api/student/{student_id}`

Operation to read student resource

Parameters

Try it out

Name	Description
student_id * required `integer` *(path)*	201

Responses

Code	Description	Links

200	Request Successful
	Media type
	application/json
	Controls `Accept` header.
	Example Value Schema
	`{`
	`"student_id": 101,`
	`"first_name": "Narendra",`
	`"last_name": "Mishra",`
	`"roll_number": "MA19M010"`
	`}`
		No links

404	Student not found	*No links*
500	Internal server error	*No links*

PUT `/api/student/{student_id}`

Operation to update the student resource

Parameters Try it out

Name	Description
student_id * required `integer` *(path)*	101

Request body **application/json**

Example Value Schema

```
{
  "first_name": "Narendra",
  "last_name": "Mishra",
  "roll_number": "MA19M010"
}
```

Responses

Code	Description	Links
200	Successfully updated Media type application/json Controls `Accept` header. **Example Value** Schema	*No links*

```
{
  "student_id": 101,
  "first_name": "Narendra",
  "last_name": "Mishra",
  "roll_number": "MA19M010"
}
```

Code	Description	Links
400	Bad request Media type application/json **Example Value** Schema	*No links*

```
{
  "error_code": "string",
  "error_message": "string"
}
```

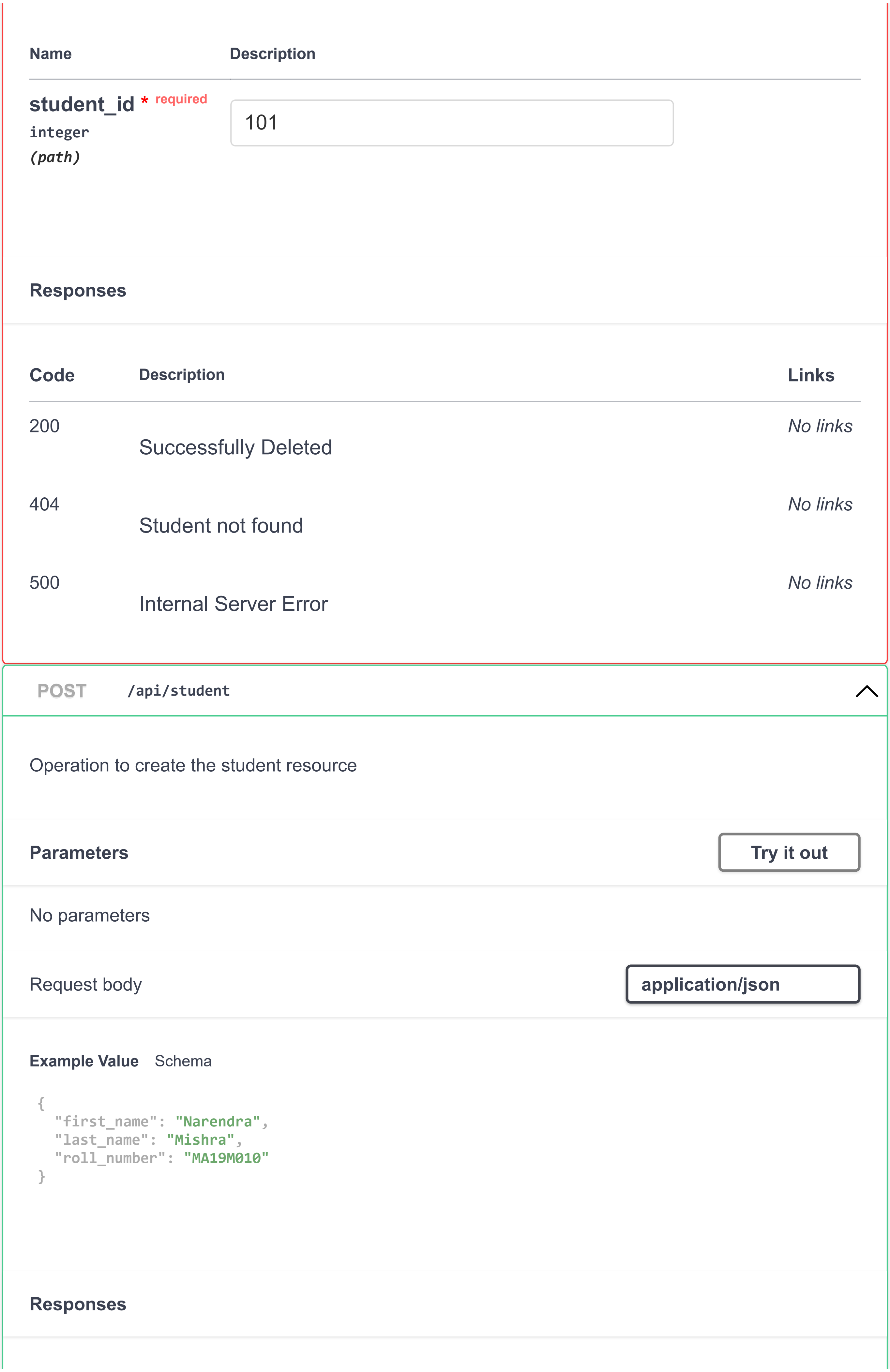
Code	Description	Links
404	Student not found	*No links*
500	Internal Server Error	*No links*

