Meecrowave Testing

JUnit

Coordinates:

```
<dependency>
  <groupId>org.apache.meecrowave</groupId>
  <artifactId>meecrowave-junit</artifactId>
  <version>${meecrowave.version}</version>
  <scope>test</scope>
</dependency>
```

Rules and Runners

Meecrowave provides two flavors of JUnit integration: standalone or runners/rules. The standalone one will ensure there is a single container for the whole JVM. It also fits standalone environments where you want to control the lifecycle. The other one will follow the JUnit lifecycle (per class or test rule).

Here how to use the standalone flavor:

```
@RunWith(MonoMeecrowave.Runner.class)
public class MonoMeecrowaveRuleTest {
    /* or
    @ClassRule
    public static final MonoMeecrowave.Rule RULE = new MonoMeecrowave.Rule();
    */

    @MonoMeecrowave.Runner.ConfigurationInject
    private Meecrowave.Builder config;

    @Test
    public void test() throws IOException {
        // use "http://localhost:" + config.getHttpPort()
    }
}
```

When using the standalone, `@MonoMeecrowave.Runner.ConfigurationInject` allows to still access the configuration and random HTTP port.

For the configuration, the standalone runner will use a global configuration shared by all tests. To load it it will use a standard `ServiceLoader` on type `org.apache.meecrowave.Meecrowave$ConfigurationCustomizer`.

And here is the one bound to the JUnit lifecycle

```
public class MeecrowaveRuleTest {
    @ClassRule // started once for the class, @Rule would be per method
    public static final MeecrowaveRule RULE = new MeecrowaveRule();

    @Test
    public void test() throws IOException {
        // use "http://localhost:" + RULE.getConfiguration().getHttpPort()
    }
}
```

As usual with JUnit rules, you can decide whereas the Meecrowave instance is bound to the entire test class or a method by using @ClassRule or @Rule.

JUnit 5

JUnit 5 integrates a new `Extension` system. It is not yet very well supported by IDEs but you can already use it with Gradle and Maven (see http://junit.org/junit5/docs/current/user-guide/#running-tests).

The usage has two annotations: `@MeecrowaveConfig` which remaps most of the configuration of Meecrowave and `@MonoMeecrowaveConfig` which is close to `MonoMeecrowave.Runner` in term of usage.

```
@MeecrowaveConfig /*(some config)*/
@TestInstance(PER_CLASS)
public class MeecrowaveConfigTest {
    @ConfigurationInject
    private Meecrowave.Builder config;

    @Test
    public void run() throws MalformedURLException {
        final String base = "http://localhost:" + config.getHttpPort();
        // asserts
    }
}
```

Or

```
@MonoMeecrowaveConfig
public class MeecrowaveConfigTest {
    // ...
}
```

JUnit 5 integration provides an `@AfterFirstInjection` method and `@AfterLastTest` which can be used to setup/reset some environment using injections once for a set of test methods. The methods must not have any parameter.

when not using `@TestInstance(PER_CLASS)`, container is started per test method. Generally speaking you should try to align the scope of your container to the scope of validity of your beans. For a library it is generally the class (so `@MeecrowaveConfig @TestInstance(PER_CLASS)`) and for an application the whole test set (so `@MonoMeecrowaveConfig`). Note that using an `Extension` you can adjust mocks or spy beans dynamically without a container restart. Having the longest life time for the container will make your test suite faster to execute.

Arquillian Container

Container dependency:

```
<dependency>
  <groupId>org.apache.meecrowave</groupId>
  <artifactId>meecrowave-arquillian</artifactId>
  <version>${meecrowave.version}</version>
  <scope>test</scope>
</dependency>
```

For the configuration check Core configuration.

