

Spring Unit Testing

Friday, November 6, 2020 9:43 AM

- junit is the most popular Unit Testing Framework in Java Echo System.
- It is flexible and scalable via third party
- It's main components are
 - Test Runner
 - Test Reporter
 - Test Assertions
- It is possible in junit to extend or replace any part of it
 - We created our own Assertions.

Springify junit

- Spring doesn't aim at replacing junit, it aims at making it better.
- Even Unit Testing can leverage the benefit of Spring Dependency Injection and AOP
- We can use Spring to dependency inject component in our Test Cases
- We can create the context in Spring Test and Use it

```
public class SpringXmlConfigTests {  
    ApplicationContext context;  
  
    @Before  
    public void setUp() throws Exception {  
        context=new ClassPathXmlApplicationContext("classpath:config/test.config.xml");  
    }  
  
    @Test  
    public void canAccessBeanByItsName() {  
        Object atm = context.getBean("myAtm");  
  
        assertNotNull(atm);  
        assertEquals(ATM.class, atm);  
    }  
}
```



Spring Support for junit

- Spring Provides its own TestRunner for junit
- The advantage is the SpringTestRunner automatically injects dependency to a junit Tests using standard @AutoWired
- You don't need to create the contexts

```
@TestRunner( SpringTestRunner.class)  
@Component  
@Configuration( MyTestConfig.class)  
class MySpringifiedAccountRepositoryTest{  
  
    @Autowired AccountRepository repository;  
}
```

Important

- Test Support is not present in the core spring package you need to import another Spring Project

```
<dependency>  
    <groupId>org.springframework</groupId>  
    <artifactId>spring-test</artifactId>  
    <version>${spring.version}</version>  
    <scope>test</scope>  
</dependency>
```