

# Spring Boot ve Testler



- Spring Boot, **entegrasyon ve birim testlerinin** yazılmasını ve çalıştırılmasını da kolaylaştırır
- Aktive etmek için **test starter**'ın eklenmesi yeterlidir

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-test</artifactId>  
</dependency>
```

# Spring Boot ve Örnek Entegrasyon Birim Testi

```
@RunWith(SpringRunner.class)
@SpringBootTest
public class PetClinicIntegrationTests {

    @Autowired
    private PetClinicService petClinicService;

    @Autowired
    private WebApplicationContext wac;

    @Test
    public void testFindOwners() {
        List<Owner> owners = petClinicService.findOwners();
        MatcherAssert.assertThat(owners.size(), Matchers.equalTo(10));
    }
}
```

# Entegrasyon Testleri ve REST Servislerine Erişim

- Spring Boot entegrasyon testini çalıştırırken web container'ıda **gömülü olarak** çalıştırabilmektedir
- Bu sayede uygulamanın **web katmanına erişimi** entegrasyon testleri üzerinden kontrol etmek mümkün olmaktadır

# Entegrasyon Testleri ve REST Servislerine Erişim

```
@RunWith(SpringRunner.class)
```

```
@SpringBootTest(webEnvironment=WebEnvironment.RANDOM_PORT)
```

```
public class PetClinicControllerIntTests {
```

```
@LocalServerPort
```

```
private int port;
```

```
@Autowired
```

```
private TestRestTemplate testRestTemplate;
```

```
private URL base;
```

```
@Before
```

```
public void setUp() throws Exception {  
    base = new URL("http://localhost:" + port + "/hello");  
}
```

```
@Test
```

```
public void testWelcome() throws IOException {  
    String response = testRestTemplate.getForObject(  
                                                                    base.toString(), String.class);  
    MatcherAssert.assertThat(response,  
                             Matchers.containsString("Welcome to PetClinic WebApp!"));  
}
```

Spring RestTemplate sınıfının üzerine entegrasyon testleri için ilave kabiliyetler koyar

# TestRestTemplate

- Arka planda işi RestTemplate'a **delege** eder
- **Basic auth** header bilgilerini test içerisinde daha kolay set etmeyi sağlar
- Apache Http Client 4.3.2 ve üstü mevcut ise **Client kütüphane** olarak otomatik bunu kullanır
- Default olarak **cookie ve http redirect'leri göz ardı** edecek biçimde ayarlanmıştır

# Mock Mvc ile Controller Metotlarının Test Edilmesi

- Spring Boot classpath'de Servlet API sınıfları mevcut ise default olarak **Mock nesnelerden oluşan bir Servlet Container ortamı** oluşturmaktadır
- Bu ortam üzerinden Tomcat gibi herhangi bir web container olmaksızın **MVC controller metotları** entegrasyon testlerine tabi tutulabilir

# Mock Mvc ile Controller Metotlarının Test Edilmesi

WebApplicationContext'i Mock Servlet ortamı ile oluşturur, eğer Servlet API sınıfları classpath'de mevcut değil ise normal bir ApplicationContext oluşturur

```
@RunWith(SpringRunner.class)
@SpringBootTest(webEnvironment=WebEnvironment.MOCK)
@AutoConfigureMockMvc
public class PetClinicControllerUnitTests {

    @Autowired
    private MockMvc mvc;

    @Test
    public void testWelcome() throws Exception {
        mvc.perform(MockMvcRequestBuilders.get("/hello")
            .accept(MediaType.APPLICATION_JSON))
            .andExpect(MockMvcResultMatchers.status().isOk())
            .andExpect(MockMvcResultMatchers.content()
                .string(Matchers.containsString(
                    "Welcome to PetClinic WebApp!"))));
    }
}
```

MockMvc nesnesinin otomatik konfigürasyonunu gerçekleştir



# Mock Mvc ile Controller Metotlarının Test Edilmesi

```
@RunWith(SpringRunner.class)
```

```
@WebMvcTest(PetClinicRestController.class)
```

```
public class PetClinicControllerUnitTests {
```

```
@Autowired
```

```
private MockMvc mvc;
```

```
@Test
```

```
public void testWelcome() throws Exception {  
    mvc.perform(MockMvcRequestBuilders.get("/hello")  
        .accept(MediaType.APPLICATION_JSON))  
        .andExpect(MockMvcResultMatchers.status().isOk())  
        .andExpect(MockMvcResultMatchers.content()  
            .string(Matchers.containsString(  
                "Welcome to PetClinic WebApp!"))));  
}
```

```
}
```

MockMvc nesnesinin otomatik konfigürasyonunu gerçekleştir Sadece ilgili Controller bean'ini initialize eder

# İletişim

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