Here is a sample:

```
<?xml version="1.0" encoding="UTF-8"?>
<arquillian xmlns="http://jboss.org/schema/arquillian"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://jboss.org/schema/arquillian
http://jboss.org/schema/arquillian/arquillian_1_0.xsd">
  <container qualifier="meecrowave" default="true">
    <configuration>
      <property name="antiResourceLocking">false</property>
      <property name="arquillianProtocol">Servlet 3.1</property>
      <property name="cdiConversation">false</property>
      <property name="clientAuth"></property>
      <property name="conf"></property>
      <property name="cxfServletParams">
        hide-service-list-page=true
      </property>
      <property name="defaultSSLHostConfigName"></property>
      <property name="deleteBaseOnStartup">true</property>
      <property name="dir"></property>
      <property name="host">localhost</property>
      <property name="http2">false</property>
      <property name="httpPort">-1</property>
      <property name="httpsPort">8443</property>
      <property name="initializeClientBus">true</property>
      <property name="injectServletContainerInitializer">true</property>
```

```
      <property name="jaxrsAutoActivateBeanValidation">true</property>
      <property name="jaxrsDefaultProviders"></property>
      <property name="jaxrsLogProviders">false</property>
      <property name="jaxrsMapping">/*</property>
      <property name="jaxrsProviderSetup">true</property>
      <property name="jaxwsSupportIfAvailable">true</property>
      <property name="jsonbBinaryStrategy"></property>
      <property name="jsonbEncoding">UTF-8</property>
      <property name="jsonbIJson">false</property>
      <property name="jsonbNamingStrategy"></property>
      <property name="jsonbNulls">false</property>
      <property name="jsonbOrderStrategy"></property>
      <property name="jsonbPrettify">false</property>
      <property name="jsonpBufferStrategy">QUEUE</property>
      <property name="jsonpMaxReadBufferLen">65536</property>
      <property name="jsonpMaxStringLen">10485760</property>
      <property name="jsonpMaxWriteBufferLen">65536</property>
      <property name="jsonpPrettify">false</property>
      <property name="jsonpSupportsComment">false</property>
      <property name="keepServerXmlAsThis">false</property>
      <property name="keyAlias"></property>
      <property name="keystoreFile"></property>
      <property name="keystorePass"></property>
      <property name="keystoreType">JKS</property>
      <property name="loggingGlobalSetup">true</property>
      <property name="loginConfig">authMethod=BASIC;realmName=app</property>
      <property name="meecrowaveProperties">meecrowave.properties</property>
      <property name="properties">
        jpa.property.openjpa.RuntimeUnenhancedClasses=supported
        jpa.property.openjpa.jdbc.SynchronizeMappings=buildSchema
      </property>
      <property name="quickSession">true</property>
      <property name="realm"
>org.apache.catalina.realm.JAASRealm:configFile=jaas.config;appName=app</property>
      <property name="roles">
        admin=admin
        limited=admin,other
      </property>
      <property name="scanningExcludes"></property>
      <property name="scanningIncludes"></property>
      <property name="scanningPackageExcludes"></property>
      <property name="scanningPackageIncludes"></property>
      <property name="securityConstraints"
>collection=sc1:/api/*:POST;authRole=**|collection=sc2:/priv/*:GET;authRole=*</propert
y>
      <property name="serverXml"></property>
      <property name="sharedLibraries"></property>
      <property name="skipHttp">false</property>
      <property name="ssl">false</property>
      <property name="sslProtocol"></property>
      <property name="stopPort">-1</property>
```

```
      <property name="tempDir">/tmp/meecrowave_16133794495335</property>
      <property name="tomcatAccessLogPattern"></property>
      <property name="tomcatAutoSetup">true</property>
      <property name="tomcatFilter"></property>
      <property name="tomcatJspDevelopment">false</property>
      <property name="tomcatNoJmx">true</property>
      <property name="tomcatScanning">true</property>
      <property name="tomcatWrapLoader">true</property>
      <property name="useLog4j2JulLogManager">true</property>
      <property name="useShutdownHook">false</property>
      <property name="useTomcatDefaults">true</property>
      <property name="users">
        admin=adminpwd
        other=secret
      </property>
      <property name="webResourceCached">true</property>
      <property name="webSessionCookieConfig"></property>
      <property name="webSessionTimeout"></property>
      <property name="webXml"></property>
    </configuration>
  </container>
</arquillian>
```