

# Exercice – Recherche et Documentation d’une API existante

## Objectif pédagogique

Cet exercice a pour objectif de vous entraîner à **analyser une API existante** et à **rédiger une documentation claire et structurée**, compréhensible par des développeurs débutants **et** des profils fonctionnels.

Vous ne devez **pas développer d’API**. Vous devez apprendre à :

- rechercher une API publique sur Internet,
  - comprendre sa documentation officielle,
  - identifier et décrire précisément ses endpoints,
  - reformuler l’information de manière claire et synthétique.
- 

## Organisation du travail

- Travail **en groupe de 2 à 3 étudiants**
  - Une seule API par groupe
  - Une seule documentation finale à rendre
- 

## Étape 1 – Recherche d’une API

Chaque groupe doit choisir **une API publique existante**, par exemple :

- API météo
- API de films / séries
- API sportive
- API de musique
- API de données ouvertes (open data)

 Contraintes :

- L’API doit être **documentée publiquement**
- Elle doit proposer **au moins 4 endpoints**
- Elle doit utiliser le protocole HTTP (API REST ou assimilée)

Vous devez indiquer clairement :

- le nom de l’API,
  - l’éditeur / l’organisation,
  - l’URL de la documentation officielle.
-

## Étape 2 – Compréhension globale de l'API

Dans une section introductive (10–15 lignes), expliquez :

- à quoi sert cette API,
- à qui elle s'adresse,
- dans quel contexte on l'utiliserait (application mobile, site web, analyse de données, etc.).

## Étape 3 – Documentation des endpoints (travail principal)

Vous devez **documenter chaque endpoint important de l'API** en respectant **le format imposé ci-dessous**.

**Pour chaque endpoint, vous devez fournir :**

1. **Nom de l'endpoint** (clair et explicite)
2. **URL complète**
3. **Verbe HTTP** (GET, POST, PUT, PATCH, DELETE)
4. **Description fonctionnelle** (ce que fait l'endpoint)
5. **Paramètres possibles**
6. paramètres d'URL (path)
7. paramètres de requête (query)
8. paramètres dans le corps de la requête (body)
9. **Résultat attendu**
10. type de réponse (liste, objet, message)
11. exemple simplifié de réponse
12. **Codes de réponse possibles** (200, 400, 404, etc.)

### Exemple de documentation attendue

 Cet exemple est fictif et sert uniquement de modèle.

#### Endpoint : Récupérer la liste des films

- **URL :** `https://api.movies-example.com/films`
- **Verbe HTTP :** `GET`

#### Description

Permet de récupérer la liste des films disponibles dans le catalogue.

#### Paramètres

Nom	Type	Obligatoire	Description
<code>title</code>	string	Non	Filtre les films par titre (recherche partielle)

Nom	Type	Obligatoire	Description
<code>min_rating</code>	number	Non	Note minimale du film
<code>max_rating</code>	number	Non	Note maximale du film
<code>page</code>	integer	Non	Numéro de page (par défaut : 1)

### Résultat attendu

L'API retourne une liste paginée de films.

```
{
  "page": 1,
  "results": [
    {
      "id": 42,
      "title": "Inception",
      "rating": 8.7,
      "release_date": "2010-07-16"
    }
  ]
}
```

### Codes de réponse

- `200 OK` : requête réussie
- `400 Bad Request` : paramètres invalides

### Endpoint : Récupérer le détail d'un film

- URL : `https://api.movies-example.com/films/{id}`
- Verbe HTTP : `GET`

### Description

Retourne les informations détaillées d'un film à partir de son identifiant.

### Paramètres

Nom	Type	Obligatoire	Description
<code>id</code>	integer	Oui	Identifiant unique du film

### Résultat attendu

```
{
  "id": 42,
  "title": "Inception",
```

```
"rating": 8.7,  
"overview": "Film de science-fiction..."  
}
```

#### Codes de réponse

- 200 OK
- 404 Not Found : film inexistant

## Étape 4 – Qualité rédactionnelle

Votre documentation doit être :

- structurée (titres, tableaux, sections claires),
- rédigée en français correct et professionnel,
- compréhensible par un étudiant n'ayant jamais utilisé l'API.

## Format du rendu

Un document unique au format :

- **PDF** ou
- **Markdown (.md)**

Nom du fichier :

NomGroupe\_Documentation\_API.pdf



## Barème indicatif (/20)


Critère	Points
Choix pertinent de l'API	3
Compréhension globale de l'API	4
Qualité de la documentation des endpoints	8
Clarté et structuration du document	3
Qualité rédactionnelle	2

## Interdictions

- Copier-coller intégralement la documentation officielle
  - Utiliser du jargon technique non expliqué
  - Fournir uniquement des captures d'écran
- 

## Compétences évaluées

- Lecture et compréhension de documentation technique
  - Synthèse et reformulation
  - Communication écrite
  - Rigueur et structuration
- 

 **Cet exercice simule une situation réelle où un individu doit comprendre et expliquer une API qu'il n'a pas conçue.**

Bon travail 