

JAN LEANDER MATUTES

BSIT - 3A

Module 3.1 Activity: Building a Full CRUD API with Pydantic

main.py

```
main.py X
main.py > ...
1 from typing import List, Optional
2 from fastapi import FastAPI, HTTPException, Query, Path
3 from pydantic import BaseModel
4
5 app = FastAPI(title="Module3.1", version="1.0")
6
7 class StudentCreate(BaseModel):
8     name: str
9     course: str
10    gpa: Optional[float] = None
11
12 class StudentInDB(StudentCreate):
13     id: int
14
15 student_db: List[StudentInDB] = []
16 student_id_counter = 0
17
18 def _find_student_index(student_id: int) -> Optional[int]:
19     for idx, student in enumerate(student_db):
20         if student.id == student_id:
21             return idx
22     return None
23
24 @app.post("/students", response_model=StudentInDB, status_code=201)
25 def create_student(student: StudentCreate):
26     global student_id_counter
27     student_id_counter += 1
28     new_student = StudentInDB(id=student_id_counter, **student.dict())
29     student_db.append(new_student)
30     return new_student
31
32 @app.get("/students", response_model=List[StudentInDB])
33 def read_students(course: Optional[str] = Query(None, description="Filter by course")):
34     if course:
35         return [s for s in student_db if s.course.lower() == course.lower()]
36     return student_db
37
38 @app.get("/students/{student_id}", response_model=StudentInDB)
39 def read_student(student_id: int = Path(..., description="The ID of the student to retrieve")):
40     idx = _find_student_index(student_id)
41     if idx is None:
42         raise HTTPException(status_code=404, detail="Student not found")
43     return student_db[idx]
44
45 @app.put("/students/{student_id}", response_model=StudentInDB)
46 def update_student(student: StudentCreate, student_id: int = Path(..., description="The ID of the student to update")):
47     idx = _find_student_index(student_id)
48     if idx is None:
49         raise HTTPException(status_code=404, detail="Student not found")
50     updated = StudentInDB(id=student_id, **student.dict())
51     student_db[idx] = updated
52     return updated
53
54 @app.delete("/students/{student_id}")
55 def delete_student(student_id: int = Path(..., description="The ID of the student to delete")):
56     idx = _find_student_index(student_id)
57     if idx is None:
58         raise HTTPException(status_code=404, detail="Student not found")
59     student_db.pop(idx)
60     return {"message": "Student deleted successfully"}
61
```

default

POST	/students	Create Student	⌵
GET	/students	Read Students	⌵
GET	/students/{student_id}	Read Student	⌵
PUT	/students/{student_id}	Update Student	⌵
DELETE	/students/{student_id}	Delete Student	⌵

1. POST: Create 2-3 different students (e.g., some in "BSIT" and some in "BSCS").

POST /students Create Student

Parameters

Cancel

Reset

No parameters

Request body required

application/json

Edit Value | Schema

```
{  "name": "Arkin Gonzaga",  "course": "BSCS",  "gpa": 1.28}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \  'http://127.0.0.1:8000/students' \  -H 'accept: application/json' \  -H 'Content-Type: application/json' \  -d '{  "name": "Arkin Gonzaga",  "course": "BSCS",  "gpa": 1.28}'
```

Request URL

http://127.0.0.1:8000/students

Server response

CodeDetails

201

Response body

```
{  "name": "Arkin Gonzaga",  "course": "BSCS",  "gpa": 1.2,  "id": 3}
```

Download

Response headers

POST

/students

Create Student

Parameters

Cancel

Reset

No parameters

Request body

required

application/json

Edit Value

Schema

```
{
  "name": "Charlene Angel Custodio",
  "course": "BSIT",
  "gpa": 1.00
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/students' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "Charlene Angel Custodio",
    "course": "BSIT",
    "gpa": 1.00
  }'
```

Request URL

http://127.0.0.1:8000/students

Server response

Code

Details

201

Response body

```
{
  "name": "Charlene Angel Custodio",
  "course": "BSIT",
  "gpa": 1,
  "id": 2
}
```

Download

Response headers

POST

/students

Create Student

Parameters

Cancel

Reset

No parameters

Request body

required

application/json

Edit Value

Schema

```
{
  "name": "Jan Leander Matutes",
  "course": "BSIT",
  "gpa": 1.48
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/students' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "Jan Leander Matutes",
    "course": "BSIT",
    "gpa": 1.48
  }'
```

