Exercices pratiques REST

Emmanuel BRUNO
Université de Toulon, LIS UMR CNRS 7020
emmanuel.bruno@univ-tln.fr

2026-11-01

Table des matières

1 Objectifs d'apprentissage 1
2 Outils nécessaires 1
3 curl - Options de base 2
4 Exercice 1: Explorer les APIs REST avec curl 2
4.1 Requêtes basiques 2
4.1.a Position actuelle de l'ISS 2
4.1.b Astronautes dans l'espace 2
5 jq - Filtrer et formater JSON 3
5.1 Syntaxe de base 3
5.2 Filtres avancés 3
5.3 Requêtes avec paramètres et authentification 3
5.4 Manipulation des headers et formats 4
5.5 Exercices pratiques 4
6 Exercice 2: Explorer OpenAPI 5
7 Exercice 3. Client Java avec Jersey 5
8 Exercice 4 Client Quarkus (Optionnel) 5

1 Objectifs d'apprentissage

À la fin de ce TP, vous devez être capable de : - Interroger des APIs REST avec différents outils - Comprendre et utiliser la documentation OpenAPI - Implémenter des clients REST en Java

2 Outils nécessaires

- curl : un outil en ligne de commande pour transférer des données avec des URL
- jq : un outil pour formater et filtrer des données JSON
- Postman : un outil pour tester des APIs REST (optionnel).

3 curl - Options de base

```
# Verbosité et debug
-v, --verbose     # Affiche les détails de la requête/réponse
-s, --silent      # Mode silencieux, supprime la barre de progression

# Méthodes HTTP
-X POST           # Spécifie la méthode HTTP (GET, POST, PUT, DELETE...)
-d, --data        # Envoie des données dans le corps de la requête

# Headers et authentification
-H, --header      # Ajoute un header personnalisé
-u, --user        # Basic auth (username:password)

# Format et données
-i                # Inclut les headers dans la sortie
```

```
# POST avec JSON et Bearer token
curl -X POST \
     -H "Content-Type: application/json" \
     -H "Authorization: Bearer YOUR_TOKEN" \
     -d '{"name":"John","age":30}' \
     "https://api.example.com/users"
```

4 Exercice 1: Explorer les APIs REST avec curl

4.1 Requêtes basiques

L'API ISS (Station Spatiale Internationale) ne nécessite pas d'authentification. http://open-notify.org/Open-Notify-API/ISS-Location-Now/

4.1.a Position actuelle de l'ISS

```
#| echo: true
#| output: true
curl -i http://api.open-notify.org/iss-now.json
```

4.1.b Astronautes dans l'espace

```
#| echo: true
#| output: true
curl -i "http://api.open-notify.org/astros.json"
```

5 jq - Filtrer et formater JSON

`jq` est un utilitaire en ligne de commande pour traiter du JSON. Il permet de filtrer, transformer et formater les données JSON de manière élégante.

5.1 Syntaxe de base

```
# Formatage simple
curl -s "http://api.open-notify.org/iss-now.json" | jq '.'

# Sélection de champs
curl -s "http://api.open-notify.org/iss-now.json" | jq '.iss_position.latitude'
```

5.2 Filtres avancés

```
# Filtrer un tableau
curl -s "http://api.open-notify.org/astros.json" | jq '.people[] | select(.craft=="ISS")'

# Transformation de données
curl -s "http://api.open-notify.org/astros.json" | \
  jq '.people[] | {astronaut: .name, vessel: .craft}'

# Comptage et statistiques
curl -s "http://api.open-notify.org/astros.json" | \
  jq '.people | group_by(.craft) | map({craft: .[0].craft, count: length})'
```

5.3 Requêtes avec paramètres et authentification

Il faut généralement ajouter des paramètres à une requête pour obtenir des informations spécifiques. En particulier, une authentification sous la forme d'une clé API (token) est souvent nécessaire.

Créer un compte et obtenu un token pour https://openweathermap.org/appid#start puis https://home.openweathermap.org/api_keys Puis faites des requêtes pour obtenir la météo d'une ville. Attention, cela prend du temps.

```
curl -s "https://api.openweathermap.org/data/2.5/weather?q=Toulon,fr&appid=$TOKEN" | jq
```

Le résultat est en JSON, pour le formater, on peut utiliser `jq`. La commande `jq` permet de formater et filtrer des données JSON.

Il est aussi possible d'utiliser des options curl pour envoyer des paramètres.

```
curl -v "https://api.openweathermap.org/data/2.5/weather" \
     -G \
     --data-urlencode "q=Toulon,fr" \
     --data-urlencode "units=metric" \
     --data-urlencode "appid=$TOKEN"
```

5.4 Manipulation des headers et formats

Pour spécifier le format de la réponse ou envoyer des headers spécifiques, on peut utiliser des options curl.

```
# Demander du JSON
curl -v -H "Accept: application/json" "https://api.github.com/users/octocat"

# Demander un format spécifique de GitHub
curl -v -H "Accept: application/vnd.github.v3+json" "https://api.github.com/users/octocat"
```

Découvrez l'api Github en utilisant votre propre token. https://docs.github.com/en/rest/authentication/authenticating-to-the-rest-api

```
# Bearer Token
curl -v -H "Authorization: Bearer YOUR_TOKEN" "https://api.github.com/user"
```

5.5 Exercices pratiques

1. Utilisez curl pour explorer l'API OpenWeatherMap en pour afficher la météo de plusieurs villes
2. Compléter le script shell suivant pour afficher la latitude et la longitude de l'ISS toutes les 30secondes.

```
# Script pour suivre l'ISS
while true; do
    curl ... | \
    jq ...
    sleep 30
done
```

3. Décrouvez des API publiques utiles :

- NASA : https://api.nasa.gov/
- openstreetmap : `curl -s "https://nominatim.openstreetmap.org/search?q=Eiffel+Tower&format=json"|jq`
- SNCF: https://ressources.data.sncf.com/api/v1/console/datasets/1.0/search/

- datalab et etalanb : https://api.gouv.fr/v1/apis

6 Exercice 2: Explorer OpenAPI

OpenAPI est une spécification pour décrire des APIs REST. Elle permet de générer des clients dans de nombreux langages. Elle permet aussi de générer de la documentation et des tests.

1. Visitez https://petstore.swagger.io/
2. Examinez la structure du document OpenAPI
3. Testez les endpoints via l'interface Swagger UI

7 Exercice 3. Client Java avec Jersey

A partir des exemples vus en cours et de la documentation de Jersey (https://eclipse-ee4j.github.io/jersey.github.io/documentation/latest31x/client.html), créez un client REST pour l'API REST de votre choix.

8 Exercice 4 Client Quarkus (Optionnel)

Quarkus est un framework Java pour les applications cloud-native. Il propose un client REST basé sur RESTEasy. A partir du tutoriel https://quarkus.io/guides/rest-client créer un client REST pour l'API de votre choix.