

SpringMVC interview

1-SpringMVC Nedir?

- **SpringMVC**, Spring Framework'ün bir modülüdür ve Model-View-Controller (MVC) tasarım desenini temel alarak web uygulamaları geliştirmeyi sağlar.
- Amaç, iş mantığını (Model), kullanıcı arayüzünden (View) ayırmak ve her iki katman arasındaki iletişimi Controller ile yönetmektir.

2. Spring WebMVC Modülünün Ana Bileşenleri Nelerdir?

1. **DispatcherServlet**: Gelen tüm HTTP isteklerini karşılar.
2. **Handler Mapping**: İsteği uygun bir Controller'a yönlendirir
3. **Controller**: İstekleri işleyip uygun yanıtı döndürür.
4. **View-View Resolver**: Controller tarafından döndürülen yanıtı doğru bir View ile eşleştirir.
5. **Model**: Uygulama verilerini içerir ve View katmanına taşır.
6. **ModelAndView**: Controllerdan model ve görünüm doayasının ismi bu bileşen il viewresolvera gönderilir

3. DispatcherServlet Nedir ve Nasıl Çalışır?

- **DispatcherServlet**, SpringMVC'nin önkontrolcüsüdür.
- **Çalışma Adımları**:
 1. Tüm HTTP isteklerini alır.
 2. İsteği bir **Handler Mapping** ile uygun bir Controller'a yönlendirir.
 3. Controller'dan dönen veriyi işler ve bir **View Resolver** ile uygun bir görünüm döndürür.

4. Handler Mapping ve View Resolver Nedir?

DispatcherServlet'in bileşenleridir.

- **Handler Mapping**:

İstek URL'sine göre hangi Controller'ın çalıştırılacağını belirler.

Örnek: `/users` isteği `UserController` ile eşleştirilir.
- **View Resolver**:

Controller'dan dönen görünüm ismini fiziksel bir dosya ile eşleştirir.

Örnek: `user` görünümü `WEB-INF/view/user.jsp` dosyasına yönlendirilir.

5. ModelAndView Nedir ve Nasıl Kullanılır?

Hem veriyi işler hemde görünümü bir arada taşıyan bir classtır

6. SpringMVC'de Yapılandırma Nasıl Yapılır?

Abstract bir classın metodları ile DispatcherServlet konfigüre edilir ve başlatılır. `@EnableWebMvc` ve `@Configuration` anotasyonları ile yapılandırma yapılır. HandlerMapping, ViewResolver ve database propertyleri (JDBC, Hibernate) için yapılandırma yapılır.

7. Spring @Controller Nedir?

- `@Controller`, bir sınıfın HTTP isteklerini karşılayan metodları tanımladığını belirtir.
- Bu classta metodlar genellikle view(JSP,HTML,...) döndürür.
- ModelAndView veya String döndürülür.

8. @Controller ve @RestController Anotasyonları Arasındaki Fark Nedir?

- `@RestController`, bir sınıfın Rest mimarisine göre HTTP isteklerini karşılayan metodları tanımladığını belirtir. Yani bu classta Restful Servisler geliştirilecek.
- Bu classta metodlar genellikle XML,TXT,**JSON**,HTML... döndürür.
- `@ResponseBody`: obje \longleftrightarrow JSON format

9. Spring MVC Dependency Injection Türleri Nelerdir?

1. **Constructor Injection:** Bağımlılıklar kurucu metoduyla enjekte edilir. Tek bir constructor varsa Autowired opsiyoneldir. Daha güvenli, bağımlılıkları zorunlu kılar, daha kolay test edilir vs. Bu sebeple Spring tarafından önerilir
2. **Setter Injection:** Bağımlılıklar setter metodları ile enjekte edilir.
3. **Field Injection:** Doğrudan alan üzerinde enjekte edilir (`@Autowired`).

10. Spring @RequestMapping Nedir?

`@RequestMapping("/users")`

- Class seviyesinde ise tüm metodlar için, metod seviyesinde ise sadece o metod için geçerlidir.

11. Spring

@GetMapping/@PutMapping/@PostMapping/@PatchMapping/@DeleteMapping Nedir?

Http Metodlarını eşleştirmek controllerdaki metodlarla eşleştirmek için kullanılır.

12. Spring @RequestParam Nedir?

- URL'deki sorguda değişken parametreleri yakalamak için kullanılır.

baseUrl/users/query?name=Ali

```
public MAV getUsers(@RequestParam String name){  
}
```

13. Spring @PathVariable Nedir?

- URL'deki path parametreleri yakalamak için kullanılır.

baseUrl/users/Ali/12

@GetMapping(/users/{name}/{age})

```
public MAV getUsers(@PathVariable("name") String name, @PathVariable("age") int
yas){
}
```