

# API Rest en Python con FastAPI

## Instalación de dependencias

Comenzamos instalando las dependencias necesarias:

```
pip install fastapi uvicorn sqlalchemy mysql-connector-python
```

## Compilación

Compilamos con el siguiente comando, mi archivo en este caso se llama fastAPI.py:

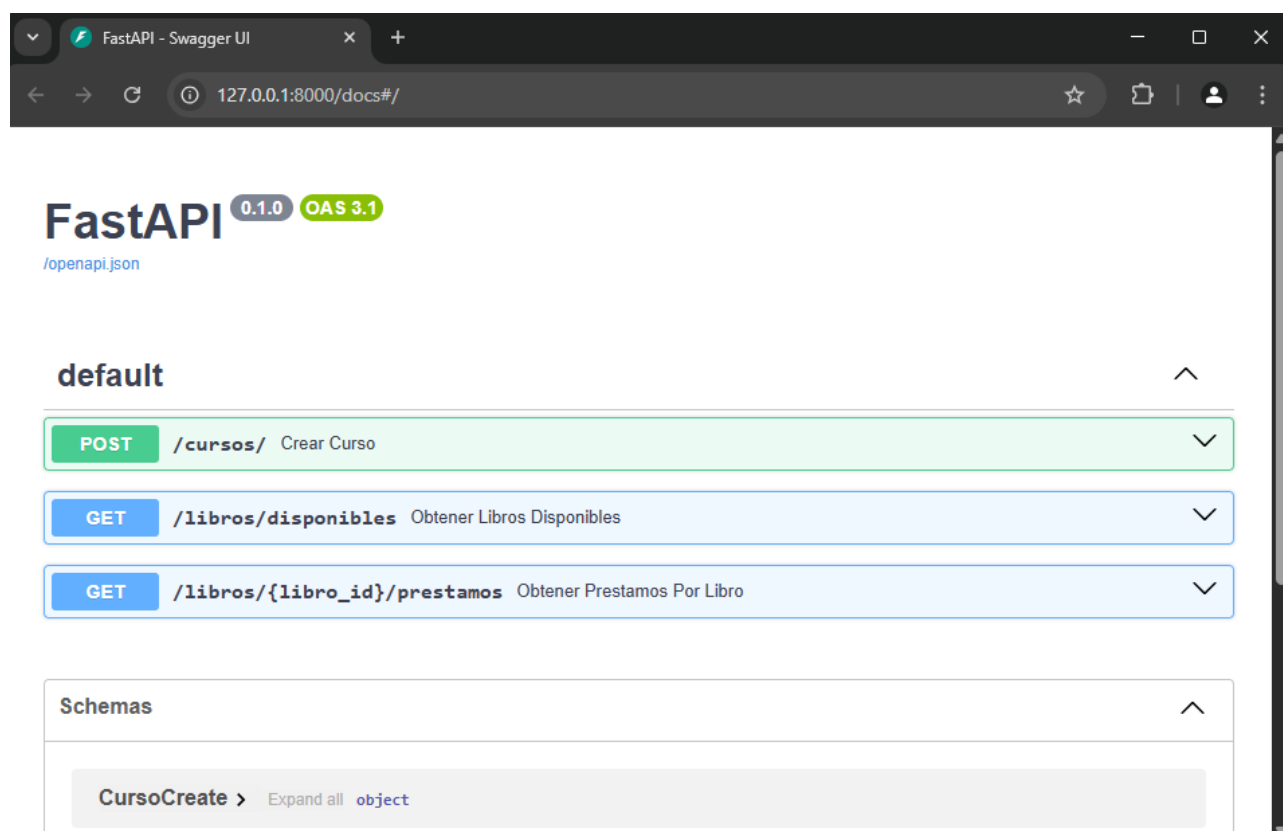
```
uvicorn fastAPI:app --reload
```

Y accedemos a:

<http://127.0.0.1:8000/docs#/>

Ya está listo.

## Funcionamiento



Accedo al endpoint de crear cursos y hago un post:

The screenshot shows the Swagger UI interface for a FastAPI application. The browser address bar displays the URL `127.0.0.1:8000/docs#/default/crear_curso_cursos_post`. The interface is titled **POST /cursos/ Crear Curso**.

**Parameters:** No parameters are defined for this endpoint.

**Request body:** Required, with a media type of `application/json`. The request body is a JSON object:

```
{
  "descripcion": "nuevo curso"
}
```

**Execute:** A blue button to execute the request. **Clear:** A button to clear the request body.

**Responses:**

**Curl:** The following curl command is generated:

```
curl -X 'POST' \
  'http://127.0.0.1:8000/cursos/' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "descripcion": "nuevo curso"
  }'
```

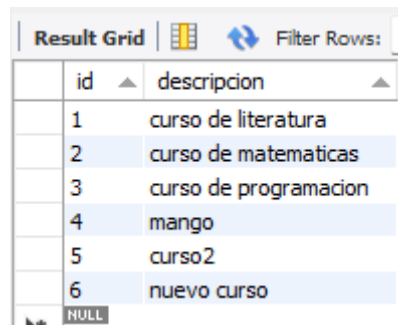
**Request URL:** `http://127.0.0.1:8000/cursos/`

**Server response:**

Code	Details
200	<p><b>Response body:</b></p> <pre>{   "descripcion": "nuevo curso",   "id": 6 }</pre> <p><b>Response headers:</b></p> <pre>content-length: 36 content-type: application/json date: Fri, 07 Mar 2025 19:12:28 GMT server: uvicorn</pre>

