

3.12.3 Accéder au service REST exposé

Introduction

Deux solutions sont présentés ici :

- Un accès en JQuery. Dans ce cas c'est du code javaScript sur le navigateur de l'utilisateur qui fait la requête REST.
- Dans ce cas c'est le code Java sur le serveur qui fait la requête REST.
 - Un accès avec RestTemplate.
 - Un accès avec le client Jersey

Dans les exemples ci-dessous on utilise le flux produit par l'URL en `/users` qui renvoie une liste de `User`

Accès JQuery

```
<div id="users" />
<script>
  //
    (function($) {
      $(document).ready(function() {
        $.getJSON("#{welcomeController.restURL}", function(data) {
          var html = "&lt;ul&gt;";
          $.each(data, function(i, item) {
            html += "&lt;li&gt;" + item.displayName + "&lt;/li&gt;";
          });
          html += "&lt;/ul&gt;";
          $(html).appendTo("#users");
        });
      });
    })(jQuery);
  //]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="73 511 511 525" data-label="Text"><p>On positionne une balise <code>&lt;div id="users" /&gt;</code> qui sera destinataire des résultats.</p></div><div data-bbox="73 533 916 558" data-label="Text"><p>La fonction JQuery <code>getJSON()</code> appelle l'URL REST (dont la valeur nous est ici donnée par le contrôleur JSF). Lors de l'itération sur la liste on construit une liste html. Cette liste est ensuite insérée dans la balise <code>&lt;div&gt;</code> cible : <code>$(html).appendTo("#users")</code></p></div><div data-bbox="73 579 306 601" data-label="Section-Header"><h3>Accès RestTemplate</h3></div><div data-bbox="73 609 922 635" data-label="Text"><p>Dans la vue (JSF ici) on utilise un composant de type liste (au sens html du terme) qui fait appel au contrôleur pour obtenir une liste d'objets (au sens java du terme) :</p></div><div data-bbox="82 650 550 711" data-label="Text"><pre>&lt;h:form&gt;
  &lt;p:dataList var="user" value("#{welcomeController.restUsers}"&gt;
    #{user.displayName}
  &lt;/p:dataList&gt;
&lt;/h:form&gt;</pre></div><div data-bbox="73 736 274 750" data-label="Text"><p>Voyons le code Java du contrôleur :</p></div><div data-bbox="82 765 572 826" data-label="Text"><pre>public List&lt;User&gt; getRestUsers() {
    RestTemplate template = new RestTemplate();
    List&lt;User&gt; ret = template.getForObject(getRestURL(), List.class);
    return ret;
}</pre></div><div data-bbox="73 851 797 865" data-label="Text"><p>Ce cas est très simple. Pour avoir plus d'information sur l'utilisation de RestTemplate Cf. <a href="#">La documentation Spring sur RestTemplate</a></p></div><div data-bbox="73 886 297 909" data-label="Section-Header"><h3>Accès Jersey Client</h3></div><div data-bbox="73 916 424 931" data-label="Text"><p>Dans cet exemple, on récupère un utilisateur à partir de son ID.</p></div>
```

```
//Initialisation du client
ClientConfig clientConfig = new DefaultClientConfig();
Client client = Client.create(clientConfig);

//Configurer l'authentification du client si le service REST l'exige
client.addFilter(new HTTPBasicAuthFilter(adminUsername, adminPassword));

//Creation d'une instance WebResource
WebResource webResource = client.resource(UriBuilder.fromUri(http://urlDeMonWebService/racine).build());

// Récupérer un user à partir de son Id au format JSON
ClientResponse response = webResource.path("user")
    .path(idDuUser)
    .type(MediaType.APPLICATION_JSON_TYPE)
    .accept(MediaType.APPLICATION_JSON_TYPE)
    .get(ClientResponse.class);

//Transformation du JSON
User myUser = response.getEntity(User.class);
```