luv2code

Database Integration Testing

Database Initialization and Cleanup

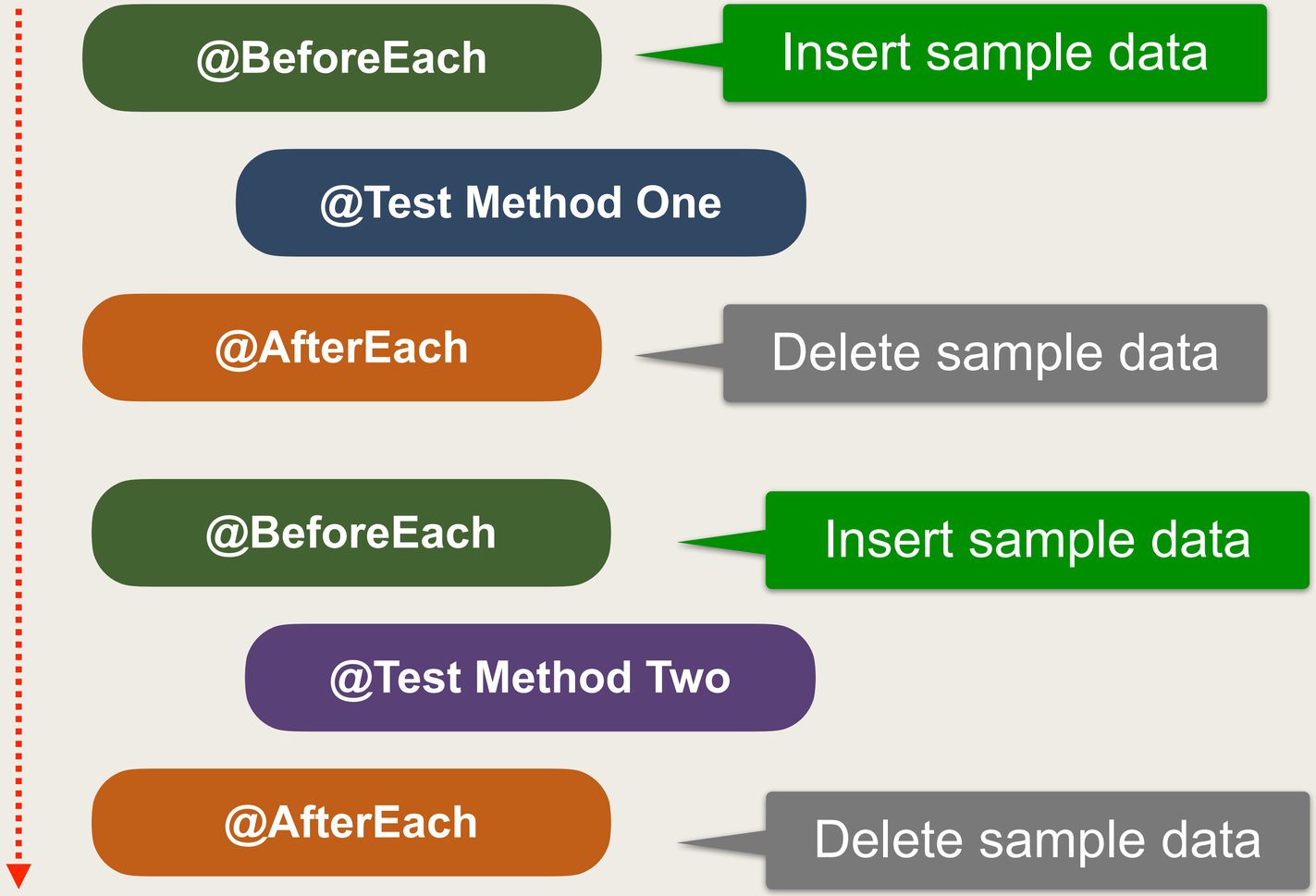
- When we are performing integration testing with a database
  - Each test should run from a known state

- Before each test, perform initialization
  - Insert sample data

- After each test, perform cleanup
  - Delete the sample data

Testing Approach

- Each test should run from a known state



@Before and @AfterEach

StudentAndGradeServiceTest.java

```
import org.springframework.jdbc.core.JdbcTemplate;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
...

@TestPropertySource("/application.properties")
@SpringBootTest
public class StudentAndGradeServiceTest {

   @Autowired
   private JdbcTemplate jdbc;

   @BeforeEach
   public void setupDatabase() {
       jdbc.execute("insert into student(id, firstname, lastname, email_address) " +
               "values (1, 'Eric', 'Roby', 'eric.roby@luv2code_school.com')");
   }


   @AfterEach
   public void setupAfterTransaction() {
       jdbc.execute("DELETE FROM student");
   }

}
```

From the Spring Framework

Insert sample data

Delete sample data

StudentAndGradeServiceTest.java

```
...
public class StudentAndGradeServiceTest {

   @Autowired
   private JdbcTemplate jdbc;

   @BeforeEach
   public void setupDatabase() {
       jdbc.execute("insert into student(id, firstname, lastname, email_address) " +
               "values (1, 'Eric', 'Roby', 'eric.roby@luv2code_school.com')");
   }

   @Test
   public void isStudentNullCheck() {

       assertTrue(studentService.checkIfStudentIsNull(1));

       assertFalse(studentService.checkIfStudentIsNull(0));
   }

   @AfterEach
   public void setupAfterTransaction() {
       jdbc.execute("DELETE FROM student");
   }

}
```

Returns true since
id 1 exists in database

Returns false since
id 0 does not exist in database