Chapitre 3 - Outils et fil rouge

- Bienvenue au bar API-ritif □□
- curl, Bruno et Postman
- TP : La chasse au trésor

Bienvenue au bar API-ritif ?

Tout au long de ce cours, nous allons fréquenter un établissement très particulier : **API-ritif**, le bar de l'école de dev. On n'y entre pas par la porte, mais par HTTP.

> L'adresse exacte du bar est donnée en cours.

> Consommation responsable : les cocktails d'API-ritif sont 100 % virtuels.
> L'abus de requêtes est sanctionné par le barman (voir `429`).

La carte

Cocktail	Description du barman
Hello World	Sans alcool. Le premier de tout développeur.
Null Pointer Mojito	Sans alcool : contient `null` alcool.
Segfault Sunrise	Un classique. Plante rarement, mais sans prévenir.
Pull Request Colada	Doux et crémeux. Nécessite deux approbations.
Blue Screen Lagoon	Bleu. Très bleu.
Java Martini	Un espresso martini. Long à démarrer, puis très efficace.
Merge Conflict	Deux alcools qui ne s'entendent pas. À résoudre à la main.
Kernel Panic	Le plus fort de la carte. Redémarrage conseillé après.
404 Spritz	Introuvable, comme son nom l'indique.
Sudo Rhum	Pour ceux qui ont les droits.
Infinite Loop	On en reprend un, puis un autre, puis un autre…

Le barman murmure qu'il existe d'autres cocktails, qui ne figurent pas sur la première page de la carte…

Les règles de la maison

- Toutes les routes commencent par `/v1`.
- Le bar parle **JSON**. Pensez au header `Content-Type: application/json` quand vous envoyez des données.
- Les listes sont **paginées** : paramètres `?page=` et `?per_page=`.
- Certaines routes demandent d'être client du bar : envoyez votre clé dans le header `X-API-Key`.
- Les erreurs sont renvoyées au format `application/problem+json` : lisez-les, elles sont souvent bavardes.
- Le barman n'est pas une machine : trop de requêtes d'un coup et il vous demande de patienter.

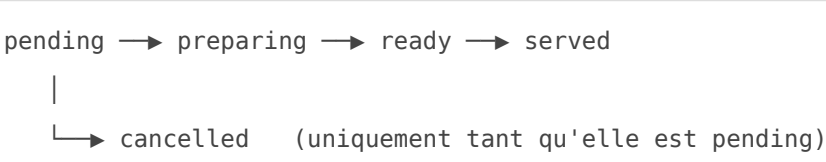
Les principales ressources

Ressource	Description
`/v1/`	L'accueil du bar
`/v1/customers`	Les clients (inscription)
`/v1/me`	Votre profil
`/v1/cocktails`	La carte
`/v1/cocktails/{id}/reviews`	Les avis sur un cocktail
`/v1/orders`	Vos commandes

La documentation complète de l'API (le contrat OpenAPI) sera étudiée au chapitre 6. D'ici là, explorez !

La vie d'une commande

Une commande ne reste pas figée : elle avance toute seule dans le temps.



> Comme dans un vrai bar : une fois que le barman a commencé à préparer la commande, il est trop tard pour annuler !

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

TP : La chasse au trésor

Objectif

Le barman d'API-ritif a caché un trésor quelque part dans son API. Pour le trouver, il faudra lire attentivement chaque réponse : codes de statut, headers, body… Rien n'est là par hasard.

Chaque étape franchie vous donne un **flag** : il se trouve dans le **header de la réponse** (`X-Flag`). Il y a **9 flags** à activer.

> TP noté, utilisation de curl ou bruno au choix

Consignes

Règles du jeu

> Inscrivez-vous avec le pseudo **`prenom-nom`**, sans accents ni espaces (ex. : `helene-dupre`). C'est lui qui permet au formateur de vous noter : un pseudo fantaisiste, c'est un TP sans note.

- Chaque réponse de l'API contient l'indice qui mène à l'étape suivante (header `X-Hint`). Lisez **tout**, y compris les headers (`curl -i`).
- Quand vous franchissez une étape, la réponse contient son **flag dans le header `X-Flag`**, pas dans le body. Sans `curl -i`, vous ne verrez ni les flags ni les indices.
- Dès que vous avez une clé d'API, soumettez chaque flag trouvé :

```
curl -i http://api-ritif.portlatch.com/v1/hunt/flag \
  -H "X-API-Key: $KEY" \
  --json '{"code": "RITIF-XXXX-XXXXXXXX"}'
```

- Les flags se valident **dans l'ordre**, et ne s'échangent pas : chaque flag (sauf le premier) est lié à votre clé.

>

> Bloqué ? Relisez la page des codes de statut et celle des verbes HTTP. Un message d'erreur est souvent un indice.

Les 9 flags à activer

Chaque étape ci-dessous renvoie un flag dans le **header de la réponse**. Soumettez-les dans cet ordre :

1. Lire les headers de l'accueil du bar.
2. Vous inscrire comme client du bar.
3. Vous présenter au barman : d'abord sans clé, puis avec.
4. Trouver le cocktail secret de la carte et le consulter.
5. Commander un thé.
6. Commander un cocktail et suivre sa commande jusqu'à ce qu'elle soit prête.
7. Annuler une commande à temps. Découvrez au passage ce que répond `OPTIONS`.
8. Obtenir un cocktail en XML.
9. Faire déborder le barman jusqu'au `429`, respecter le délai demandé, puis refaire une requête.

Point de départ

Tout commence à l'accueil du bar :

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

Le reste, c'est à vous de le trouver. Santé ! □□