Introdução ao cURL

Transferência de Dados e Manipulação de Requisições HTTP

O **cURL** (Client URL) é uma ferramenta de linha de comando robusta e uma biblioteca (às vezes referida como libcurl) usada para transferência de dados utilizando diversos protocolos, como **HTTP, HTTPS, FTP, FTPS, SCP, SFTP, LDAP, DICT, TELNET, e FILE**.

Ele é amplamente empregado em automação, integrações entre sistemas e testes de API.

O cURL está presente nativamente em sistemas **Linux, macOS**, e pode ser instalado facilmente no **Windows**.

Ele suporta autenticação HTTP, cookies, manipulação de cabeçalhos, proxy, compressão e conexões seguras via SSL/TLS.

Conceitos Fundamentais

Antes de explorar comandos avançados, é crucial compreender os aspectos técnicos fundamentais do cURL:

1. **Requisições HTTP/HTTPS**: Suporte completo a métodos como **GET, POST, PUT, PATCH, DELETE, OPTIONS e HEAD**.
2. **Manipulação de Headers**: Permite adicionar, remover e modificar cabeçalhos HTTP.
3. **Autenticação e Segurança**: Suporte a **Basic Auth, Digest Auth, Bearer Tokens, OAuth, Certificados SSL e chaves privadas**.
4. **Envio e Recebimento de Dados**: Suporte a **formulários multipart, JSON, XML, e arquivos binários**.
5. **Redirecionamento e Proxy**: Gerenciamento de **redirecionamentos HTTP (301, 302, 307)** e suporte a proxies HTTP e SOCKS.
6. **Performance e Depuração**: Permite medição de tempo de resposta e depuração de problemas com opções verbosas.

Comandos Avançados e Otimização de Uso

A seguir, veja comandos altamente técnicos do cURL com exemplos práticos e ajustes de desempenho.

1. Requisição GET com Timeout e Verbosidade

Obtendo dados de uma API e limitando o tempo de resposta:

```
curl -X GET https://jsonplaceholder.typicode.com/posts/1 \
    --connect-timeout 5 \
    --max-time 10 \
    --verbose
```

2. Requisição GET com Headers Customizados e Compressão

```
curl -X GET https://api.exemplo.com/dados \
    -H "Authorization: Bearer SEU_TOKEN" \
    -H "Accept-Encoding: gzip, deflate" \
    --compressed
```

3. Requisição POST com JSON e Headers Dinâmicos

```
curl -X POST https://jsonplaceholder.typicode.com/posts \
    -H "Content-Type: application/json" \
    -H "User-Agent: Mozilla/5.0" \
    -d '{"title": "Novo Post", "body": "Conteúdo do post", "userId": 1}'
```

4. Requisição PUT com Upload de Arquivo JSON

```
curl -X PUT https://jsonplaceholder.typicode.com/posts/1 \
    -H "Content-Type: application/json" \
    -T dados.json
```

5. Requisição DELETE com Autenticação via Basic Auth

```
curl -X DELETE https://api.exemplo.com/recursos/1 \
    -u usuario:senha \
    --verbose
```

6. Download de Arquivo com Resume

```
curl -O -C - https://exemplo.com/arquivo.zip
```

7. Upload de Arquivo via Formulário Multipart

```
curl -X POST https://api.exemplo.com/upload \
    -F "file=@/caminho/do/arquivo.txt" \
    -F "descricao=Arquivo de teste"
```

8. Medindo Tempo de Resposta da Requisição

```
curl -o /dev/null -s -w 'Tempo total: %{time_total}s\n' https://api.exemplo.com/dados
```

9. Testando Headers e Debugging Completo

```
curl -X GET https://api.exemplo.com/dados \
    -H "Accept: application/json" \
    --include \
    --trace-ascii log.txt
```

10. Uso de Proxy SOCKS5

```
curl -x socks5h://127.0.0.1:9050 https://check.torproject.org
```

Considerações

O cURL é essencial para automação e testes de serviços web. Dominar seus argumentos e opções permite desde depuração até operações complexas com segurança e autenticação avançada. Recursos como **limitação de banda (--limit-rate), redirecionamento (-L), e manipulação de cookies (-c, -b)** permitem maximizar seu potencial.

Para mais informações, consulte a documentação oficial em https://curl.se/docs/

EducaCiência FastCode para a comunidade