Spring Boot Web Test Support

Full Stack Test

```java
@RunWith(SpringRunner.class)
@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
public class EmbeddedServerIT {

    @Autowired
    private TestRestTemplate restTemplate;

    @Test
    public void testJsp() {
        ResponseEntity<String> entity = this.restTemplate.getForEntity("/", String.class);

        assertThat(entity.getStatusCode()).isEqualTo(HttpStatus.OK);
        assertThat(entity.getBody()).contains("Thymeleaf");
    }

}
```

Context Only Test

```java
@RunWith(SpringRunner.class)
@SpringBootTest
@AutoConfigureMockMvc
public class PublicPlayerControllerIT {

    @Autowired
    private MockMvc mockMvc;

    @Test
    public void testJsp() throws Exception {
        mockMvc.perform(MockMvcRequestBuilders.get("/"))

                .andDo(MockMvcResultHandlers.print())

                .andExpect(status().isOk())
                .andExpect(content().string(containsString("Thymeleaf")));
    }

}
```

Controller Unit Test

```java
@RunWith(SpringRunner.class)
@WebMvcTest(controllers = AdminPlayerController.class, secure = false)
public class AdminPlayerControllerTest {

    @Autowired
    private MockMvc mockMvc;

    @MockBean
    private IPlayerService service;

    @Test
    public void testJsp() throws Exception {
        when(service.findAll()).thenReturn(PlayerBuilder.getInstance().createPlayers());

        mockMvc.perform(MockMvcRequestBuilders.get("/admin/players/list/"))

                .andDo(MockMvcResultHandlers.print())

                .andExpect(status().isOk())
                .andExpect(view().name("players/list"));
    }

}
```