

Using properties without .properties file

You can inject properties directly into your tests without writing them into any `.properties` file by using Spring's `@TestPropertySource` annotation. This annotation allows you to define properties inline directly within your test class.

Here's an example:

```
```java
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.TestPropertySource;

@SpringBootTest
@TestPropertySource(properties = {
 "myapp.some-property=value1",
 "myapp.another-property=value2"
})
public class MyTest {

 @Value("${myapp.some-property}")
 private String someProperty;

 @Value("${myapp.another-property}")
 private String anotherProperty;

 @Test
 public void testProperties() {
 // Assert the injected properties
 assert "value1".equals(someProperty);
 assert "value2".equals(anotherProperty);
 }
}
```
```

In this example:

- We use `@TestPropertySource` to specify inline properties directly within the test class.
- The `properties` attribute of `@TestPropertySource` allows you to define key-value pairs of properties.
- We inject the properties using `@Value` annotation and assert their values in the test method.

This approach is useful for testing scenarios where you need to provide specific property values for your tests without relying on external `.properties`

files.