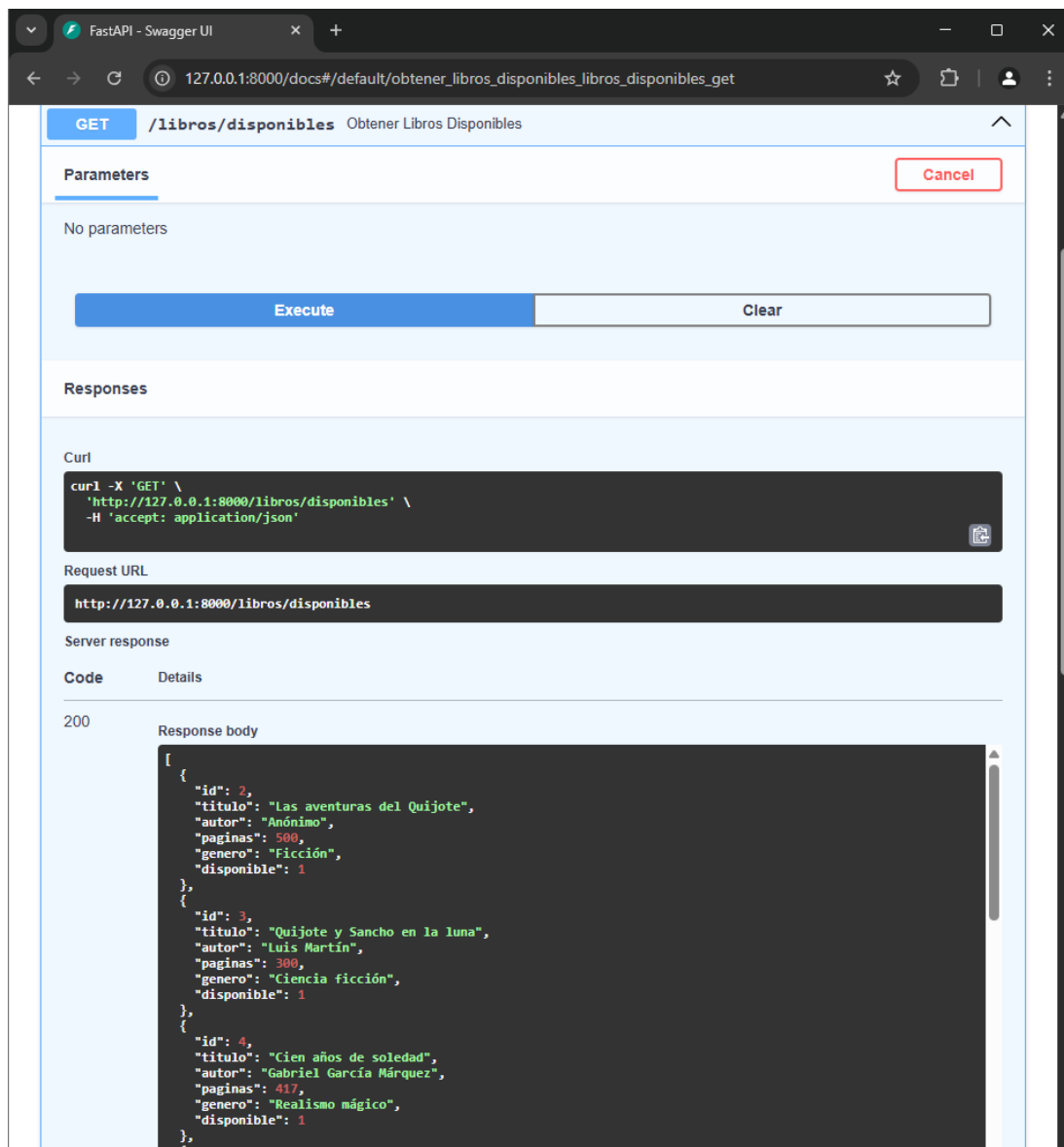
Oscar Montero Hinojosa  
**Acceso a Datos**

Y como vemos se ha insertado correctamente:



	id	descripcion
	1	curso de literatura
	2	curso de matematicas
	3	curso de programacion
	4	mango
	5	curso2
	6	nuevo curso
	NULL	

Ahora probamos el get de libros disponibles:



GET /libros/disponibles Obtener Libros Disponibles

Parameters

No parameters

Execute Clear

Responses

Curl

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

Request URL

```
http://127.0.0.1:8000/libros/disponibles
```

Server response

Code Details

200

Response body

```
[
  {
    "id": 2,
    "titulo": "Las aventuras del Quijote",
    "autor": "Anónimo",
    "paginas": 500,
    "genero": "Ficción",
    "disponible": 1
  },
  {
    "id": 3,
    "titulo": "Quijote y Sancho en la luna",
    "autor": "Luis Martín",
    "paginas": 300,
    "genero": "Ciencia ficción",
    "disponible": 1
  },
  {
    "id": 4,
    "titulo": "Cien años de soledad",
    "autor": "Gabriel García Márquez",
    "paginas": 417,
    "genero": "Realismo mágico",
    "disponible": 1
  },
  {
    "id": 5,
    "titulo": "Cien años de soledad",
    "autor": "Gabriel García Márquez",
    "paginas": 417,
    "genero": "Realismo mágico",
    "disponible": 1
  }
]
```

Y como vemos, funciona.

Y el tercer endpoint también funciona:

GET

/libros/{libro\_id}/prestamos

Obtener Prestamos Por Libro

^

Parameters

Cancel

Name	Description
libro_id * required integer (path)	<input type="text" value="1"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/libros/1/prestamos' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/libros/1/prestamos
```

Server response

Code	Details
200	<div><div>Response body<div><pre>[   {     "id_libro": 1,     "id_lector": 1,     "fechamita": "2025-03-07",     "fechabaja": null   } ]</pre></div><div><div>Download</div></div></div></div> <div>Response headers</div>