

GABRIELE ROMANATO

Effettuare richieste HTTP con Bash

Bash è uno strumento potente e versatile per automatizzare compiti quotidiani, inclusa l'interazione con servizi web tramite richieste HTTP. In questo articolo esploreremo come utilizzare i comandi `curl` e `wget` per effettuare richieste HTTP direttamente da un terminale Bash.

Prerequisiti

Prima di iniziare, assicurati di avere accesso a un terminale Bash e di avere installati i seguenti strumenti:

- **curl**: Un comando versatile per effettuare richieste HTTP.
- **wget**: Un'utilità utile per il download di file e richieste HTTP.

Entrambi sono disponibili nella maggior parte delle distribuzioni Linux e possono essere installati facilmente utilizzando il gestore di pacchetti della tua distribuzione.

Effettuare Richieste HTTP con `curl`

`curl` è uno degli strumenti più popolari per interagire con API REST e servizi web. Ecco alcuni esempi di utilizzo:

1. Effettuare una Richiesta GET

```
curl https://api.example.com/data
```

Questo comando effettua una richiesta GET verso l'URL specificato e restituisce la risposta.

2. Effettuare una Richiesta POST

```
curl -X POST -d "param1=valore1&param2=valore2"  
https://api.example.com/submit
```

La flag `-X POST` specifica il metodo HTTP, mentre `-d` invia i dati del corpo della richiesta.

3. Aggiungere Intestazioni Personalizzate

```
curl -H "Authorization: Bearer YOUR_TOKEN"  
https://api.example.com/protected
```

Con la flag `-H`, puoi specificare intestazioni HTTP personalizzate.

Effettuare Richieste HTTP con `wget`

`wget` è principalmente utilizzato per scaricare file, ma può anche gestire richieste HTTP. Ecco alcuni esempi:

1. Effettuare una Richiesta GET

```
wget https://api.example.com/data -O output.json
```

Il parametro `-O` permette di salvare la risposta in un file.

2. Effettuare una Richiesta con Intestazioni

```
wget --header="Authorization: Bearer YOUR_TOKEN"  
https://api.example.com/protected
```

Utilizza `--header` per aggiungere intestazioni personalizzate.

Consigli Utili

Ecco alcuni suggerimenti per ottimizzare l'uso di `curl` e `wget`:

- Utilizza l'opzione `-s` con `curl` per nascondere il progresso e altre informazioni superflue.
- Combina `jq` con `curl` per formattare le risposte JSON.
- Verifica i codici di stato HTTP per gestire correttamente errori e successi.

Conclusione

Effettuare richieste HTTP con Bash è semplice e potente grazie a strumenti come `curl` e `wget`. Questi strumenti ti permettono di automatizzare interazioni con API e di gestire trasferimenti di dati in modo efficiente. Prova gli esempi forniti e adatta i comandi alle tue esigenze specifiche.