Spring MVC JUnit Mock Standalone Configuration

Maven Dependencies

```
<properties>
	<spring.test>4.1.4.RELEASE</spring.test>
	<junit.version>4.12</junit.version>
	<hamcrest.version>1.3</hamcrest.version>
	<mockito.all>2.0.2-beta</mockito.all>
</properties>

<dependencies>
<dependency>
	<groupId>org.springframework</groupId>
	<artifactId>spring-test</artifactId>
	<version>${spring.test}</version>
	<scope>test</scope>
</dependency>
<dependency>
	<groupId>junit</groupId>
	<artifactId>junit</artifactId>
	<version>${junit.version}</version>
	<scope>test</scope>
</dependency>
<dependency>
	<groupId>org.hamcrest</groupId>
	<artifactId>hamcrest-library</artifactId>
	<version>${hamcrest.version}</version>
	<scope>test</scope>
</dependency>
<dependency>
	<groupId>org.mockito</groupId>
	<artifactId>mockito-all</artifactId>
	<version>${mockito.all}</version>
	<scope>test</scope>
</dependency>
</dependencies>
```

Example Usage

```
import org.junit.Before;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.mockito.InjectMocks;
import org.mockito.runners.MockitoJUnitRunner;
import org.springframework.test.web.servlet.MockMvc;
import org.springframework.test.web.servlet.setup.MockMvcBuilders;
import org.springframework.web.servlet.ModelAndView;

import java.util.List;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertNotEquals;
import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;

@RunWith(MockitoJUnitRunner.class)
public class GumrukServiceUtilTest {
     @InjectMocks
     private AdminController adminController;

     private MockMvc mockMvc;

     @Before
     public void setup() {
         // Setup Spring test in standalone mode
         this.mockMvc= MockMvcBuilders.standaloneSetup(adminController).build();
     }

     @Test
     public void testKayitliBelgeVerileri() throws Exception {
         ModelAndView modelAndView = mockMvc.perform(get("/yonetim")).
             andExpect(status().isOk()).andReturn().getModelAndView();
         assertEquals("yonetim/index", modelAndView.getViewName());
     }

}
```

At Line 19 we have to use `@InjectMocks` annotation if we want to use standalone initialization specified at line 27.