*Sources - https://www.baeldung.com/spring-rest-hal*

*Hateoas - HATEOAS acronyms for Hypermedia as the Engine of Application State. The term hypermedia refers to content that contains a link to other forms of media like images, movies, and text. Using HATEOAS, a client interacts with a network application, whose application server provides information dynamically through Hypermedia.*

*Spring-HATEOAS*

*Spring-HATEOAS is the library of APIs. We can use these APIs for creating REST representations that follow the HATEOAS principle while working with Spring MVC.*

* *It supports hypermedia formats like HAL.*
* *It provides Link builder API to create links pointing to MVC controller methods.*
* *Model classes for the link, resource representation models.*

*Spring Boot does the following tasks:*

* *Configure HAL support*
* *Register support for entity links*
* *Wire up message converter support*

*Suppose, we have requested a GET request for localhost:8080/users/1, it returns the details of user id 1. Along with this, it also returns a field called link that contains a link (localhost:8080/users) of all users so that consumers can retrieve all the users. This concept is called HATEOAS.*

*Step1: Open the pom.xml and add the spring-boot-starter-hateoas dependency.*

*<dependency>d*

*<groupId>org.springframework.boot</groupId>*

*<artifactId>spring-boot-starter-hateoas</artifactId>*

*</dependency>*

*Step 2: Open UserResource.java and copy the retrieveUser() method.*

*Step 3: Paste the method and make the following changes:*

* *Create a constructor of Resource class.*

1. Resource<User> resource = **new** Resource<User>(User)

Remember that import the Resource class of org.springframework.hateoas package.

* Add a link to retrieve all users by using the **ControllerLinkBuilder** class. It enables us to create a link from methods.
* Import the ControllerLinkBuilder.

1. **import** **static** org.springframework.hateoas.mvc.ControllerLinkBuilder.

* Use the method **linkTo()** of ControllerLinkBuilder class. It creates a new ControllerLinkBuilder with a base of mapping annotated to the given controller class.

1. ControllerLinkBuilder linkTo=linkTo(methodOn(**this**.getClass().retrieveAllUsers());

methodOn() is a wrapper for **DummyInvocationUtils.methodOn(class, Object)** to be available in case you work with the static import of ControllerLinkBuilder.

* Add this link to the resource with the name which we want to use inside the HATEOAS.

1. resource.add(linkTo.withRel("all-users"));

**withRel(String rel)** is the method that creates the link built by the current builder instance with the given rel. The parameter rel must not be null.

* Return the **resource** instead of the user.
* Change the return type of method to **Resource**.