Request URL

http://127.0.0.1:8000/students

Server response

Code

Details

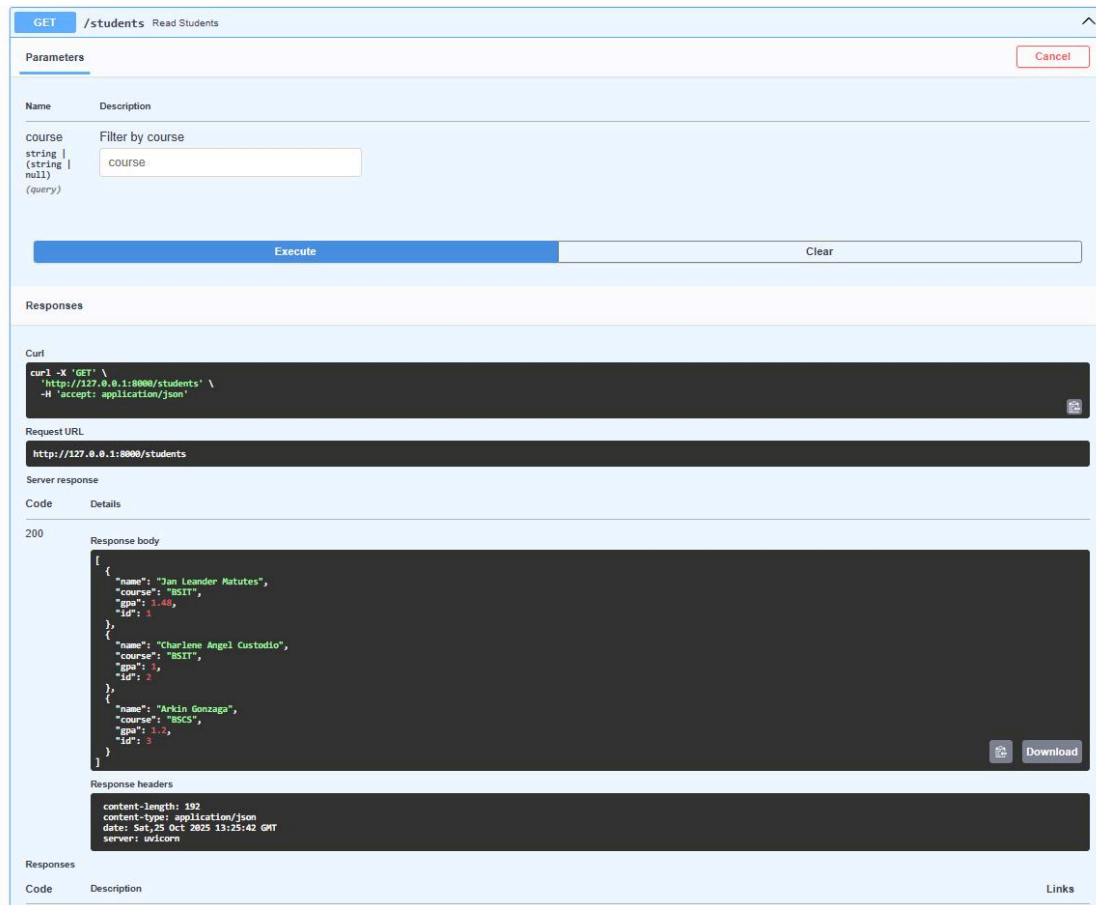
201

Response body

```
{
  "name": "Jan Leander Matutes",
  "course": "BSIT",
  "gpa": 1.48,
  "id": 1
}
```

Download

2. GET (All): Call the /students endpoint to see all students you created.



GET /students Read Students

Parameters

Cancel

Name	Description
course	Filter by course
string (string null) (query)	<input type="text" value="course"/>

