

捲曲

這裡有一個結合了幾個 curl 範例的 bash 腳本：

```
#!/bin/bash
```

```
# 基本 GET 請求
```

```
echo " 基本 GET 請求:"
```

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

```
# 帶自定義標頭的 GET 請求
```

```
echo "\n 帶自定義標頭的 GET 請求:"
```

```
curl -H "User-Agent: MyUserAgent/1.0" http://example.com
```

```
# POST 請求
```

```
echo "\nPOST 請求:"
```

```
curl -X POST -d "key1=value1&key2=value2" http://example.com/post
```

```
# 帶 JSON 資料的 POST 請求
```

```
echo "\n 帶 JSON 資料的 POST 請求:"
```

```
curl -X POST -H "Content-Type: application/json" -d "{\"key1\":\"value1\",\"key2\":\"value2\"}" http://example.com/post
```

```
# 使用代理
```

```
echo "\n 使用代理:"
```

```
curl -x http://proxy.example.com:8080 http://example.com
```

```
# 認證
```

```
echo "\n 基本認證:"
```

```
curl -u username:password http://example.com/auth
```

```
# 下載文件
```

```
echo "\n 下載文件:"
```

```
curl -O http://example.com/file.zip
```

```
# 跟隨重定向
```

```
echo "\n 跟隨重定向:"
```

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

```
# 將輸出保存到文件
```

```
echo "\n 將輸出保存到文件:"
```

```
curl http://example.com -o output.txt
```

```
# 自定義 HTTP 方法
```

```
echo "\n 自定義 HTTP 方法 (DELETE):"
```

```
curl -X DELETE http://example.com/delete/resource
```

```
# 詳細輸出
```

```
echo "\n 詳細輸出:"
```

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

```
# 連接超時
```

```
echo "\n 連接超時:"
```

```
curl --connect-timeout 5 http://example.com
```

```
echo "\n 腳本執行完成。"
```

將這個腳本保存為 .sh 擴展名，例如 curl_examples.sh。如果你在 Unix 類似的環境或在 Windows 子系統 Linux (WSL) 中運行，請確保它是可執行的：

```
chmod +x curl_examples.sh
```

然後，你可以運行它：

```
./curl_examples.sh
```

Windows 注意事項： - 如果你在 Windows Cmd 或 PowerShell 中運行這個腳本，你需要使用 bash 或 wsl (如果你已經安裝了 WSL)，或者確保 curl 在你的 PATH 中以便直接執行：- bash curl_examples.sh 或 wsl ./curl_examples.sh - 記住，這個腳本使用了 bash 語法，所以它在 Windows 命令提示符或 PowerShell 中不會原生兼容，除非通過 WSL 或 bash 環境運行。然而，curl 命令本身與 Windows 版本的 curl 兼容。