**Understanding HATEOAS**

HATEOAS (Hypertext as the Engine of Application State) is a [constraint of the REST application achitecture](http://en.wikipedia.org/wiki/HATEOAS).

A hypermedia-driven site provides information to navigate the site's [REST](http://spring.io/understanding/rest) interfaces dynamically by including hypermedia links with the responses. This capability differs from that of SOA-based systems and WSDL-driven interfaces. With SOA, servers and clients usually must access a fixed specification that might be staged somewhere else on the website, on another website, or perhaps distributed by email.

**Examples:**

The following code represents a Customer object.

**class** **Customer** **{**

String name**;**

**}**

A simple JSON presentation is traditionally rendered as:

{

"name" : "Alice"

}

The customer data is there, but the data contains nothing about its relevant links.

A HATEOAS-based response would look like this:

{

"name": "Alice",

"links": [ {

"rel": "self",

"href": "http://localhost:8080/customer/1"

} ]

}

This response not only has the person's name, but includes the self-linking URL where that person is located.

* **rel** means relationship. In this case, it's a self-referencing hyperlink. More complex systems might include other relationships. For example, an order might have a "rel":"customer" relationship, linking the order to its customer.
* **href** is a complete URL that uniquely defines the resource.

It's possible to build more complex relationships. With HATEOAS, the output makes it easy to glean how to interact with the service without looking up a specification or other external document.