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/students' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/students
```

Server response

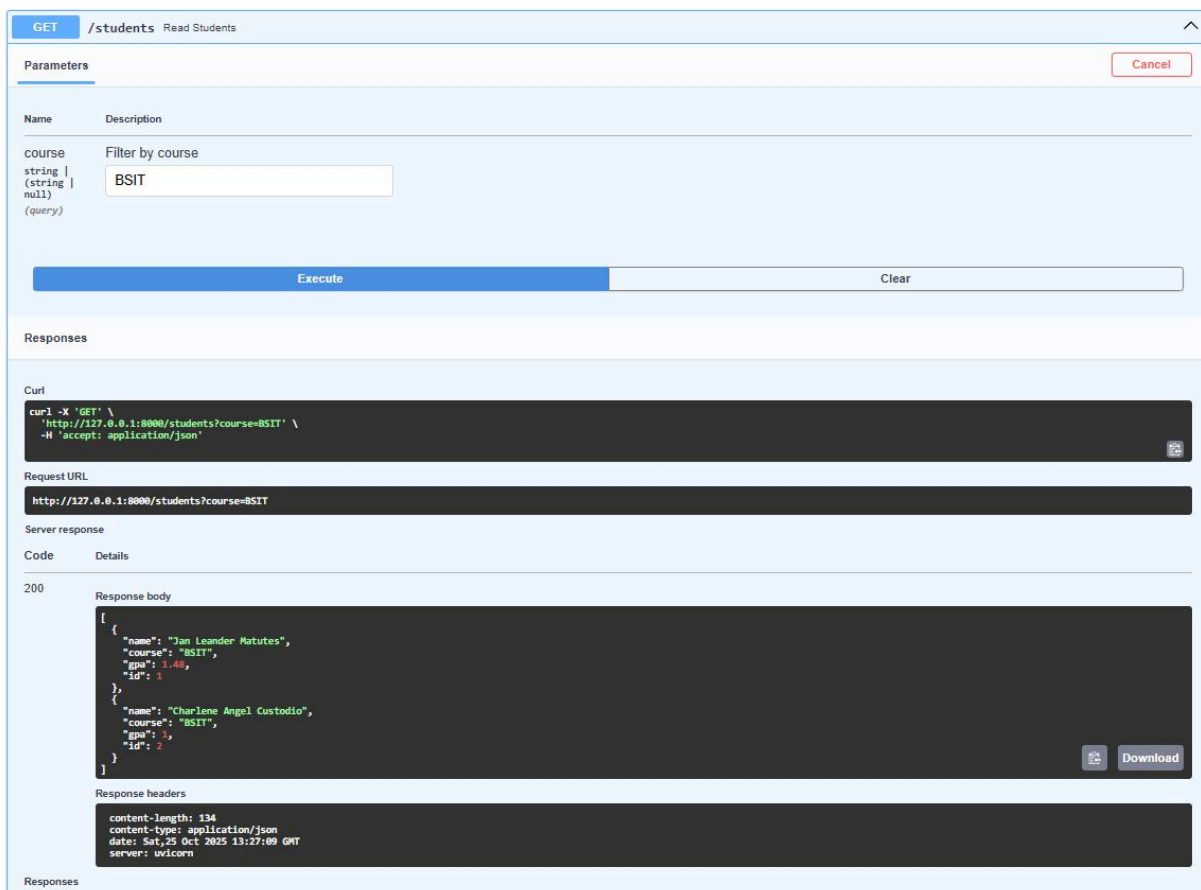
Code	Details
200	<p>Response body</p> <pre>[{ "name": "Jan Leander Matutes", "course": "BSIT", "gpa": 1.48, "id": 1 }, { "name": "Charlene Angel Custodio", "course": "BSIT", "gpa": 1, "id": 2 }, { "name": "Arkin Gonzaga", "course": "BSCS", "gpa": 1.2, "id": 3 }]</pre> <p>Response headers</p> <pre>content-length: 192 content-type: application/json date: Sat, 25 Oct 2025 13:25:42 GMT server: uvicorn</pre>

Responses

Code	Description
------	-------------

Links

3. GET (Query): Call the /students?course=BSIT endpoint to test the filtering.



GET /students Read Students

Parameters

Cancel

Name	Description
course	Filter by course
string (string null) (query)	<input type="text" value="BSIT"/>

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/students?course=BSIT' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/students?course=BSIT
```

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "name": "Jan Leander Matutes", "course": "BSIT", "gpa": 1.48, "id": 1 }, { "name": "Charlene Angel Custodio", "course": "BSIT", "gpa": 1, "id": 2 }]</pre> <p>Response headers</p> <pre>content-length: 134 content-type: application/json date: Sat, 25 Oct 2025 13:27:09 GMT server: uvicorn</pre>

Responses

Code	Description
------	-------------

4. GET (One): Get one of the students by their ID (e.g., /students/1).

GET

/students/{student_id}

Read Student

Cancel

Name	Description
student_id * required	The ID of the student to retrieve
integer (path)	<input type="text" value="2"/>

