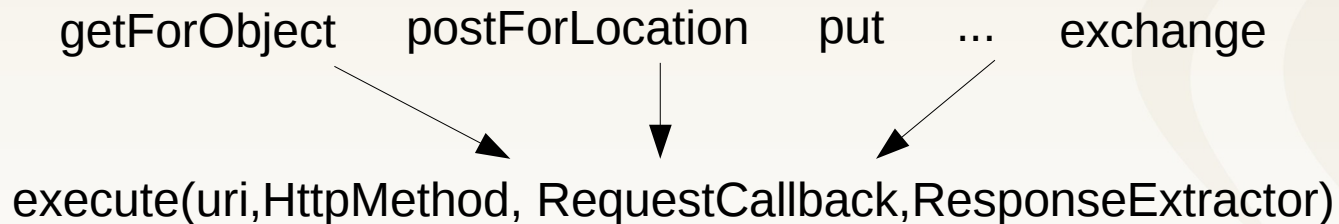


RestTemplate ile REST Servislerine Erişim



RestTemplate Execute

- RestTemplate sınıfındaki bütün metotlar arka tarafta işi **execute metoduna** delege ederler



- RestTemplate.execute() metodu ile HTTP **request** ve **response** nesneleri üzerinde **doğrudan işlem** yapılabilir

RestTemplate Kullanım

Örneği: Execute

```
RequestCallback requestCallback = new RequestCallback() {

    @Override
    public void doWithRequest(ClientHttpRequest request)
        throws IOException {
        request.getHeaders()
            .set("Authorization", "Basic aHR0cHdhdGNoOmY=");
    }
};

ResponseExtractor<Vet> responseExtractor = new ResponseExtractor<Vet>() {

    @Override
    public Vet extractData(ClientHttpResponse response)
        throws IOException {
        HttpStatus statusCode = response.getStatusCode();
        InputStream inputStream = response.getBody();
        //...
        return vet;
    }
};

Vet vet = restTemplate.execute("http://localhost:8080/mvc/vets/1",
    HttpMethod.GET, requestCallback, responseExtractor);
```

RestTemplate ve JSON ObjectMapper Kullanımı

- REST servisinden dönen JSON String'i istemci tarafında Jackson **ObjectMapper** kullanılarak **manuel biçimde** de DTO'ya dönüştürülebilir

```
String url = "http://localhost:8080/mvc/rest/owners/1";  
String jsonContent = restTemplate.getForObject(url, String.class);  
  
ObjectMapper objectMapper = new ObjectMapper();  
OwnerDTO owner = objectMapper.readValue(  
    jsonContent.getBytes(), OwnerDTO.class);
```

RestTemplate ve ClientHttpRequestInterceptor

- **ClientHttpRequestInterceptor** nesneleri vasıtası ile istemci tarafında HTTP request öncesi ve HTTP response döndükten sonra ilave işlemler yapılabilir
- Örneğin **BasicAuthorizationInterceptor** yardımı ile HTTP isteğine basic authentication header değerleri yerleştirilebilir

BasicAuthorization Interceptor Örneği

```
RestTemplate restTemplate = new RestTemplate();

BasicAuthorizationInterceptor basicAuthorizationInterceptor =
    new BasicAuthorizationInterceptor("user1", "secret");
restTemplate.setInterceptors(
    Arrays.asList(basicAuthorizationInterceptor));

Owner owner = restTemplate
    .getForObject("http://localhost/owners/1", Owner.class);
```

RestTemplate ve HTTP Client Konfigürasyonu

- RestTemplate default constructor ile yaratılırsa Java'nın **built-in http client kütüphanesi** kullanılır
- **ClientHttpRequestFactory** alan constructor'ı ile bu değiştirilebilir
- Bu sayede **Apache Commons HttpClient** kullanılabilir
- Apache Commons HttpClient **credentials desteği** sunar

RestTemplate ve HTTP Client Konfigürasyonu

```
CredentialsProvider credentialsProvider = new BasicCredentialsProvider();

credentialsProvider.setCredentials(
    AuthScope.ANY,
    new UsernamePasswordCredentials("user1", "secret"));

HttpClient httpClient = HttpClientBuilder.create()
    .setDefaultCredentialsProvider(credentialsProvider).build();

RestTemplate restTemplate = new RestTemplate(
    new HttpClientHttpRequestFactory(httpClient));

Vet vet = new Vet("Kenan", "Sevindik", 1999);

restTemplate.postForObject("http://localhost:8080/petclinic/mvc/vet",
    vet, Void.class);
```


İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

