@RunWith

The `@RunWith` annotation in JUnit is used to define a custom test runner class for running tests. By default, JUnit uses the `BlockJUnit4ClassRunner` to execute test cases, but you can specify a different test runner using `@RunWith`.

For example, if you're using Spring-based tests with JUnit, you typically use the `SpringJUnit4ClassRunner` to enable Spring's testing support. Here's an example:

```java
@RunWith(SpringJUnit4ClassRunner.class)
@SpringBootTest
public class MySpringTest {
    // Test methods
}
```

In this example, `@RunWith(SpringJUnit4ClassRunner.class)` tells JUnit to use Spring's test runner for executing tests, and `@SpringBootTest` indicates that this is a Spring Boot test and Spring Boot's features should be enabled for testing.

Similarly, if you're using Mockito for mocking objects in tests, you can use the `MockitoJUnitRunner`:

```java
@RunWith(MockitoJUnitRunner.class)
public class MyMockitoTest {
    // Test methods
}
```

Using `@RunWith` allows you to integrate different test frameworks or customize the test execution environment according to your needs.