## 2. Wat is een resource?

De belangrijkste **abstractie van informatie** in REST is een [bron](https://restfulapi.net/resource-naming/). Alle informatie die we kunnen noemen, kan een bron zijn. Een REST-bron kan bijvoorbeeld een document of afbeelding, een tijdelijke service, een verzameling andere bronnen of een niet-virtueel object (bijvoorbeeld een persoon) zijn.

De status van de resource, op een bepaald moment, wordt de **resourcerepresentatie genoemd**.

De resourcerepresentaties bestaan uit:

* de **gegevens**
* de **metadata** die de gegevens beschrijven
* en de **hypermedialinks** die de clients kunnen helpen bij de overgang naar de volgende gewenste staat.

Een REST API bestaat uit een verzameling van onderling verbonden bronnen. Deze set bronnen staat bekend als het ***resourcemodel*** van de REST API.

### 2.1. Resource-id’s

REST gebruikt bron-id’s om elke bron te identificeren die betrokken is bij de interacties tussen de client en de serveronderdelen.

### 2.2. Hypermedia

De gegevensindeling van een representatie wordt een [mediatype genoemd](https://www.iana.org/assignments/media-types/media-types.xhtml). Het mediatype identificeert een specificatie die definieert hoe een representatie moet worden verwerkt.

**Een RESTful API ziet eruit als** [***hypertekst***](https://restfulapi.net/hateoas/)**.** Elke adresseerbare informatie-eenheid draagt een adres, hetzij expliciet (bijv. Link- en id-attributen) of impliciet (bijvoorbeeld afgeleid van de definitie en representatiestructuur van het mediatype).

*Hypertekst* (of hypermedia) betekent de **gelijktijdige presentatie van informatie en controles**, zodat de informatie de betaalbaarheid wordt waardoor de gebruiker (of automaat) keuzes verkrijgt en acties selecteert.

Vergeet niet dat hypertekst geen HTML (of XML of JSON) in een browser hoeft te zijn. Machines kunnen koppelingen volgen wanneer ze de gegevensindeling en relatietypen begrijpen.

- Roy Fielding

### 2.3. Zelf beschrijvend

Verder **moeten resourcerepresentaties zelfbeschrijvend zijn**: de klant hoeft niet te weten of een resource een werknemer of een apparaat is. Het moet werken op basis van het mediatype dat aan de bron is gekoppeld.

Dus in de praktijk zullen we veel **aangepaste mediatypen** maken - meestal één mediatype dat aan één bron is gekoppeld.

Elk mediatype definieert een standaardverwerkingsmodel. HTML definieert bijvoorbeeld een weergaveproces voor hypertekst en het browsergedrag rond elk element.

Mediatypen hebben geen relatie met de bronmethoden GET/PUT/POST/DELETE/… anders dan het feit dat sommige mediatype-elementen een procesmodel definiëren dat gaat als “ankerelementen met een attribuut maken een hypertekstkoppeling die, wanneer geselecteerd, een ophaalverzoek (GET) aanroept op de URI die overeenkomt met het kenmerk -encoded.”href``CDATA``href