

curl, Bruno et Postman

Pour parler à une API, pas besoin d'écrire du code : des outils permettent d'envoyer des requêtes HTTP à la main.

curl

`curl` est l'outil en ligne de commande de référence. Il est installé par défaut sur Linux, macOS et Windows 10+.

```
curl http://api-ritif.portlatch.com/v1/cocktails
```

Les options utiles

Option	Rôle
<code>-i</code>	Affiche les headers de la réponse
<code>-v</code>	Mode verbeux : affiche toute la requête et toute la réponse
<code>-X METHODE</code>	Choisit la méthode HTTP (<code>-X DELETE</code>)
<code>-H "Nom: valeur"</code>	Ajoute un header
<code>-d '...'</code>	Envoie un body (et passe en <code>POST</code> par défaut)
<code>--json '...'</code>	Envoie du JSON, avec les headers <code>Content-Type</code> et <code>Accept</code> adaptés
<code>-s</code>	Mode silencieux (masque la barre de progression)

Exemples

```
# Lire les headers de la réponse
curl -i http://api-ritif.portlatch.com/v1/cocktails/3

# Envoyer du JSON
curl -i -X POST http://api-ritif.portlatch.com/v1/orders \
  -H "Content-Type: application/json" \
  -H "X-API-Key: votre-cle" \
  -d '{"cocktail_id": 3}'
```

```
# La même chose, en plus court
curl -i http://api-ritif.portlatch.com/v1/orders \
  -H "X-API-Key: votre-cle" \
  --json '{"cocktail_id": 3}'

# Supprimer
curl -i -X DELETE http://api-ritif.portlatch.com/v1/orders/42 -H "X-API-Key: votre-cle"
```

Pour rendre le JSON lisible, passez la réponse dans un formateur : `curl -s ... | python3 -m json.tool` ou `curl -s ... | jq`.

Sous Windows PowerShell, `curl` peut être un alias vers `Invoke-WebRequest`, qui n'a pas les mêmes options. Utilisez `curl.exe`, ou Git Bash.

Plutôt que de copier votre clé dans chaque commande, stockez-la dans une variable : `export KEY=votre-cle`, puis `-H "X-API-Key: $KEY"`.

Bruno

[Bruno](#) est un client d'API graphique, open source et sans compte. On y enregistre ses requêtes dans des **collections**.

Sa particularité : une collection est un **dossier de fichiers texte**. On peut donc la versionner dans git et la partager avec son équipe comme du code.

Prise en main

- Installez Bruno depuis usebruno.com/downloads.
- Créez une collection `api-ritif` dans un dossier de votre choix.
- Créez un **environnement** avec deux variables : `baseUrl` = `http://api-ritif.portlatch.com/v1` et `apiKey` = votre clé.
- Créez une requête `GET {{baseUrl}}/cocktails` et envoyez-la.
- Dans l'onglet **Headers** d'une requête, ajoutez `X-API-Key: {{apiKey}}`.

Et Postman ?

Postman est le client d'API le plus connu, et vous le croiserez sûrement en entreprise. Il fonctionne sur le même principe que Bruno.

On ne l'utilise pas dans ce cours : il demande un compte, et stocke les collections dans son cloud plutôt que dans votre dépôt git.

Revision #4

Created 2 October 2026 20:00:44 by Thibaud FRICHET

Updated 13 October 2026 14:37:46 by Thibaud FRICHET