



# 7/13 ~ 7/17 2주차 - 7

## REST를 위한 단위 테스트

- RestController의 단위 테스트에는 com.jayway.jsonpath가 주로 사용된다.
- spring boot 에서는 spring-boot-starter-test에서 이미 포함되고 있으므로 추가 설정은 필요없다.

## 주요 표현 예

표현	의미
\$	전달받은 json 객체의 root element
\$.name	root element의 name 속성
\$.hobby[2]	root element 중 hobby 속성이 배열이고 2번 index 요소
\$.hobby[2].name	root element의 hobby 속성 중 2번 index의 name 속성
\$.hobby.length()	root element의 hobby 속성의 길이

## 테스트 해보기

- 전체 국가 조회

```

@Autowired
RestController controller;

MockMvc mock;

@BeforeEach
public void setup() {
    mock = MockMvcBuilders.standaloneSetup(controller).build();
}

@Test
public void selectAllTest() throws Exception {
    mock.perform(get("/api/countries").param("per", "5").param("page", "1"))
        .andExpect(status().isOk())
        // 전달된 content type 검증
        .andExpect(header().string("content-type", "application/json"))
        // 배열의 개수 검증
        .andExpect(jsonPath("$.data.length()", equalTo(5)))
        // 데이터 검증
        .andExpect(jsonPath("$.data[0].code", equalTo("ABW")))
        .andDo(print());
}

```

- 특정 국가 정보 조회

```

@Test
public void selectTest() throws Exception {
    mock.perform(get("/api/countries/KOR"))
        .andExpect(status().isOk())
        // Country의 속성 개수
        .andExpect(jsonPath("$.data.length()", equalTo(16)))
        // Country 속성 값 검증
        .andExpect(jsonPath("$.data.name", equalTo("South Korea")))
        // 데이터 검증 - 배열의 개수
        .andExpect(jsonPath("$.data.cities.length()", equalTo(70)))
        // 데이터 검증 - 값
        .andExpect(jsonPath("$.data.cities[0].district", equalTo("Seoul")))
        .andDo(print());
}

```

- 국가 정보 추가

- DML 테스트 이기 때문에 @Transactional이 사용되었다.
- Country 객체를 전달하기 위해 Jackson Databind를 이용해 JSON 형태의 문자열로 만들었다.

JSON 형태의 데이터를 전달하기 위해 **contentType**에 **MediaType.APPLICATION\_JSON**으로 설정되었으며 **content**로 **json** 문자열을 넘겨준다.

```
@Test
@Transactional
public void insertTest() throws Exception {
    Country country = new Country();
    country.setCode("TTT");
    country.setName("TestCountry");
    country.setContinent("Asia");
    country.setRegion("Eastern Asia");
    country.setSurfaceArea(994340);
    country.setPopulation(46844000L);
    country.setLocalName("Test Local Name");
    country.setGovernmentForm("Republic");
    country.setCode2("TT");

    // Jackson Databind 사용(객체 ⇄ JSON 문자열 변환)
    ObjectMapper mapper = new ObjectMapper();
    String jsonStr = mapper.writeValueAsString(country);

    mock.perform(post("/api/countries").contentType(MediaType.APPLICATION_JSON).content(jsonStr))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$.success", is(true)))
        .andExpect(jsonPath("$.data", is(1)))
        .andDo(print());
}
```

- 국가 정보 수정

```

@Test
@Transactional
public void updateTest() throws Exception {
    Country country = new Country();
    country.setCode("KOR");
    country.setName("TestCountry");
    country.setContinent("Asia");
    country.setRegion("Eastern Asia");
    country.setSurfaceArea(99434D);
    country.setPopulation(46844000L);
    country.setLocalName("Test Local Name");
    country.setGovernmentForm("Republic");
    country.setCode2("TT");

    // Jackson Databind 사용
    ObjectMapper mapper = new ObjectMapper();
    String jsonStr = mapper.writeValueAsString(country);

    mock.perform(put("/api/countries").contentType(MediaType.APPLICATION_JSON).content(jsonStr))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$.success", is(true)))
        .andExpect(jsonPath("$.data", is(1)))
        .andDo(print());
}

```

- 국가 정보 삭제

```

@Test
@Transactional
public void deleteTest() throws Exception {
    // 도시가 없는 국가만 삭제가 가능하다.
    mock.perform(delete("/api/countries/ATA"))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$.success", is(true)))
        .andExpect(jsonPath("$.data", is(1)))
        .andDo(print());
}

```