REST is een acroniem voor **REpresentational** **State** **Transfer** en een architecturale stijl voor **gedistribueerde hypermediasystemen**. Roy Fielding presenteerde het voor het eerst in 2000 in zijn beroemde [proefschrift](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm).

Net als andere bouwstijlen heeft REST zijn leidende principes en beperkingen. Aan deze principes moet worden voldaan als een service-interface **RESTful** moet worden genoemd.

Een web-API (of webservice) die voldoet aan de REST-architectuurstijl is een *REST API*.

## 1. Leidende principes van REST

De zes leidende principes of [beperkingen van de RESTful-architectuur](https://restfulapi.net/rest-architectural-constraints/) zijn:

### 1.1. Uniforme interface

Door het [principe van algemeenheid](https://www.d.umn.edu/~gshute/softeng/principles.html) toe te passen op de componenteninterface, kunnen we de algehele systeemarchitectuur vereenvoudigen en de zichtbaarheid van interacties verbeteren.

Meerdere architecturale beperkingen helpen bij het verkrijgen van een uniforme interface en het sturen van het gedrag van componenten.

Met de volgende vier beperkingen kunt u een uniforme REST-interface bereiken:

* **Identificatie van bronnen** : de interface moet elke bron die betrokken is bij de interactie tussen de client en de server op unieke wijze identificeren.
* **Manipulatie van bronnen door middel van representaties** - De bronnen moeten uniforme representaties hebben in de serverrespons. API-consumenten moeten deze representaties gebruiken om de bronstatus op de server te wijzigen.
* **Zelfbeschrijvende berichten** - Elke bronrepresentatie moet voldoende informatie bevatten om te beschrijven hoe het bericht moet worden verwerkt. Het moet ook informatie bevatten over de aanvullende acties die de client op de bron kan uitvoeren.
* **Hypermedia als de engine van de toepassingsstatus**: de client mag alleen de initiële URI van de toepassing hebben. De clienttoepassing moet alle andere bronnen en interacties dynamisch aansturen met behulp van hyperlinks.

### 1.2. Client-server

Het client-serverontwerppatroon dwingt de **scheiding van zorgen af**, waardoor de client en de serveronderdelen onafhankelijk van elkaar kunnen evolueren.

Door de zorgen over de gebruikersinterface (client) te scheiden van de gegevensopslag (server), verbeteren we de portabiliteit van de gebruikersinterface over meerdere platforms en verbeteren we de schaalbaarheid door de servercomponenten te vereenvoudigen.

Terwijl de client en de server evolueren, moeten we ervoor zorgen dat de interface / het contract tussen de client en de server niet wordt verbroken.

### 1.3. Staatloos

[Statelessness](https://restfulapi.net/statelessness/) schrijft voor dat elk verzoek van de client naar de server alle informatie moet bevatten die nodig is om het verzoek te begrijpen en te voltooien.

De server kan niet profiteren van eerder opgeslagen contextinformatie op de server.

Om deze reden moet de clienttoepassing de sessiestatus volledig behouden.

### 1.4. Cachebaar

De [cachebare beperking](https://restfulapi.net/caching/) vereist dat een antwoord zichzelf impliciet of expliciet labelt als cacheable of niet-cacheable.

Als het antwoord in de cache kan worden opgeslagen, krijgt de clienttoepassing het recht om de antwoordgegevens later opnieuw te gebruiken voor gelijkwaardige aanvragen en een opgegeven periode.

### 1.5. Gelaagd systeem

Met de gelaagde systeemstijl kan een architectuur worden samengesteld uit hiërarchische lagen door het gedrag van componenten te beperken.

In een gelaagd systeem kan elke component bijvoorbeeld niet verder kijken dan de directe laag waarmee ze communiceren.

### 1.6. Code op aanvraag (*optioneel*)

REST maakt het ook mogelijk om de functionaliteit van de client uit te breiden door het downloaden en uitvoeren van code in de vorm van applets of scripts.

De gedownloade code vereenvoudigt clients door het aantal functies dat vooraf moet worden geïmplementeerd te verminderen. Servers kunnen een deel van de functies leveren die aan de client worden geleverd in de vorm van code en de client hoeft alleen de code uit te voeren.