

# Integration Tests

# Integration Tests

---

- ▶ Unit tests test small units of code, for example calling a method and assert that the expected return value is what actually is returned
- ▶ Integration tests test the whole system - the integration of many units
- ▶ Spring Boot has test support for sending HTTP requests to Controllers
- ▶ It also has support for asserting that the response include expected values and that it doesn't include unexpected values

# Spring Boot Support

---

- ▶ Need to annotate the test class with `@AutoConfigureMockMvc`
- ▶ Autowire a `MockMvc` object into the Test class to perform the requests
- ▶ The response can be inspected to make sure expected values are included, and also that unexpected values are not included

# Demo 1 - Integration Tests

---

- ▶ Trigger requests in an Integration Test
- ▶ Assert that the response include expected value (and possibly that it doesn't include unexpected values)

# Exercise 1 - Integration Tests

---

- ▶ Reuse the solution from Creating a Spring Boot Application (with two methods calculate the sum of number1 and number2)
- ▶ Create an Integration test for the method that takes two request params and make sure that when sending two numbers the sum will be present in the response. Also test that another number that is not expected isn't included.
- ▶ Then create a similar test for the method that uses path variables, making sure the sum is returned and that some other irrelevant number is not part of the response

# Exercise 2 - Integration Tests

---

- ▶ If you have time - create another Controller method with a `@GetMapping` and make it return something.
- ▶ Or reuse another Spring Boot project from the course, for example a Spring MVC project.
- ▶ Then create integration tests that makes sure that the response includes the expected value, and also perhaps that it doesn't include an unexpected value