VERBOS HTTP:

**Resumo Didático**

| **Verbo** | **Ação** | **Idempotente** | **Quando Usar** |
| --- | --- | --- | --- |
| **GET** | Ler/retriever dados | Sim | Obter dados ou informações de um recurso. |
| **POST** | Criar novos recursos | Não | Criar algo novo no servidor. |
| **PUT** | Substituir completamente | Sim | Atualizar todos os dados de um recurso. |
| **PATCH** | Atualizar parcialmente | Sim | Atualizar apenas campos específicos. |
| **DELETE** | Excluir recursos | Sim | Remover um recurso existente. |
| **HEAD** | Obter metadados | Sim | Verificar existência sem o corpo da resposta. |
| **OPTIONS** | Descobrir operações | Sim | Verificar quais métodos são suportados. |

Com essa explicação, você consegue entender o papel de cada verbo HTTP e como utilizá-los no desenvolvimento e consumo de APIs no DRF!

**1. GET**

* **O que faz**: Recupera dados do servidor (operações de leitura).
* **Quando usar**: Quando você precisa obter informações de um recurso sem alterar seu estado.
* **Idempotente**: Sim (a mesma requisição sempre retorna o mesmo resultado, se os dados não mudarem).

**Exemplo prático**:

* Requisição: Recuperar todos os eBooks.

GET /api/ebooks/ HTTP/1.1

Resposta:

[

{"id": 1, "title": "Python Basics"},

{"id": 2, "title": "Advanced Django"}

]

Requisição: Recuperar detalhes de um eBook específico.

GET /api/ebooks/1/ HTTP/1.1

**2. POST**

* **O que faz**: Envia dados para o servidor para criar um novo recurso.
* **Quando usar**: Quando você precisa criar algo novo no servidor.
* **Idempotente**: Não (uma mesma requisição cria múltiplos recursos).

**Exemplo prático**:

* Requisição: Criar um novo eBook.

POST /api/ebooks/ HTTP/1.1

Content-Type: application/json

{

"title": "Python REST APIs",

"author": "Alexandre Bastos"

}

Resposta:

{

"id": 3,

"title": "Python REST APIs",

"author": "Alexandre Bastos"

}

**PUT**

* **O que faz**: Atualiza um recurso existente, substituindo-o completamente.
* **Quando usar**: Quando você quer substituir todos os dados de um recurso por novos dados.
* **Idempotente**: Sim (o mesmo resultado será alcançado com a mesma requisição).

**Exemplo prático**:

* Requisição: Atualizar um eBook existente.

PUT /api/ebooks/1/ HTTP/1.1

Content-Type: application/json

{

"title": "Python Basics Updated",

"author": "Alexandre Bastos"

}

Resposta:

{

"id": 1,

"title": "Python Basics Updated",

"author": "Alexandre Bastos"

}

**4. PATCH**

* **O que faz**: Atualiza parcialmente um recurso existente.
* **Quando usar**: Quando você quer modificar apenas alguns campos de um recurso.
* **Idempotente**: Sim.

**Exemplo prático**:

* Requisição: Atualizar o autor de um eBook.

PATCH /api/ebooks/1/ HTTP/1.1

Content-Type: application/json

{

"author": "Novo Autor"

}

PATCH /api/ebooks/1/ HTTP/1.1

Content-Type: application/json

{

"author": "Novo Autor"

}

Resposta:

{

"id": 1,

"title": "Python Basics",

"author": "Novo Autor"

}

**5. DELETE**

* **O que faz**: Remove um recurso do servidor.
* **Quando usar**: Quando você precisa excluir um recurso.
* **Idempotente**: Sim (deletar o mesmo recurso várias vezes retorna o mesmo resultado).

**Exemplo prático**:

* Requisição: Excluir um eBook.

DELETE /api/ebooks/1/ HTTP/1.1

Resposta (sem conteúdo):

HTTP/1.1 204 No Content

**6. HEAD**

* **O que faz**: Recupera apenas os cabeçalhos de uma resposta, sem o corpo.
* **Quando usar**: Quando você quer verificar a existência de um recurso ou metadados sem baixar o conteúdo.
* **Idempotente**: Sim.

**Exemplo prático**:

* Requisição: Verificar se o recurso /api/ebooks/1/ existe.

HEAD /api/ebooks/1/ HTTP/1.1

Resposta:

HTTP/1.1 200 OK

Content-Type: application/json

Content-Length: 145

**7. OPTIONS**

* **O que faz**: Retorna as opções permitidas para um recurso.
* **Quando usar**: Quando você quer descobrir quais verbos HTTP são suportados por um endpoint.
* **Idempotente**: Sim.

**Exemplo prático**:

* Requisição: Verificar as operações disponíveis no endpoint /api/ebooks/1/.

OPTIONS /api/ebooks/1/ HTTP/1.1

Resposta:

HTTP/1.1 200 OK

Allow: GET, PUT, PATCH, DELETE, OPTIONS

Content-Type: application/json