TestRestTemplate

```
package integration.tests.web.layer;

import java.net.URI;
import java.net.URISyntaxException;

import org.assertj.core.api.Assertions;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.boot.test.context.SpringBootTest.WebEnvironment;
import org.springframework.boot.test.web.client.TestRestTemplate;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpMethod;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.test.context.jdbc.Sql;

import integration.tests.web.layer.person.Person;
import integration.tests.web.layer.person.PersonRepository;

@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
class TestRestTemplateIntegrationTest {

	@Autowired
	private TestRestTemplate testRestTemplate;

	@Autowired
	private PersonRepository personRepository;

	@AfterEach
	public void cleanUp() {
		personRepository.deleteAll();
	}

	@Test
	@Sql(statements = "INSERT INTO person (name, age) VALUES ('Mark',
40)")
	public void getTest() throws URISyntaxException {
		ResponseEntity<Person> responseGetForEntity =
testRestTemplate.getForEntity(new URI("/"), Person.class);

Assertions.assertThat(responseGetForEntity.getBody().getName()).isEqualTo("
Mark");
```

```
Assertions.assertThat(responseGetForEntity.getStatusCode()).isEqualTo(HttpStatus.OK);

        Person responseGetForObject = testRestTemplate.getForObject("/", Person.class);

Assertions.assertThat(responseGetForObject.getName()).isEqualTo("Mark");
    }

    @Test
    public void postTest() {
        ResponseEntity<Person> responsePostForEntity = testRestTemplate.postForEntity("/",
                new Person("John", 62), Person.class);

Assertions.assertThat(responsePostForEntity.getBody().getName()).isEqualTo("John");

Assertions.assertThat(responsePostForEntity.getStatusCode()).isEqualTo(HttpStatus.CREATED);

        Person responsePostForObject = testRestTemplate.postForObject("/",
                new Person("Doe", 45), Person.class);

Assertions.assertThat(responsePostForObject.getName()).isEqualTo("Doe");
    }

    @Test
    @Sql(statements = "INSERT INTO person (id, name, age) VALUES (99, 'Mark', 57)")
    public void putTest() {
        testRestTemplate.put("/", new Person(99,"Len", 57));

        Person person = personRepository.findById(99).get();
        Assertions.assertThat(person.getName()).isEqualTo("Len");
    }

    @Test
    @Sql(statements = "INSERT INTO person (id, name, age) VALUES (50, 'Len', 57)")
    public void deleteTest() {
        testRestTemplate.delete("/{id}", 50);

        Long count = personRepository.count();
        Assertions.assertThat(count).isEqualTo(0);
```

```
	}

	@Test
	@Sql(statements = "INSERT INTO person (name, age) VALUES ('Mark',
57)")
	public void exchangeTest() {
		ResponseEntity<Person> response = testRestTemplate.exchange("/",
HttpMethod.GET,
				HttpEntity.EMPTY, Person.class);

Assertions.assertThat(response.getBody().getName()).isEqualTo("Mark");

Assertions.assertThat(response.getStatusCode()).isEqualTo(HttpStatus.OK);
	}
}
```