



Spring Testing

Integration Testing

Importance of Integration Testing

Integration testing helps identify issues related to the interaction between different modules or services.

Many issues, such as configuration errors, missing data, or incorrect business logic, only surface when components interact with each other.

Integration tests serve as a safeguard when changes are made to the codebase. They help ensure that new changes do not break the existing functionality, making regression testing more effective.



Using WebClient

WebTestClient provides a fluent API and supports making real HTTP requests to your application. Although **WebTestClient** is primarily associated with WebFlux applications, it can also be used with Spring MVC applications when set up correctly.

Use the `@AutoConfigureWebTestClient` to auto configure the **WebTestClient**.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-webflux</artifactId>
  <scope>test</scope>
</dependency>
```

WebTestClient Testing Methods

Common WebTestClient Response Methods:

- `exchange()`: Executes the request and returns a `WebTestClient.ResponseSpec`
- `expectStatus()`: Asserts the status code of the response.
- `expectBody()`: Asserts the body of the response.
- `expectHeader()`: Asserts the headers of the response.
- `.jsonPath("$.id").isNotEmpty()`
- `.jsonPath("$.name").isEqualTo("Jane Doe")`
- `.jsonPath("$.email").isEqualTo("jane.doe@example.com")`

