

# Spring MockMvc ve Controller Testleri



# MockMvc ile Handler Metotlarının Test Edilmesi

- MockMvc nesnesi vasıtası ile **standalone ortamda** Controller bean'larının handler metotları invokasyonu tetiklenebilir
- **MockMvc.perform()** ile test ortamında bir web request'i yapılabilir
- **MockMvcRequestBuilders** ile HttpServletRequest nesnesi oluşturulabilir

## Metotlarının Test Edilmesi

- Request sonucunda **MvcResult** nesnesi döner
- Dönen MvcResult içerisinde request, response, **ModelAndView**, **Exception** nesnelere erişilebilir
- **ResultHandler** vasıtası ile dönen response içeriği **görüntülenebilir**
- **ResultMatcher** vasıtası ile dönen response üzerinde **assertion**'lar yapılabilir

# MockMvc ile Handler Metotlarının Test Edilmesi

```
@RunWith(SpringJUnit4ClassRunner.class)
@WebAppConfiguration("webapp")
@ContextConfiguration("/appcontext/beans-*.xml")
public class WebAppContextMvcTests {

    @Autowired
    private WebApplicationContext wac; // cached

    private MockMvc mockMvc;

    @Before
    public void setup() {
        this.mockMvc = MockMvcBuilders.webAppContextSetup(this.wac).build();
    }

    @Test
    public void testGetVets() throws Exception {
        MvcResult result = mockMvc.perform(MockMvcRequestBuilders
                                           .get("/vets")).andReturn();
    }
}
```

# RequestBuilder ve ResultHandler Kullanımı

**RequestBuilder** :MockHttpServletRequest  
oluşturmaya yarar  
MockMvcRequestBuilders ile kullanılır

```
@Test
public void testGetVets() throws Exception {

    mockMvc.perform(MockMvcRequestBuilders.get("/vets"))
        .andDo(MockMvcResultHandlers.print());
}
```

**ResultHandler** :Dönen response üzerinde bir action  
execute etmeyi sağlar  
MockMvcResultHandlers ile kullanılır

# ResultMatcher Kullanımı

```
@Test
public void testFindOwners() throws Exception {
    mockMvc.perform(MockMvcRequestBuilders
        .get("/owners").param("lastName", "Davis"))
        .andExpect(MockMvcResultMatchers.view().name("ownersList"))
        .andExpect(MockMvcResultMatchers.model()
            .attributeExists("owners"));
}
```

**ResultMatcher** :Dönen response üzerinde bir “expectation”ın kontrol edilmesini sağlar  
MockMvcResultMatchers ile kullanılır

# İletişim

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

