# RestApi nedir?

Rest http protokolü ile url üzerinden veri alışverişini sağlayan bir yapıdır. Client’in gönderdiği request e karşılık bir json(javascript object notation) dosası döner.

http durum kodları:

100-199:bilgi

200-299:başarılı

300-399:redirection

400-499:client error

401:Authetication

404:file not found

500-599:server error

JSON:Json içerisindeki nesnenin değişken adları çift tırnak içerisinde yazılır ve : sonrası değeri yazılır. Integer değerler tırnak içinde yazılmazken, string değerler tırnak içinde yazılır. Boolen değerler de tırnak içinde yazılmazlar ve null değer alan değişkenler için null değerleri de tırnak içine alınmadan yazılır.Ör:

{

“id”:14,

“firstName”:”Mario”,

“lastName”:”Rossi”,

“active”: true,

“courses”:null

}

İç içe geçmiş JSON objeleri:

{

“id”:14,

“firstName”:”Mario”,

“lastName”:”Rossi”,

“active”: true,

“address”:

{

“street”:”100 Main st”,

“city”:”Philadelphia”,

“state”:”pennsylvania”,

“zip”:”19103”,

“country”:”USA”

}

}

Json içerisinde dizi kullanımı:

{

“id”:14,

“firstName”:”Mario”,

“lastName”:”Rossi”,

“active”: true,

“languages”: [“Java”,”C#”,”Python”,”Javascript”]

}

# HTTP ve Crud operasyonları:

POST yeni bir entity yaratma

GET entity listesi veya tek bir entityi okuma

PUT var olan bir entityi güncelleme

DELETE var olan bir entityi silme

http response mesajı:

Response line: server protokolü ve durum kodu

Header değişkenleri: response metadata

Message body: mesaj içeriği

# MIME Content types:

Mail-Extension(çok amaçlı internet poste uzantısı) gönderilen mesajın gerçek içeriğini tanımlar.

Ör:

Text/html,text/plain

# @PathVaraible

Requestlerle gelen değişken değerlerini method parametreleriyle eşleştirmeye yarar. Örneğin Id’ye göre arama yapıldığında, requestte gelen id değerini method parametresindeki idye eşleme yapılır apideki varaiblle ismiyle metod parametresi aynı isimle tanımlanmalı.

# Rest Exception Handling

Tanımladığımız Exception türünde veya belirli bir exception türünde hataları yakalamak için hata yakalanınca yapılacak işlerin bulunduğu metod üzerinde ExceptionHandler anotasyonu koymak gerekir. Parametre kısmına da ister belirlediğimiz bir hata türü ister belirli bir hata türünde parametre ekleyip ilgili hata oluştuğunda yapılacak işleri belirleyebiliriz.

# Global Exception Handling

## @ControllerAdvice

Bir filtre gibi çalışır, apilere gelen istekleri önceden işlemek için ve istisnaları ele almak için yanıtları sonradan işlemek için de kullanılabilir. Global hata yakalama yapmak için güzeldir. Bu anotasyon AOP için gerçek dünya kullanım örneğidir.

# Spring Rest Api Design

Api design için öncelikle Api’nin kullanıcılarını ve Api’yi nasıl kullanacaklarını bilmeliyiz. Bu bilgilere göre de Api’yi tasarlamalıyız.

İlk adım olarak Api gereksinimlerini gözden geçirmek gerekir. Ana kaynakları ve entityleri gerekli şekilde tanımlamak gerekir. Ardında bir kaynağa işlem atamak için http kullanmak gerekir.

# Spring Data JPA

Spring data jpa JpaRepository interfacesini sağlar, bu interface sayesinde Entity ile Primarykey bilgisini vererek kolayca Repository katmanını oluşturabiliriz.

# JPQL

# Query DSL

# Defining custom methods (low level coding)

# Spring Data Rest

Bu kütüphaneyi mavenda pom.xml dosyasına ekledikten sonra spring otomatik olarak RestController ve Service katmanlarını oluşturur. CRUD endpointleri default olarak entity adının küçük harfle başlayıp sonuna s takısı alması şeklinde oluşturulur.

## HATEOAS: Hypermedia as the Engine of Application State

Responslardaki meta-data ları işleyen bir yapı

## RepositoryRestResource

Bu anotaston repository katmanına eklenir ve spring data restin otomatik olarak oluşturacağı endpointlerin default isimlendirmesini gerçekleştirir.

@RepositoryRestResource(path = "members")

## Base-path

Endpointlerin başına default bir isim eklemek için kullanılır

spring.data.rest.base-path=/api

## Default page size

Veri tabanından çekilen veriler için tek seferde kaç veri çekileceğini sınırlamak için kullanılan configürasyon kodu

spring.data.rest.default-page-size=3

## Sorting

<http://localhost:8080/api/employee?sort=firstName,desc>

bu komut ile gelen verileri soyada göre azalan sıralama yapabiliriz, istersek asc olarak artan veya soyada göre de bu işlemleri gerçekleştirebiliriz, data rest ile bu özellikler kolayca sağlanmış olur.