DELETE `/api/student/{student_id}`

Operation to delete the course resource

Parameters

Try it out



Name	Description
student_id * required integer *(path)*	101

Responses

Code	Description	Links
200	Successfully Deleted	*No links*
404	Student not found	*No links*
500	Internal Server Error	*No links*

POST /api/student

Operation to create the student resource

Parameters

No parameters

Request body

application/json

Example Value Schema

```
{
  "first_name": "Narendra",
  "last_name": "Mishra",
  "roll_number": "MA19M010"
}
```

Responses

Code	Description	Links
201	Successfully Created Media type application/json Controls `Accept` header. **Example Value** Schema `{ "student_id": 101, "first_name": "Narendra", "last_name": "Mishra", "roll_number": "MA19M010" }`	*No links*
400	Bad request Media type application/json **Example Value** Schema `{ "error_code": "string", "error_message": "string" }`	*No links*
409	Student already exist	*No links*
500	Internal Server Error	*No links*

GET `/api/student/{student_id}/course`

URL to get the list of enrollments, the student is enrolled in. This path belongs to the Enrollment table.

Parameters

Try it out

Name	Description
student_id * required `integer` *(path)*	101

Responses

Code **Description** **Links**

200

Request Successful

No links

Media type

application/json

Controls `Accept` header.

Example Value Schema

```
[
  {
    "enrollment_id": 10,
    "student_id": 101,
    "course_id": 201
  }
]
```

400

Invalid Student Id

No links

Media type

application/json

Example Value Schema

```
{
  "error_code": "string",
  "error_message": "string"
}
```

404

Student is not enrolled in any course

No links

500

Internal Server Error

No links

POST `/api/student/{student_id}/course`

Add student enrollment aka enroll the student to the course. This path belongs to the Enrollment table.

Parameters

Try it out

Name	Description
student_id * required integer *(path)*	101

Request body

application/json

Example Value Schema

```
{
  "course_id": 12345
}
```

Responses

Code	Description	Links
201	Enrollment successful	*No links*

Media type

application/json

Controls `Accept` header.

Example Value Schema

```
[
  {
    "enrollment_id": 10,
    "student_id": 101,
    "course_id": 201
  }
]
```

Code	Description	Links
400	Bad request	*No links*

Media type

application/json

Example Value Schema

```
{
  "error_code": "string",
  "error_message": "string"
}
```

Code	Description	Links
404	Student not found	*No links*
500	Internal Server Error	*No links*

DELETE `/api/student/{student_id}/course/{course_id}`

URL to delete enrollment of the student in the course. This path belongs to the Enrollment table.

Parameters

Try it out

Name	Description
student_id * required `integer` *(path)*	101
course_id * required `integer` *(path)*	10

Responses

Code	Description	Links
200	Successfully deleted	*No links*
400	Invalid Student Id or Course Id. Media type **application/json** **Example Value** Schema	*No links*

```
{
  "error_code": "string",
  "error_message": "string"
}
```

Code	Description	Links
404	Enrollment for the student not found	*No links*
500	Internal Server Error	*No links*

VALID {···}