


Open · Opened 2 days ago by  **Rubén Montero**

GET-POST-PUT-DELETE

Resumen

- Repasaremos qué es CRUD
- Veremos que los principales verbos HTTP tienen una correspondencia 1:1 con las operaciones CRUD
- Crearemos un nuevo *endpoint* que responde de forma distinta en función del verbo HTTP recibido

Descripción

Un concepto esencial que aparece en distintas disciplinas de la informática es el de [CRUD](#). Son siglas en inglés para:

- **C**reate
- **R**ead
- **U**pdate
- **D**elete

...y definen las operaciones básicas de acceso a datos.

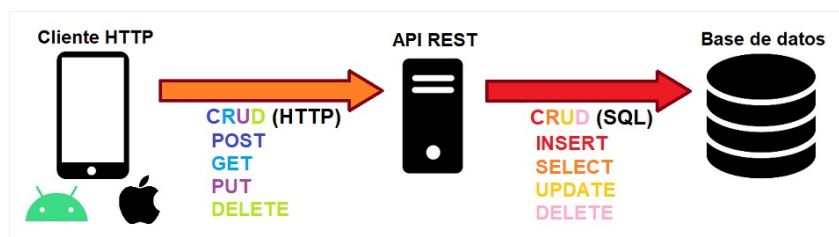
Correspondencia con métodos HTTP

Cuando estamos diseñando un API REST, las peticiones HTTP tienen por objeto **alterar** los datos de una base de datos.

Los métodos HTTP principales *encajan* con cada una de las operaciones CRUD así:

- **POST** : Create
- **GET** : Read
- **PUT** : Update
- **DELETE** : Delete

Una visión global



¿Base de datos?

Lo veremos pronto.

La tarea

Vamos a crear un nuevo *endpoint* que, simplemente, entregue al [cliente HTTP](#) una respuesta distinta en función del verbo HTTP usado en la petición.

De momento **no** hay conexión a ninguna base de datos.

Bastará con añadir este código a `endpoints.py` :

```
@csrf_exempt
def http_methods_example(request):
    if request.method == "GET":
        message = "Reading some data, huh?"
    elif request.method == "POST":
        message = "This should create a new thing"
    elif request.method == "PUT":
        message = "You can update any element with this"
```

```
elif request.method == "DELETE":
    message = "This will remove one or many elements"
else:
    # Esta línea nunca se ejecutará si el verbo HTTP no es GET, POST, PUT o DELETE
    return JsonResponse({"error": "HTTP method not allowed"}, status=405)
# Flujo normal
return JsonResponse({"message": message})
```

...y en `urls.py` **mapearemos** la función a `/http_methods`, así:

```
urlpatterns = [
    # ...
    path('http_methods', endpoints.http_methods_example),
]
```

¿Cuál es la respuesta HTTP si ejecutas los siguientes comandos en un terminal (cmd.exe)?

```
curl -X POST localhost:8000/http_methods
```

```
curl -X GET localhost:8000/http_methods
```

```
curl -X PUT localhost:8000/http_methods
```

```
curl -X DELETE localhost:8000/http_methods
```

Por último

Comprueba que tu código pasa el *test* asociado a la tarea.

Haz `commit` y `push` para subir los cambios al repositorio.



[Rubén Montero @ruben.montero](#) changed milestone to [%Sprint 4](#) 2 days ago