

Os principais métodos ou verbos HTTP são:

POST	GET	PUT	PATCH	DELETE
criar novos recursos	ler ou recuperar um recurso	substituir ou atualizar um recurso	modificar parcialmente um recurso	excluir um recurso

São chamados de verbos no modelo em que fazem pares com os substantivos indicados no nó do serviço. Cada verbo HTTP forma seu par no CRUD referente aos dados:

POST	GET	PUT	PATCH	DELETE
CREATE	READ	UPDATE		DELETE

Combinando os verbos com os dois formatos de URI's teríamos:

método	POST	GET	PUT	PATCH	DELETE
crud	CREATE	READ	UPDATE		DELETE
coleção (/clientes)	201: criado header location com o link	200: ok lista de clientes	405: não permitido		405: não permitido
item específico (/clientes/{id})	404 ou 409 (not found ou conflito - recurso já existe)	200 ou 204 (vazio) 404 não encontrado	200 ou 204 (vazio) 404 não encontrado		200 ou 404 não encontrado

Os métodos HTTP podem ser classificados em **safe**, **idempotent** ou **cacheable**:

método	POST	GET	PUT	PATCH	DELETE
safe	não	sim	não	não	não
idempotent	não	sim	idempotent	não	idempotent
cacheable	Only if freshness information is included	sim	não	não	não
req has body	sim	não	sim	sim	Talvez
resp has body	sim	sim	não	sim	Talvez
Allow in forms	sim	sim	não	não	não

## Exemplos

- POST

POST /test HTTP/1.1

Host: foo.example

Content-Type: application/x-www-form-urlencoded

Content-Length: 27

field1=value1&field2=value2

OU -----

POST /test HTTP/1.1

Host: foo.example

Content-Type: multipart/form-data;boundary="boundary"

--boundary

Content-Disposition: form-data; name="field1"

value1

--boundary

Content-Disposition: form-data; name="field2"; filename="example.txt"

value2

--boundary--

- GET

GET /index.html

- PUT

PUT /new.html HTTP/1.1

Host: example.com

Content-type: text/html

Content-length: 16

<p>New File</p>

HTTP/1.1 201 Created

Content-Location: /new.html