

- Guia Rápido de curl
 - 0. Abra uma janela de Terminal
 - 1. HTTP GET
 - 2. Guardar o resultado num Ficheiro
 - 3. HTTP POST
 - 4. HTTP HEAD
 - 5. Enviar Dados JSON
 - 6. HTTP PUT
 - 7. Adicionar Cabeçalhos
 - 8. Seguir Redirecionamentos
 - 9. Modos Verbose e Silencioso
 - 10. Usar Cookies

TeSP STI • Ano Letivo 2023/24 • Jaime Martins

Guia Rápido de curl

O **curl** é uma ferramenta versátil, de terminal, para transferir dados através de URLs. É amplamente utilizado para testar métodos HTTP, descarregar ficheiros e interagir com APIs.

Está disponível nativamente no Windows 10 e 11, ou em <https://curl.se/windows/>. Para outros sistemas operativos: <https://curl.se/download.html>.

Vamos em seguida explorar como, no papel de **Cliente**, se pode usar o **curl** para fazer solicitações a um **Servidor**, através de vários métodos HTTP.

0. Abra uma janela de Terminal

Antes de continuar com o tutorial, é necessário abrir um aplicativo de Terminal próprio para o Sistema Operativo que está a usar. Todos os comandos seguintes são escritos em terminal.

1. HTTP GET

Para obter o conteúdo de uma página web ou recurso:

```
curl http://example.com
```

Este comando obtém o conteúdo HTML de **example.com** e exibe-o no terminal.

2. Guardar o resultado num Ficheiro

Para guardar o resultado num ficheiro, em vez de o exibir:

```
curl http://example.com -o nomeficheiro.html
```

Em que **nomeficheiro.html** corresponde ao ficheiro onde queremos guardar o resultado. A opção **-o** (de *output* em inglês) especifica o ficheiro de saída.

3. HTTP POST

Para enviar dados num pedido POST:

```
curl -X POST -d "param1=valor1&param2=valor2" http://example.com/recurso
```

A opção **-X** especifica o método do pedido (POST), enquanto **-d** indica os dados a serem enviados.

4. HTTP HEAD

O método HEAD é utilizado para obter o cabeçalho de um recurso, o que é útil para ler metadados ou verificar a existência do mesmo, sem ter de o obter na íntegra.

```
curl -I http://example.com/recurso
```

5. Enviar Dados JSON

Para APIs que requerem dados de entrada no formato JSON:

```
curl -X POST -H "Content-Type: application/json" -d '{"chave1":"valor1",  
"chave2":"valor2"}' http://example.com/api
```

A opção **-H** adiciona um cabeçalho para especificar o tipo de conteúdo, como JSON.

6. HTTP PUT

Para atualizar um recurso com PUT:

```
curl -X PUT -d "param=valor" http://example.com/recurso
```

Também podemos enviar os parâmetros através de um ficheiro:

```
curl -X PUT -d @ficheiro.txt http://example.com/resource
```

7. Adicionar Cabeçalhos

Para incluir cabeçalhos adicionais na solicitação:

```
curl -H "X-Meu-Cabeçalho: 123" -H "Outro-Cabeçalho: abc" http://example.com
```

8. Seguir Redirecionamentos

Algumas solicitações HTTP resultam em redirecionamentos. Para seguir automaticamente:

```
curl -L http://example.com
```

A opção **-L** diz ao **curl** para seguir redirecionamentos até alcançar o destino final.

9. Modos Verbose e Silencioso

- **Modo verbose** – fornece informações detalhadas do pedido e resposta:

```
curl -v http://example.com
```

- **Modo silencioso** – suprime toda a saída, exceto o corpo principal da resposta:

```
curl -s http://example.com
```

10. Usar Cookies

Para enviar um pedido com cookies:

```
curl -b "nomeCookie=valorCookie" http://example.com
```

A opção **-b** envia cookies juntamente com o pedido.