

Using @Tag to filter test executions

So we have tests. Actually, we have a few tests, and they are located in several classes. We want to run just a subset of them. Say we want to run Sanity tests only when a new version of the System Under Test is deployed to provide a quick feedback loop for the dev team. We use the built-in Junit 5 [Tag annotation](#). Here's how it's done:

Define a @Tag annotation

The first step is to add a @Tag annotation to each test method. For example:

```
1 @Test
2 @DisplayName("Sanity test")
3 @Tag ("Sanity")
4 public void sanityTest(){
5     Report.info("This is a sanity test");
6 }
```

Once we have defined `sanityTest` to carry the Sanity tag, we will be able to run maven CLI in the following way:

```
1 mvn test -Dgroups=Sanity
```

You may add more than one tag:

```
1 @Test
2 @DisplayName("Sanity and Regression test")
3 @Tag("Sanity")
4 @Tag("Regression")
5 public void sanityAndRegression(){
6     Report.info("This is a sanity and regression test");
7 }
```

and this code will run with either `-Dgroups="Sanity"` or `-Dgroups="Regression"`.

You may also run more than one group set:

```
1 mvn test -Dgroups=Sanity,Regression
```

And finally, you can create a "sort of" enum class to help you with the string of the tags:

```
1 public class Suites {
2     public static final String Sanity = "Sanity";
3     public static final String Regression = "Regression";
4 }
```

and then the final code will look like this:

```
1 @Test
2 @DisplayName("Sanity and Regression test")
3 @Tag(Suites.Sanity)
4 @Tag(Suites.Regression)
5 public void sanityAndRegression(){
6     Report.info("This is a sanity and regression test");
7 }
```