Execute

Clear

Responses

Curl

curl -X 'GET' \ 'http://127.0.0.1:8000/students/2' \ -H 'accept: application/json'

Request URL

http://127.0.0.1:8000/students/2

Server response

Code	Details
200	<div><div>Response body</div><div>{ "name": "Charlene Angel Custodio", "course": "BSIT", "gpa": 1, "id": 2 }</div><div><div>Download</div></div></div> <div><div>Response headers</div><div>content-length: 67 content-type: application/json date: Sat, 25 Oct 2025 13:27:42 GMT server: uvicorn</div></div>

Responses

Code	Description	Links
200	Successful Response	No links

5. PUT: Update one of the students (e.g., change the gpa or course for /students/1).

PUT

/students/{student_id}

Update Student

Cancel

Reset

75%

Name	Description
student_id * required	The ID of the student to update
integer (path)	<input type="text" value="1"/>

Request body * required

application/json

Edit Value | Schema

{ "name": "Jan Leander Matutes", "course": "BSCS", "gpa": 1.00 }

Execute

Clear

Responses

Curl

curl -X 'PUT' \ 'http://127.0.0.1:8000/students/1' \ -H 'accept: application/json' \ -H 'Content-Type: application/json' \ -d '{ "name": "Jan Leander Matutes", "course": "BSCS", "gpa": 1.00 }'

Request URL

http://127.0.0.1:8000/students/1

Server response

Code	Details
200	<div><div>Response body</div><div>{ "name": "Jan Leander Matutes", "course": "BSCS", "gpa": 1, "id": 1 }</div><div><div>Download</div></div></div>

6. DELETE: Delete one of the students (e.g., /students/2).

DELETE

/students/{student_id} Delete Student

Cancel

Parameters

Name	Description
student_id * required	The ID of the student to delete
integer (path)	<input type="text" value="3"/>

ExecuteClear

Responses

Curl

curl -X 'DELETE' \n'http://127.0.0.1:8000/students/3' \n-M 'accept: application/json'

Request URL

http://127.0.0.1:8000/students/3

Server response

Code	Details
200	<div><div>Response body</div><div>{\n "message": "Student deleted successfully"\n}</div><div>Download</div></div> <div><div>Response headers</div><div>content-length: 42\ncontent-type: application/json\ndate: Sat, 25 Oct 2025 13:29:48 GMT\nserver: uvicorn</div></div>

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

"string"

7. GET (Invalid): Try to GET, PUT, or DELETE a student that doesn't exist (e.g., /students/99) to confirm you get a 404 "Not Found" error.

PUT

/students/{student_id} Update Student

Cancel

Reset

Parameters

Name	Description
student_id * required	The ID of the student to update
integer (path)	<input type="text" value="99"/>

Request body required

application/json

Edit Value | Schema

{\n "name": "Jan Leander Matutes",\n "course": "BSCS",\n "gpa": 1.00\n}

ExecuteClear

Responses

Curl

curl -X 'PUT' \n'http://127.0.0.1:8000/students/99' \n-H 'accept: application/json' \n-H 'Content-Type: application/json' \n-d '{\n "name": "Jan Leander Matutes",\n "course": "BSCS",\n "gpa": 1.00\n}'

Request URL

http://127.0.0.1:8000/students/99

Server response

Code	Details
404	<div><div>Error: Not Found</div><div>undocumented</div><div><div>Response body</div><div>{\n "detail": "Student not found"\n}</div><div>Download</div></div></div>

GET

/students/{student_id}

Read Student

Parameters

Cancel

Name	Description
student_id * required <small>integer (path)</small>	The ID of the student to retrieve

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/students/99' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/students/99
```

Server response

Code	Details
404 <small>undocumented</small>	Error: Not Found

Response body

```
{
  "detail": "Student not found"
}
```

Download

Response headers

```
content-length: 30
content-type: application/json
date: Sat, 25 Oct 2025 13:31:12 GMT
server: uvicorn
```

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header.

DELETE

/students/{student_id}

Delete Student

Parameters

Cancel

Name	Description
student_id * required <small>integer (path)</small>	The ID of the student to delete

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://127.0.0.1:8000/students/99' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/students/99
```

Server response

Code	Details
404 <small>undocumented</small>	Error: Not Found

Response body

```
{
  "detail": "Student not found"
}
```

Download

Response headers

```
content-length: 30
content-type: application/json
date: Sat, 25 Oct 2025 13:32:04 GMT
server: uvicorn
```

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header.

Example Value

Schema