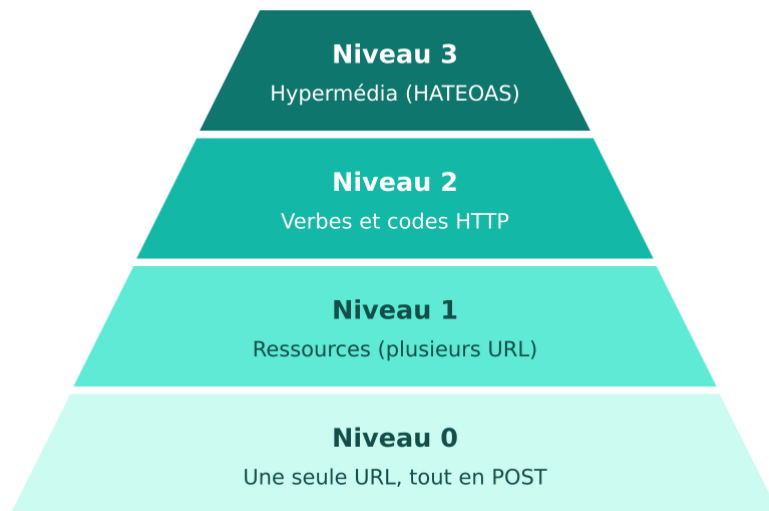


Le modèle de maturité de Richardson

En 2008, [Leonard Richardson](#) propose un modèle en 4 niveaux pour mesurer à quel point une API exploite le web. Chaque niveau ajoute un des principes REST.

Modèle de maturité de Richardson



Fil conducteur : dans l'API d'une bibliothèque, **annuler la réservation 42**.

Niveau 0 - Le marécage du POX (Plain Old XML)

Une seule URL, une seule méthode (`POST`), et l'action est décrite dans le body. HTTP ne sert que de tuyau. C'est le cas de SOAP et de XML-RPC (POX : *Plain Old XML*).

```
POST /api
```

```
Content-Type: application/json
```

```
{"action": "cancelReservation", "reservationId": 42}
```

```
HTTP/1.1 200 OK
```

```
{"success": false, "error": "Livre déjà retiré"}
```

Niveau 1 - Les ressources

Chaque ressource a sa propre URL, mais on utilise encore `POST` pour tout, avec des verbes dans les URL.

```
POST /api/reservations/42/cancel
```

Niveau 2 - Les verbes et codes HTTP

On utilise la bonne méthode HTTP pour chaque action, et les bons codes de statut. C'est le niveau de la grande majorité des API « REST ».

```
DELETE /v1/reservations/42
```

```
HTTP/1.1 409 Conflict
```

Niveau 3 - Les contrôles hypermédias

Les réponses contiennent des **liens** vers les actions possibles. Le client n'a plus besoin de deviner les URL : il suit les liens. C'est **HATEOAS**.

```
{
  "id": 42,
  "status": "waiting",
  "_links": {
    "self": {"href": "/v1/reservations/42"},
    "cancel": {"href": "/v1/reservations/42", "method": "DELETE"},
    "book": {"href": "/v1/livres/3"}
  }
}
```

Si le livre a déjà été retiré, le lien `cancel` n'apparaît plus : le client sait qu'il ne peut plus annuler, sans connaître les règles métier.

Selon Fielding, seul le niveau 3 est vraiment REST. On y revient en détail au chapitre 9.

Revision #5

Created 2 October 2026 20:00:43 by Thibaud FRICHET

Updated 11 October 2026 20:27:10 by Thibaud FRICHET