

Les liens hypermédias

Sur un site web, vous ne tapez jamais d'URL à la main : vous arrivez sur la page d'accueil et vous cliquez sur des liens. Le site peut réorganiser toutes ses adresses, vous ne vous en rendez pas compte.

HATEOAS (*Hypermedia As The Engine Of Application State*) applique la même idée aux API : chaque réponse contient des **liens** vers les ressources liées et les actions possibles. Le client n'a besoin de connaître qu'une seule URL, le point d'entrée.

C'est la dernière contrainte de l'interface uniforme de REST, et le niveau 3 du modèle de Richardson (chapitre 2).

Partir du point d'entrée

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

```
{
  "name": "API-ritif",
  "version": "1.0.0",
  "_links": {
    "self": {"href": "/v1/"},
    "cocktails": {"href": "/v1/cocktails"},
    "customers": {"href": "/v1/customers", "method": "POST"},
    "me": {"href": "/v1/me"},
    "orders": {"href": "/v1/orders"},
    "docs": {"href": "/docs"}
  }
}
```

On suit `cocktails`, puis le lien `self` d'un cocktail :

```
{
  "id": 3,
  "name": "Segfault Sunrise",
  "price": 9.5,
```

```
"_links": {
  "self":    {"href": "/v1/cocktails/3"},
  "reviews": {"href": "/v1/cocktails/3/reviews"},
  "order":   {"href": "/v1/orders", "method": "POST"}
}
```

Le cocktail indique lui-même où trouver ses avis et comment le commander. Aucune URL n'a été construite par le client.

Anatomie d'un lien

- La **clé** est la **relation** (*rel*) : elle dit ce que représente le lien par rapport à la ressource courante.
- `href` est l'adresse.
- `method` (extension d'API-ritif) précise la méthode HTTP quand ce n'est pas `GET`.

Les relations courantes

Certaines relations sont standardisées par l'IANA, et tout le monde les comprend :

Relation	Sens
<code>self</code>	La ressource elle-même
<code>next</code> , <code>prev</code> , <code>first</code> , <code>last</code>	Navigation dans une liste paginée
<code>collection</code>	La collection à laquelle appartient l'élément
<code>item</code>	Un élément d'une collection
<code>related</code>	Une ressource liée
<code>edit</code>	L'adresse pour modifier la ressource

Les autres sont propres à votre métier : `cancel`, `reviews`, `order`, `pay`... Choisissez des noms explicites et documentez-les.

Ce que ça change pour le client

- Le serveur peut **changer ses URL** sans casser les clients, puisqu'ils suivent les liens.
- Le client **découvre** ce qui est possible au lieu de le deviner.

- Les **règles métier** restent côté serveur : si une action n'est pas permise, son lien n'apparaît pas (page « Des liens qui dépendent de l'état »).

Vous utilisez HATEOAS depuis le chapitre 4 sans le savoir : parcourir une liste en suivant `_links.next`, c'est exactement ça.

Revision #3

Created 2 October 2026 20:00:58 by Thibaud FRICHET

Updated 12 October 2026 22:02:48 by Thibaud FRICHET